



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE TELECOMUNICACIONES

**INFORME FINAL DEL TRABAJO DE INTEGRACIÓN
CURRICULAR, PROYECTO DE INVESTIGACIÓN**

TEMA:

**“TRADUCTOR DE LENGUAJE DE SEÑAS PARA PERSONAS CON
DISCAPACIDAD AUDITIVA BASADO EN VISIÓN ARTIFICIAL”**

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

Línea de investigación: Desarrollo, aplicación de software y cibersecurity (seguridad cibernética)

AUTOR:

Hinojosa González David Santiago

DIRECTOR:

Ing. Cuzme Rodríguez Fabián Geovanny, MSc

Ibarra, 2024

**UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA**

IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003767694		
APELLIDOS Y NOMBRES:	HINOJOSA GONZÁLEZ DAVID SANTIAGO		
DIRECCIÓN:	AV. ATAHUALPA 35-120 Y NAZACOTA PUENTO		
EMAIL:	davidhinojosag@gmail.com		
TELÉFONO FIJO:	062-511-547	TELF. MOVIL	0995493539

DATOS DE LA OBRA	
TÍTULO:	TRADUCTOR DE LENGUAJE DE SEÑAS PARA PERSONAS CON DISCAPACIDAD AUDITIVA BASADO EN VISIÓN ARTIFICIAL
AUTOR (ES):	HINOJOSA GONZÁLEZ DAVID SANTIAGO
FECHA: AAAAMMDD	2024/12/02
SOLO PARA TRABAJOS DE INTEGRACIÓN CURRICULAR	
CARRERA/PROGRAMA:	<input checked="" type="checkbox"/> GRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN TELECOMUNICACIONES
DIRECTOR:	ING. CUZME RODRÍGUEZ FABIÁN GEOVANNY, MSC
ASESOR:	ING. MICHILENA CALDERÓN JAIME ROBERTO, MSC

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, DAVID SANTIAGO HINOJOSA GONZÁLEZ, con cédula de identidad Nro. 1003767694, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 3 días del mes de diciembre de 2024

EL AUTOR:



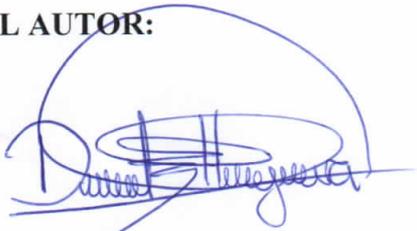
Hinojosa González David Santiago

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 3 días, del mes de diciembre de 2024

EL AUTOR:



.....

Hinojosa González David Santiago

**CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE
INTEGRACIÓN CURRICULAR**

Ibarra, 2 de diciembre de 2024

ING. CUZME RODRÍGUEZ FABIÁN GEOVANNY. MSC
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.



ING. CUZME RODRÍGUEZ FABIÁN GEOVANNY, MSC

C.C.: 1311527012

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “TRADUCTOR DE LENGUAJE DE SEÑAS PARA PERSONAS CON DISCAPACIDAD AUDITIVA BASADO EN VISIÓN ARTIFICIAL” elaborado por HINOJOSA GONZÁLEZ DAVID SANTIAGO, previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



Ing. Cuzme Rodríguez Fabián Geovanny, MSc

C.C.: 1311527012



Ing. Michilena Calderón Jaime Roberto, MSc

C.C.: 1002198438

DEDICATORIA

Quiero dedicar este trabajo de titulación a mis padres Vinicio y Tamara, a mis hermanos Martín y Emilia, y a toda mi familia, quienes con su amor y sacrificio han sido la base de mis logros y sueños, por estar a mi lado apoyándome a lo largo de este camino, por siempre brindarme de sus consejos y enseñanzas, ya que gracias a ellos soy la persona que soy.

A mi abuelito Jaime, ya que su ausencia física no disminuye su presencia en mi corazón, le dedico este logro porque merecía estar presente en este paso de mi vida.

A mi novia Samantha, por su ánimo en cada momento, por estar a mi lado, por creer en mí y por hacer que cada obstáculo sea más sencillo con su apoyo.

Finalmente, a Oliver, por ser mi fiel compañía desde el día cero con el cual inició este recorrido, su lealtad fue imprescindible en cada momento.

AGRADECIMIENTO

En primer lugar, agradezco a Dios por guiarme y poner en mi vida a personas que han sido fundamentales en este camino.

A mis padres y hermanos, quienes con su sacrificio, apoyo y valiosos consejos, han sido los pilares que me han permitido culminar mis estudios.

A mis amigos, gracias por su apoyo constante, por su atención y por la preocupación que demostraron en cada etapa de este recorrido.

Expreso mi más profundo agradecimiento a mi director Msc. Fabián Cuzme y a mi asesor Msc. Jaime Michilena, por compartirme sus conocimientos, por su orientación y el tiempo que dedicaron a mi formación académica, contribuyendo de manera significativa a mi aprendizaje.

Estoy profundamente agradecido con todas las personas que me han acompañado a lo largo de este trayecto, brindándome su presencia y apoyo para alcanzar este logro.

RESUMEN

La visión artificial es una rama dentro de la inteligencia artificial, con la cual un dispositivo busca simular la visión humana permitiendo extraer información relevante de imágenes mediante un proceso de entrenamiento previo. Por lo cual, la visión artificial puede ser aplicada a distintos campos para brindar varios beneficios. En este contexto, en el presente documento se presenta el desarrollo de un prototipo funcional de traducción de lenguaje de señas basado en visión artificial que permite a las personas con discapacidad auditiva comunicarse con personas que desconocen este lenguaje, promoviendo así la inclusión social. Para ello el sistema es implementado en la placa NVIDIA Jetson Nano para realizar la detección de letras del ASL (Lenguaje de Señas Americano) mediante una inferencia en tiempo real, haciendo uso de librerías destinadas a tareas de visión artificial y compatibles con el lenguaje de programación Python, como OpenCV y MediaPipe, además de la Red Neuronal Convolucional (CNN) “yolov8n.pt” del modelo YOLOv8, misma que fue entrenada con el dataset “Real-Time-Sign-Language”, que ofrece diversas imágenes de cada letra del alfabeto ASL. El proceso de desarrollo del presente proyecto incluye la detección en tiempo real de los gestos de la mano y la reproducción de la traducción mediante un sintetizador de voz. Este trabajo representa un aporte en accesibilidad y aprendizaje del lenguaje de señas, siendo una base para ser optimizada mediante modelos más complejos en futuros proyectos.

Palabras clave: Visión artificial, Red Neuronal Convolucional (CNN), traducción de lenguaje de señas, discapacidad auditiva, prototipo, inclusión social.

ABSTRACT

Computer vision is a branch within artificial intelligence, with which a device seeks to simulate human vision allowing to extract relevant information from images through a previous training process. Therefore, machine vision can be applied to different fields to provide several benefits. In this context, this paper presents the development of a functional prototype for sign language translation based on artificial vision that allows people with hearing disabilities to communicate with people who do not know this language, thus promoting social inclusion. For this purpose, the system is implemented on the NVIDIA Jetson Nano board to perform ASL (American Sign Language) letter detection by means of real-time inference, making use of libraries designed for artificial vision tasks and compatible with the Python programming language, such as OpenCV and MediaPipe, in addition to the Convolutional Neural Network (CNN) “yolov8n. pt” of the YOLOv8 model, which was trained with the ‘Real-Time-Sign-Language’ dataset, which offers several images of each letter of the ASL alphabet. The development process of the present project includes the real-time detection of hand gestures and the reproduction of the translation by means of a speech synthesizer. This work represents a contribution in accessibility and learning of sign language, being a base to be optimized through more complex models in future projects.

Keywords: Computer vision, Convolutional Neural Network (CNN), sign language translation, hearing impairment, prototype, social inclusion.

LISTA DE SIGLAS

CNN. Convolutional Neural Network (Red Neuronal Convolutacional)

IA. Inteligencia Artificial

mAP. Mean Average Precision (Métrica de precisión promedio en detección de objetos)

GPU. Graphics Processing Unit (Unidad de Procesamiento Gráfico)

CPU. Central Processing Unit (Unidad Central de Procesamiento)

TPU. Tensor Processing Unit (Unidad de Procesamiento de Tensores)

IDE. Integrated Development Environment (Entorno de Desarrollo Integrado)

RAM. Random Access Memory (Memoria de Acceso Aleatorio)

SSD. Solid State Drive (Unidad de Estado Sólido)

WIFI. Wireless Fidelity

USB. Universal Serial Bus

ISO. International Organization for Standardization

IEC. International Electrotechnical Commission

IEEE. Institute of Electrical and Electronics Engineers

NIC. Network Interface Card (Tarjeta de Interfaz de Red)

CSI. Camera Serial Interface (Interfaz Serial de Cámara)

ASL. American Sign Language (Lengua de Señas Americana)

LSEC. Lengua de Señas Ecuatoriana

BSL. Lengua de Señas Británica

ÍNDICE DE CONTENIDOS

CAPÍTULO I	23
ANTECEDENTES	23
1.1 Problema	23
1.2 Justificación	25
1.3 Objetivos	28
<i>1.3.1 Objetivo General</i>	28
<i>1.3.2 Objetivos Específicos</i>	28
CAPÍTULO II.....	29
FUNDAMENTACIÓN TEÓRICA	29
2.1 Discapacidad auditiva	29
<i>2.1.1 Clasificación otológica</i>	30
<i>2.1.2 Clasificación de acuerdo al momento de aparición</i>	30
<i>2.1.3 Derechos de personas con discapacidad auditiva</i>	30
2.2 Lenguaje de señas	32
<i>2.2.1 Sistemas de lenguaje de señas a nivel mundial</i>	34
2.3 Inteligencia artificial	35
<i>2.3.1 Visión artificial</i>	36
<i>2.3.2 Etapas de un sistema de visión artificial</i>	36
<i>2.3.3 Campos de aplicación de visión artificial</i>	37
<i>2.3.4 Redes neuronales convolucionales</i>	38
<i>2.3.5 Capa convolucional</i>	39

	13
2.3.6 <i>Capa de Pooling</i>	41
2.3.7 <i>Capa completamente conectada</i>	42
2.4 Hardware y software para visión artificial	43
2.4.1 <i>Hardware para visión artificial</i>	43
2.4.2 <i>Software para visión artificial</i>	46
CAPÍTULO III	48
METODOLOGÍA	48
3.1 Metodología	48
3.1.1 <i>Modelo en Cascada</i>	48
3.2 Análisis	49
3.2.1 <i>Revisión de trabajos relacionados</i>	50
3.2.2 <i>Análisis de trabajos relacionados</i>	51
3.4 Requerimientos	52
3.4.1 <i>Requerimientos del sistema</i>	54
3.4.2 <i>Requerimientos de Hardware y Software</i>	55
3.4.3 <i>Requerimientos del Dataset</i>	57
3.5 Selección de Hardware	58
3.5.1 <i>Placa embebida</i>	58
3.5.2 <i>Cámara para captura de gestos realizados con las manos</i>	59
3.6 Selección de Software	61
3.6.1 <i>Lenguaje de programación</i>	61
3.6.2 <i>CNN preentrenada</i>	62

	14
3.6.3 <i>Plataforma compatible con tareas de entrenamiento de IA</i>	63
3.7 Selección de Dataset	65
3.8 Diseño del sistema	66
3.8.1 <i>Arquitectura del sistema</i>	67
3.8.2 <i>Diagrama de conexiones del prototipo</i>	69
3.8.3 <i>Diagrama de bloques del sistema</i>	71
CAPÍTULO IV	83
EVALUACIÓN Y RESULTADOS	83
4.1 Proceso de entrenamiento	83
4.1.1 <i>Carga de Dataset en Google Colab</i>	83
4.1.2 <i>Preparación de requerimientos para entrenamiento en Google Colab</i> ...	86
4.1.3 <i>Definición de parámetros de entrenamiento</i>	87
4.1.4 <i>Entrenamiento del modelo de visión artificial</i>	88
4.1.5 <i>Adquisición del modelo entrenado</i>	90
4.2 Proceso de detección de gestos	92
4.2.1 <i>Importar librerías</i>	93
4.2.2 <i>Inicialización del procesador de voz</i>	96
4.2.3 <i>Carga de modelo entrenado</i>	97
4.2.4 <i>Lectura de cámara</i>	98
4.2.5 <i>Declarar detector de manos</i>	100
4.2.6 <i>Extracción de información de la mano</i>	101
4.2.7 <i>Recorte de la mano</i>	103

4.2.8	<i>Extracción de resultados del modelo de YOLOv8</i>	104
4.2.9	<i>Visualización de anotaciones de resultados</i>	105
4.2.10	<i>Conversión de resultados a texto</i>	106
4.2.11	<i>Inferencia en tiempo real</i>	108
4.3	Pruebas	111
4.3.1	<i>Pruebas individuales de periféricos</i>	113
4.3.2	<i>Pruebas de validación de detección de mano</i>	121
4.3.3	<i>Pruebas de detección de letras del alfabeto ASL</i>	133
4.3.4	<i>Pruebas de integración del sistema completo en funcionamiento</i>	138
4.4	Limitaciones del sistema	149
4.4.1	<i>Tamaño del objeto delante de la mano</i>	149
4.4.2	<i>Distancia máxima de detección de la mano</i>	151
4.4.3	<i>Detección de la letra “Z”</i>	152
4.5	Análisis Costo/Beneficio	153
4.6	Discusión	159
	CONCLUSIONES	162
	RECOMENDACIONES	164
	REFERENCIAS	166
	ANEXOS	172
	ANEXO A: Revisión de trabajos relacionados	172
	ANEXO B: Análisis del número óptimo de épocas en el entrenamiento del modelo de visión artificial	178

ANEXO C: Imágenes de las pruebas de validación de detección de mano en diferencias escenarios y condiciones	194
ANEXO D: Resultados detallados de las pruebas de detección de letras del alfabeto ASL.....	202
ANEXO E: Resultados individuales de las pruebas de integración con personas con discapacidad auditiva	216
ANEXO F: Resultados individuales de las pruebas de integración con personas sin discapacidad auditiva	249

ÍNDICE DE TABLAS

Tabla 1 Comparación de características técnicas entre Raspberry Pi 3 y NVIDIA Jetson Nano	45
Tabla 2 Ficha bibliográfica de trabajos relacionados a visión artificial aplicada al reconocimiento de gestos con la mano	50
Tabla 3 Acrónimos utilizados en los requerimientos necesarios.....	53
Tabla 4 Prioridades de los requerimientos del prototipo de visión artificial.....	54
Tabla 5 Tabla de requerimientos del sistema	54
Tabla 6 Tabla de requerimientos de Hardware y Software del sistema	55
Tabla 7 Requerimientos del Dataset	57
Tabla 8 Comparación de diferentes placas embebidas	59
Tabla 9 Comparación de diferentes cámaras	60
Tabla 10 Comparación de lenguajes de programación.....	61
Tabla 11 Comparación de CNNs preentrenadas de YOLOv8.....	63
Tabla 12 Comparación de plataformas compatibles con tareas de entrenamiento de IA	64
Tabla 13 Comparación de Datasets	65
Tabla 14 Pruebas realizadas para la validación del sistema de traducción de lenguaje de señas	112
Tabla 15 Resultados de las pruebas individuales de periféricos del sistema.....	121
Tabla 16 Resultados de las pruebas de validación de detección de mano.....	132
Tabla 17 Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 1	135
Tabla 18 Aspectos evaluados en la detección de letras del alfabeto ASL tras 10 repeticiones.....	136

Tabla 19 Tipos de entornos para las pruebas de integración	139
Tabla 20 Características de los participantes en las pruebas de integración	140
Tabla 21 Resultados de precisión de traducción para usuarios con discapacidad auditiva	141
Tabla 22 Resultados de tiempo de respuesta para usuarios con discapacidad auditiva	143
Tabla 23 Resultados de precisión de traducción para usuarios sin discapacidad auditiva	145
Tabla 24 Resultados de tiempo de respuesta para usuarios sin discapacidad auditiva	146
Tabla 25 Comparación de precisión de traducción y tiempo de respuesta entre usuarios con y sin discapacidad auditiva en los cuatro entornos evaluados	148
Tabla 26 Costos de hardware para el sistema de traducción de lenguaje de señas	154
Tabla 27 Costos de software para el sistema de traducción de lenguaje de señas	155
Tabla 28 Costos de ingeniería para el sistema de traducción de lenguaje de señas	157
Tabla 29 Costos totales para el sistema de traducción de lenguaje de señas.....	158

ÍNDICE DE FIGURAS

Figura 1 Abecedario de Lengua de Señas Americana (ASL).....	33
Figura 2 Etapas de un sistema de visión artificial	37
Figura 3 Operación básica de convolución	40
Figura 4 Proceso de obtención de matriz de activación	41
Figura 5 Operación de Max-Pooling	42
Figura 6 Raspberry Pi 3	44
Figura 7 NVIDIA Jetson Nano.....	45
Figura 8 Arquitectura del prototipo de traducción de lenguaje de señas basado en visión artificial.....	69
Figura 9 Diagrama de conexiones correspondiente a la vista superior del prototipo....	70
Figura 10 Diagrama de conexiones correspondiente a la vista frontal del prototipo	71
Figura 11 Diagrama de bloques del sistema.....	73
Figura 12 Diagrama de flujo para el proceso de entrenamiento.....	77
Figura 13 Diagrama de flujo para el proceso de detección de gestos	82
Figura 14 Dataset Real-Time-Sign-Language almacenado en GitHub.....	84
Figura 15 Contenido del archivo “data.yaml” del Dataset Real-Time-Sign-language .	85
Figura 16 Carga de Dataset en Google Colab y modificación de rutas de conjuntos de entrenamiento y validación.....	85
Figura 17 Selección de T4 GPU como acelerador de hardware.....	86
Figura 18 Instalación de Ultralytics en Google Colab para uso de CNNs preentrenadas de YOLOv8	87
Figura 19 Comando para realizar el entrenamiento en Google Colab	89
Figura 20 Información presentada durante la ejecución del comando de entrenamiento	90

Figura 21 Comando para comprimir la carpeta que contiene al modelo entrenado.....	90
Figura 22 Comando para realizar la descarga de la carpeta comprimida al computador	91
Figura 23 Archivo .pt correspondiente al modelo de visión artificial entrenado	92
Figura 24 Instalación librerías en el IDE PyCharm.....	94
Figura 25 Comando para realizar la instalación de la librería Ultralytics	95
Figura 26 Librerías importadas en el código para realizar la inferencia	95
Figura 27 Líneas de código para inicializar el procesador de voz	97
Figura 28 Línea de código para cargar el modelo de visión artificial entrenado	97
Figura 29 Líneas de código para inicializar la captura de video asíncrona en el segundo hilo.....	99
Figura 30 Líneas de código para la implementación del hilo de captura de video	100
Figura 31 Línea de código para declarar el detector de manos	101
Figura 32 Líneas de código para extraer información de la mano	102
Figura 33 Líneas de código para realizar el recorte de la mano detectada.....	104
Figura 34 Líneas de código para la extracción de resultados del modelo de visión artificial.....	104
Figura 35 Líneas de código para visualizar los resultados y traducción de lenguaje de señas	105
Figura 36 Líneas de código para definir la función de conversión de resultados a texto	107
Figura 37 Líneas de código para realizar la conversión de los resultados a texto.....	108
Figura 38 Líneas de código para la finalización del ciclo de inferencia y limpieza de recursos.....	109

Figura 39 Inferencia en tiempo real para la traducción y reproducción de lenguaje de señas	111
Figura 40 Prueba de la cámara mediante el indicador de encendido	114
Figura 41 Prueba de la cámara mediante captura de imagen	115
Figura 42 Prueba de la cámara mediante inferencia de traducción de lenguaje de señas	116
Figura 43 Prueba del monitor mediante detección del periférico en la placa embebida	117
Figura 44 Prueba del monitor mediante inferencia de traducción de lenguaje de señas	118
Figura 45 Prueba de los parlantes mediante sonido de prueba proporcionado por el sistema	119
Figura 46 Sistema de traducción de lenguaje de señas en funcionamiento simultáneo	120
Figura 47 Cambios en el código para dibujar puntos de la mano y cuadro delimitador	123
Figura 48 Pruebas de validación de detección de mano en posición recta.....	123
Figura 49 Pruebas de validación de detección de mano en posición recta y en ambiente oscuro	124
Figura 50 Pruebas de validación de detección de mano en posición recta y fondo con ruido visual	125
Figura 51 Pruebas de validación de detección de mano en posición recta y usuario alejado de la cámara	126
Figura 52 Pruebas de validación de detección de mano en posición con ángulo.....	127
Figura 53 Pruebas de validación de detección de mano parcialmente visible	128

Figura 54 Pruebas de validación de detección de mano cerrada en puño	129
Figura 55 Pruebas de validación de detección de mano en movimiento.....	130
Figura 56 Pruebas de validación de seguimiento de la mano.....	131
Figura 57 Detección de letra B del alfabeto ASL.....	134
Figura 58 Detección de la mano con un objeto de 13 cm delante	150
Figura 59 Falla en la detección de la mano con un objeto de 14 cm delante	150
Figura 60 Detección de la mano a una distancia de 2,55 metros	152
Figura 61 Falla en la detección de la mano a una distancia de 2,56 metros.....	152

CAPÍTULO I

ANTECEDENTES

En este capítulo se hablará de la problemática encontrada en la comunicación de las personas que padecen discapacidad auditiva, con las personas que desconocen el lenguaje de señas, de igual forma se establecerán los objetivos los cuales se buscarán conseguir y ejecutar en la culminación de este proyecto, y por último se detallará la justificación por la cual se realiza este prototipo.

1.1 Problema

Actualmente, en Ecuador, se alberga una población diversa en donde es posible encontrar personas que padecen varios tipos de discapacidades, tales como discapacidades físicas, intelectuales, auditivas, visuales, psicosociales y otras condiciones de salud (Cevallos, 2017). Esta diversidad refleja la realidad de la sociedad y la importancia de abordar las necesidades de este grupo de personas de manera equitativa y justa.

Dentro del país, existe un total de 66538 personas que presentan discapacidad auditiva, las cuales representan el 14,12% de la población ecuatoriana, de dicho porcentaje, 30257 personas corresponden al género femenino, 36272 al género masculino y 9 personas se denominan LGTBI (CONADIS, 2022).

En cuanto a la ciudad de Ibarra, existe un total de 1164 personas las cuales tienen discapacidad auditiva, representando el 18,50% de las personas con incapacidad auditiva (CONADIS, 2022), siendo así el tercer tipo de discapacidad con mayor porcentaje entre las demás discapacidades mencionadas anteriormente, solo por detrás de la física e intelectual, del porcentaje mencionado anteriormente de personas que tienen discapacidad auditiva, 543 corresponden al género femenino y 621 al género masculino. Estas personas

se comunican principalmente a través de lo que perciben visualmente, por lo que su forma natural de expresión es a través de una lengua visual y gestual, como la lengua de señas, en lugar de utilizar un lenguaje oral basado en lo auditivo. (Sánchez y otros, 2019)

En la sociedad actual, la comunicación desempeña un papel fundamental en la vida de las personas, permitiéndoles no solo expresar sus pensamientos y emociones, sino también acceder a información vital y participar plenamente en una amplia gama de interacciones y actividades. Sin embargo, este aspecto esencial de la vida puede resultar extremadamente desafiante para las personas con discapacidad auditiva, quienes a menudo encuentran barreras significativas al intentar comunicarse con aquellos que no están familiarizados con el lenguaje de señas.

La comunicación es un proceso esencial en la sociedad y constituye la base de cualquier organización social. Va más allá de simplemente transmitir mensajes; es una interacción humana entre personas y grupos que permite la formación de identidades y definiciones. Los derechos de comunicación se fundamentan en la idea de un flujo libre de información, siendo un derecho interactivo, igualitario y libre de discriminación. (Zepeda & Prieto, 2007)

Aunque en la actualidad se han presentado grandes avances tecnológicos, existe una brecha significativa en la inclusión de personas que presentan discapacidad auditiva en la sociedad. La falta de herramientas de comunicación efectiva y soluciones tecnológicas accesibles y precisas contribuyen considerablemente a esta problemática, en donde las soluciones tecnológicas existentes, como las aplicaciones de traducción de lenguaje de señas y dispositivos de asistencia auditiva, generalmente cuentan con costos prohibitivos, lo que limita su adopción.

La falta de comunicación puede traer consigo la exclusión social como consecuencia, de tal forma que, cause limitaciones en cuanto a oportunidades educativas y laborales de las personas con discapacidad auditiva. Por esta razón, se ha identificado la necesidad de desarrollar un “Traductor de lenguaje de señas basado en Visión Artificial” la cual permitirá que exista una comunicación eficiente y optima entre las personas que sufren de esta discapacidad con las personas que no conocen el lenguaje de señas.

1.2 Justificación

La necesidad de desarrollar un “Traductor de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial” surge de una realidad innegable, en donde las personas con discapacidad auditiva enfrentan obstáculos y barreras significativas en la comunicación y participación en la sociedad. Debido a la falta de soluciones tecnológicas existentes dentro del país, esta población continúa lidiando con desafíos que limitan su acceso a la educación, al empleo y a sus actividades desarrolladas en su vida cotidiana.

Esta necesidad cobra más relevancia al considerar la importancia del derecho de la comunicación en la vida de cualquier persona, pues la comunicación no es solo una herramienta para compartir o transmitir información, es la base de la expresión de pensamientos, ideologías, emociones y deseos. Es crucial garantizar el acceso universal a los bienes culturales y las condiciones necesarias para su calidad, así como asegurar que todas las personas con discapacidad adquieran las competencias básicas necesarias para continuar con sus estudios, integrarse activamente en la sociedad y ejercer sus derechos y deberes como ciudadanos libres y responsables (Luján, 2020). Por ello las personas con discapacidad auditiva merecen, al igual que cualquier otro ciudadano, la oportunidad de expresarse sin tener que restringirse a las barreras existentes en la sociedad. Sin esta

posibilidad, este grupo de personas se enfrentan a una exclusión social que afecta a su calidad de vida.

El uso del lenguaje de señas por parte de personas con discapacidad auditiva promueve su participación social e inclusión, fomentando su independencia, el respeto por sus derechos y el cumplimiento de sus deberes. Por lo tanto, la lengua de señas se ha convertido en una herramienta valiosa que beneficia y enriquece a niños, niñas y adolescentes en su desarrollo cognitivo, permitiéndoles expresar sus emociones y establecer relaciones satisfactorias. (Franco, 2023)

Este proyecto tiene como objetivo romper las barreras que generalmente limitan las oportunidades para la discapacidad auditiva al relacionarse de manera directa con la visión artificial, promoviendo una sociedad más inclusiva y equitativa con cada una de las personas que la conforman. La labor de una persona con discapacidad es una lucha incesante que supone un escenario tanto espiritual como espacial por el reconocimiento de su autonomía de la inclusión y por la no discriminación en la sociedad (Luján, 2020). La tecnología tiene la capacidad de ser una poderosa herramienta para lograrlo al permitir una comunicación óptima entre las personas que tienen discapacidad auditiva y aquellas que desconocen el lenguaje de señas, lo cual no es únicamente relevante desde una perspectiva de derechos humanos, sino que, adicionalmente contribuye a una sociedad más diversa y enriquecedora.

Generar soluciones basadas en la tecnología para abordar la exclusión social de personas con discapacidad auditiva es de suma importancia. La tecnología proporciona acceso a la información de manera equitativa, facilitando la comunicación efectiva a través de aplicaciones, traducción de lenguaje de señas y redes sociales. Adicionalmente, la tecnología desempeña un papel clave en la inclusión al ofrecer sistemas de transcripción en tiempo real y de forma accesible. Asimismo, la tecnología contribuye a

la sensibilización de la sociedad. En conjunto, estas soluciones tecnológicas promueven la inclusión social y mejoran la calidad de vida de las personas que tienen discapacidad auditiva.

Adicionalmente, el presente proyecto busca abordar una necesidad crítica y contribuye directamente a la construcción de una sociedad equitativa y accesible. Al enfocarse en la adaptabilidad e inclusión, se alinea con los valores fundamentales de igualdad y diversidad, su potencial impacto en la mejora de la calidad de vida de las personas con discapacidad auditiva y su participación activa en la sociedad destacan su importancia y relevancia en el contexto presentado actualmente.

En la ley orgánica de discapacidades, se menciona el artículo número 16, literal 4, en la cual menciona: “El acceso y uso de todas las formas de comunicación visual, auditiva, sensorial y a otras que permitan la inclusión de personas con discapacidad”, con esto podemos determinar que la inclusión para las personas que padecen de este tipo de discapacidad es de suma importancia para el bienestar de quienes sobrellevan esta discapacidad. De igual forma en ley orgánica de discapacidades, en el artículo 64 menciona que: “La autoridad nacional encargada de las telecomunicaciones dictará las normas y regulará la implementación de herramientas humanas, técnicas y tecnológicas necesarias en los medios de comunicación audiovisual para que las personas con discapacidad auditiva ejerzan su derecho de acceso a la información”, con lo anteriormente expuesto se puede evidenciar que el ente regulador para la implementación de herramientas tecnológicas para la ayuda de las personas con discapacidad son profesionales relacionados con la rama de Telecomunicaciones (Giler, 2012).

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un prototipo funcional de traducción de lenguaje de señas basado en visión artificial que permitirá a las personas con discapacidad auditiva comunicarse con personas que desconocen este lenguaje.

1.3.2 Objetivos Específicos

1. Analizar la fundamentación teórica correspondiente a las principales necesidades y desafíos de las personas con discapacidad auditiva al comunicarse con personas que no conocen el lenguaje de señas.
2. Realizar un dataset que contenga gestos de lenguaje de señas, capturados a través de sistemas de detección de manos para proporcionar una base sólida y diversificada de datos.
3. Crear un prototipo de traducción de lenguaje de señas mediante el uso de librerías que son aplicadas para el seguimiento y detección de objetos, las cuales ayudaran para la creación de una relación directa hombre-maquina.
4. Evaluar el funcionamiento del prototipo, valorando la capacidad, la precisión y la velocidad para el reconocimiento de una variedad de gestos de manos.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se presenta la fundamentación teórica, el cual contendrá los conocimientos que se necesitan para el desarrollo del presente proyecto destinado a personas con discapacidad auditiva, en el cual se hablará de las necesidades de las personas que padecen discapacidad auditiva, las bases correspondientes a la lengua de señas, las diversas aplicaciones de la visión artificial, en qué consisten las redes neuronales convolucionales (CNN) y cómo se estructuran, y la infraestructura tecnológica necesaria, explorando tanto hardware como software, destinada la implementación de sistemas de visión artificial.

2.1 Discapacidad auditiva

La discapacidad auditiva se define como la pérdida o anomalía en la función anatómica y/o fisiológica del sistema auditivo, con consecuencias directas en la capacidad de escuchar, esto se traduce en un déficit en el acceso al lenguaje oral (García, 2017). La discapacidad auditiva abarca desde una pérdida superficial hasta una profunda de la capacidad auditiva, siendo imperceptible en términos de características físicas evidentes.

Las personas que padecen discapacidad auditiva dependen en gran medida de la visualidad, lo que se refleja en su preferencia por el lenguaje visual gestual, como la lengua de señas, en lugar del lenguaje oral auditivo. La pérdida significativa de la audición durante las etapas iniciales de la vida puede tener impactos sustanciales en el desarrollo y la adquisición del lenguaje oral en un niño o niña, afectando así su proceso de enseñanza y aprendizaje.

2.1.1 Clasificación otológica

Esta categorización se basa en la localización de la lesión que provoca la pérdida auditiva. A continuación, se proporciona una explicación más detallada de cada categoría:

- **Hipoacusia de conducción o transmisión:** Se trata de una pérdida auditiva originada por una alteración en el oído externo o medio, afectando la parte mecánica del oído y dificultando que el sonido estimule adecuadamente las células del órgano de Corti.
- **Sordera central:** También denominada agnosia auditiva, implica la pérdida de reconocimiento del estímulo auditivo debido a lesiones en las vías auditivas centrales.
- **Sordera mixta:** Se refiere a aquella en la que la patología afecta tanto la vía de conducción del sonido como la de percepción.

2.1.2 Clasificación de acuerdo al momento de aparición

- **Hipoacusia prelocutiva:** Se refiere a la pérdida auditiva que está presente al nacer o que surge antes de que el niño adquiriera el lenguaje, generalmente antes de los 2-3 años de vida. En casos de sorderas graves o profundas, esto impide que el niño aprenda a hablar de manera adecuada.
- **Hipoacusia postlocutiva:** Esta categoría aborda la pérdida auditiva que se manifiesta después de que el individuo ha adquirido el lenguaje. Este tipo de pérdida auditiva puede generar de forma progresiva alteraciones en la pronunciación y entonación, así como afectaciones en la calidad de la voz.

2.1.3 Derechos de personas con discapacidad auditiva

Las personas con discapacidad auditiva tienen derechos reconocidos tanto en la Constitución de la República como en diversos tratados internacionales. Estos documentos no solo se relacionan con la Declaración Universal de los Derechos

Humanos, sino también con el Protocolo de la Convención sobre los Derechos de las Personas con Discapacidad y la Convención Interamericana para la Eliminación de Todas las Formas de Discriminación contra las Personas con Discapacidad (Cevallos-Sánchez, 2017).

Estos derechos también están contemplados en la Constitución de la República del Ecuador y en los Derechos del Buen Vivir. El Artículo 16, numeral 4, establece que este grupo tiene acceso prioritario a la comunicación visual, auditiva, sensorial, y otros medios que faciliten su inclusión en diferentes procesos (Constitución de la República del Ecuador, 2015). La Ley Orgánica de Discapacidades en Ecuador tiene como finalidad asegurar la plena vigencia, difusión y ejercicio de los derechos de las personas con discapacidad, tal como se establece en la Constitución de la República (CONADIS, 2012). A continuación, se presentan los derechos para personas con discapacidad.

- **Derecho a la salud.**

Es responsabilidad del Estado garantizar a las personas con discapacidad el acceso al derecho a la salud, asegurando servicios que incluyan la promoción, prevención, atención especializada continua y prioritaria, así como la habilitación y rehabilitación funcional e integral de salud. Estos servicios deben ser accesibles tanto en instituciones públicas como en privadas que brinden atención médica.

- **Derecho a la educación.**

El Estado se compromete a garantizar que las personas con discapacidad puedan acceder, mantenerse y completar su formación dentro del Sistema Nacional de Educación y el Sistema de Educación Superior. Esto conlleva ofrecerles la oportunidad de recibir educación, formación y/o capacitación, ya sea en instituciones educativas especializadas o en centros de educación regular, según sus necesidades particulares.

- **Derecho al trabajo.**

Las personas con discapacidad tienen el derecho de obtener empleo remunerado en condiciones de equidad y no deben tener discriminación en ninguna etapa de las prácticas laborales. Esto abarca los procesos de aplicación, selección, contratación, formación, compensación de personal y demás condiciones estipuladas tanto en el ámbito público como privado.

- **Derecho de la accesibilidad.**

Se garantizará que las personas con discapacidad puedan acceder y utilizar los bienes y servicios disponibles en la sociedad, eliminando cualquier barrera que impida o dificulte su participación plena e integración social. En cualquier proyecto, tanto público como privado y de acceso general, ya sea en áreas urbanas o rurales, se deberán incluir accesos, vías de circulación, información e instalaciones adecuadas para facilitar la inclusión de personas con discapacidad.

- **Derecho a la seguridad social.**

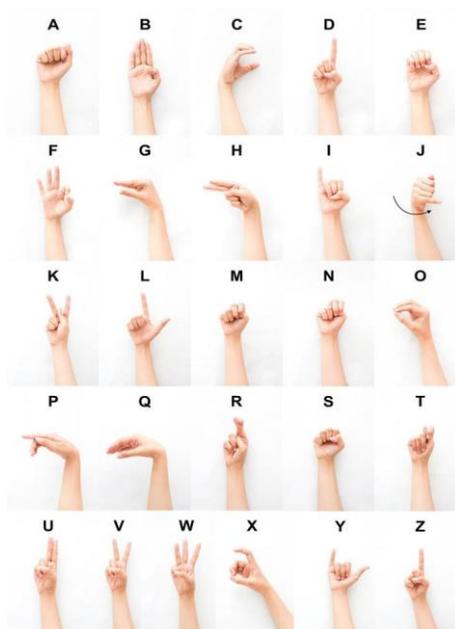
El derecho a la seguridad social es inalienable, y es deber fundamental del Estado asegurar y hacer efectiva su plena aplicación, especialmente para las personas con discapacidad que necesiten atención continua, así como para las personas y familias que se encarguen de su cuidado.

2.2 Lenguaje de señas

El lenguaje de señas, también llamado lengua de signos, es el idioma natural que utiliza gestos espaciales y percepción visual, permitiendo que las personas con discapacidad auditiva se comuniquen dentro de su entorno social. Este sistema de comunicación se basa en movimientos y expresiones realizadas con las manos, ojos, rostro, boca y cuerpo. Las personas con discapacidad auditiva son quienes más utilizan y se benefician de esta forma de comunicación (Peña et al., 2021).

Figura 1

Abecedario de Lengua de Señas Americana (ASL)



Fuente: (Benítez et al., 2021)

La antigüedad de la lengua de señas se remonta en los mismos inicios de la historia humana y continúa siendo utilizada por comunidades de oyentes. Un ejemplo de su aplicación se observa en las comunidades amerindias de las Grandes Llanura en América del Norte, estos grupos empleaban un sistema de señas para facilitar la comunicación entre distintas etnias que hablaban lenguajes diversos, Este método persistió durante un extenso período, incluso después de la llegada de la conquista europea (Ortiz-Farfán & Camargo-Mendoza, 2020).

Abad Charles-Michel de l'Épée, preocupado por la educación de las personas sordas con las que tenía contacto, estableció una escuela para sordos en París en 1755 (J. Varela et al., 2017). Se le atribuye la estandarización de la lengua de signos y la redacción de un libro que abordaba esta lengua y su enfoque educativo para los sordos. En la actualidad, diversas lenguas de señas, originadas en las lenguas francesa, británica, alemana, entre otras, presentan variaciones tanto en el léxico como en la gramática.

2.2.1 *Sistemas de lenguaje de señas a nivel mundial*

Según la Federación Mundial de sordos, alrededor de 70 millones de personas en todo el mundo padecen discapacidad auditiva, con más del 80 por ciento de ellas residiendo en países en desarrollo. Como grupo, estas personas utilizan más de 300 lenguas de señas distintas (ONU, 2021).

- **Lengua de Señas Americana (ASL).**

Como indica Reza et al. (2018), aproximadamente 500,000 personas en Estados Unidos emplean el Lenguaje de Señas Americano (ASL). Considerado como una lengua natural, muchas personas sordas lo aprenden como su primera lengua durante la infancia, proporcionándoles un medio afectivo de comunicación. En contraste con los lenguajes basado en sonidos, las expresiones verbales en ASL se llevan a cabo mediante símbolos manuales conocidos comúnmente como “signos”.

- **Lengua de Señas Británica (BSL).**

El Lenguaje de Señas Británico (BSL) es un sistema de comunicación gestual empleado en el Reino Unido y es la lengua primaria o preferida por la comunidad sorda en ese país. Al igual que en todas las lenguas de señas, la gran mayoría de los gestos manuales no son comprensibles en otros sistemas de señas, ya que la forma de expresar un número o construir oraciones varía entre distintos idiomas de señas. El BSL se caracteriza como un “lenguaje espacial”, dado que implica “mover signos en el espacio” (Albanie et al., 2021).

- **Lengua de Señas Mexicana (LSM).**

Como indica L.-Dellamary (2019), la lengua de Señas Mexicana (LSM) es el sistema de comunicación utilizado por las personas con discapacidad auditiva en México. Está compuesta por signos visuales con una estructura lingüística única. Aunque la LSM no guarda relación con otros idiomas en México, es similar a la Lengua de Señas

Francesa, mostrando ciertas similitudes con la Lengua de Señas Americana y otras lenguas influenciadas.

- **Lengua de Señas Ecuatoriana (LSEC).**

La lengua de señas empleada en Ecuador, conocida como LSEC, es el medio de comunicación predominante para la comunidad sorda en el país. Debido a un prolongado intercambio entre la comunidad sorda ecuatoriana, instructores y personas de otras nacionalidades, cerca del 30% del vocabulario de la LSEC proviene de la Lengua de Señas Americana, mientras que aproximadamente el 20% tiene origen en la Lengua de Señas Española. (Ureta-Andrade & Donoso-Cedeño, 2022).

2.3 Inteligencia artificial

Según Rouhiainen (2018), la inteligencia artificial (IA) se define como la capacidad de las máquinas para utilizar algoritmos, aprender de los datos y aplicar ese conocimiento en la toma de decisiones, imitando el proceso humano. A diferencia de los seres humanos, los dispositivos con IA no necesitan descanso y pueden procesar grandes volúmenes de información de manera simultánea, con una proporción de errores mucho menor al realizar tareas similares a las que realizan las personas.

La IA tiene el potencial de proporcionar sugerencias y predicciones sobre aspectos significativos de la vida cotidiana, impactando en áreas como la salud, el bienestar, la educación, el empleo y las relaciones interpersonales. Esta tecnología cambiará la dinámica empresarial al ofrecer ventajas competitivas a aquellas empresas que adopten y apliquen estas herramientas de manera rápida y eficiente. Otro beneficio de la IA es que permitirá que máquinas y robots realicen tareas que los humanos consideren difíciles, monótonas o peligrosas (Estupiñán et al., 2021).

2.3.1 *Visión artificial*

La visión artificial, también denominada visión por computador, es una rama dentro de la inteligencia artificial. Su propósito central es programar una computadora para que interprete la información contenida en una imagen (Horna & Manzano Ramos, 2021).

La finalidad principal de la visión artificial es capacitar a una computadora para simular la visión humana, permitiéndole extraer información relevante de las imágenes mediante un proceso de entrenamiento previo que le posibilite responder a preguntas planteadas con anterioridad. (Kakani et al., 2020).

2.3.2 *Etapas de un sistema de visión artificial*

En el desarrollo de la solución de un problema específico mediante el empleo de visión artificial, se llevan a cabo cuatro etapas esenciales. Estas fases están diseñadas para maximizar la eficiencia en el uso de tiempo y recursos (Tinajero, 2020b). A continuación, se detallan cada una de las fases que integran este proceso.

1. **Captura:** En esta etapa se obtiene la información, que corresponde a las imágenes capturadas.
2. **Procesamiento:** Mediante la aplicación de diversos algoritmos, se identifican las áreas de interés y se eliminan segmentos no deseados de las imágenes, facilitando las etapas siguientes del proceso.
3. **Segmentación:** Se lleva a cabo el aislamiento de los elementos de interés del resto de la imagen.
4. **Reconocimiento:** A través de un análisis previo, se compara las características extraídas con las generadas en ese momento, estableciendo así las diferencias entre objetos.

Figura 2

Etapas de un sistema de visión artificial



Fuente: Elaboración propia

Nota. El gráfico representa las cuatro etapas fundamentales de un sistema de visión artificial.

2.3.3 Campos de aplicación de visión artificial

Borella (2022), indica que los beneficios y aplicaciones de la visión artificial son variados, pero destacan cinco áreas de suma importancia que mantendrán su relevancia en el ámbito de la inteligencia artificial.

- **Medicina:** La visión artificial está mejorando significativamente la capacidad de procesamiento de imágenes en dispositivos médicos como tomógrafos y equipos de resonancia magnética, integrándolos con bases de datos para la generación eficiente de informes médicos, permitiendo agilizar los diagnósticos y mejorando la atención al paciente con mayor rapidez y precisión.
- **Seguridad:** Los sistemas de reconocimiento facial son comúnmente utilizados en empresas que manejan información sensible y en entidades bancarias para verificar la identidad del usuario durante transacciones poco comunes.
- **Industrial:** Es aplicada en la comparación de imágenes de objetos manufacturados con modelos previos, la visión artificial automatiza el proceso de

comparación de repuestos y, por ende, puede automatizar los procedimientos de control de calidad.

- **Educación:** La visión artificial se aplica en la creación de proyectos con fines metodológicos para aquellos interesados en especializarse en el área. También se utiliza en el desarrollo de programas educativos que benefician a los niños, proporcionando retroalimentación para obtener beneficios cognitivos.

2.3.4 *Redes neuronales convolucionales*

Las redes neuronales convolucionales (CNN) buscan replicar la estructura y el funcionamiento de las redes neuronales del cerebro humano. Están compuestas por nodos interconectados que reciben, procesan y transmiten información a través de otras neuronas artificiales, las cuales son programables para optimizar la salida, adaptándose así a diversos tipos de entradas (Aguilar Alvarado & Campoverde Molina, 2020).

Las CNN pueden estar compuestas por varias decenas o incluso cientos de capas, donde cada capa tiene la tarea de aprender y reconocer diferentes características de una imagen. A lo largo del entrenamiento, se aplican filtros a las imágenes a diversas resoluciones, y la salida generada por la convolución de cada imagen se emplea como entrada para la siguiente capa. En otras palabras, las capas iniciales se especializan en detectar propiedades o formas fundamentales, y a medida que se avanza en las capas, se logra la capacidad de reconocer formas más complejas, como un rostro o una silueta.

Artola (2019), indica que las redes neuronales convolucionales están diseñadas para la identificación de objetos en imágenes. Inicialmente, el aprendizaje automático se basaba en redes neuronales simples, las cuales eran adecuadas para discernir los elementos de imágenes más pequeñas. No obstante, con el tiempo, estos algoritmos demostraron ser limitados al enfrentarse a la detección de objetos en imágenes de mayor tamaño. Estas redes se estructuran a partir de tres tipos de capas.

2.3.5 Capa convolucional

La capa convolucional desempeña un papel crucial, en la práctica, los parámetros de una capa convolucional se refieren a un conjunto de filtros entrenables. Cada filtro ocupa una región pequeña a lo largo de las dimensiones de ancho y alto, pero abarca toda la profundidad del volumen de entrada. Durante la propagación directa o hacia adelante, cada filtro a lo largo de las dimensiones de ancho y alto del volumen de entrada genera un mapa de activación bidimensional específico para ese filtro. A medida que el filtro se desplaza a lo largo del área de la entrada, se realiza un producto escalar entre los valores del filtro y los de la región de entrada a la que se aplica (Durán Suárez et al., 2021).

Es decir, la capa convolucional desempeña un doble papel al disminuir la cantidad de elementos que constituyen la red y al identificar características específicas que resultarán valiosas para el análisis de la imagen, por ello, la presente capa resulta sumamente beneficiosa en el análisis de imágenes al simplificar la complejidad del sistema y, al mismo tiempo, extraer rasgos significativos que facilitan su análisis.

La convolución de las matrices I y K , con dimensiones n , m y k , de dimensiones i , j , generalmente denotada por el símbolo de asterisco, se expresa de la siguiente forma tal y como puede representarse a continuación en la Ecuación 1:

Ecuación 1

Convolución de dos matrices.

$$[I * k](i, j) = S(i, j) = \sum_n \sum_m I(n, m) \cdot k(i - n, j - m)$$

Donde:

- S : Matriz de salida.
- I : Matriz de entrada.

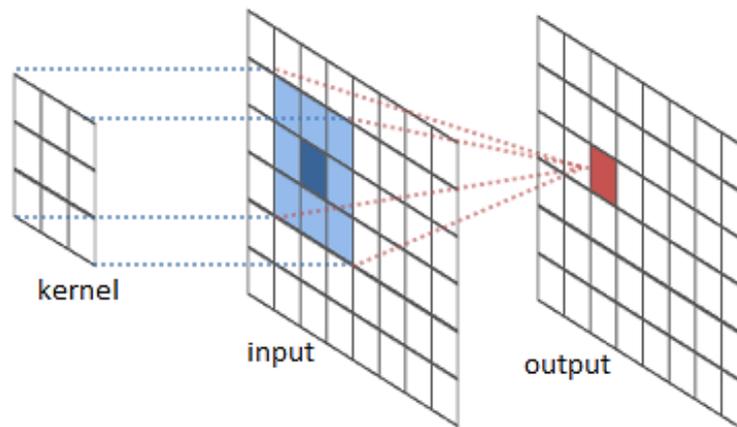
- k: Kernel de convolución.

La ecuación tiene como propósito mostrar cómo se invierte un kernel de convolución y cómo se desplaza a través de los elementos de la matriz de entrada, indicada por la letra I, mientras se realiza una multiplicación punto por punto entre los elementos.

Cuando la red procesa la imagen de entrada, el filtro se superpone sobre la imagen y se efectúa una convolución bidimensional entre los elementos de la imagen y el kernel. El resultado de esta operación se almacena en una posición específica dentro de la matriz de activación, como se presenta en la Figura 3.

Figura 3

Operación básica de convolución



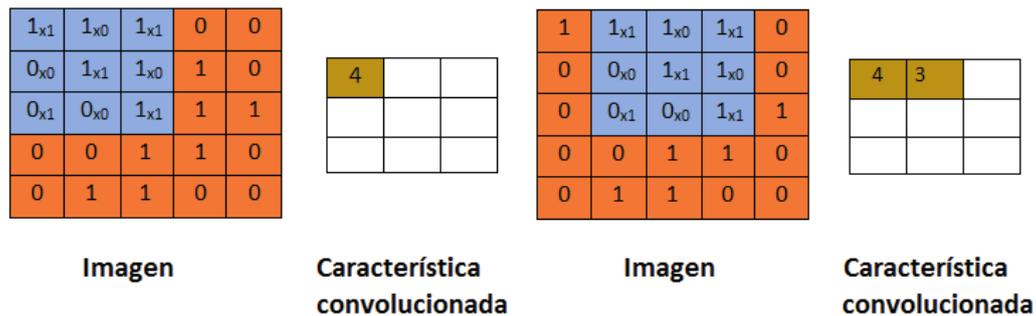
Fuente: (Massiris et al., 2020)

Luego, el filtro se desplaza una posición hacia la derecha sobre la imagen y se realiza otra vez la convolución, guardando el resultado en la posición siguiente de la matriz de activación. Este proceso iterativo continúa a lo largo de la imagen, moviéndose de izquierda a derecha y descendiendo una unidad al llegar al borde. Al completar el recorrido por toda la imagen, se genera la matriz de activación completa, la cual contiene

las características identificadas en la imagen para cada filtro. Este procedimiento se ilustra con mayor claridad en la Figura 4.

Figura 4

Proceso de obtención de matriz de activación

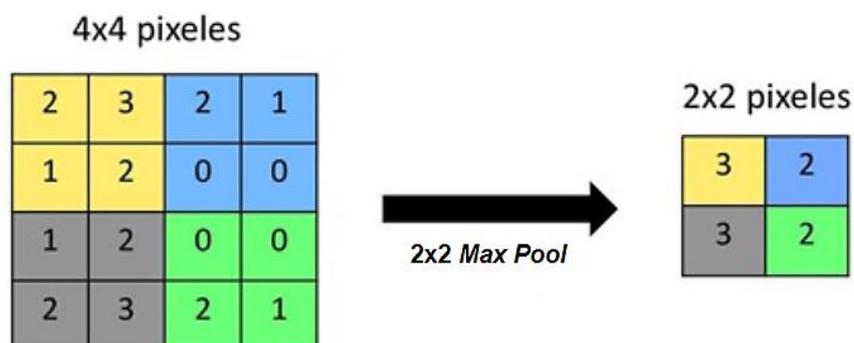


Fuente: (Carrión, 2020)

2.3.6 Capa de Pooling

El objetivo principal de la capa de Pooling es reunir una amplia variedad de características de los mapas generados cuando una imagen pasa por un filtro de convolución. En esencia, su función es reducir de manera constante las dimensiones de la imagen, lo que lleva a una disminución en la cantidad de parámetros y reduce la carga computacional. Esta reducción tiene como fin evitar que la red neuronal se sobreajuste (Badillo et al., 2021).

Según Mera-Jiménez & Ochoa-Gómez (2021), la forma más utilizada de Pooling es el “Max-Pooling”. En este método, se aplica un filtro “Max” a una subregión de la imagen donde ya se ha creado la matriz de píxeles, identificando el píxel con el valor máximo en esa área. Este proceso se ilustra claramente en la Figura 5.

Figura 5*Operación de Max-Pooling*

Fuente: (Durán Suárez et al., 2021)

2.3.7 Capa completamente conectada

También llamada “Full connected layer”, presenta una estructura idéntica a las capas convencionales de una red neuronal, pero con la distinción de que su arquitectura está completamente interconectada. Cada neurona de esta capa establece conexiones con todas las neuronas de la capa anterior, específicamente, con todos los mapas de características de dicha capa (Andrade Carrera et al., 2021).

La función principal de las capas completamente conectadas es realizar una especie de consolidación de la información adquirida hasta ese momento, que será utilizada en cálculos posteriores para la clasificación definitiva.

Según Gutiérrez Leguizamón et al. (2022), a diferencia de las capas convolucionales en las CNN, esta capa no hace uso de la propiedad de conectividad local. En lugar de ello, una capa completamente conectada se vincula con la totalidad del volumen de entrada, generando un elevado número de conexiones. En contraste, las capas convolucionales se conectan con una región local única en la entrada, y varias neuronas comparten parámetros en estas capas convolucionales.

2.4 Hardware y software para visión artificial

Los sistemas de visión artificial son sistemas que dependen de componentes computacionales y de captura para llevar a cabo sus funciones. Estos sistemas requieren hardware que facilite la adquisición de imágenes, así como software que permita programar tareas específicas para un determinado tipo de sistema de visión artificial, junto con librerías que asistan en el procesamiento de imágenes (Pérez et al., 2021).

2.4.1 Hardware para visión artificial

Pachón Espinel (2019) menciona que, los elementos clave del hardware en un sistema de visión artificial incluyen la iluminación, los lentes, el sensor de imagen, el procesador de visión y los sistemas de comunicación. La función de la iluminación es destacar las características del objeto que se va a analizar, facilitando que la cámara las capte de manera clara. El lente recoge la imagen y la transmite al sensor en forma de luz. Este sensor, situado en una cámara de visión artificial, transforma la luz en una imagen digital que luego es enviada al procesador para su análisis. El procesador de visión artificial utiliza algoritmos para gestionar la imagen, extraer la información necesaria, realizar la inspección correspondiente y tomar decisiones.

La Raspberry Pi 3 se considera esencial para la implementación de sistemas de visión artificial económicos y portátiles, debido a su capacidad de procesamiento y velocidad, comparables a las de una computadora personal. Desde una perspectiva técnica, cuenta con un procesador ARM Cortex A53 de cuatro núcleos a 1.2 GHz y 64 bits, que ofrece un rendimiento diez veces superior al de la Raspberry Pi original y un 50% más que la Raspberry Pi 2. Esta versión incluye cuatro puertos USB, 40 pines GPIO, un puerto HDMI, un puerto Ethernet, un conector de audio y video compuesto, interfaces para cámara y pantalla, ranura para tarjetas microSD y un núcleo gráfico VideoCore IV

3D. La Raspberry Pi 3 mantiene un factor de forma idéntico a la Pi 2 y es totalmente compatible con la Raspberry Pi 1 y la Raspberry Pi 2 (Tinajero, 2020).

Figura 6

Raspberry Pi 3



Fuente: (Duque, 2021)

Por otra parte, la plataforma NVIDIA Jetson Nano representa una oportunidad para explorar el campo de la inteligencia artificial. Se trata de una computadora compacta que posibilita la ejecución simultánea de diversas redes neuronales, facilitando tareas como la clasificación de imágenes o la detección de objetos. Su atractivo radica en su costo accesible, además, la Jetson Nano está equipada con una GPU que posibilita la ejecución de aplicaciones de visión artificial, generando un interés considerable en cuanto a su rendimiento (Varela, 2021).

En cuanto a sus características técnicas, la NVIDIA Jetson Nano cuenta con GPU de 128 núcleos, un procesador ARM Cortex A57 de cuatro núcleos, memoria LPDDR4 de 4GB y 64 bits, puerto para tarjeta microSD y codificador de video de 4K a 30 cuadros, puerto HDMI, puerto Ethernet y conexiones GPIO de 40 pines (Amaya, 2018).

Figura 7*NVIDIA Jetson Nano*

Fuente: (Coanqui Apaza, 2022)

A continuación, en la Tabla 1 se presenta una comparación de las características técnicas relevantes de la Raspberry Pi 3 y la NVIDIA Jetson Nano, centrándose en sus capacidades para aplicaciones de visión artificial.

Tabla 1

Comparación de características técnicas entre Raspberry Pi 3 y NVIDIA Jetson Nano

Características	Raspberry Pi 3	NVIDIA Jetson Nano
Procesador	ARM Cortex A53	ARM Cortex A57
GPU	VideoCore IV	128-core Maxwell GPU
Memoria RAM	1GB LPDDR2	4GB LPDDR4
Almacenamiento	Ranura microSD	Ranura microSD
USB	4 puertos USB 2.0	4 puertos USB 3.0
Conectividad	Ethernet, Wi-Fi, Bluetooth	Ethernet, Wi-Fi
GPIO	40 pines GPIO	40 pines GPIO

Interfaz de cámara	CSI (Interfaz de cámara serie)	MIPI CSI-2
Salida de video	HDMI	HDMI
Rendimiento GPU	Básico para tareas ligeras	Potente para tareas intensivas en GPU
Consumo de energía	2.5W	5-10W

Fuente: Elaboración propia con datos tomados de Tinajero (2020)

Ambas placas tienen capacidades para ejecutar tareas de visión artificial, pero la NVIDIA Jetson Nano está especialmente diseñada para cargas de trabajo más intensivas en GPU y tareas de aprendizaje profundo. Su GPU Maxwell de 128 núcleos proporciona un rendimiento significativamente superior en comparación con la GPU VideoCore IV de la Raspberry Pi 3. La Jetson Nano también tiene una mayor cantidad de memoria RAM (4GB LPDDR4) en comparación con la Raspberry Pi 3 (1 GB LPDDR3), lo cual es beneficioso para aplicaciones más exigentes en términos de memoria.

2.4.2 Software para visión artificial

El software especializado en visión artificial emplea algoritmos y modelos matemáticos para analizar y procesar imágenes o videos, con el propósito de imitar la habilidad humana de percibir y comprender el entorno. Estos programas pueden llevar a cabo una variedad de tareas, como reconocimiento de objetos, detección de rostros, seguimiento de movimientos y clasificación de imágenes. Para alcanzar estos objetivos, se aplican técnicas como el aprendizaje profundo, redes neuronales convolucionales y otros métodos avanzados (Alvear, 2017).

Fresno (2020), Indica que existen diversas librerías de código abierto para el desarrollo de visión artificial, a continuación, se presentan tres de ellas que son completas y adecuadas para una variedad de propósitos.

- **OpenCV:** Es la librería más reconocida y abarca más de 500 algoritmos. Cuenta con extensa documentación y está disponible para sistemas operativos como Linux, Android y Windows.
- **Sherlock:** Se trata de una librería que se basa en una interfaz gráfica para simplificar la programación. Las herramientas integradas hacen que sea útil para desarrollar aplicaciones sin requerir de un conocimiento profundo de visión artificial.
- **Halcon:** Esta librería está diseñada para el desarrollo de aplicaciones industriales de visión artificial y es creada y comercializada por MVTec. Su complejidad se sitúa en un nivel medio-alto.

Para desarrollar software de visión artificial, se pueden emplear diversos lenguajes de programación, entre los cuales destacan C++, Java y Python, cada uno con características particulares presentadas a continuación.

- **C++:** Es un lenguaje compilado y orientado a objetos que cuenta con una amplia variedad de librerías, lo que facilita la labor del programador.
- **Java:** Ampliamente utilizado, es un lenguaje de programación que puede funcionar en diversas plataformas. Sin embargo, carece de una librería de visión artificial suficientemente aprobada y funcional.
- **Python:** Este lenguaje es fácilmente portable a otras plataformas, posee una extensa biblioteca estándar y es conocido por su sintaxis sencilla, que es limpia y se asemeja al inglés.

CAPÍTULO III

METODOLOGÍA

En el presente capítulo, se detalla el enfoque metodológico para el desarrollo del prototipo, así como el análisis de los requerimientos necesarios para su implementación. Se discute el proceso de recolección de información del lenguaje de señas, la determinación de la cantidad de muestras requeridas y la metodología para el entrenamiento y modelado del sistema, con el fin de lograr el reconocimiento y la traducción de las letras del Alfabeto del Lenguaje de Señas Americano (ASL) en tiempo real. Se destaca la adopción de un enfoque secuencial en Cascada para desarrollar el proyecto, que se abordará en detalle en este capítulo.

3.1 Metodología

El propósito principal de este proyecto es desarrollar un sistema de visión artificial que pueda traducir en tiempo real el Alfabeto del Lenguaje de Señas Americano (ASL). Se opta por emplear el modelo secuencial en Cascada, previamente mencionado, para guiar tanto el diseño como la implementación del prototipo. Además, se llevará a cabo un análisis para seleccionar el hardware y software adecuados que permitan un funcionamiento óptimo del sistema de visión artificial en tiempo real.

3.1.1 *Modelo en Cascada*

El enfoque metodológico del modelo en cascada implica organizar de manera secuencial las diferentes fases a seguir, lo que lo hace adecuado tanto para el desarrollo de software como de hardware (Delgado Olivera et al., 2021). Por lo tanto, este modelo secuencial se seleccionó para la implementación del sistema de visión artificial en tiempo real en este prototipo. A continuación, se describirán las etapas que se seguirán en el desarrollo del presente proyecto:

- **Análisis:** Durante esta etapa se llevará a cabo un análisis de los requisitos de software y hardware necesarios para el desarrollo del prototipo y sistema de visión artificial. En este análisis, se desglosarán los problemas complejos en tareas secuenciales para una comprensión detallada de los requisitos.
- **Diseño:** En la etapa de Diseño, es necesario enfocarse en la creación de una solución que cumpla con las necesidades y estrategias identificadas durante la etapa de análisis. Por lo tanto, es crucial desarrollar la arquitectura del sistema y elaborar un plan de diseño detallado.
- **Implementación:** Durante la etapa de Implementación, se lleva a cabo la ejecución de la arquitectura diseñada para el sistema de visión artificial. Esto implica acciones como la adquisición del Dataset, la instalación de librerías y complementos necesarios, la programación del software y el entrenamiento del modelo de visión artificial. El objetivo principal en esta etapa es lograr un funcionamiento óptimo en conjunto de las partes involucradas en la presente fase.
- **Pruebas:** Esta etapa se enfoca en la extracción de resultados al poner en funcionamiento el prototipo que se fundamenta en visión artificial, de tal forma que permita determinar si se ha podido desarrollar una solución óptima y efectiva a las necesidades y problemáticas mencionadas en el presente proyecto.
- **Entrega:** Esta es la etapa final, en donde una vez se haya podido conseguir óptimos resultados en cuanto al funcionamiento del sistema de visión artificial se concluirá el proyecto.

3.2 Análisis

El presente punto busca analizar los requerimientos esenciales del prototipo en términos de hardware, software y Dataset para garantizar un rendimiento óptimo. Esto implica la identificación de librerías y complementos necesarios, así como la selección

de la placa embebida destinada a ejecutar tareas de visión artificial y la cámara digital para capturar los gestos de señas realizados a través de la mano del usuario. Se empleará una técnica de investigación la cual consiste en la revisión de trabajos relacionados al tema del presente proyecto.

3.2.1 Revisión de trabajos relacionados

La revisión de trabajos relacionados con el tema del presente proyecto está enfocada en la identificación de requerimientos necesarios para el diseño y la implementación del traductor de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial. Se ha utilizado una metodología de revisión de trabajos de investigación, a partir de la cual se ha recopilado información relevante mediante fichas bibliográficas, como se muestra en el ANEXO A. En la Tabla 2, se presenta una ficha bibliográfica básica de los cinco proyectos tomados para esta técnica de investigación.

Tabla 2

Ficha bibliográfica de trabajos relacionados a visión artificial aplicada al reconocimiento de gestos con la mano

Ficha bibliográfica 1	
Autor: Alarcón Nelson.	Editorial: Universidad de Guayaquil.
Título: “Traductor de Lenguaje de señas basado en Visión artificial para personas con discapacidad auditiva”.	
Año: 2022.	Ciudad: Guayaquil.
Ficha bibliográfica 2	

Autores: Pigou Lionel, Dieleman Snder, **Editorial:** Ghent University.

Kinde rmans, Pieter, Schrauwen,

Bnejamin.

Título: “Sign Language Recognition Using Convolutional Neural Networks”.

Año: 2015.

Ciudad: Ghent.

Ficha bibliográfica 3

Autor: Luis Sivisapa.

Editorial: Universidad de las Fuerzas Armadas.

Título: “Visión Artificial aplicada para el reconocimiento del lenguaje de señas”.

Año: 2014

Ciudad: Loja

Ficha bibliográfica 4

Autor: Lucía Cheung, Carlos Medina.

Editorial: UTP-Ridda2.

Título: “Implementación y Análisis de un Detector de Manos Basado en Visión Artificial”.

Año: 2013.

Ciudad: Ciudad de Panamá.

Ficha bibliográfica 5

Autor: Valeria Andrade.

Editorial: Universidad Nacional de Chimborazo.

Título: “Diseño e implementación de un sistema traductor de lengua de señas mediante inteligencia artificial para personas con discapacidad auditiva”.

Año: 2022.

Ciudad: Riobamba.

3.2.2 *Análisis de trabajos relacionados*

- En el trabajo relacionado 1, se llevó a cabo el desarrollo del prototipo en una NVIDIA Jetson Nano, que ofrece un beneficio significativo al contar con una

tarjeta gráfica. Adicionalmente se indica la utilización de Python para el desarrollo del software, específicamente la versión 3.8, debido a la compatibilidad presentada con las librerías utilizadas en el desarrollo del presente trabajo.

- En el trabajo relacionado 2, el entrenamiento del modelo de visión artificial fue realizado en la plataforma virtual Google Colab, en donde se seleccionó la opción de trabajar con los recursos computacionales de GPU y TPU ofrecidos. Adicionalmente el procesamiento de las imágenes es estandarizado con un tamaño de 640x640 pixeles debido a su compatibilidad con el modelo de IA escogido.
- En el trabajo relacionado 3, el sistema de visión artificial es desarrollado en un entorno de distribución Linux, además la principal librería de visión artificial empleada es OpenCV, ya que es compatible con Python, el lenguaje de programación utilizado en el proyecto mencionado, y presenta facilidades enfocadas al desarrollo de visión artificial.
- En el trabajo relacionado 4, se realiza una partición del Dataset en dos conjuntos: el grupo de entrenamiento, destinado a entrenar el modelo de visión artificial, y el grupo de validación, diseñado para evaluar el rendimiento del modelo durante el proceso de entrenamiento. En dicha partición se considera que alrededor del 80% de las muestras sean almacenadas en el grupo de entrenamiento y que alrededor del 20% se almacenen en el grupo de validación.
- En el trabajo relacionado 5, se emplea el IDE PyCharm para la programación en Python. Adicionalmente se utiliza Tensorflow para el proceso de entrenamiento de la red neuronal.

3.4 Requerimientos

En esta sección se definen los requerimientos necesarios, tanto de hardware y software como del Dataset, correspondientes al traductor de lenguaje de señas para

personas con discapacidad auditiva basado en visión artificial. Este apartado tiene como objetivo establecer los elementos funcionales para conseguir un funcionamiento óptimo de todo el sistema.

El establecimiento de los requerimientos necesarios para desarrollar este proyecto se basa en el estándar ISO/IEC/IEEE 29148:2018, el cual describe los procesos empleados en las actividades de ingeniería que derivan en los requisitos para sistemas y productos de software a lo largo de su ciclo de vida. Además, proporciona directrices sobre el formato de los elementos de información requeridos y relacionados.

A continuación, en la Tabla 3 se presentan los acrónimos utilizados para facilitar la comprensión y gestión de los términos relacionados con los requisitos. Esta tabla abarca los requerimientos tanto de hardware y software, así como los requerimientos del Dataset.

Tabla 3

Acrónimos utilizados en los requerimientos necesarios

Acrónimo	Descripción
SiRS	Requerimientos del sistema
SHRS	Requerimientos de Hardware y Software
SdRS	Requerimientos del Dataset

En la Tabla 4 se indican y describen las prioridades de los requerimientos, que se utilizarán en las comparaciones de los requerimientos del prototipo de visión artificial. Dichas prioridades corresponden a alta, media y baja.

Tabla 4*Prioridades de los requerimientos del prototipo de visión artificial*

Prioridad	Descripción
Alta	Se refiere a un requerimiento esencial para el desarrollo y funcionamiento del sistema, su ausencia tendría un impacto significativo en la eficiencia y efectividad del mismo.
Media	Requerimiento que puede ser excluido solo en circunstancias de fuerza mayor, aunque su omisión podría afectar parcialmente a la funcionalidad y rendimiento del sistema.
Baja	Requerimientos que pueden ser ignorados, su ausencia no tendría un impacto significativo en el funcionamiento general del sistema.

3.4.1 Requerimientos del sistema

En esta sección se definen los ítems correspondientes a los requerimientos del sistema, los cuales han sido analizados a través de la técnica de revisión de trabajos de investigación del ANEXO A presentado. En la tabla 5 se detallan los requerimientos descritos y sus etiquetas de prioridad.

Tabla 5*Tabla de requerimientos del sistema*

Requerimientos del Sistema (SiRS)				
N°	Requerimiento	Prioridad		
		Alta	Media	Baja
Requerimientos funcionales				
SiRS1	Detectar la letra del ASL realizada con la mano	X		
SiRS2	Realizar la inferencia en tiempo real	X		

SiRS3	Mostrar en pantalla la precisión de detección	X
SiRS4	Visualizar el seguimiento de la mano	X
Requerimientos de uso		
SiRS5	Adaptabilidad a diferentes condiciones de ambiente	X
SiRS6	Portabilidad para usarse en distintas ubicaciones	X
SiRS7	Facilidad de manejo y entendimiento	X
Requerimientos de interfaces		
SiRS8	Interfaces USB para la conexión de periféricos	X
SiRS9	Interfaz Ethernet para permitir conexión alámbrica a internet	X
SiRS10	Interfaz HDMI para conectar el sistema a un monitor	X
SiRS11	Interfaz para adaptar tarjeta microSD	X
Requerimientos físicos		
SiRS12	Sistema de disipación térmica	X

3.4.2 *Requerimientos de Hardware y Software*

En la presente sección se detallan todos los requerimientos necesarios de Hardware y Software del sistema. En la Tabla 6 se detallan estos requerimientos y sus respectivas etiquetas de prioridad.

Tabla 6

Tabla de requerimientos de Hardware y Software del sistema

Requerimientos de Hardware y Software (SHRS)		
N°	Requerimiento	Prioridad
		Alta Media Baja

Requerimientos lógicos

SHRS1	Compatibilidad con tecnologías de conexión a internet	X	
SHRS2	Compatibilidad con sistemas operativos Linux	X	
SHRS3	Integración con diversos periféricos		X
SHRS4	Disponibilidad de soporte y actualizaciones	X	
SHRS5	Compatibilidad con lenguajes de programación de alto nivel	X	

Requerimientos de Hardware

SHRS6	Placa embebida	X	
SHRS7	Cámara USB mínimo resolución HD	X	
SHRS8	Monitor o pantalla digital	X	
SHRS9	Conexión a internet alámbrica o inalámbrica		X
SHRS10	Tarjeta de memoria microSD mínimo 64 Gb	X	
SHRS11	Case compatible con la placa embebida		X
SHRS12	Fuente de alimentación eléctrica	X	
SHRS13	Ranura para tarjeta microSD		X
SHRS14	Memoria RAM mínimo 4GB		X
SHRS15	Rendimiento de CPU mayor o igual a 13,5 Gflop		X

Requerimientos de Software

SHRS16	Licencia de libre uso		X
SHRS17	Compatibilidad con Python	X	
SHRS18	IDE de programación		X

SHRS19	Compatibilidad con librerías de visión artificial	X
SHRS20	Herramientas de carga de sistema operativo	X
SHRS21	Uso de modelos preentrenados	X
SHRS22	Complementos para optimización de recursos	X
SHRS23	Compatibilidad con biblioteca PyTorch	X
SHRS24	Plataforma alojada en la nube compatible con tareas de entrenamiento de IA	X
Requerimientos de CNN preentrenada		
SHRS25	Bajo consumo de recursos computacionales	X
SHRS26	Buena precisión de detección de CNN preentrenada	X
SHRS27	Inferencia en tiempo real fluida	X

3.4.3 *Requerimientos del Dataset*

En la presente sección se detallan todos los requerimientos necesarios del Dataset. Al ser un conjunto de muestras utilizadas para entrenar el modelo de visión artificial, debe tener ciertos requerimientos para obtener resultados óptimos de entrenamiento y detectar gestos realizados con las manos para cada letra del alfabeto ASL. En la Tabla 7 se detallan cada uno de estos requerimientos y sus respectivas etiquetas de prioridad.

Tabla 7

Requerimientos del Dataset

Requerimientos del Dataset (SdRS)				
N°	Requerimiento	Prioridad		
		Alta	Media	Baja
SdRS1	Imágenes capturadas en diferentes ambientes	X		

SdRS2	Variedad de gestos y poses	X	
SdRS3	Estandarización de un tamaño de pixeles específico		X
SdRS4	Particionamiento del dataset en conjunto de entrenamiento y conjunto de validación	X	
SdRS5	Distribución y particionamiento adecuado de conjuntos	X	
SdRS6	Asignación de clases a cada letra del alfabeto ASL	X	
SdRS7	Disponer de etiquetas de posición del objeto a detectar para cada imagen	X	
SdRS8	Archivo que almacene las clases del Dataset		X

3.5 Selección de Hardware

En este apartado, se selecciona el hardware específico mediante una ponderación de los requerimientos presentados anteriormente. A continuación, se expone la selección de la placa embebida que mejor cumpla estos criterios, junto con la cámara digital necesaria para capturar los gestos realizados con las manos.

3.5.1 Placa embebida

La placa embebida debe contar con la capacidad de abarcar la cantidad de procesamiento que conllevan las tareas de visión artificial. Por esta razón, para lograr una selección adecuada, en la Tabla 8 se realiza una comparación de diferentes placas embebidas considerando los requerimientos presentados anteriormente.

Tabla 8*Comparación de diferentes placas embebidas*

Requisitos	Placa embebida			
	NVIDIA Jetson Nano	Raspberry Pi 3	Intel Movidius Myriad X	Google Coral Dev Board
SHRS2	SI	SI	SI	SI
SHRS3	SI	SI	SI	SI
SHRS4	SI	SI	NO	SI
SHRS5	SI	SI	SI	SI
SHRS6	SI	SI	SI	SI
SHRS9	SI	SI	NO	NO
SHRS13	SI	SI	SI	SI
SHRS14	SI	NO	NO	NO
SHRS15	SI	NO	NO	NO
Total SI	9	7	5	6

En la Tabla 8 se puede identificar que la placa embebida más adecuada, en función a los requerimientos establecidos, corresponde a la NVIDIA Jetson Nano debido a sus características que la diferencian de las demás, las cuales principalmente son su memoria RAM de 4Gb y su rendimiento de CPU 472 GFLOP, destacando su potencial para realizar tareas destinadas a IA y visión artificial.

3.5.2 Cámara para captura de gestos realizados con las manos

La cámara debe poseer una resolución mínima de HD. Según Caballero (2017), una calidad de imagen aceptable se logra con una resolución de al menos HD, es decir, igual o superior a 1280x720 píxeles. Es necesario disponer de una cámara con esta

resolución como mínimo para capturar de manera efectiva los gestos realizados con la mano, con el objetivo de detectar la mayor cantidad de detalles posibles. Además, debe ser compatible con distribuciones Linux y con librerías de visión artificial. En la Tabla 9 se realiza una comparación de cámaras, considerando los requerimientos presentados anteriormente.

Tabla 9

Comparación de diferentes cámaras

Cámara	Requisitos						Total SI
	SHRS2	SiRS5	SiRS6	SHRS7	SiRS13	SHRS19	
Logitech c170	SI	NO	SI	NO	NO	NO	2
Logitech c922 Pro	SI	SI	SI	SI	SI	SI	6
Microsoft							
Lifecam HD-3000	NO	SI	SI	SI	NO	SI	4

En la Tabla 9 se puede identificar que la cámara más adecuada, en función a los requerimientos establecidos, corresponde a la Logitech c922 Pro principalmente por su compatibilidad con librerías enfocadas a tareas de visión artificial y su resolución HD. Otro punto para considerar es que la presente cámara cuenta con compatibilidad con sistemas operativos de distribuciones Linux.

3.6 Selección de Software

En este apartado, se selecciona el software específico mediante una ponderación de los requerimientos presentados anteriormente. A continuación, se expone la selección del lenguaje de programación, la CNN preentrenada, el IDE de programación y la plataforma compatible con tareas de entrenamiento de IA.

3.6.1 Lenguaje de programación

El lenguaje de programación con el que se desarrollará el proyecto de visión artificial debe contar con compatibilidad con librerías enfocadas a visión artificial, PyTorch, contar con licencia de libre uso y ser un lenguaje de programación de alto nivel, además de proporcionar facilidad de manejo y entendimiento al momento de realizar la programación para llevar a cabo el presente proyecto de visión artificial. En la Tabla 10 se realiza una comparación de tres lenguajes de programación en específico considerando los requerimientos presentados anteriormente.

Tabla 10

Comparación de lenguajes de programación

Lenguaje de programación	Requisitos						Total SI
	SiRS4	SHRS5	SiRS7	SHRS16	SHRS19	SHRS23	
Python	SI	SI	SI	SI	SI	SI	6
C++	SI	NO	SI	SI	SI	NO	4
Java	SI	NO	NO	SI	SI	NO	3

En la Tabla 10 se puede identificar que el lenguaje de programación más adecuado para llevar a cabo el presente proyecto, en función a los requerimientos establecidos, corresponde a Python principalmente por su facilidad de manejo y entendimiento, además

de contar con una licencia de libre uso y de presentar disponibilidad de varias librerías de visión artificial como OpenCV, Numpy y Mediapipe. Cabe destacar que Python ofrece la opción de trabajar con PyTorch y Ultralytics, las cuales son herramientas que ofrecen una amplia gama de beneficios a la hora de entrenar modelos de visión artificial de manera eficiente y detectar objetos en tiempo real.

3.6.2 CNN preentrenada

Se hará uso de las CNNs preentrenadas que dispone Ultralytics, específicamente las ofrecidas por YOLOv8, debido a que están enfocadas especialmente en la detección de objetos. La selección de la CNN preentrenada se debe realizar al considerar requerimientos como, buena precisión de detección de CNN preentrenada, bajo consumo de recursos computacionales e inferencia en tiempo real fluida.

Los parámetros asociados con los requerimientos presentados para la selección de la CNN preentrenada de YOLOv8, los cuales se encuentran detallados en el sitio web oficial de Ultralytics, son los siguientes:

- **Params (M):** Representa el número de parámetros o pesos de la red neuronal, expresado en millones. Un mayor número de parámetros implica un mayor consumo de recursos computacionales durante el entrenamiento y la inferencia en tiempo real.
- **mAP:** Esta métrica evalúa la precisión general del modelo de detección de objetos. Un valor de mAP más alto indica una mayor precisión en la detección de objetos.
- **Speed (ms):** Representa el tiempo promedio que tarda el modelo en realizar una inferencia, expresado en milisegundos. Un tiempo de inferencia menor indica una mayor eficiencia al realizar inferencias en tiempo real.

En la Tabla 11 se realiza una comparación de CNNs preentrenadas de YOLOv8 considerando los requerimientos y parámetros presentados anteriormente.

Tabla 11

Comparación de CNNs preentrenadas de YOLOv8

CNN preentrenada	Requisitos		
	SHRS25	SHRS26	SHRS27
	Params(M)	mAP	Speed (ms)
yolov8n	3,2	37,3	0,99
yolov8s	11,2	44,9	1,20
yolov8m	25,9	50,2	1,83
yolov8l	43,7	52,9	2,39
yolov8x	68,2	53,9	3,53

En la Tabla 11 se puede identificar que la CNN preentrenada de YOLOv8 más adecuada para llevar a cabo el presente proyecto, en función a los requerimientos establecidos, corresponde a yolov8n. Esta elección se fundamenta en varios aspectos. En primer lugar, al considerar que la inferencia se realizará en tiempo real, es de suma importancia que el modelo pueda realizarla en el menor tiempo posible. Además, es posible conocer que las otras alternativas de CNNs presentan un consumo de recursos más elevado, lo cual afectaría al requerimiento de la inferencia mencionado inicialmente.

3.6.3 Plataforma compatible con tareas de entrenamiento de IA

La plataforma en la que se lleve a cabo el entrenamiento del modelo de visión artificial con la CNN preentrenada de YOLOv8 debe brindar al desarrollador características de rendimiento ante altas cargas de procesamiento, por lo que con ella se debe proporcionar un rendimiento de GPU considerable para entrenar modelos de visión

artificial de tal forma que permita realizar dicho entrenamiento de forma más rápida. La plataforma debe brindar al desarrollador facilidad de manejo y entendimiento, disponibilidad de soporte y actualizaciones, compatibilidad con Python y principalmente garantizar el uso de modelos preentrenados.

En la Tabla 12 se realiza una comparación de plataformas que permiten realizar tareas de entrenamiento de visión artificial considerando los requerimientos presentados anteriormente.

Tabla 12

Comparación de plataformas compatibles con tareas de entrenamiento de IA

Plataforma	Requisitos							Total
	SHRS	SiR	SHRS	SHRS	SHRS	SHRS	SHR	
	4	S7	17	18	20	22	S25	SI
Coolaboratory	SI	SI	SI	SI	SI	NO	SI	6
Google Colab	SI	SI	SI	SI	SI	SI	SI	7
Amazon SageMaker	SI	NO	NO	SI	SI	SI	SI	5
Entrenamiento local	SI	NO	SI	SI	SI	SI	NO	5

En la Tabla 12 se puede identificar que la plataforma más adecuada para el entrenamiento de modelos de visión artificial, según los criterios establecidos, es Google Colab. Esta plataforma ofrece acceso a recursos computacionales de alto rendimiento, como una GPU virtual NVIDIA Tesla T4, lo que resulta en un notable aumento en el rendimiento de la GPU. Como resultado, el tiempo de entrenamiento del modelo de visión artificial se reduce significativamente en comparación con el uso de un equipo físico.

Además, al ser una plataforma alojada en la nube, Google Colab no utiliza los recursos computacionales del dispositivo personal, lo que ayuda a reducir su carga. Adicionalmente esta plataforma ofrece una amplia gama de recursos educativos, facilidad de uso y compatibilidad con Python, además de la opción de utilizar de modelos de CNN preentrenados para el entrenamiento de manera eficiente.

3.7 Selección de Dataset

El Dataset utilizado para el entrenamiento del modelo de visión artificial debe cumplir con ciertos requerimientos para garantizar resultados óptimos en la detección de gestos realizados con las manos para cada letra del alfabeto ASL. Este conjunto de datos seleccionado debe incluir imágenes capturadas en diferentes ambientes y poses, así como un particionamiento adecuado en conjunto de entrenamiento y conjunto de validación, junto con etiquetas de posición del objeto a detectar para cada imagen. Estos requerimientos son necesarios ya que influyen directamente en la precisión de detección de gestos debido a que el Dataset es utilizado para el entrenamiento del modelo de visión artificial.

En la Tabla 13 se realiza una comparación entre varios Datasets de uso libre almacenados en repositorios de GitHub, considerando los requerimientos presentados anteriormente.

Tabla 13

Comparación de Datasets

Dataset	Requisitos								Total SI
	SdRS	SdRS	SdRS	SdRS	SdRS	SdRS	SdRS	SdRS	
	1	2	3	4	6	7	8		
Real-Time-Sign-Language	SI	SI	NO	SI	SI	SI	SI	SI	7

sign_language_dat assets	SI	SI	NO	NO	NO	NO	NO	3
Modeling- Intensification-for- SLG	NO	SI	NO	SI	NO	SI	NO	4
American-Sign- Language- Recognition-using- Deep-Neural- Network	NO	SI	NO	NO	SI	SI	SI	5

En la Tabla 13 se puede identificar que el Dataset de imágenes más adecuado para llevar a cabo el presente proyecto, en función a los requerimientos establecidos, corresponde al Real-Time-Sign-Language, el cual dispone de muestras de imágenes de gestos realizados con la mano de cada letra correspondiente al alfabeto ASL. Este Dataset cumple con los criterios establecidos para ser utilizado al momento de llevar a cabo el entrenamiento de visión artificial. Adicionalmente, el Dataset seleccionado se distingue de las otras alternativas por tener un archivo “.yaml” el cual es necesario para el entrenamiento, por lo que a su vez cuenta con asignación de clases para cada letra del alfabeto ASL.

3.8 Diseño del sistema

En el presente apartado se describe el diseño del sistema, considerando las selecciones de requerimientos tomadas según las comparaciones presentadas en las tablas de los ítems 3.5, 3.6 y 3.7, así como la revisión bibliográfica de trabajos relacionados con visión artificial aplicada a la traducción de lenguaje de señas. Para una comprensión más clara del diseño y funcionamiento del sistema, se emplearán diagramas de bloque y de flujo.

3.8.1 *Arquitectura del sistema*

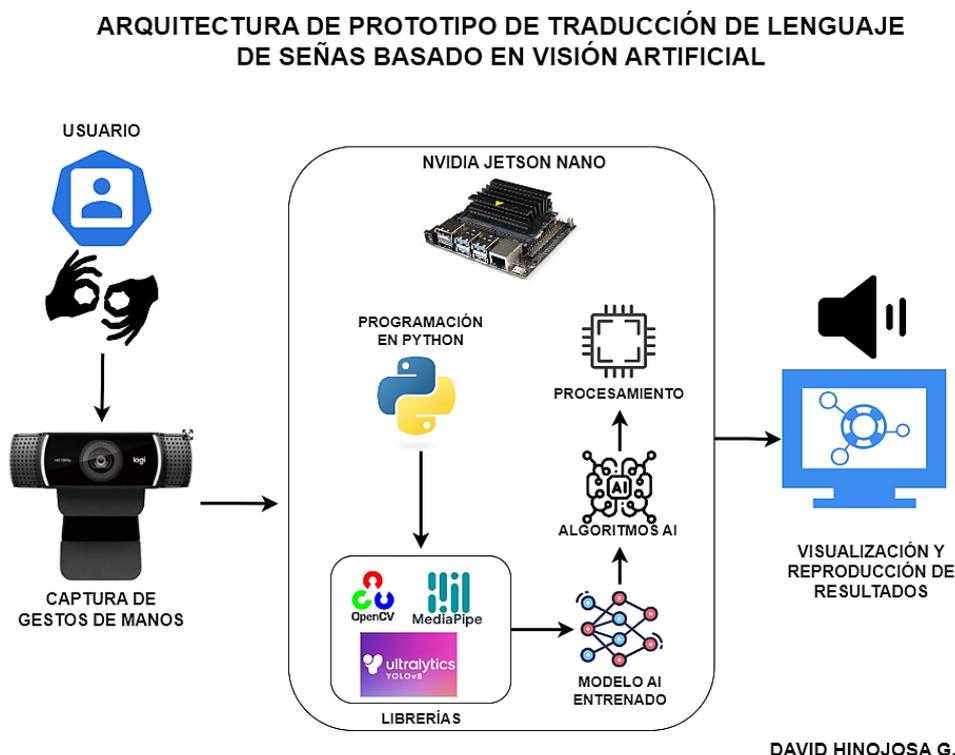
La arquitectura del prototipo de traducción de lenguaje de señas basado en visión artificial se muestra en la Figura 8, la cual consta de varios componentes fundamentales que interactúan entre sí para capturar, procesar y traducir los gestos realizados con las manos en tiempo real. A continuación, se describe cada componente y su función en el sistema:

- **Usuario:** Este componente corresponde a las personas con discapacidad auditiva que utilizan el prototipo para traducir sus gestos de lenguaje de señas.
- **Captura de gestos con las manos:** Este componente se relaciona directamente con la cámara, la cual es una cámara Logitech c922, seleccionada en el ítem 3.5.2. Siendo utilizada para capturar los gestos de las manos del usuario.
- **NVIDIA Jetson Nano:** La placa embebida actúa como el núcleo de procesamiento del sistema, en la cual se ejecutan los algoritmos de AI y el modelo entrenado en Google Colab. Este componente consta de los siguientes subcomponentes:
 - **Programación en Python:** Python se selecciona como lenguaje de programación principal debido a su compatibilidad con diversas librerías de visión artificial.
 - **Librerías:** Se emplean librerías de visión artificial como OpenCV, MediaPipe y Ultralytics YOLOv8 para la detección de manos y traducción de lenguaje de señas en tiempo real. Adicionalmente, se utiliza Google Text-to-Speech para la síntesis de voz, permitiendo que cada letra detectada sea anunciada de manera audible.

- **Modelo AI entrenado:** El modelo entrenado en Google Colab se carga en la placa embebida NVIDIA Jetson Nano para realizar las inferencias en tiempo real.
- **Algoritmos AI:** Se aplican los algoritmos de inteligencia artificial, utilizando la CNN preentrenada de YOLOv8 para detectar y clasificar los gestos de las manos.
- **Procesamiento:** Se realizan tareas de procesamiento de las imágenes capturadas por la cámara, como el redimensionamiento y normalización de las imágenes, durante la inferencia en tiempo real.
- **Visualización y reproducción de resultados:** Los resultados de la traducción de los gestos de lenguaje de señas son visualizados a través de un monitor, y adicionalmente, se reproduce el sonido de la letra detectada a través de parlantes o dispositivos de salida de audio.

Figura 8

Arquitectura del prototipo de traducción de lenguaje de señas basado en visión artificial



3.8.2 Diagrama de conexiones del prototipo

En la presente sección se detallan las conexiones del sistema de traducción de lenguaje de señas implementadas en la placa embebida NVIDIA Jetson Nano. Los diagramas de conexiones se presentan en vistas superior y frontal para ofrecer una comprensión completa de la configuración del hardware.

3.8.2.1 Vista superior

En el diagrama de conexiones correspondiente a la vista superior se muestran las conexiones físicas de cada uno de los componentes conectados, el cual se presenta en la Figura 9. A continuación, se detallan cada uno de ellos:

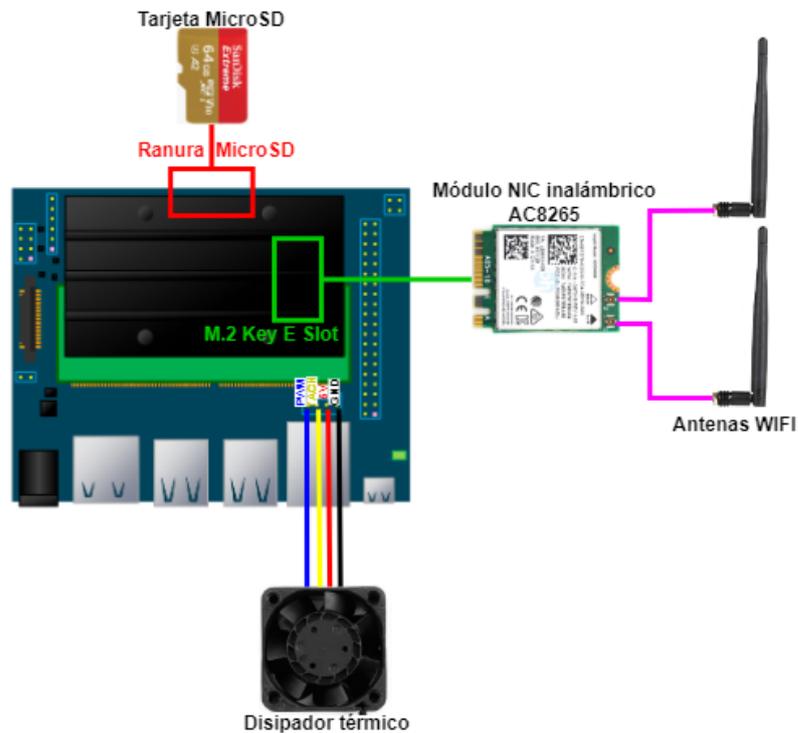
- **Tarjeta MicroSD:** En ella se almacena el sistema operativo, la cual es insertada en la ranura para tarjetas MicroSD de la placa embebida.

- **Módulo NIC inalámbrico AC8265:** El presente módulo es conectado en la ranura M.2 Key Slot en la Jetson Nano, el cual proporciona conectividad inalámbrica a internet.
- **Antenas WiFi:** Son conectadas al módulo NIC para la recepción de señales WiFi.
- **Disipador térmico:** Es conectado a los pines PWM, TACH, 5V y GND de la NVIDIA Jetson Nano.

Figura 9

Diagrama de conexiones correspondiente a la vista superior del prototipo

DIAGRAMA DE CONEXIONES - VISTA SUPERIOR



DAVID HINOJOSA G.

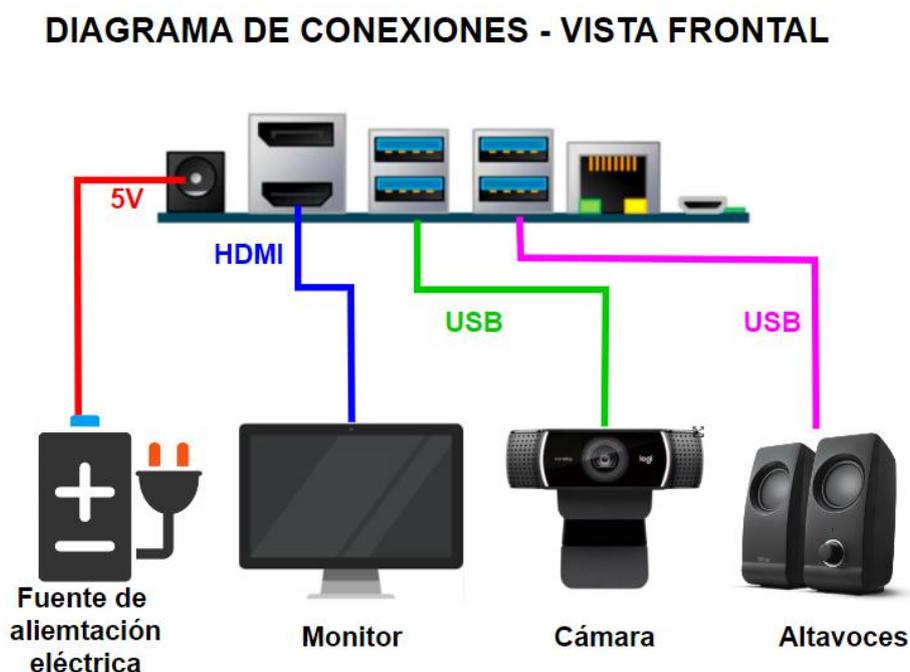
3.8.2.2 Vista frontal

En el diagrama de conexiones correspondiente a la vista frontal se muestran las conexiones físicas de cada uno de los componentes conectados, el cual se presenta en la Figura 10. A continuación, se detallan cada uno de ellos:

- **Fuente de alimentación eléctrica (5V):** Se conecta al puerto de alimentación de 5 Voltios de la NVIDIA Jetson Nano para proporcionar la energía necesaria para su funcionamiento y componentes de hardware conectados.
- **Monitor:** El monitor se conecta a la NVIDIA Jetson Nano a través de un cable HDMI para permitir la visualización de los resultados de la traducción de lenguaje de señas.
- **Cámara:** Se conecta a la NVIDIA Jetson Nano mediante un puerto USB.
- **Altavoces:** Se conectan a la placa embebida a través de un puerto USB.

Figura 10

Diagrama de conexiones correspondiente a la vista frontal del prototipo

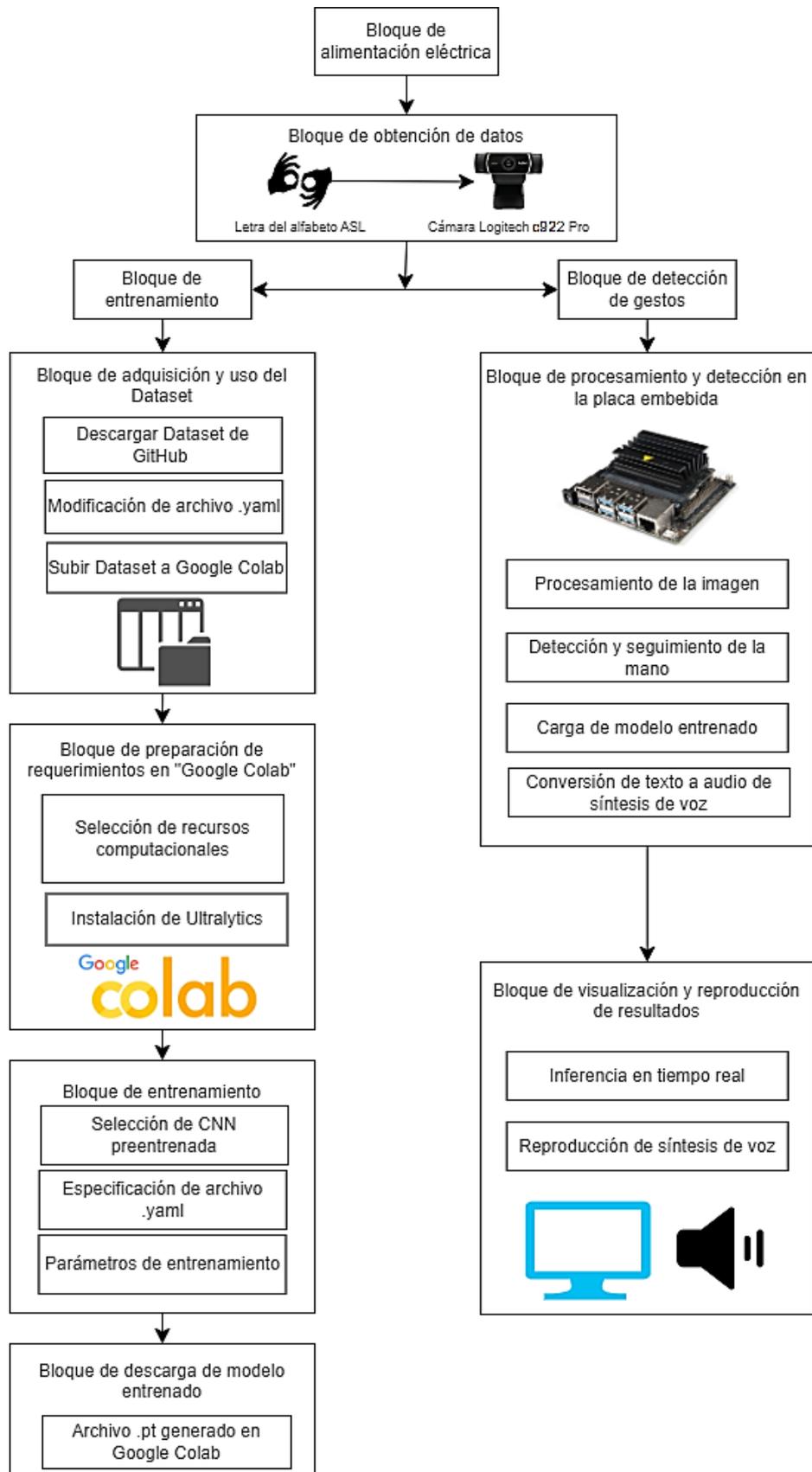


DAVID HINOJOSA G.

3.8.3 Diagrama de bloques del sistema

En el presente proyecto se desarrolla un prototipo de traducción de lenguaje de señas, específicamente con el uso del alfabeto de Lenguaje de Señas Americano (ASL), para personas con discapacidad auditiva basado en visión artificial, por lo cual el sistema

del presente proyecto se divide en dos procesos. El primer proceso corresponde al entrenamiento del modelo de visión artificial realizado en la plataforma alojada en la nube Google Colab, mientras que el segundo proceso corresponde al proceso de detección de gestos de las manos para cada letra del alfabeto ASL realizado en la placa embebida NVIDIA Jetson Nano. En la Figura 11 se puede visualizar el diagrama de bloques del sistema, donde se presentan los procesos mencionados anteriormente. Después, se examinará en detalle cada uno de estos.

Figura 11*Diagrama de bloques del sistema*

3.8.3.1 Bloque de entrenamiento

El bloque de entrenamiento se lleva a cabo principalmente en Google Colab debido a que esta plataforma ofrece acceso a recursos computacionales de alto rendimiento. Además, al ser una plataforma alojada en la nube, Google Colab no utiliza los recursos computacionales del dispositivo personal, lo que ayuda a reducir su carga.

Como se indicó anteriormente en la Figura 11, el bloque de entrenamiento se divide en varios subbloques, los cuales son: subbloque de adquisición y uso del Dataset, subbloque de preparación de requerimientos en Google Colab, subbloque de entrenamiento y subbloque de descarga del modelo entrenado. A continuación, se detalla el funcionamiento de cada uno de estos subbloques:

- **Subbloque de adquisición y uso del Dataset:** En este subbloque se adquiere el Dataset seleccionado en el ítem 3.7, el cual corresponde al Real-Time-Sign-Language. Este Dataset de uso libre dispone de muestras de imágenes de gestos realizados con la mano de cada letra correspondiente al alfabeto ASL.

El Dataset Real-Time-Sign-Language cuenta con un particionamiento adecuado en conjunto de entrenamiento (80%) y conjunto de validación (20%), junto con etiquetas de posición del objeto a detectar para cada imagen. El conjunto de entrenamiento se utiliza para que el algoritmo de aprendizaje automático ajuste los parámetros del modelo, mientras que el conjunto de validación evalúa el rendimiento del modelo durante el entrenamiento.

En el archivo “.yaml”, el cual almacena las clases del Dataset, es posible verificar las clases definidas para cada una de las 26 letras del alfabeto ASL, además de las rutas correspondientes a los directorios de los conjuntos de entrenamiento y validación debido a que el presente archivo es esencial para el proceso de entrenamiento que se explicará posteriormente.

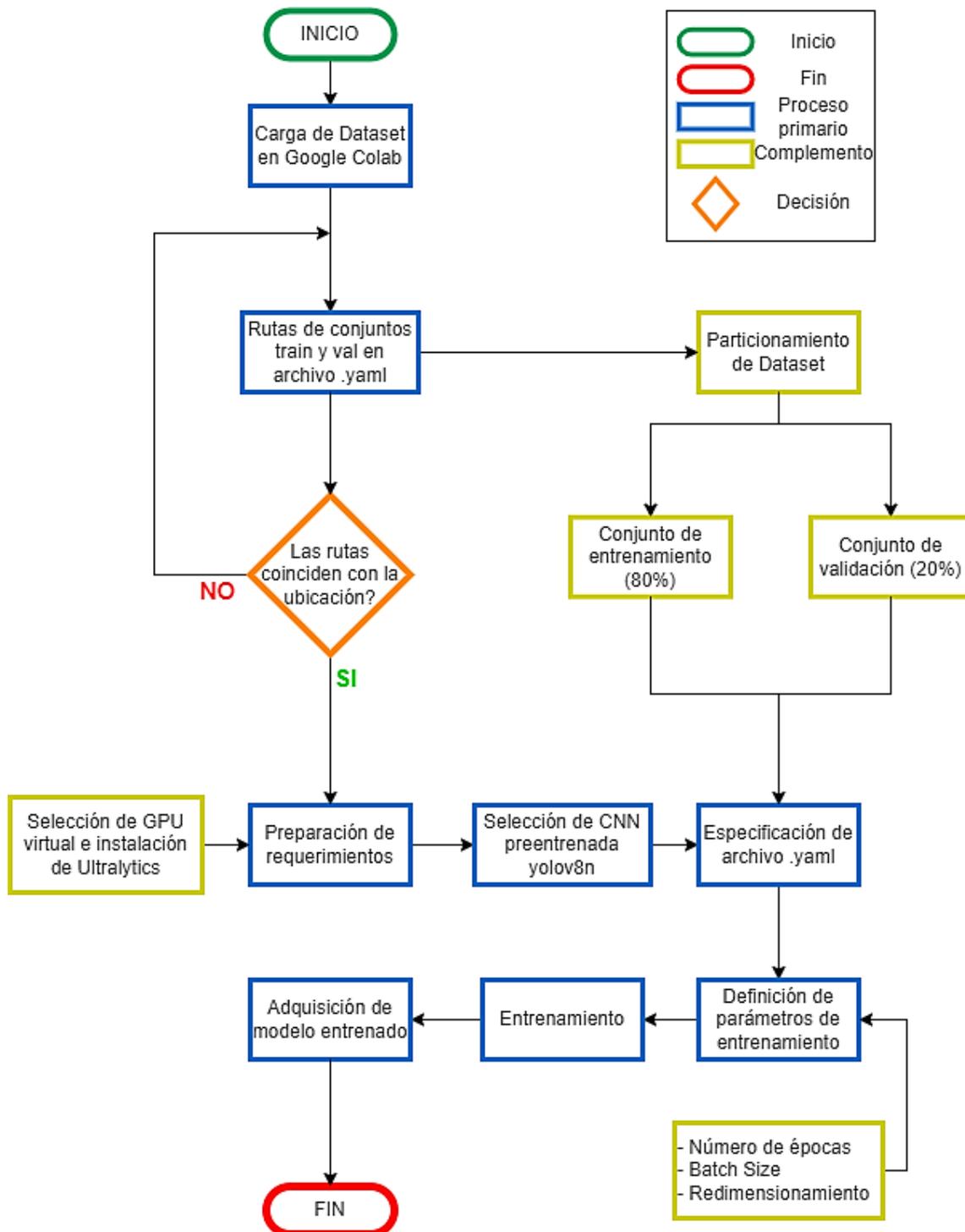
- **Subbloque de preparación de requerimientos en Google Colab:** En este subbloque se seleccionarán los recursos computacionales que ofrece Google Colab para aumentar el rendimiento a la hora de realizar el entrenamiento del modelo de visión artificial. La selección de la GPU virtual NVIDIA Tesla T4 ofrecerá un notable aumento en el rendimiento de la GPU. Adicionalmente, en este subbloque se llevará a cabo la instalación de Ultralytics para poder hacer uso de las CNNs preentrenadas que ofrece YOLOv8, debido a que están enfocadas especialmente en la detección de objetos.
- **Subbloque de entrenamiento:** En este subbloque, se realizará el entrenamiento del modelo de visión artificial mediante la CNN preentrenada de YOLOv8 y seleccionada en el ítem 3.6.2, la cual corresponde a la yolov8n. Para llevar a cabo el entrenamiento se deben considerar aspectos como: especificación del Dataset a implementar, número de épocas de entrenamiento que se realizarán, tamaño del lote que se utilizará durante el entrenamiento (batch size) y el redimensionamiento en píxeles de las imágenes tomadas como muestras que se almacenan en el Dataset.
- **Subbloque de descarga del modelo entrenado:** Este subbloque toma lugar una vez el entrenamiento haya finalizado. Se descarga un archivo “.pt” que contiene el modelo de la red neuronal entrenada y almacena los parámetros óptimos aprendidos durante el entrenamiento. Este archivo se utilizará después para detectar y traducir los gestos de lenguaje de señas durante la inferencia en tiempo real.

Una vez detallados y especificados los subbloques correspondientes al bloque de entrenamiento, en la Figura 12 se muestra el diagrama de flujo para el proceso de entrenamiento del sistema de traducción de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial.

El proceso de entrenamiento de este diagrama de flujo comienza con la carga del Dataset Real-Time-Sign-Lenguaje, llevada a cabo en la plataforma alojada en la nube Google Colab. Se especifican las rutas de los conjuntos de entrenamiento y validación en el archivo “.yaml”, que almacena las clases del Dataset. Este Dataset está particionado en un conjunto de entrenamiento que almacena el 80% de las muestras totales y un conjunto de validación que almacena el otro 20% de las muestras. Si las rutas especificadas coinciden con la ubicación de los conjuntos, se seleccionan los recursos computacionales de Google Colab para el entrenamiento, en caso contrario, se revisan posibles errores en las rutas del archivo “.yaml”. Una vez configurado el entorno, se procede al entrenamiento del modelo de visión artificial utilizando la CNN preentrenada de YOLOv8, específicamente la yolov8n. Se definen los parámetros clave para el entrenamiento, como número de épocas, el tamaño del lote (batch size) y el redimensionamiento de las imágenes en píxeles. Al finalizar el entrenamiento, se descarga un archivo “.pt” que contiene el modelo de la red neuronal entrenada y los parámetros óptimos aprendidos durante este proceso. Este archivo “.pt” se utiliza posteriormente para la detección y traducción de los gestos del lenguaje de señas durante la inferencia en tiempo real.

Figura 12

Diagrama de flujo para el proceso de entrenamiento



3.8.3.2 Bloque de detección de gestos

Como se indicó en la Figura 11, el bloque de detección se divide en varios subbloques, los cuales son: subbloque de alimentación eléctrica, subbloque de obtención de datos, subbloque de procesamiento y detección en la placa embebida y subbloque de visualización y reproducción de resultados. A continuación, se detalla el funcionamiento de cada uno de estos subbloques:

- **Subbloque de alimentación eléctrica:** El presente subbloque define las formas en las que se alimenta el prototipo y el hardware del sistema, siendo la placa embebida NVIDIA Jetson Nano y la cámara Logitech C922 Pro.
- **Subbloque de obtención de datos:** En el presente subbloque se capturan los gestos de lenguaje de señas del alfabeto ASL realizados con las manos a través de la cámara Logitech C922 Pro.
- **Subbloque de procesamiento y detección en la placa embebida:** El presente subbloque es llevado a cabo en la placa embebida NVIDIA Jetson Nano, por lo que a continuación se detallan sus correspondientes subprocesos:
 - **Procesamiento de la imagen:** Se desarrolla el procesamiento de la imagen proporcionada del bloque de obtención de datos, en dicho procesamiento se hace uso de la librería OpenCV dado a que esta es especializada en visión artificial y proporcionará herramientas para el procesamiento de la imagen, se especifica la resolución de la imagen en 1280x720 correspondiente a la visualización de todo el entorno capturado por la cámara Logitech C922 Pro y un recorte de la imagen 640x640 para la mano que se detectará, debido que esta resolución es un requerimiento mencionado por Ultralytics para hacer uso de la CNN preentrenada yolov8n. Al recortar la mano para la resolución 640x640, se realiza una

interpolación cúbica ya que al hacer un cambio en el tamaño de la imagen se suele perder información al reducirla en un tamaño menor, por eso se implementa la interpolación cúbica para mejorar la calidad y precisión de la imagen resultante sin pérdida significativa de la información.

- Detección y seguimiento de la mano: En el presente sistema se lleva a cabo la detección y seguimiento de la mano mediante la implementación de la librería Mediapipe, esta librería es especializada en la detección de objetos, como la mano en este caso, para este proceso se especifica que la mano sea detectada al definir un umbral de detección del 90%, indicando que la mano sea detectada si se cuenta con un porcentaje de confianza de detección igual o superior al especificado para evitar confusiones por parte del algoritmo a la hora de realizar la inferencia en tiempo real. Adicionalmente se dibujan puntos en la mano para extraer sus correspondientes posiciones y extraer dichos píxeles con el fin de que en el recorte del mano, explicado en el subproceso anterior, se haga un seguimiento de la mano en los movimiento que esta tenga a lo largo de la inferencia en tiempo real.
- Carga de modelo entrenado: Se realiza la carga del modelo entrenado, siendo el archivo “.pt” generado tras la culminación del proceso de entrenamiento, para realizar las predicciones sobre las imágenes de la inferencia en tiempo real. Estas predicciones se visualizarán cuando el modelo presente un porcentaje de coincidencia igual o mayor al 80% para eliminar falsos positivos y garantizar que la traducción sea llevada de forma óptima.

- Conversión de texto a audio de síntesis de voz: En este proceso se utiliza la librería Google Text-to-Speech para transformar el texto correspondiente a la letra detectada en una señal de audio. Esta señal luego es reproducida a través de parlantes u otros dispositivos de salida de audios conectados al sistema.
- **Subbloque de visualización y reproducción de resultados:** En este bloque se visualiza la traducción de lenguaje de señas mediante la inferencia en tiempo real a través de un monitor, donde se visualizarán las predicciones de los gestos realizados con las manos en anotaciones sobre el recorte. En estas anotaciones será posible visualizar la letra resultante con su correspondiente porcentaje de coincidencia. Adicionalmente, se lleva a cabo la reproducción de los resultados a través del audio de síntesis de voz, complementando la visualización de los resultados al proporcionar una representación vocal de las señas detectadas, permitiendo que los usuarios escuchen en tiempo real el resultado de la traducción de los gestos.

Una vez detallados y especificados los subbloques indicados anteriormente, en la Figura 13 se muestra el diagrama de flujo para el proceso de detección de gestos del sistema de traducción de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial.

El proceso de detección de gestos de este diagrama de flujo comienza con la importación de librerías especializadas en visión artificial, a continuación se crea una cola para gestionar las tareas de síntesis de voz y se inicia el procesador de voz que se encargará de enunciar la letra detectada, luego, se carga el modelo de visión artificial entrenado desde el archivo “.pt” generado en el proceso de entrenamiento, seguido por la lectura de la cámara Logitech c922 Pro con una resolución de imagen de 1280x720

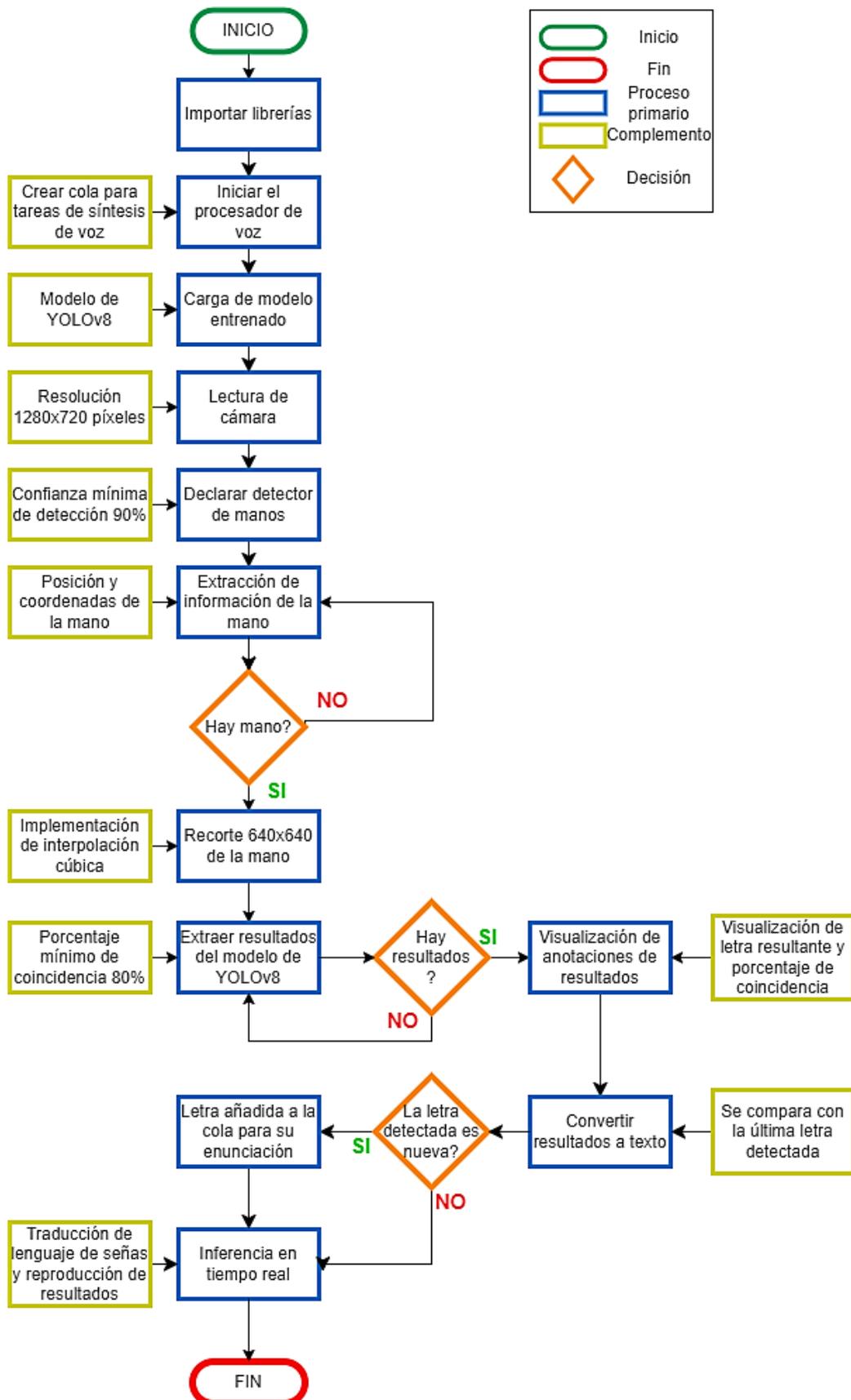
píxeles para capturar el entorno completo. Luego, se realiza la detección de la mano con un umbral de detección del 90%, indicando que se detectará la mano si la confianza de detección es igual o superior al valor especificado.

A continuación, se dibujan puntos en la mano para extraer su posición y píxeles correspondientes. Si se detecta una mano, se procede al recorte de esta a una resolución de 640x640 píxeles, ya que el modelo fue entrenado con imágenes de esta resolución. En caso contrario, se reitera el proceso de detección de la mano. La información extraída se compara con los parámetros aprendidos del modelo de visión artificial, extrayendo resultados si el porcentaje de coincidencia es igual o mayor al 80%, lo que ayuda a eliminar falsos positivos y garantiza una traducción óptima. Si se obtienen resultados, se muestran anotaciones que incluyen la letra resultante y el porcentaje de coincidencia. Estos resultados son luego convertidos a texto para compararlos con la última letra detectada, si la letra detectada es nueva, se añade a la cola de síntesis de voz para su enunciación.

Finalmente, la traducción de lenguaje de señas se realiza en la inferencia en tiempo real, mostrando las predicciones de los gestos realizados y reproduciendo la letra detectada a través del sistema de voz. Este flujo asegura una traducción óptima y eficiente del lenguaje de señas, permitiendo la visualización y reproducción de los gestos detectados realizados con la mano.

Figura 13

Diagrama de flujo para el proceso de detección de gestos



CAPÍTULO IV

EVALUACIÓN Y RESULTADOS

En este capítulo se detallará la implementación del presente proyecto, se evaluará que tan eficiente es el funcionamiento del sistema, con diferentes pruebas las cuales permitirán saber en qué parámetros el funcionamiento se está dando de forma correcta y en que parámetros se deberá mejorar, para poder garantizar la funcionabilidad del prototipo como herramienta de uso diario en personas que padecen de esta discapacidad auditiva.

4.1 Proceso de entrenamiento

En esta sección se describe el proceso de entrenamiento del modelo de visión artificial realizado en Google Colab, utilizando la CNN preentrenada de YOLOv8 para el traductor de lenguaje de señas dirigido personas con discapacidad auditiva. Este proceso se sigue de acuerdo con el diagrama de flujo para el entrenamiento presentado en la Figura 12, en donde se incluye la selección de recursos computaciones que ofrece Google Colab y la configuración de parámetros necesarios para el entrenamiento. El resultado final es un archivo “.pt” que contiene el modelo de la red neuronal entrenada.

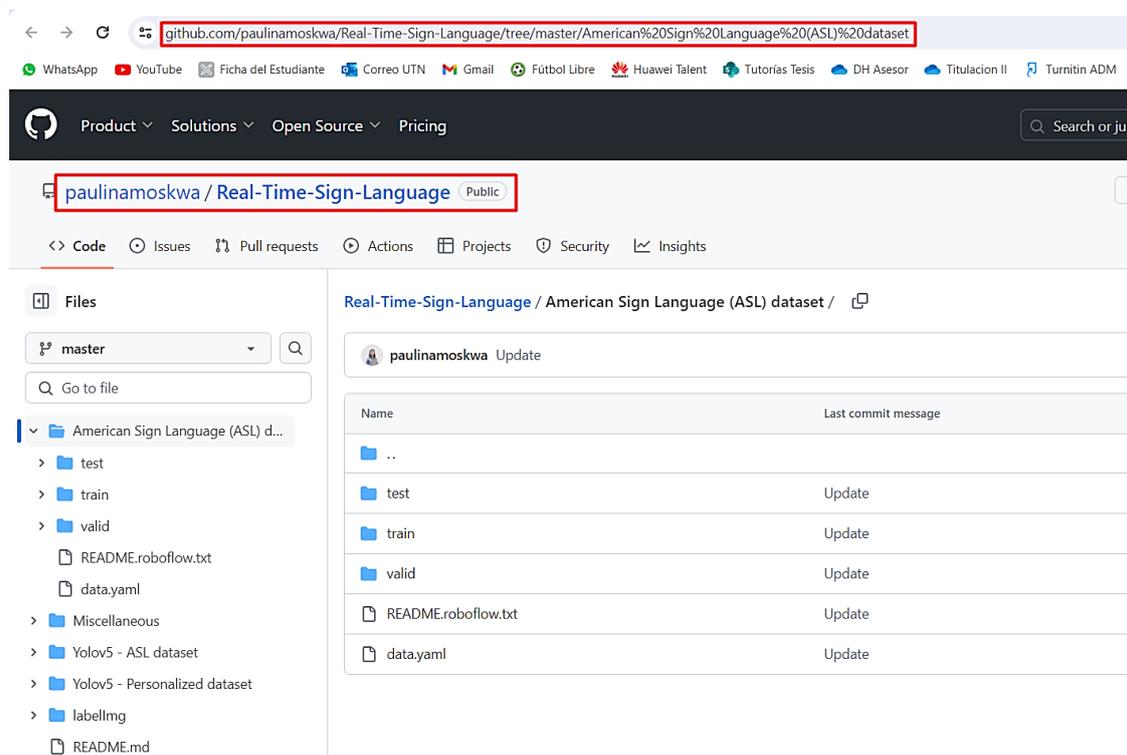
4.1.1 Carga de Dataset en Google Colab

Para el proceso de entrenamiento, se utiliza el Dataset Real-Time-Sign-Language, el cual debe descargarse desde su repositorio en GitHub, como se muestra en la Figura 14. Este repositorio contiene los conjuntos de entrenamiento y validación, que incluyen las muestras de los gestos de lenguaje de señas realizados con la mano. Estos conjuntos permiten que el modelo aprenda a detectar las letras del alfabeto ASL realizadas con la mano. El Dataset utilizado se accede a través del siguiente enlace:

[https://github.com/paulinamoskwa/Real-Time-Sign-Language/tree/master/American%20Sign%20Language%20\(ASL\)%20dataset](https://github.com/paulinamoskwa/Real-Time-Sign-Language/tree/master/American%20Sign%20Language%20(ASL)%20dataset)

Figura 14

Dataset Real-Time-Sign-Language almacenado en GitHub



Como se muestra en la Figura 15, en el archivo “*data.yaml*” se puede verificar que se encuentren incluidas las clases para cada una de las 26 letras del alfabeto ASL. Además, este archivo contiene las rutas correspondientes a los directorios de los conjuntos de entrenamiento y validación, permitiendo que el algoritmo de entrenamiento acceda a las muestras almacenadas en estos directorios en el proceso de entrenamiento.

Figura 15

Contenido del archivo “data.yaml” del Dataset Real-Time-Sign-language



```

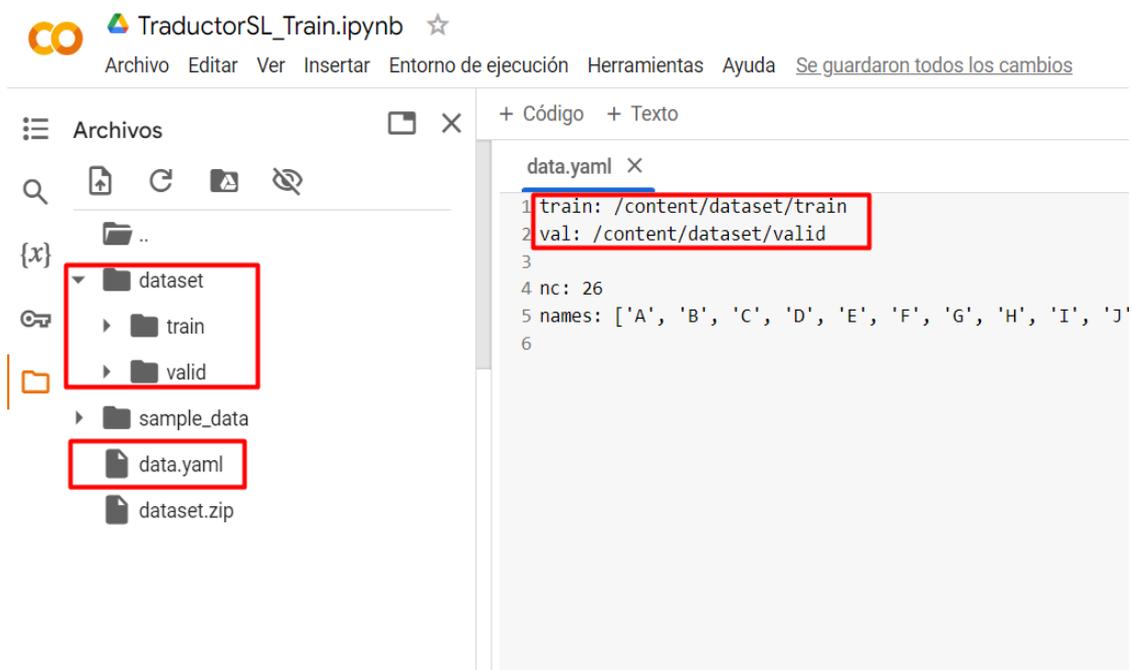
1 train: ../train/images
2 val: ../valid/images
3
4 nc: 26
5 names: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

```

A continuación, se carga el Dataset Real-Time-Sign-Language descargado previamente en el apartado de archivos de Google Colab. Luego, se actualizan las rutas de los repositorios de los conjuntos de entrenamiento y validación en el archivo “data.yaml”, editando y modificando dichas rutas para reflejar las ubicaciones actuales donde estos conjuntos están almacenados.

Figura 16

Carga de Dataset en Google Colab y modificación de rutas de conjuntos de entrenamiento y validación



```

1 train: /content/dataset/train
2 val: /content/dataset/valid
3
4 nc: 26
5 names: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
6

```

4.1.2 Preparación de requerimientos para entrenamiento en Google Colab

Para llevar a cabo el entrenamiento de la CNN preentrenada de YOLOv8 en Google Colab, primeramente, se deben seleccionar los recursos computacionales que esta plataforma alojada en la nube ofrece, con el fin de maximizar el rendimiento durante el proceso de entrenamiento del modelo de visión artificial. En el apartado de acelerador de hardware, se selecciona la opción “T4 GPU” debido a su capacidad para mejorar significativamente el rendimiento de la GPU, permitiendo aprovechar su alto rendimiento y reducir el tiempo total de entrenamiento.

Figura 17

Selección de T4 GPU como acelerador de hardware

Configuración del notebook

Tipo de entorno de ejecución

Python 3

Acelerador de hardware ?

CPU T4 GPU A100 GPU L4 GPU V100 GPU (deprecated)

TPU (deprecated) TPU v2

¿Quieres acceder a GPU premium? [Compra unidades de procesamiento adicionales](#)

Ejecutar automáticamente la primera celda o sección en cualquier ejecución

Omitir el resultado de las celdas al guardar este notebook

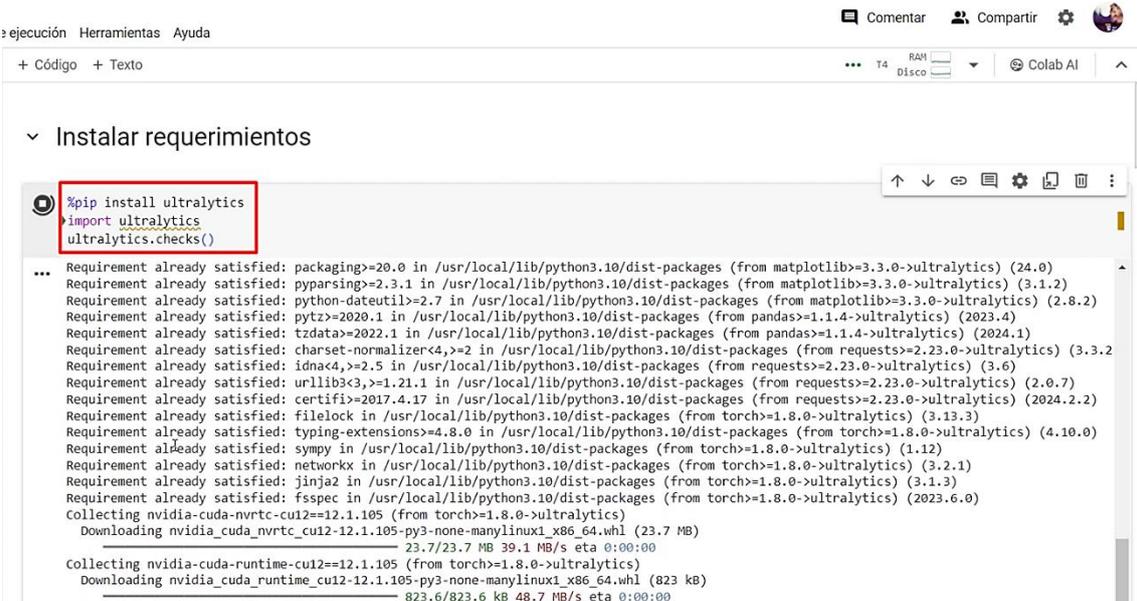
Cancelar Guardar

Una vez cargado el Dataset en la plataforma y editado el archivo “*data.yaml*”, se procede a instalar Ultralytics. Esta instalación proporciona todos los complementos necesarios para la utilizar las funciones y clases de Ultralytics, necesarias para el proceso de entrenamiento del modelo de visión artificial. Además, permite hacer uso de las CNNs preentrenadas que ofrece YOLOv8. En la Figura 18 se detallan los comandos a implementar para llevar a cabo la instalación de Ultralytics e importar su librería en el

entorno de trabajo, junto con una línea de verificación que confirma que la instalación se ha realizado correctamente y que todos los requisitos necesarios están presentes.

Figura 18

Instalación de Ultralytics en Google Colab para uso de CNNs preentrenadas de YOLOv8



```

ejecución Herramientas Ayuda
+ Código + Texto
... T4 RAM Disco Colab AI

Instalar requerimientos

%pip install ultralytics
import ultralytics
ultralytics.checks()

Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (24.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.4->ultralytics) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.4->ultralytics) (2024.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (2024.2.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.13.3)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (4.10.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.2.1)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.1.3)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (2023.6.0)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
    23.7/23.7 MB 39.1 MB/s eta 0:00:00
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
    823.6/823.6 kB 48.7 MB/s eta 0:00:00
  
```

4.1.3 Definición de parámetros de entrenamiento

A continuación, se implementa el comando necesario para llevar a cabo el entrenamiento del modelo de visión artificial utilizando la librería de Ultralytics, que permitirá realizar la traducción de lenguaje de señas. En este comando se deben considerar los siguientes parámetros:

- **Train:** Indica la tarea a realizar a través de la librería de Ultralytics, para el presente caso corresponde a la tarea de entrenamiento utilizando YOLOv8.
- **Model:** En el presente parámetro se especifica la CNN preentrenada de YOLOv8 a utilizar, la cual para el presente caso corresponde a yolov8n.pt.
- **Data:** En el presente parámetro se especifica el Dataset a utilizar durante el entrenamiento, el cual debe estar en el formato “.yaml”.

- **Epochs:** En este parámetro se indica el número de épocas que se utilizarán para el entrenamiento. Una época se define como un ciclo completo a través de todo el conjunto de datos de entrenamiento.
- **Batch:** Este parámetro especifica el tamaño de lote de datos que se procesará en cada paso de entrenamiento. Para el presente caso, el tamaño de lote corresponde a 2, indicando que el modelo procesará dos imágenes a la vez, siendo útil para reducir el tiempo de entrenamiento.
- **Imgsz:** En el presente parámetro se especifica el tamaño de redimensionamiento en píxeles de las imágenes de entrada que se utilizarán durante el entrenamiento. Para el presente caso, se redimensionarán a 640x640 píxeles, conforme a las especificaciones de Ultralytics al utilizar la CNN preentrenada yolov8n.pt.

4.1.4 Entrenamiento del modelo de visión artificial

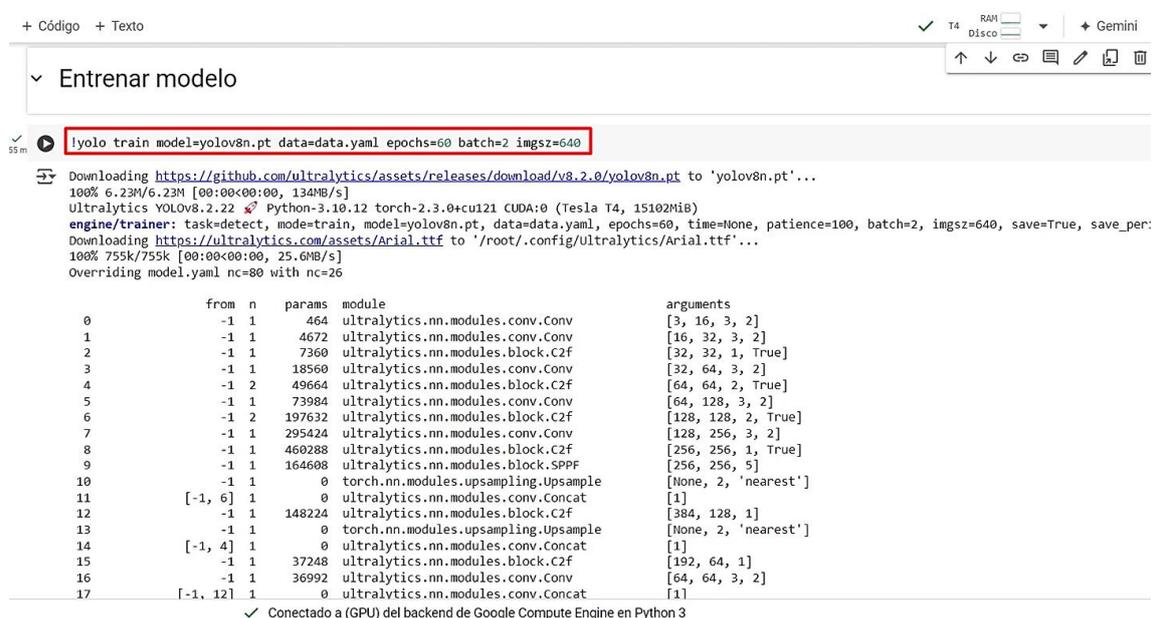
Durante el proceso de entrenamiento, el parámetro de épocas desempeña un papel crucial en la calidad y eficiencia del modelo entrenado. En cada época, el modelo procesa cada muestra del conjunto de datos una vez. Por lo tanto, el número de épocas influye significativamente en el rendimiento del modelo. Al aumentar el número de épocas, el modelo puede alcanzar una mayor precisión en la detección y clasificación de objetos, mientras que un menor número de épocas puede resultar en un modelo subentrenado, lo que se traduce en un rendimiento inferior.

Por eso, se entrena el modelo considerando diferentes épocas, para desarrollar un modelo de visión artificial eficiente y preciso. Para determinar el número de épocas adecuado para implementar en el proceso de entrenamiento, se ha realizado un análisis presentado en el ANEXO B, en el que se concluye que 60 épocas son las adecuadas a implementar.

Se realiza el entrenamiento del modelo de visión artificial a través del comando indicado en la Figura 19 para llevar a cabo esta tarea mediante la librería de Ultralytics YOLOv8, en el que se deben detallar los parámetros especificados en el punto 4.1.3. Adicionalmente, en la figura mencionada, se puede observar el proceso de descarga e inicialización de la CNN preentrenada “*yolov8n.pt*”.

Figura 19

Comando para realizar el entrenamiento en Google Colab



```

+ Código + Texto
v Entrenar modelo
yolo train model=yolov8n.pt data=data.yaml epochs=60 batch=2 imgsz=640
Downloading https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8n.pt to 'yolov8n.pt'...
100% 6.23M/6.23M [00:00<00:00, 134MB/s]
ultralytics YOLOv8.2.22 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8n.pt, data=data.yaml, epochs=60, time=None, patience=100, batch=2, imgsz=640, save=True, save_per:
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100% 755k/755k [00:00<00:00, 25.6MB/s]
overriding model.yaml nc=80 with nc=26

      from n  params module                        arguments
  0           -1  1      464 ultralytics.nn.modules.conv.Conv          [3, 16, 3, 2]
  1           -1  1     4672 ultralytics.nn.modules.conv.Conv          [16, 32, 3, 2]
  2           -1  1     7360 ultralytics.nn.modules.block.C2f           [32, 32, 1, True]
  3           -1  1    18560 ultralytics.nn.modules.conv.Conv          [32, 64, 3, 2]
  4           -1  2    49664 ultralytics.nn.modules.block.C2f           [64, 64, 2, True]
  5           -1  1    73984 ultralytics.nn.modules.conv.Conv          [64, 128, 3, 2]
  6           -1  2   197632 ultralytics.nn.modules.block.C2f           [128, 128, 2, True]
  7           -1  1   295424 ultralytics.nn.modules.conv.Conv          [128, 256, 3, 2]
  8           -1  1   460288 ultralytics.nn.modules.block.C2f           [256, 256, 1, True]
  9           -1  1   164608 ultralytics.nn.modules.block.SPPF          [256, 256, 5]
 10          -1  1           0 torch.nn.modules.upsampling.Upsample       [None, 2, 'nearest']
 11          [-1, 6] 1           0 ultralytics.nn.modules.conv.Concat          [1]
 12          -1  1   148224 ultralytics.nn.modules.block.C2f           [384, 128, 1]
 13          -1  1           0 torch.nn.modules.upsampling.Upsample       [None, 2, 'nearest']
 14          [-1, 4] 1           0 ultralytics.nn.modules.conv.Concat          [1]
 15          -1  1    37248 ultralytics.nn.modules.block.C2f           [192, 64, 1]
 16          -1  1    36992 ultralytics.nn.modules.conv.Conv          [64, 64, 3, 2]
 17          [-1, 12] 1           0 ultralytics.nn.modules.conv.Concat          [1]

✓ Conectado a (GPU) del backend de Google Compute Engine en Python 3

```

En la Figura 20, se presenta la información generada durante la ejecución del comando de entrenamiento del modelo de visión artificial. La salida muestra detalles específicos para cada época sobre varios parámetros y métricas, tales como la época actual respecto a las 60 épocas totales, la memoria GPU utilizada durante el entrenamiento, la pérdida de clasificación (que se refiere a la precisión del modelo al asignar la clase correcta a cada objeto detectado), el número de instancias procesadas en esa época, y el tiempo total tomado para procesar todas las imágenes, junto con la velocidad en imágenes por segundo. Este proceso se repite para cada una de las 60 épocas, permitiendo monitorear el rendimiento del modelo de manera continua.

Figura 20

Información presentada durante la ejecución del comando de entrenamiento

Entrenar modelo

```
!yolo train model=yolov8n.pt data=data.yaml epochs=60 batch=2 imgs=640
```

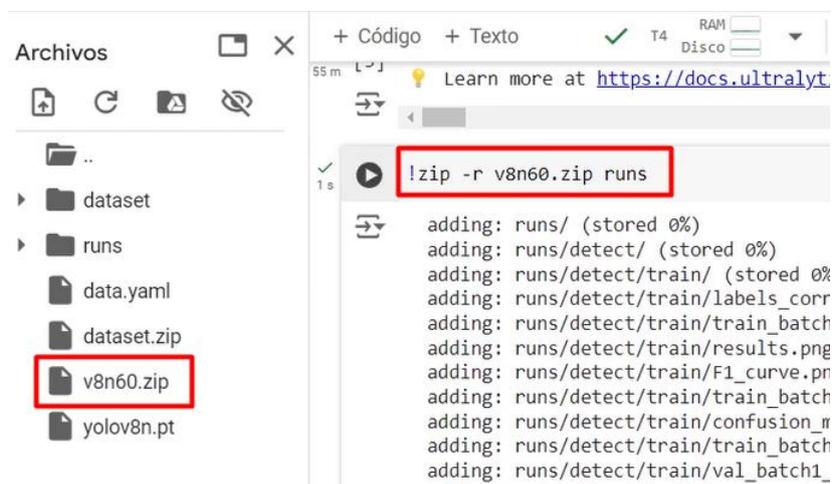
	all	144	144	0.819	0.796	0.884	0.708
Epoch 17/60	GPU_mem	0.489G	0.7619	1.724	1.226	5	Size: 640: 100% 756/756 [01:13<00:00, 10.35it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 36/36 [00:01<00:00, 19.12it/s]
	all	144	144	0.72	0.803	0.859	0.681
Epoch 18/60	GPU_mem	0.489G	0.7666	1.669	1.225	3	Size: 640: 100% 756/756 [01:12<00:00, 10.39it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 36/36 [00:01<00:00, 18.29it/s]
	all	144	144	0.743	0.834	0.872	0.702
Epoch 19/60	GPU_mem	0.489G	0.7539	1.617	1.227	6	Size: 640: 100% 756/756 [01:12<00:00, 10.49it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 36/36 [00:03<00:00, 11.59it/s]
	all	144	144	0.762	0.872	0.897	0.7
Epoch 20/60	GPU_mem	0.489G	0.7599	1.564	1.221	4	Size: 640: 100% 756/756 [01:12<00:00, 10.48it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 36/36 [00:01<00:00, 18.97it/s]
	all	144	144	0.791	0.801	0.885	0.713
Epoch 21/60	GPU_mem	0.489G	0.7656	1.54	1.222	4	Size: 640: 100% 756/756 [01:12<00:00, 10.38it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 36/36 [00:01<00:00, 18.80it/s]
	all	144	144	0.787	0.863	0.906	0.721

4.1.5 Adquisición del modelo entrenado

Tras el proceso de entrenamiento del modelo de visión artificial, se adquiere el modelo entrenado. Para ello, se debe implementar el comando indicado en la Figura 21, el cual comprime en una carpeta “.zip” el directorio donde se encuentra almacenado el modelo entrenado (/content/runs/). Esta acción se realiza con el propósito de facilitar la posterior descarga del modelo

Figura 21

Comando para comprimir la carpeta que contiene al modelo entrenado



```
+ Código + Texto T4 RAM Disco
Learn more at https://docs.ultralyt
!zip -r v8n60.zip runs
adding: runs/ (stored 0%)
adding: runs/detect/ (stored 0%)
adding: runs/detect/train/ (stored 0%)
adding: runs/detect/train/labels_corr
adding: runs/detect/train/train_batch
adding: runs/detect/train/results.png
adding: runs/detect/train/F1_curve.pr
adding: runs/detect/train/train_batch
adding: runs/detect/train/confusion_rr
adding: runs/detect/train/train_batch
adding: runs/detect/train/val_batch1_
```

Posteriormente, se implementa el comando indicado en la Figura 22, que permite importar la carpeta “.zip” al computador local desde Google Colab, facilitando así la transferencia del modelo entrenado a una ubicación local para su posterior uso.

Figura 22

Comando para realizar la descarga de la carpeta comprimida al computador



The image shows a snippet of a Google Colab interface. At the top, there are tabs for '+ Código' and '+ Texto'. Below them, a code cell is visible with a play button icon and a '0 s' timer. The code cell contains two lines of Python code: `from google.colab import files` and `files.download('v8n60.zip')`. This code is enclosed in a red rectangular box. Below the code cell, there is a progress bar for the download of 'v8n60.zip', which is partially filled with blue, indicating the download is in progress.

Una vez descargada la carpeta comprimida “.zip” desde Google Colab al computador local, se puede encontrar el modelo de visión artificial entrenado en el directorio especificado en la Figura 23, que se encuentra en la ruta ‘runs/detect/train/weights’. Dentro de esta carpeta se encuentran dos archivos con extensión “.pt”: “best.pt” y “last.pt”.

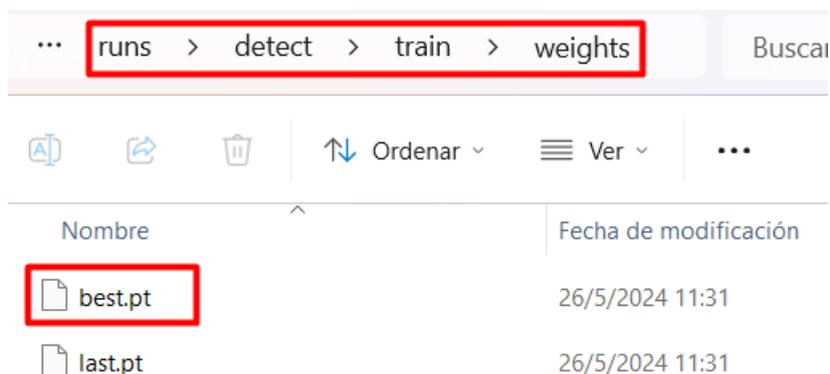
El archivo “best.pt” representa el modelo con el mejor rendimiento durante el proceso de entrenamiento, basado en las métricas de evaluación indicadas anteriormente en la Figura 20. Por otro lado, “last.pt” contiene el estado del modelo en la última época del entrenamiento.

Para implementar el modelo entrenado en el presente proyecto, se selecciona “best.pt” en lugar de “last.pt”, ya que de esta forma se garantiza que se está utilizando la versión óptima del modelo, debido a que este archivo refleja el mejor punto de rendimiento alcanzado durante el entrenamiento, asegurando así que el modelo sea lo más

eficiente y preciso posible para las tareas de detección de gestos realizados con la mano en el presente proyecto.

Figura 23

Archivo .pt correspondiente al modelo de visión artificial entrenado



4.2 Proceso de detección de gestos

En esta sección se describe el proceso de detección de gestos, el cual se lleva a cabo principalmente en la placa embebida NVIDIA Jetson Nano. Se detalla el código del sistema para la detección de gestos realizados con la mano. Este proceso se sigue según el diagrama de flujo para detectar gestos de la Figura 13, donde se incluyen las librerías para visión artificial, el procesamiento de la imagen, la detección de la mano y la visualización y reproducción de los resultados. El resultado final es la traducción de lenguaje de señas mediante una inferencia en tiempo real.

Para llevar a cabo la traducción de lenguaje de señas en tiempo real, se utilizará un código para la detección y seguimiento de manos obtenido del siguiente repositorio de GitHub: <https://github.com/AprendeIngenia/Piedra-Papel-o-Tijera-con-IA/blob/main/SeguimientoManos.py>.

Este código se encargará de identificar y rastrear las manos en cada fotograma de video, proporcionando las coordenadas y puntos de referencia necesarios para la

inferencia. Utilizando la librería MediaPipe, el código detectará las manos presentes, asegurando que sean correctamente identificadas. En el código que se realice para la inferencia, se importará y utilizará el seguimiento de manos para tomar la región de interés en cada fotograma y ejecutar el modelo entrenado, permitiendo así la traducción en tiempo real de los gestos de lenguaje de señas.

4.2.1 Importar librerías

Para iniciar el proceso de traducción de lenguaje de señas mediante la inferencia en tiempo real, como primer paso se realiza la instalación de las librerías necesarias en el IDE PyCharm, como se muestra en la Figura 24. Estas librerías son:

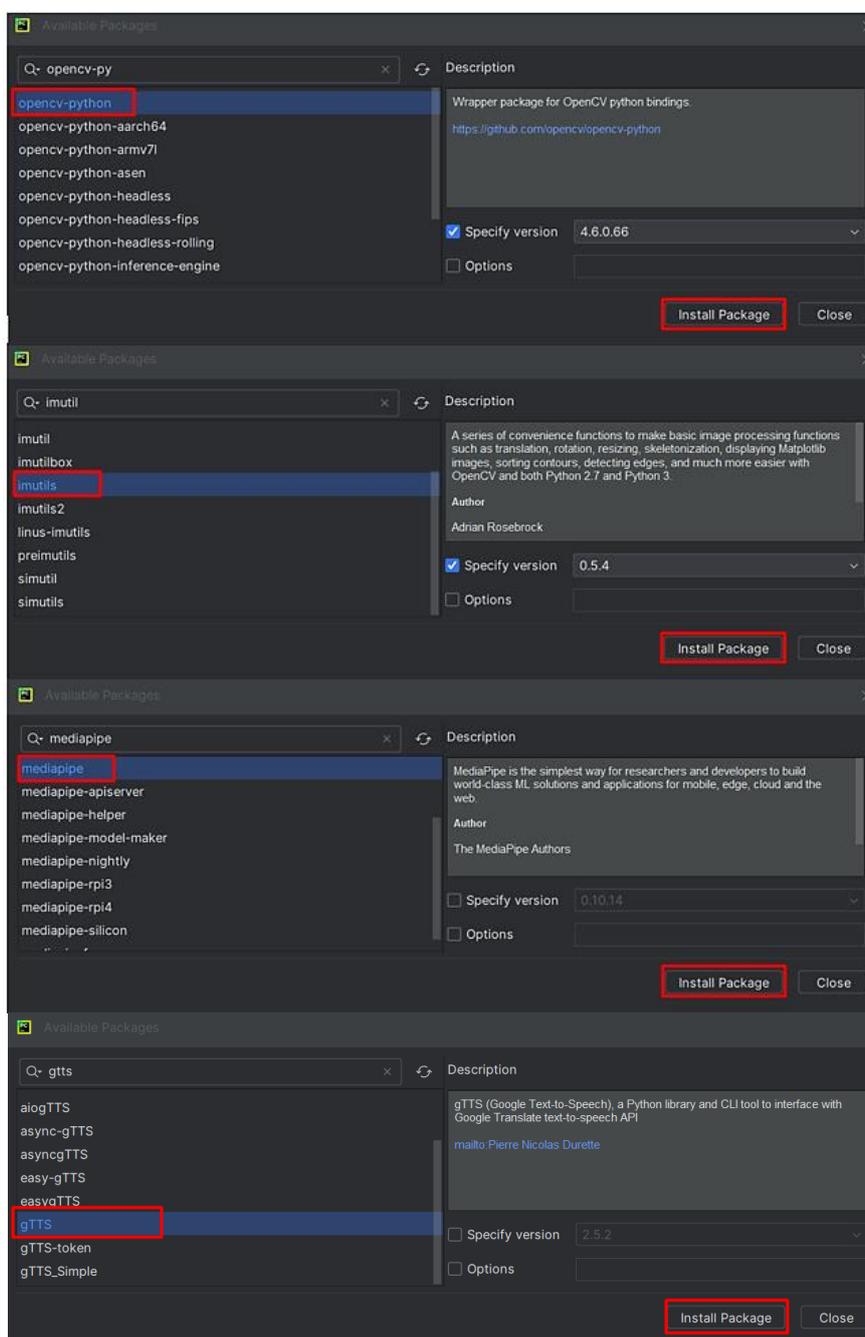
- **OpenCV:** Esta librería se utiliza para el procesamiento de imágenes. En el código se encarga de capturar los fotogramas de la cámara, realizar transformaciones de color y mostrar los resultados procesados en tiempo real.
- **Imutils:** Esta librería proporciona funciones que simplifican el trabajo con imágenes. En el código puede ser utilizada para realizar operaciones como el redimensionamiento, la rotación y la traslación de imágenes, facilitando el manejo de los fotogramas capturados.
- **Threading:** Esta librería es utilizada para ejecutar tareas en paralelo en hilos separados. Es utilizada para capturar video, procesar la detección de gestos y sintetizar voz de manera simultánea, optimizando así el rendimiento del sistema.
- **MediaPipe:** Esta librería se emplea para la detección y el seguimiento de las manos. En el código se encarga de identificar las manos en cada fotograma, extrayendo las coordenadas y puntos de referencia necesarios para la inferencia del modelo de traducción de lenguaje de señas.
- **gTTS (Google Text-to-Speech):** Esta librería permite convertir texto en habla utilizando el motor de síntesis de voz de Google. En el código se emplea para

generar audio a partir del texto de las letras detectadas, que luego se reproducen para anunciar la letra identificada.

- **Os:** Esta librería proporciona una forma de interactuar con el sistema operativo. En el código, se utiliza para ejecutar comandos del sistema, como la reproducción del audio generado por gTTS.

Figura 24

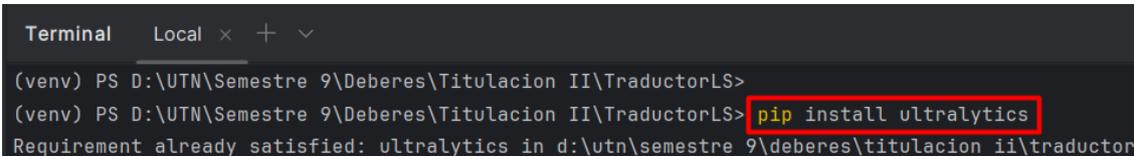
Instalación librerías en el IDE PyCharm



Posteriormente, se realiza la instalación de Ultralytics en el terminal local de PyCharm utilizando el comando que se puede visualizar en la Figura 25, para implementar el modelo entrenado. En el código, la librería Ultralytics se encarga de cargar y ejecutar el modelo de visión artificial previamente entrenado para la traducción de lenguaje de señas.

Figura 25

Comando para realizar la instalación de la librería Ultralytics

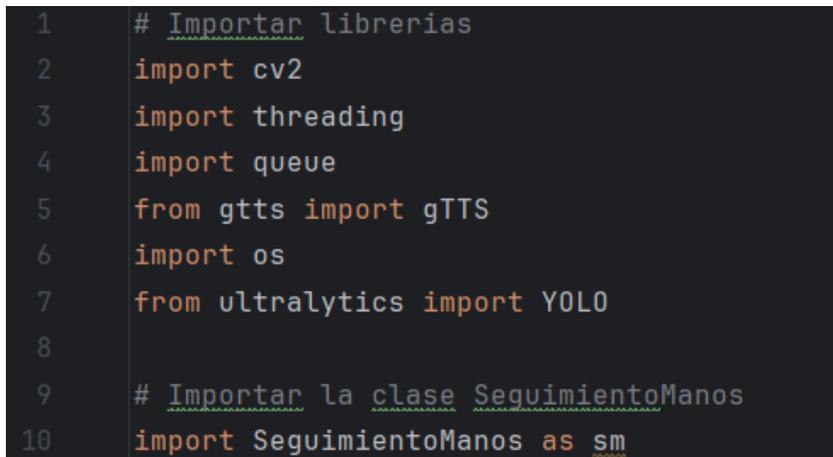


```
Terminal Local x + v
(venv) PS D:\UTN\Semestre 9\Deberes\Titulacion II\TraductorLS>
(venv) PS D:\UTN\Semestre 9\Deberes\Titulacion II\TraductorLS> pip install ultralytics
Requirement already satisfied: ultralytics in d:\utn\semestre 9\deberes\titulacion ii\traductorl
```

Una vez se hayan instalado todas las librerías necesarias, se procede a implementar las líneas de código indicadas en la Figura 26 para importar las librerías para la inferencia. Además, se importa la clase “*SeguimientoManos*” para hacer uso del código descargado previamente de GitHub, el cual permite la detección y el seguimiento de las manos.

Figura 26

Librerías importadas en el código para realizar la inferencia



```
1 # Importar librerías
2 import cv2
3 import threading
4 import queue
5 from gtts import gTTS
6 import os
7 from ultralytics import YOLO
8
9 # Importar la clase SeguimientoManos
10 import SeguimientoManos as sm
```

4.2.2 Inicialización del procesador de voz

En la sección del código visualizada en la Figura 27, se implementa el sistema de síntesis de voz que enuncia las letras detectadas. Para ello se utiliza la librería gTTS, que convierte el texto en audio. Esta sección del código comienza creando una cola de tareas (“*voice_queue*”), la cual se utilizará para gestionar las letras detectadas que deben ser pronunciadas.

A continuación, se define la función “*voice_processor()*”, que es ejecutada en un hilo separado. Esta función se encarga de procesar la cola de síntesis de voz en un bucle continuo. Cuando se detecta una letra, esta se agrega a la cola. El procesador de voz toma las letras de la cola, una a una, las convierte en audio utilizando gTTS, y reproduce el sonido con el comando del sistema operativo mpg321. Si en la cola se encuentra un valor None, el bucle se rompe y el hilo termina su ejecución.

Finalmente, se inicia un hilo dedicado exclusivamente al procesador de voz (“*voice_thread*”). Este hilo se ejecuta de manera paralela al resto del código, lo que permite que la inferencia del lenguaje de señas continúe sin interrupciones mientras se realiza la síntesis de voz. De esta manera, el sistema puede pronunciar las letras detectadas en tiempo real, mejorando la fluidez y la interacción con el usuario.

Figura 27

Líneas de código para inicializar el procesador de voz

```

12 # Crear cola para tareas de síntesis de voz
13 voice_queue = queue.Queue()
14
15 # Hilo de Procesamiento de voz
16 # Función para procesar la cola de síntesis de voz usando gTTS
17 usage
18 def voice_processor():
19     while True:
20         texto = voice_queue.get()
21         if texto is None:
22             break
23         tts = gTTS(texto, lang='es')
24         tts.save("temp_audio.mp3")
25         os.system("mpg321 temp_audio.mp3")
26         voice_queue.task_done()
27
28 # Iniciar el hilo del procesador de voz
29 voice_thread = threading.Thread(target=voice_processor)
30 voice_thread.start()

```

4.2.3 Carga de modelo entrenado

Una vez inicializado el procesador de voz, se procede a cargar el modelo de visión artificial entrenado previamente en Google Colab. Esto se realiza mediante el fragmento de código detallado en la Figura 28.

Mediante la línea de código “`model = YOLO('model.pt')`”, se carga el modelo utilizando la librería Ultralytics. El archivo “`model.pt`” corresponde al modelo entrenado previamente en Google Colab y guardado en el computador local. Al cargar este modelo, se habilita la capacidad del sistema para detectar y traducir gestos de lenguaje de señas.

Figura 28

Línea de código para cargar el modelo de visión artificial entrenado

```

48
49 # Cargar modelo entrenado
50 model = YOLO('model.pt')
51

```

4.2.4 Lectura de cámara

En la presente sección se implementa la lectura de la cámara para capturar los fotogramas durante la inferencia. En la porción de código detallada en la Figura 29, se gestiona la captura de video de manera asíncrona mediante la clase “*VideoCaptureAsync*”, que corresponde al segundo hilo implementado en el sistema, específicamente para la captura de video. Este hilo asegura que la captura de fotogramas desde la cámara ocurra de forma continua y en paralelo con otras tareas, como la inferencia del modelo y la síntesis de voz, optimizando el rendimiento del sistema.

En este fragmento de código, se define la clase “*VideoCaptureAsync*” y su constructor (“*__init__*”). Este constructor inicializa la fuente de video (“*src*”), que generalmente es la cámara conectada al sistema. Luego, se configura la captura de video con “*cv2.VideoCapture(self.src)*”, lo que establece la conexión con la cámara. Posteriormente, se define la resolución de la captura a 1280x720 píxeles mediante “*self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)*” y “*self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)*”, lo que garantiza que los fotogramas capturados tengan una calidad adecuada para el procesamiento posterior. A continuación, se realiza la captura del primer fotograma usando “*self.cap.read()*”, y se almacenan tanto el fotograma como el estado de la captura en “*self.frame*” y “*self.ret*”, respectivamente. Finalmente, se inicializan dos variables de control: “*self.stopped*”, que se utiliza para detener el hilo de captura cuando sea necesario, y “*self.lock*”, que es un mecanismo de sincronización que previene conflictos entre la captura y el procesamiento de fotogramas cuando se ejecutan en paralelo en diferentes hilos.

Figura 29

Líneas de código para inicializar la captura de video asíncrona en el segundo hilo

```

34 # Hilo de captura de video
35 # Lectura de la cámara
1 usage
36 class VideoCaptureAsync:
37     def __init__(self, src=0):
38         self.src = src
39
40         # Iniciar captura de video
41         self.cap = cv2.VideoCapture(self.src)
42
43         # Definir resolución a 1280x720 pixeles
44         self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, value: 1280)
45         self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, value: 720)
46
47         # Capturar el primer fotograma
48         self.ret, self.frame = self.cap.read()
49         self.stopped = False
50         self.lock = threading.Lock()

```

Después del fragmento de código explicado anteriormente, se continúa con la porción de código, detallada en la Figura 30, que se enfoca en gestionar la captura de video en segundo plano. La función “*start(self)*” inicia un hilo que ejecuta la función “*update(self)*”, la cual captura continuamente los fotogramas de la cámara mientras otros procesos del programa, como la inferencia, se ejecutan simultáneamente. Dentro de “*update(self)*”, se emplea un candado (“*lock*”) para proteger las variables que almacenan los fotogramas capturados, asegurando que no haya interferencias entre hilos.

La función “*read(self)*” permite acceder al fotograma actual de manera segura, utilizando también el bloqueo para evitar problemas de concurrencia. Finalmente, la función “*stop(self)*” detiene la captura de video cuando ya no es necesaria, asegurando un cierre ordenado del proceso de captura. Este enfoque asegura que la captura de video sea eficiente y no interfiera con otras tareas críticas del sistema, como la traducción en tiempo real de lenguaje de señas.

Figura 30

Líneas de código para la implementación del hilo de captura de video

```

52     # Iniciar hilo de captura de video
      1 usage
53     def start(self):
54         threading.Thread(target=self.update, args=()).start()
55         return self
56
57     # Función para actualizar fotogramas en segundo plano
      1 usage
58     def update(self):
59         while not self.stopped:
60             ret, frame = self.cap.read()
61             with self.lock:
62                 self.ret, self.frame = ret, frame
63
64     # Función para leer el fotograma actual
      1 usage
65     def read(self):
66         with self.lock:
67             frame = self.frame.copy()
68             return self.ret, frame
69
70     # Función para detener la captura de video
      1 usage
71     def stop(self):
72         self.stopped = True
73
74     # Iniciar captura de video en segundo plano
75     cap = VideoCaptureAsync().start()

```

4.2.5 Declarar detector de manos

Se declara el detector de manos utilizando el código descargado previamente del repositorio de GitHub. Esto se realiza mediante el fragmento de código detallado en la Figura 31, el cual es “*detector = sm.detectormanos(Confdeteccion=0.9)*”.

En esta línea de código, “*sm.detectormanos*” se refiere a una clase definida en el archivo “*SeguimientoManos.py*”, la cual es responsable de la detección y seguimiento de las manos en cada fotograma de video. Al declarar el detector de manos de esta clase con “*Confdeteccion=0.9*”, se está especificando un umbral de confianza del 90% para la detección de manos, es decir, solo se considerarán detecciones con un porcentaje de confianza de detección igual o superior al especificado, evitando así confusiones y falsas

detecciones de manos. Esta clase es utilizada en el código de la inferencia ya que proporciona las herramientas necesarias para identificar y detectar las manos de manera precisa.

Figura 31

Línea de código para declarar el detector de manos

```

76
77     # Declarar detector de manos
78     detector = sm.detectormanos(Confdeteccion=0.9)
79

```

4.2.6 Extracción de información de la mano

En la presente sección del código, se extrae la información necesaria de la mano para realizar la inferencia en tiempo real. Esto se realiza mediante el siguiente fragmento de código detallado en la Figura 32.

A partir de esta sección, se implementa el tercer hilo, que es el hilo principal encargado del procesamiento e inferencia en tiempo real. Este hilo coordina la captura de fotogramas, la detección de manos, y la inferencia del modelo para identificar las letras del lenguaje de señas.

El código inicia con un bucle “*while True*”, que garantiza la ejecución continua del procesamiento de video hasta que se decida detenerlo. En la primera línea dentro del bucle, “*ret, frame = cap.read()*”, se realiza la lectura del fotograma actual utilizando la función “*read()*” de la clase “*VideoCaptureAsync*”. Esta función retorna un booleano *ret* que indica si la captura fue exitosa, y “*frame*”, que es el fotograma capturado. Si no se logra capturar un fotograma (“*if not ret:*”), el bucle se interrumpe y finaliza la ejecución.

A continuación, se procede a la extracción de la información de la mano detectada en el fotograma actual. Para ello, mediante la línea de código “*frame = detector.encontrarmanos(frame, dibujar=False)*”, la función

“*detector.encontrarmanos(frame, dibujar=False)*” se utiliza para detectar las manos en el fotograma capturado. Esta función, definida en la clase “*detector manos*” en el archivo “*SeguimientoManos.py*”, utiliza la librería MediaPipe para identificar la presencia de manos. Al establecer “*dibujar=False*”, se evita que se dibujen las marcas de detección en el fotograma.

Mediante la línea de código “*lista1, bbox, mano = detector.encontrarposicion(frame, ManoNum=0, dibujarPuntos=False, dibujarBox=False, color=[0,255,0])*”, la función “*detector.encontrarposicion(frame, ManoNum=0, dibujarPuntos=False, dibujarBox=False, color=[0,255,0])*” obtiene la posición de una sola mano en el fotograma. Esta función devuelve:

- “*lista1*”: Una lista de coordenadas de los puntos clave de la mano detectada.
- “*bbox*”: Las coordenadas del cuadro delimitador que encierra la mano.
- “*mano*”: Un indicador del número de manos detectadas en el fotograma.

En esta función, “*ManoNum=0*” especifica que se analiza la primera mano detectada, “*DibujarPuntos=False*” y “*DibujarBox=False*” indican que no se dibujarán los puntos ni el cuadro delimitador en el fotograma, y “*color=[0,255,0]*” define el color que se usaría si se optara por dibujar.

Figura 32

Líneas de código para extraer información de la mano

```

94 # Hilo principal para procesamiento e inferencia
95 while True:
96     # Lectura de fotograma actual
97     ret, frame = cap.read()
98     if not ret:
99         break
100
101     # Extraer información de la mano
102     frame = detector.encontrarmanos(frame, dibujar=False)
103
104     # Posición de una sola mano
105     lista1, bbox, mano = detector.encontrarposicion(frame, ManoNum=0, dibujarPuntos=False, dibujarBox=False, color=[0, 255, 0])

```

4.2.7 Recorte de la mano

En la presente sección del código, se realiza el recorte de la mano detectada en el fotograma para enfocarse en la región de interés. Esto se hace mediante fragmento de código detallado en la Figura 33, que continúa dentro del bucle “*While*” anterior.

Este proceso se inicia con una verificación de que se ha detectado una mano (“*if mano == 1:*”). Si se cumple esta condición, se procede a extraer las coordenadas del recuadro delimitador que rodea la mano, utilizando la línea de código “*xmin, ymin, xmax, ymax = bbox*”. Aquí, “*xmin*” y “*ymin*” representan las coordenadas de la esquina superior izquierda del cuadro, mientras que “*xmax*” y “*ymax*” indican las coordenadas de la esquina inferior derecha.

Para asegurar que la mano esté completamente capturada dentro del recorte y evitar que posibles movimientos de la mano queden fuera del cuadro, se asigna un margen de 60 píxeles alrededor de las coordenadas del recuadro delimitador. Esto se realiza en las líneas de código “*xmin, ymin = max(xmin - 60, 0), max(ymin - 60, 0)*” y “*xmax, ymax = min(xmax + 60, frame.shape[1]), min(ymax + 60, frame.shape[0])*”, donde se asegura que las nuevas coordenadas no excedan los límites del fotograma capturado.

Una vez ajustadas las coordenadas, se recorta la región del fotograma que contiene la mano usando “*recorte = frame[ymin:ymax, xmin:xmax]*”. Esto crea una subimagen que contiene únicamente la mano, excluyendo el resto del entorno.

Finalmente, para adaptar la imagen recortada a las dimensiones requeridas por el modelo de inferencia, se redimensiona esta subimagen a 640x640 píxeles utilizando “*recorte = cv2.resize(recorte, (640, 640), interpolation=cv2.INTER_CUBIC)*”. La interpolación cúbica es utilizada en este paso para preservar la calidad de la imagen durante el redimensionamiento, asegurando que los detalles cruciales no se pierdan y que

la inferencia sea lo más precisa posible. Este tamaño estandarizado es necesario ya que las imágenes utilizadas para entrenar el modelo de visión artificial fueron redimensionadas al mismo tamaño de píxeles.

Figura 33

Líneas de código para realizar el recorte de la mano detectada

```

107 # Si hay mano
108 if mano == 1:
109     # Extraer coordenadas de recuadro delimitador de la mano
110     xmin, ymin, xmax, ymax = bbox
111
112     # Asignar margen
113     xmin, ymin = max(xmin - 60, 0), max(ymin - 60, 0)
114     xmax, ymax = min(xmax + 60, frame.shape[1]), min(ymax + 60, frame.shape[0])
115
116     # Recorte de la mano
117     recorte = frame[ymin:ymax, xmin:xmax]
118
119     # Redimensionamiento a 640x640 pixeles
120     recorte = cv2.resize(recorte, dsize=(640, 640), interpolation=cv2.INTER_CUBIC)

```

4.2.8 Extracción de resultados del modelo de YOLOv8

En esta sección del código, se procede a extraer los resultados del modelo de visión artificial. Esto se hace mediante fragmento de código detallado en la Figura 34, que continúa dentro del “if” anterior.

Mediante la línea de código “*resultados = model.predict(recorte, conf = 0.80)*”, se realiza la predicción sobre el recorte de la mano utilizando el modelo de visión artificial. El parámetro “*conf=0.80*” establece un umbral de confianza de 80%, indicando que solo se considerarán las predicciones con una confianza igual o superior a este porcentaje de coincidencia para eliminar falsos positivos y garantizar que la traducción sea llevada a cabo de forma óptima.

Figura 34

Líneas de código para la extracción de resultados del modelo de visión artificial

```

121
122     # Extraer resultados del modelo de YOLOv8
123     resultados = model.predict(recorte, conf=0.80)
124

```

4.2.9 Visualización de anotaciones de resultados

En la siguiente porción del código, se detalla la visualización de los resultados anotados cuando se lleva a cabo la inferencia en tiempo real. Esto se hace mediante fragmento de código detallado en la Figura 35.

Primero, el código verifica si hay resultados válidos de la inferencia utilizando la condición `if len(resultados) != 0 and resultados[0] is not None`. Aquí, `len(resultados) != 0` asegura que se haya recibido al menos un resultado, y `resultados[0] is not None` confirma que el primer resultado no sea nulo, es decir, que la inferencia haya producido un resultado válido.

Si se cumplen ambas condiciones, se procede a generar las anotaciones visuales de los resultados. Esto se logra con la línea `anotaciones = resultados[0].plot()`, donde `plot()` es una función que se encarga de superponer sobre la imagen las anotaciones correspondientes, como el cuadro delimitador y el nombre de la clase detectada (en este caso, la letra del lenguaje de señas).

Finalmente, estas anotaciones se muestran en una ventana separada denominada "RECORTE" utilizando `cv2.imshow("RECORTE", anotaciones)`. La función `cv2.imshow` abre una ventana donde se despliega la imagen con las anotaciones, permitiendo que el usuario vea en tiempo real la predicción realizada por el modelo sobre la imagen recortada de la mano y la precisión de la detección.

Figura 35

Líneas de código para visualizar los resultados y traducción de lenguaje de señas

```
125     # Si hay resultados
126     if len(resultados) != 0 and resultados[0] is not None:
127         anotaciones = resultados[0].plot()
128
129     # Mostrar las anotaciones en la ventana "RECORTE"
130     cv2.imshow( winname: "RECORTE", anotaciones)
```

4.2.10 Conversión de resultados a texto

En esta sección del código, se realiza la conversión de los resultados obtenidos en texto. Este proceso se realiza para permitir que los resultados puedan ser transformados a un formato textual, el cual posteriormente será enunciado mediante audio por el sintetizador de voz.

Antes de que esta función pueda ser utilizada es necesario definirla, como se muestra en el fragmento de código presentado en la Figura 36. La función `“resultado_a_texto(resultados)”` toma como argumento los resultados generados por el modelo de inferencia.

Dentro de la función, se recorre cada resultado individualmente mediante un bucle `“for result in resultados:”`. Para cada resultado, se verifica primero que no sea `“None”` utilizando la condición `“if result is not None:”`. Luego, se comprueba si el resultado contiene un cuadro delimitador válido (`“result.bboxes”`) y si la lista de cuadros no está vacía (`“len(result.bboxes) > 0”`). Estas verificaciones aseguran que se ha detectado algo relevante en la imagen analizada.

Si se cumplen estas condiciones, se procede a identificar la clase (letra) detectada. Esto se logra convirtiendo el índice de la clase (`“result.bboxes.cls[0]”`) a un entero (`“clase_idx”`), y posteriormente utilizándolo para obtener el nombre de la clase desde el diccionario de nombres del modelo (`“result.names[clase_idx]”`).

Finalmente, si el nombre de la clase está presente en el conjunto de nombres del modelo, la función retorna un texto que indica la letra detectada, utilizando la línea `“return f‘Letra {clase}’”`. Si no se cumplen las condiciones iniciales, la función retorna `“None”`, lo que indica que no se ha detectado ninguna letra válida.

Figura 36

Líneas de código para definir la función de conversión de resultados a texto

```

83 # Función para convertir el resultado del modelo a texto
      | usage
84 def resultado_a_texto(resultados):
85     for result in resultados:
86         if result is not None:
87             if result.bboxes is not None and len(result.bboxes) > 0:
88                 clase_idx = int(result.bboxes.cls[0])
89                 clase = result.names[clase_idx]
90                 if clase in result.names.values():
91                     return f'Letra {clase}'
92     return None

```

Una vez definida la función para convertir los resultados en texto, esta se implementa dentro del hilo principal de la inferencia, como se muestra en la porción de código presentada en la Figura 37. Este fragmento es crucial para el funcionamiento del sistema, ya que permite identificar cuando se detecta una nueva letra y asegura que la misma se enuncie correctamente.

Primero, se llama a la función “*resultado_a_texto(resultados)*” para convertir los resultados del modelo en texto, asignando el resultado a la variable *texto*. Posteriormente, se verifica si *texto* contiene un valor válido y si este texto es diferente de la última letra detectada, lo cual se hace utilizando la condición “*if texto and texto != ultima_letra:*”. Si ambas condiciones son verdaderas, significa que se ha detectado una nueva letra.

En este caso, se actualiza la variable “*ultima_letra*” con la nueva letra detectada (“*ultima_letra = texto*”). Luego, la nueva letra se coloca en la cola de voz (“*voice_queue.put(texto)*”), lo que desencadena la reproducción de audio correspondiente.

Finalmente, si no se han detectado resultados válidos (es decir, resultados está vacío o es “*None*”), se cierra la ventana de recorte utilizando “*cv2.destroyAllWindows("RECORTE")*”, lo que asegura que el sistema no mantenga ventanas abiertas innecesarias.

Este segmento de código asegura que solo se enuncien nuevas letras detectadas, evitando repeticiones innecesarias y manteniendo la eficiencia del sistema en la inferencia en tiempo real.

Figura 37

Líneas de código para realizar la conversión de los resultados a texto

```

132     # Convertir resultados a texto
133     texto = resultado_a_texto(resultados)
134     if texto and texto != ultima_letra:
135         ultima_letra = texto
136
137         # Enviar la letra a la cola de voz para ser reproducida
138         voice_queue.put(texto)
139     else:
140         cv2.destroyAllWindows("RECORTE")

```

4.2.11 Inferencia en tiempo real

En la presente sección, se explica la última parte del ciclo de procesamiento del sistema, donde se realizan las acciones finales de la inferencia y se aseguran las operaciones de limpieza al finalizar el programa. Esta sección del código asegura que el sistema funcione de manera continua y que se detenga correctamente cuando el usuario lo decida, tal y como se puede apreciar en la Figura 38.

Primero, el fotograma original, que contiene toda la imagen capturada por la cámara junto con cualquier anotación realizada, se muestra en la ventana "LENGUAJE DE SEÑAS" utilizando “`cv2.imshow()`”. Esto permite al usuario ver en tiempo real lo que el sistema está capturando y procesando.

A continuación, se implementa una condición de espera mediante “`cv2.waitKey(1)`”, que escucha si se presiona la tecla "Esc" (representada por el valor ASCII “27”). Si esta tecla es presionada, el ciclo de inferencia se interrumpe y se sale del bucle, lo que lleva a la finalización del programa.

Después de salir del bucle, se realizan varias tareas de limpieza. Primero, se detiene la captura de video utilizando “`cap.stop()`”, lo que termina el hilo de captura de video. Luego, se cierran todas las ventanas abiertas con “`cv2.destroyAllWindows()`” para liberar los recursos utilizados por las interfaces gráficas.

Finalmente, se termina el hilo del procesador de voz. Esto se logra colocando “`None`” en la cola de tareas de voz (“`voice_queue.put(None)`”), lo que indica al hilo que debe detenerse. La llamada a “`voice_thread.join()`” asegura que el hilo del procesador de voz se cierre correctamente antes de que el programa termine por completo.

Figura 38

Líneas de código para la finalización del ciclo de inferencia y limpieza de recursos

```

142     # Mostrar el fotograma original
143     cv2.imshow( winname: "LENGUAJE DE SEÑAS", frame)
144
145     # Lectura de teclado para salir del bucle
146     if cv2.waitKey(1) == 27:
147         break
148
149     # Detener la captura de video y cerrar ventanas
150     cap.stop()
151     cv2.destroyAllWindows()
152
153     # Terminar el hilo del procesador de voz
154     voice_queue.put(None)
155     voice_thread.join()

```

Una vez implementado el código explicado anteriormente, se procede a compilarlo para dar paso a la inferencia en tiempo real, tal y como se puede apreciar en la Figura 39. En esta figura, se puede observar que el sistema cuenta con dos ventanas principales: una correspondiente a la visualización de todo el entorno capturado por la cámara y otra que se despliega cuando el algoritmo detecta una mano.

La primera ventana, denominada "LENGUAJE DE SEÑAS," muestra el entorno completo capturado por la cámara. La segunda ventana, denominada "RECORTE," se

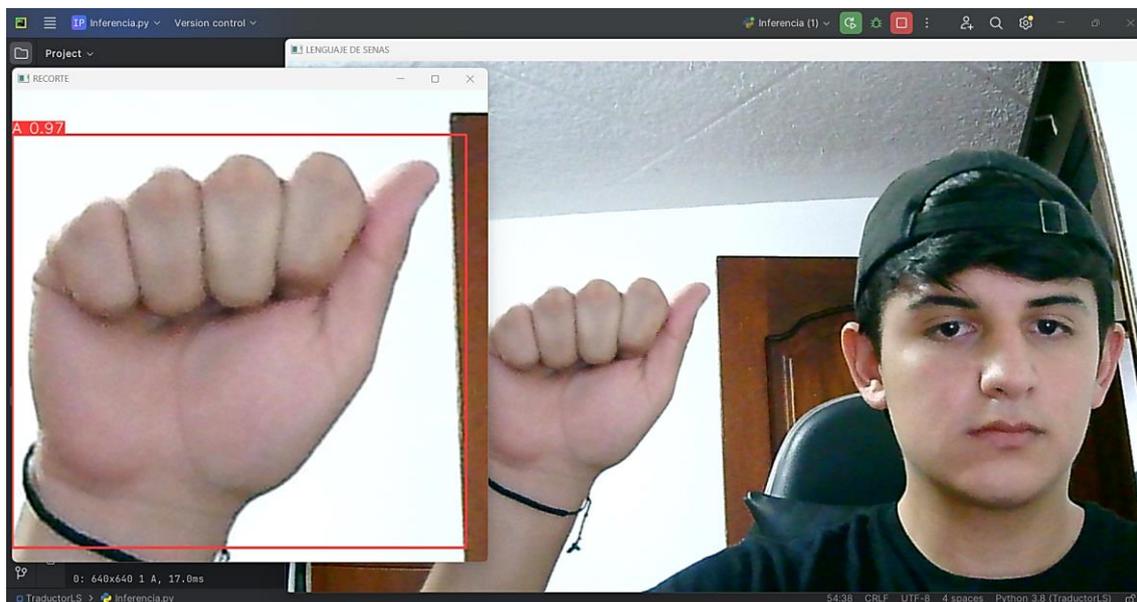
despliega cuando se detecta una mano, mostrando un recorte del área donde se encuentra la mano con una resolución de 640x640 píxeles, como se implementó en el código. Además, se realiza el seguimiento de la mano cuando se encuentra en movimiento.

Como se muestra en la Figura 39, cuando se detecta una seña del alfabeto ASL, se dibuja un cuadro delimitador alrededor de la mano y se visualizan las anotaciones de los resultados. En este caso, se ha detectado la letra "A" con un 97% de porcentaje de coincidencia. Este cuadro delimitador y la anotación proporcionan una indicación de la letra detectada, así como el nivel de confianza en la predicción realizada por el modelo de visión artificial.

Además de la visualización, el sistema incluye el sintetizador de voz que enuncia la letra detectada. Una vez que se identifica una letra diferente de la última detectada, el texto correspondiente se envía a una cola de procesamiento de voz. El sintetizador de voz, que funciona en un hilo separado, convierte este texto en audio y lo reproduce, permitiendo que la detección se anuncie verbalmente en tiempo real. Esta funcionalidad agrega una capa adicional de accesibilidad y usabilidad al sistema, facilitando la interacción con usuarios que pueden beneficiarse de la retroalimentación auditiva.

Figura 39

Inferencia en tiempo real para la traducción y reproducción de lenguaje de señas



4.3 Pruebas

En la presente sección, se detallan las pruebas realizadas para validar el correcto funcionamiento del prototipo de traducción de lenguaje de señas basado en visión artificial. Estas pruebas abarcan aspectos fundamentales del sistema, como la verificación individual de los periféricos, la detección de la mano, el reconocimiento de letras del alfabeto ASL y la integración completa del sistema en funcionamiento.

Las pruebas se resumen en la Tabla 14, donde se describen cada una de ellas y los resultados esperados, con el fin de asegurar que el sistema cumple con los requerimientos establecidos y funciona de manera eficiente en diversos escenarios de uso.

Tabla 14*Pruebas realizadas para la validación del sistema de traducción de lenguaje de señas*

ID	Prueba	Descripción	Resultado esperado
PP	Pruebas individuales de periféricos	Se verifica el funcionamiento individual de cada componente del sistema: la cámara, el monitor y los parlantes. Estas pruebas aseguran que cada periférico opere de manera adecuada antes de integrarse al sistema completo.	Los periféricos deben funcionar correctamente. La cámara debe capturar imágenes, el monitor debe mostrar los gestos, y los parlantes deben reproducir el sonido sin fallos.
PM	Pruebas de validación de detección de mano	Se comprueba que el sistema detecte y siga la mano del usuario en diferentes posiciones: mano recta, en ángulo, parcialmente visible, cerrada en puño, y en movimiento.	El sistema debe detectar la mano en todas las posiciones y seguir sus movimientos con precisión.
PA	Pruebas de detección de letras del alfabeto ASL	Se realizan los gestos correspondientes a las 26 letras del alfabeto ASL frente a la cámara. Cada gesto es repetido 10 veces para evaluar la consistencia del sistema en la detección de las letras. Se verifica que el sistema detecte cada letra en todas las repeticiones y que la traducción a audio sea correcta.	El sistema debe reconocer y traducir cada letra del alfabeto ASL con precisión, mostrando un alto porcentaje de coincidencia en las 10 repeticiones de cada gesto y reproduciendo el sonido adecuado.

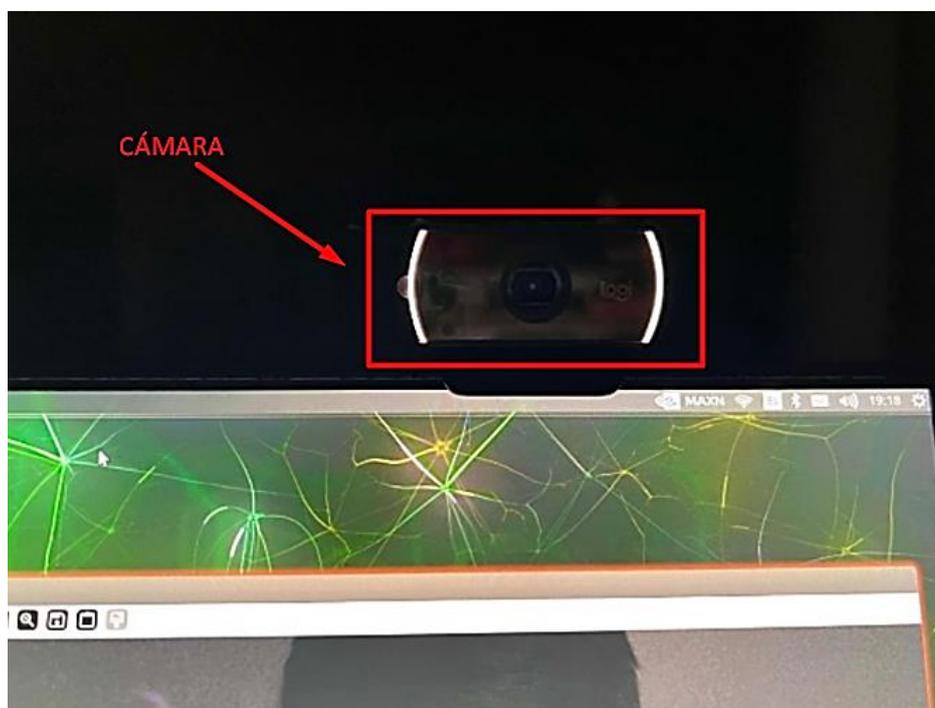
PI	Pruebas de integración del sistema completo en funcionamiento	Se evalúa el desempeño del sistema con personas con y sin discapacidad auditiva, verificando la precisión de la traducción, el tiempo de respuesta y la adaptación a diversos entornos (monocromático, ruido visual, ambientes oscuros, usuario a distancia).	El sistema debe operar de manera eficiente en todos los escenarios, detectando y traduciendo gestos de manera precisa, con tiempos de respuesta adecuados y sin perder funcionalidad en distintos entornos.
-----------	--	---	---

4.3.1 Pruebas individuales de periféricos

En la presente sección, se realizan las pruebas necesarias para verificar el correcto funcionamiento de los periféricos esenciales del sistema. Se comienza con la cámara Logitech C922 Pro. Para validar su operatividad, se conecta la cámara a la placa embebida NVIDIA Jetson Nano y se comprueba que, al ser activada, se encienden sus luces laterales, lo cual indica que está en funcionamiento y lista para capturar imágenes. Esta indicación visual, observable en la Figura 40, confirma que la cámara ha sido correctamente activada y está en óptimas condiciones para ser utilizada en el proceso de detección y traducción de gestos en lenguaje de señas.

Figura 40

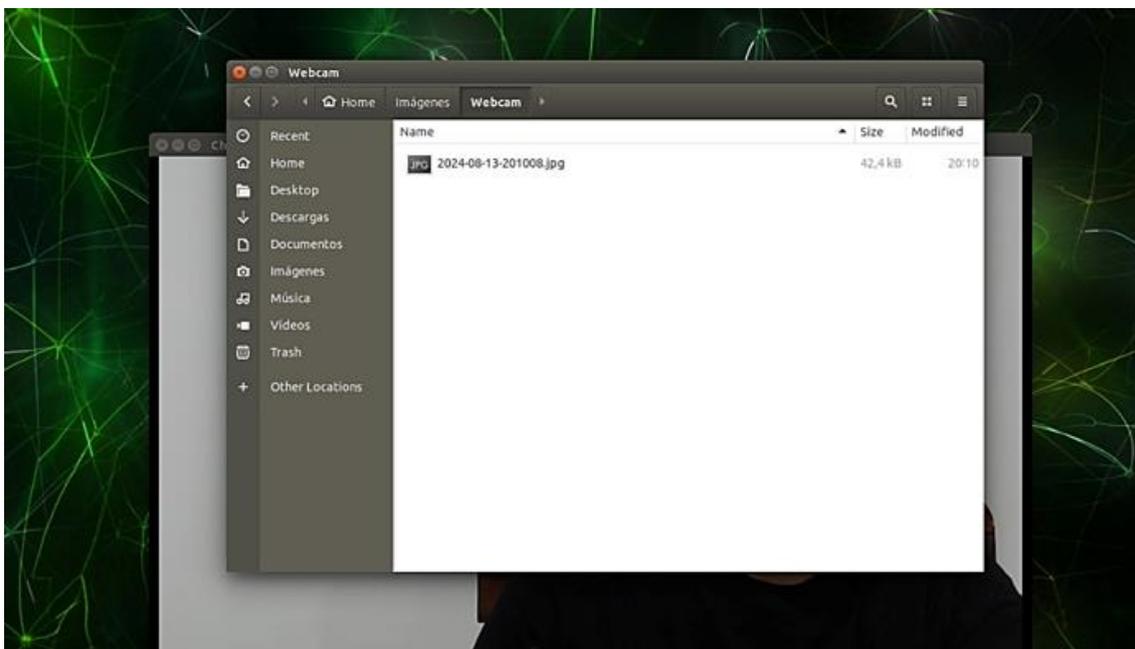
Prueba de la cámara mediante el indicador de encendido



Para complementar la validación inicial del encendido de la cámara, se llevó a cabo una segunda prueba con el objetivo de confirmar el correcto funcionamiento de la cámara Logitech C922 Pro. En esta prueba, se capturó una imagen utilizando la cámara y se verificó su almacenamiento en el sistema. Tal como se aprecia en la Figura 41, la imagen se guardó correctamente, lo que confirma que la cámara no solo está activa, sino que también puede capturar y almacenar imágenes de manera adecuada. Este procedimiento asegura que la cámara está en pleno funcionamiento y lista para ser utilizada en las etapas posteriores de detección y procesamiento de gestos en lenguaje de señas.

Figura 41

Prueba de la cámara mediante captura de imagen



Finalmente, para asegurar el correcto funcionamiento de la cámara Logitech C922 Pro, se realizó una prueba de inferencia en tiempo real del sistema de traducción de lenguaje de señas. En la Figura 42, se aprecia cómo los gestos realizados por el usuario son capturados de manera precisa por la cámara y procesados eficazmente por el sistema. La imagen muestra la representación de la mano del usuario, junto con la letra detectada, en este caso, la letra "B". Esto confirma que la cámara está plenamente operativa y cumple su función de captura de imágenes en tiempo real sin inconvenientes, garantizando que satisface los requisitos necesarios para la detección y traducción precisa de los gestos en el sistema.

Figura 42

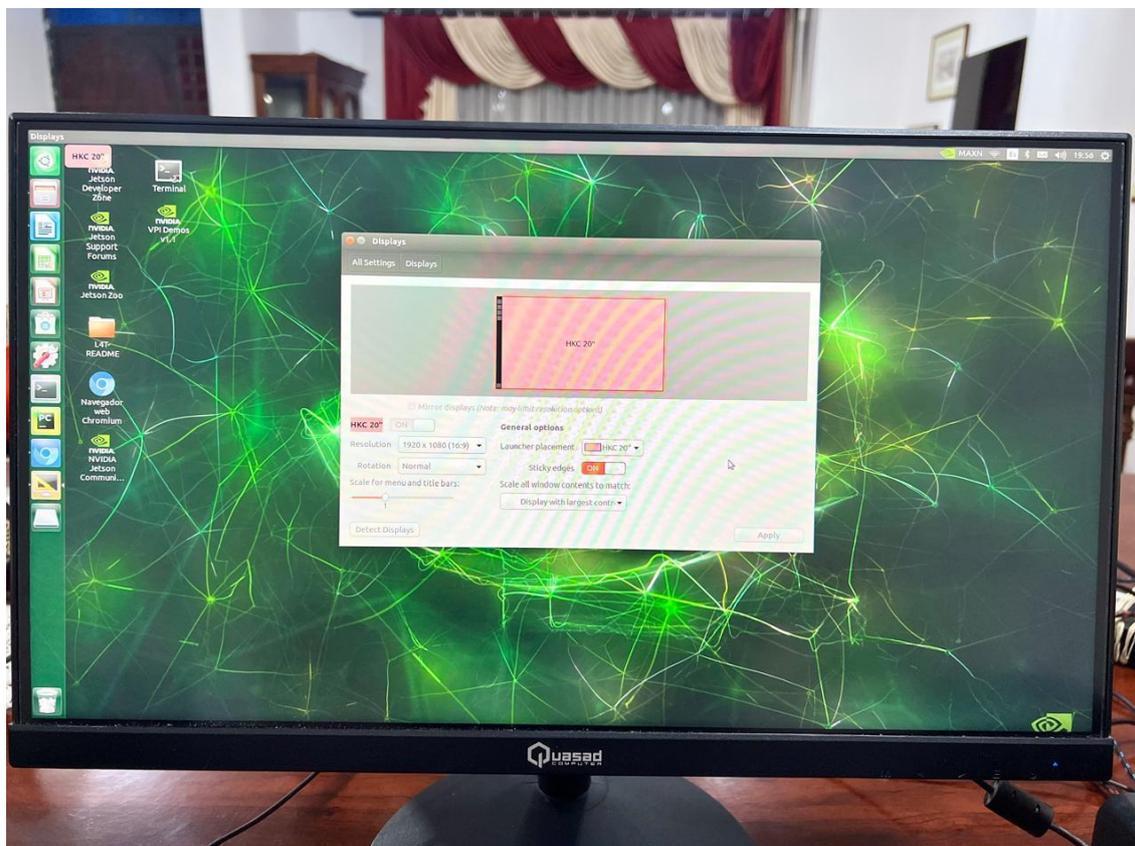
Prueba de la cámara mediante inferencia de traducción de lenguaje de señas



Posteriormente, se procede a realizar las pruebas de funcionamiento del monitor. En primera instancia, se verifica que la placa embebida NVIDIA Jetson Nano detecte el monitor de manera adecuada. Esta comprobación se realiza accediendo al apartado de pantallas en la configuración del sistema, donde se confirma que el monitor ha sido correctamente reconocido y configurado, tal como se muestra en la Figura 43. Este paso es esencial para asegurar que el monitor está debidamente conectado y listo para ser utilizado en la visualización de la traducción de los gestos de lenguaje de señas.

Figura 43

Prueba del monitor mediante detección del periférico en la placa embebida



Para concluir las pruebas del monitor, se procede a verificar que, durante la inferencia en tiempo real de la traducción de lenguaje de señas, los gestos se visualicen correctamente en este periférico. Como se muestra en la Figura 44, el monitor permite una visualización clara y completa de los gestos realizados con la mano, en este caso, la letra "A" detectada. Esto confirma que el monitor está funcionando de manera óptima y es capaz de mostrar con precisión tanto la imagen capturada como las anotaciones generadas por el sistema, lo cual es esencial para la interacción y el uso eficaz del prototipo.

Figura 44

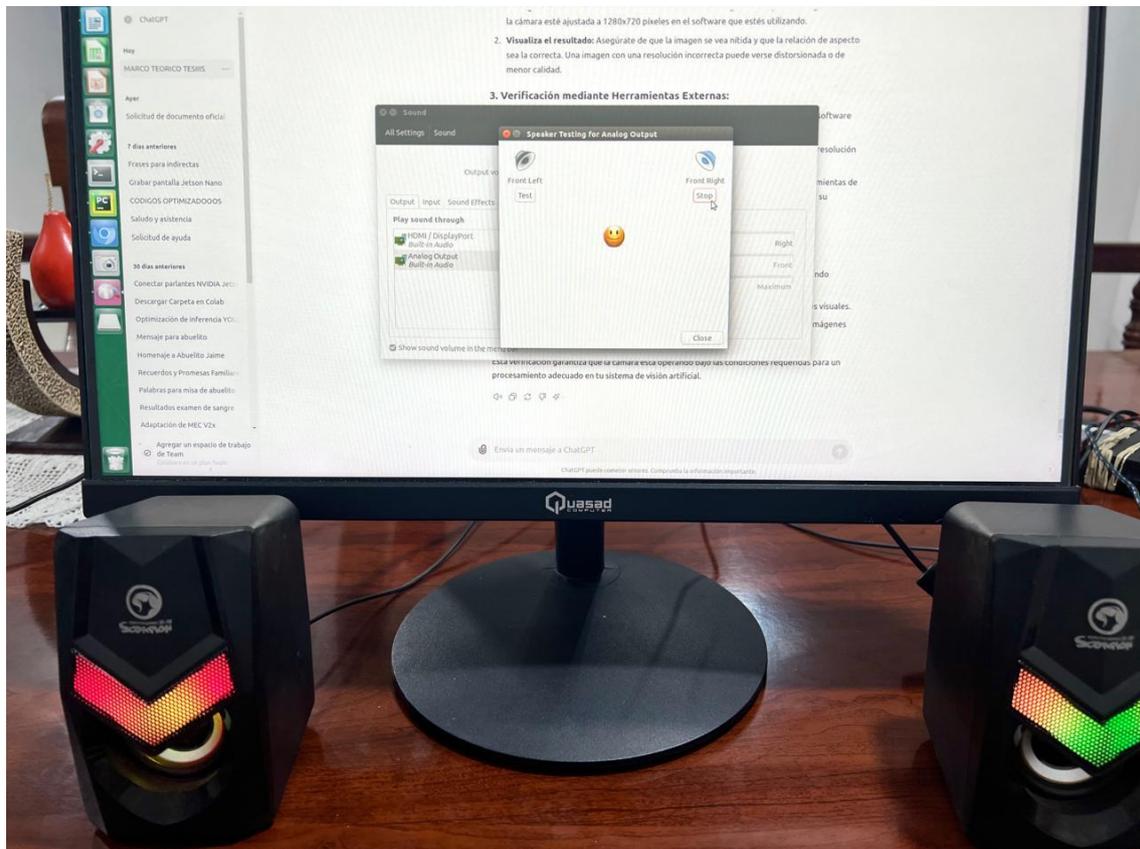
Prueba del monitor mediante inferencia de traducción de lenguaje de señas



A continuación, se llevan a cabo las pruebas de funcionamiento de los parlantes del sistema. Se verifica que los parlantes sean reconocidos correctamente por el sistema operativo de la placa embebida NVIDIA Jetson Nano. Una vez confirmada su detección, se realiza una prueba de sonido mediante una función de prueba de altavoces integrada en el sistema, la cual emite un sonido de verificación. Como se muestra en la Figura 45, los parlantes reproducen el sonido de prueba de manera correcta, lo que confirma su adecuado funcionamiento. Esta prueba garantiza que los parlantes están en óptimas condiciones para su uso en la reproducción de audio durante la traducción de lenguaje de señas en tiempo real.

Figura 45

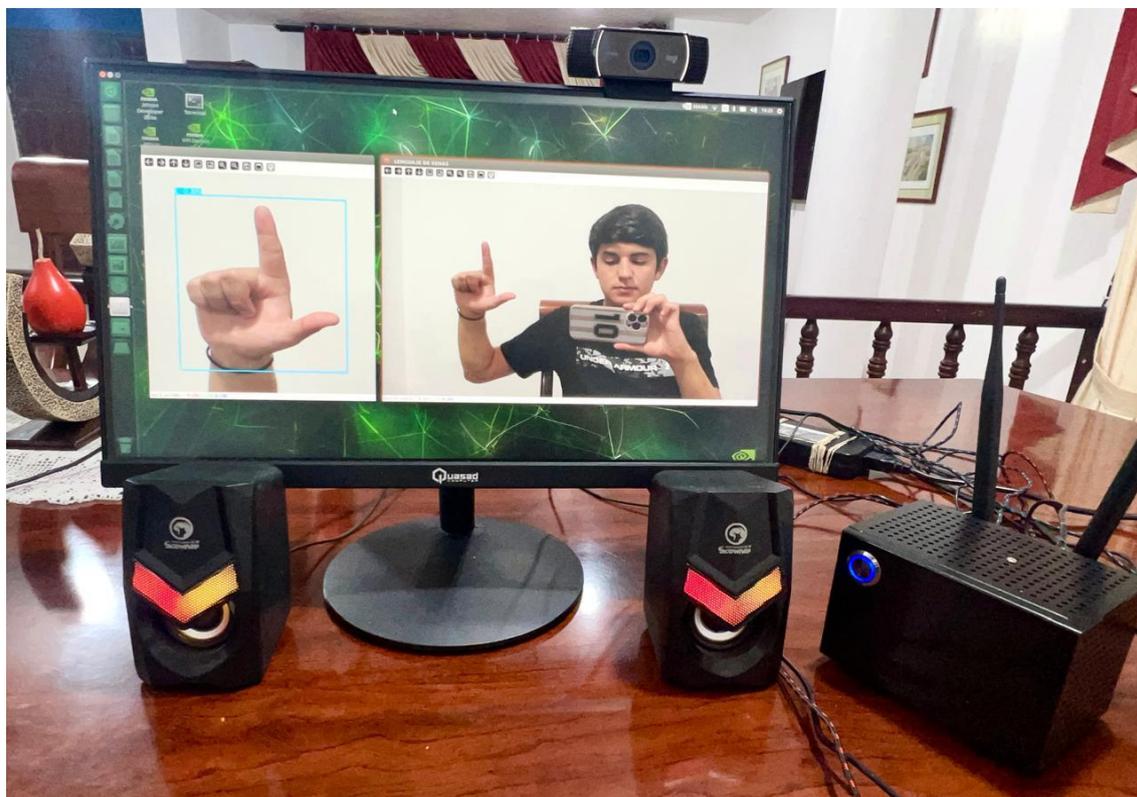
Prueba de los parlantes mediante sonido de prueba proporcionado por el sistema



Con las pruebas individuales de los periféricos completadas, se confirma que cada componente del sistema funciona correctamente. En la Figura 46, se puede observar el sistema completo de traducción de lenguaje de señas en pleno funcionamiento. La cámara captura los gestos realizados por el usuario, el monitor muestra la letra detectada, y los parlantes reproducen el sonido correspondiente. Además, se aprecia la placa embebida NVIDIA Jetson Nano, que actúa como el núcleo de procesamiento del sistema. La sincronización y correcto funcionamiento de todos estos elementos verifican que el prototipo para la traducción de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial opera de manera eficiente y cumple con los requisitos establecidos para su uso.

Figura 46

Sistema de traducción de lenguaje de señas en funcionamiento simultáneo



Los resultados obtenidos en las pruebas de funcionamiento de los periféricos del sistema se resumen en la Tabla 15, la cual detalla la verificación individual de cada componente, incluyendo la cámara, el monitor y los parlantes. Como se puede observar, todas las pruebas realizadas mostraron resultados satisfactorios, confirmando que cada periférico del prototipo de traducción de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial funciona correctamente.

Tabla 15*Resultados de las pruebas individuales de periféricos del sistema*

Pruebas individuales de periféricos (PP)			
Periférico	Prueba realizada	Resultado obtenido	Funciona correctamente (Si/No)
Cámara	PP1: Encendido y detección visual de la cámara	Luces laterales encendidas	Si
	PP2: Captura de imagen	Imagen guardada correctamente	Si
	PP3: Inferencia en tiempo real	Gestos capturados y procesados	Si
Monitor	PP4: Detección del monitor por la placa embebida	Monitor reconocido en la configuración	Si
	PP5: Visualización de gestos	Gestos visualizados claramente	Si
Parlantes	PP6: Detección y prueba de sonido	Sonido emitido correctamente	Si
Sistema completo	PP7: Funcionamiento simultáneo	Todos los periféricos funcionando juntos	Si

4.3.2 Pruebas de validación de detección de mano

La presente sección tiene como objetivo verificar que el sistema detecte correctamente la mano del usuario. Para ello, se llevan a cabo pruebas que consisten en variar la posición y la visibilidad de la mano para evaluar el rendimiento del sistema en diferentes escenarios. Se contemplan varios casos, incluyendo la mano en posición recta,

en ángulo, parcialmente visible, cerrada en puño y en movimiento. Además, se verificará que el seguimiento de la mano se realice de manera precisa mientras esta se desplaza a lo largo del entorno capturado por la cámara. Estas pruebas son esenciales para asegurar que el sistema pueda detectar y seguir con precisión la mano del usuario en diversas condiciones de uso.

Además de evaluar distintos casos específicos de posiciones de la mano, también se verificará la detección de la mano en diversas condiciones: ambiente oscuro, fondo con ruido visual y cuando el usuario se encuentra alejado de la cámara. Estas pruebas son esenciales para asegurar que el sistema detecte la mano correctamente en una amplia variedad de situaciones y entornos, lo que garantiza su adaptabilidad y robustez en condiciones que pueden variar significativamente en el uso real del prototipo.

Para llevar a cabo las pruebas de validación de detección de mano de manera más efectiva, se realizan modificaciones en el código de la inferencia. Específicamente, se cambian los valores señalados en los recuadros rojos de “*False*” a “*True*”, como se muestra en la Figura 47. Estos cambios permiten que, al compilar el código, se dibujen los 21 puntos de referencia de la mano utilizando la librería MediaPipe. Estos puntos representan las articulaciones clave de la mano, como las puntas de los dedos, las uniones de los nudillos y la base de la palma, lo que facilita el seguimiento del movimiento de la mano en términos de visión artificial. Además, se dibuja un cuadro delimitador alrededor de la mano, lo que permite verificar si la detección se ha realizado correctamente.

Estos ajustes en el código son cruciales para visualizar de manera precisa la detección y el seguimiento de la mano durante las presentes pruebas, asegurando así la fiabilidad del sistema en su capacidad de reconocer y rastrear la mano del usuario.

Figura 47

Cambios en el código para dibujar puntos de la mano y cuadro delimitador

```

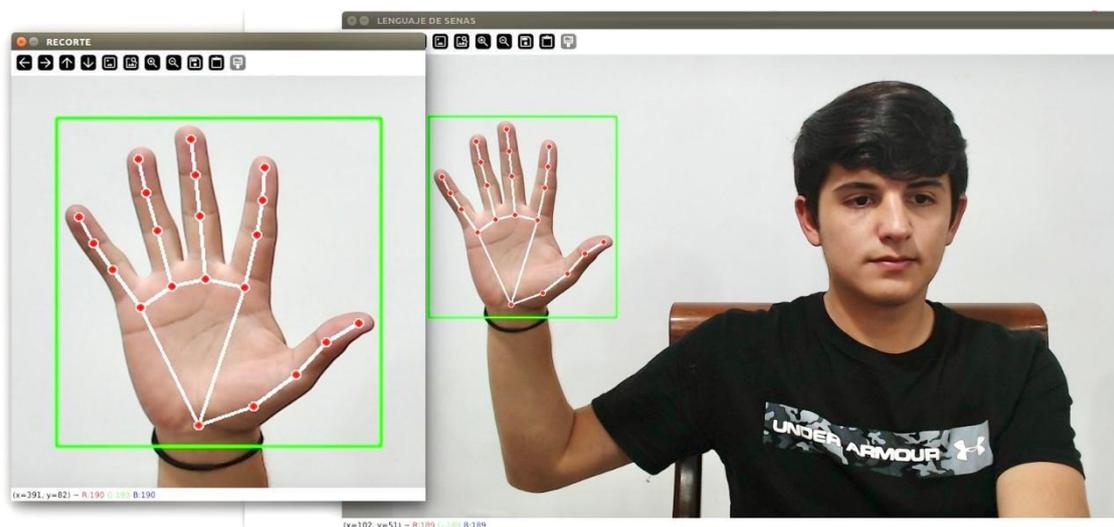
100
101 # Extraer información de la mano
102 frame = detector.encontrarmanos(frame, dibujar=True)
103
104 # Posición de una sola mano
105 lista1, bbox, mano = detector.encontrarposicion(frame, ManoNum=0, dibujarPuntos=False, dibujarBox=True, color=[0, 255, 0])
106

```

En primer lugar, se llevan a cabo pruebas de detección de la mano en posición recta en un entorno ideal, es decir con buena iluminación y con un fondo monocromático, evaluando la capacidad del sistema para identificar correctamente esta configuración básica. En la Figura 48, se puede observar claramente cómo el sistema detecta los 21 puntos de referencia que conforman la mano, los cuales corresponden a las principales articulaciones y extremidades de los dedos, así como a la base de la palma. Además, el cuadro delimitador que rodea la mano en la imagen refuerza la evidencia de que el sistema la ha identificado con precisión. La correcta visualización de estos elementos confirma que el sistema es capaz de detectar la mano en posición recta de manera efectiva, estableciendo una base sólida para continuar con pruebas en otras posiciones y movimientos.

Figura 48

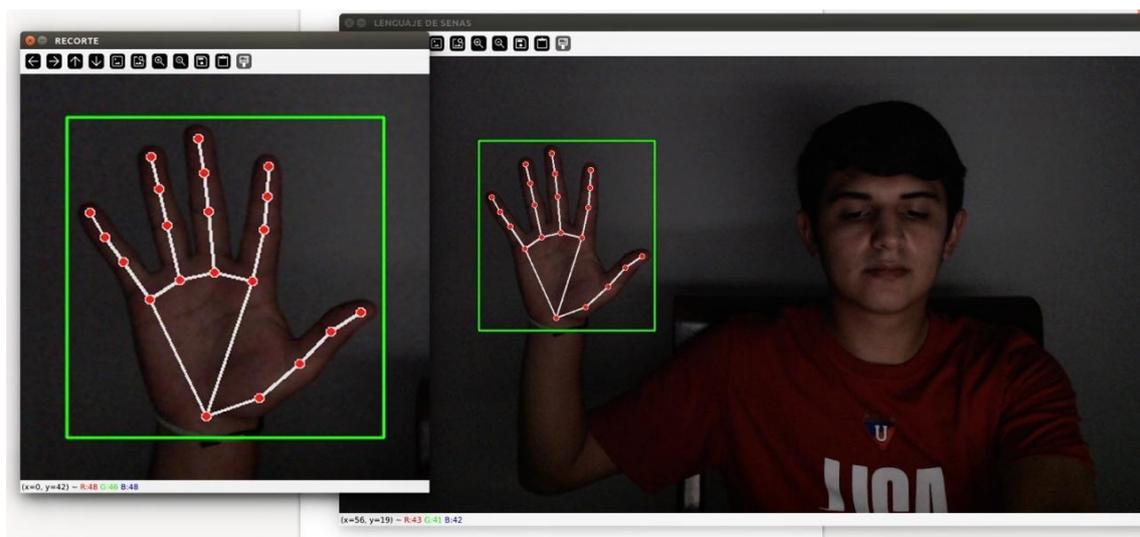
Pruebas de validación de detección de mano en posición recta



A continuación, se procede a evaluar la capacidad del sistema para detectar la mano en posición recta en un entorno con baja iluminación. Tal como se muestra en la Figura 49, el sistema sigue identificando correctamente los 21 puntos de referencia que componen la mano, a pesar de la iluminación limitada. Aun bajo estas condiciones adversas, el cuadro delimitador alrededor de la mano es claramente visible, lo que confirma que el sistema es capaz de detectarla con precisión. Esto evidencia que el prototipo puede funcionar de manera efectiva en ambientes con poca luz, garantizando su adaptabilidad a diversas condiciones.

Figura 49

Pruebas de validación de detección de mano en posición recta y en ambiente oscuro

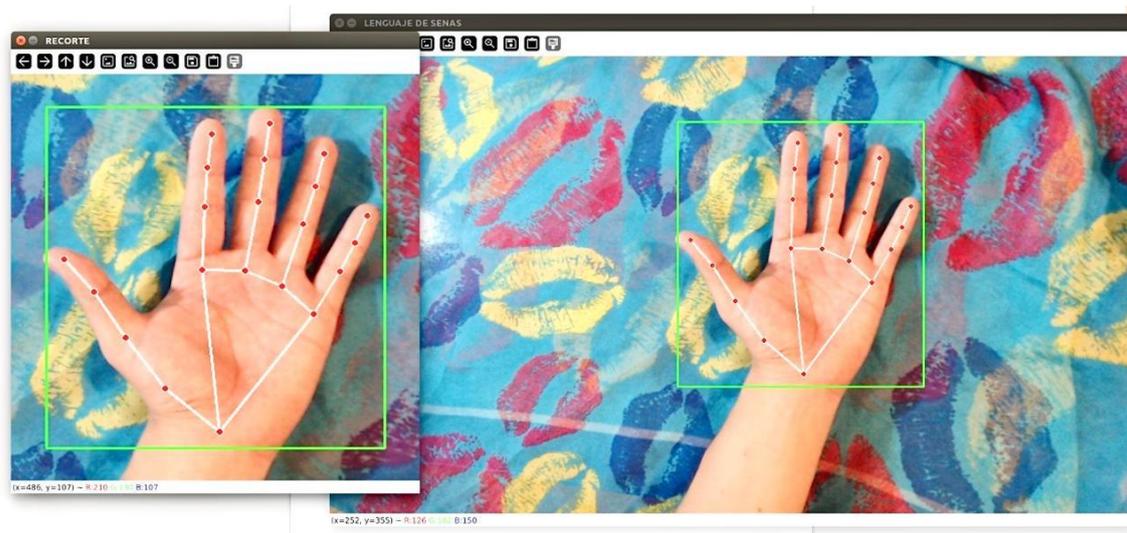


A continuación, se verifica la capacidad del sistema para detectar la mano en posición recta con un fondo que contiene ruido visual. Como se muestra en la Figura 50, el sistema continúa identificando correctamente los 21 puntos de referencia de la mano, a pesar de la complejidad y los colores llamativos del fondo. El cuadro delimitador aparece claramente alrededor de la mano, lo que confirma que el sistema distingue la mano con precisión, incluso en un entorno con elementos visuales distractores. Esto

evidencia la solidez del sistema para operar en escenarios con fondos no homogéneos, garantizando una detección efectiva en entornos visualmente variados.

Figura 50

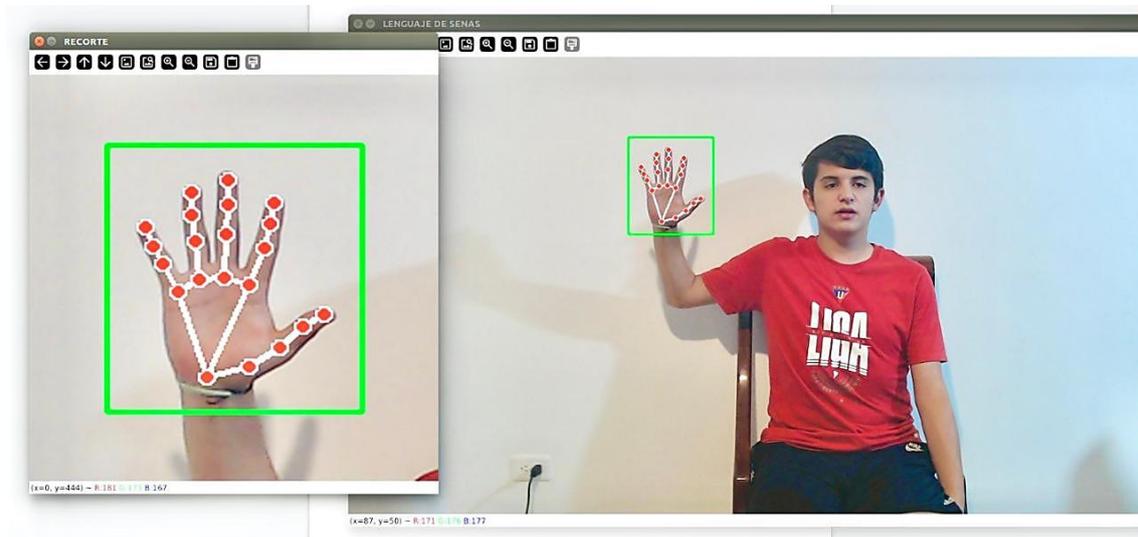
Pruebas de validación de detección de mano en posición recta y fondo con ruido visual



Luego, se verifica la capacidad del sistema para detectar la mano en posición recta cuando el usuario se encuentra más alejado de la cámara. Como se puede observar en la Figura 51, el sistema logra identificar con precisión los puntos clave de la mano y dibujar el cuadro delimitador, a pesar de la distancia. Este resultado evidencia que el sistema conserva su capacidad de detección eficaz incluso cuando el usuario no está cerca de la cámara, lo cual es fundamental para asegurar su correcto funcionamiento en entornos donde la distancia entre el usuario y la cámara puede variar considerablemente.

Figura 51

Pruebas de validación de detección de mano en posición recta y usuario alejado de la cámara



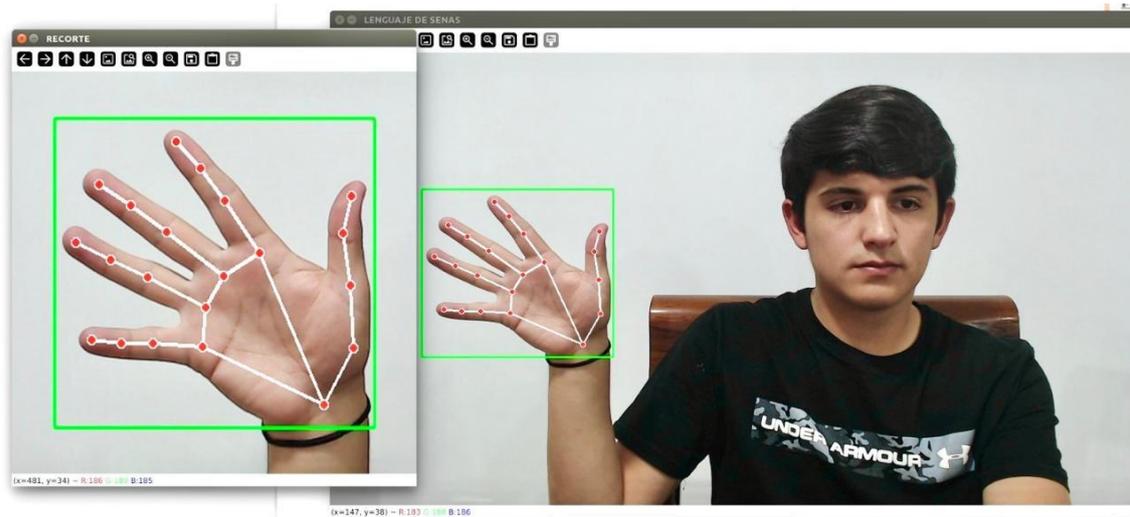
Después de completar la validación de la detección de la mano en posición recta, se procede con las pruebas utilizando configuraciones más complejas, comenzando con la mano en posición angular. Cabe destacar que, a partir de este punto, todas las configuraciones y posiciones de la mano serán evaluadas bajo diversas condiciones, tales como: ambiente oscuro, fondo con ruido visual y cuando el usuario se encuentra alejado de la cámara. Sin embargo, para evitar que la documentación se extienda innecesariamente y se torne repetitiva, en esta sección solo se presentarán los resultados obtenidos en un entorno ideal, es decir, con buena iluminación y fondo monocromático. Los resultados correspondientes a las diferentes condiciones serán presentados en la Tabla 16.

Como se aprecia en la Figura 52, el sistema detecta correctamente los 21 puntos de referencia que conforman la mano, incluso cuando se encuentra en una posición con ángulo. El cuadro delimitador alrededor de la mano sigue siendo claramente visible, lo que confirma que la mano ha sido detectada con precisión a pesar de la complejidad

añadida por la inclinación. Esto demuestra la capacidad del sistema para mantener un rendimiento eficiente en la detección de la mano, independientemente de la variación en su orientación.

Figura 52

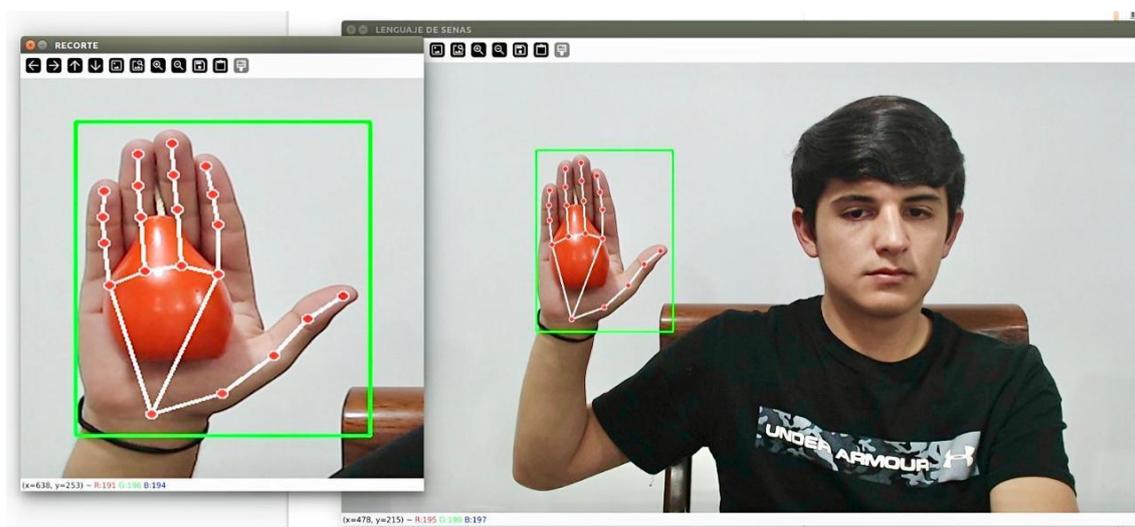
Pruebas de validación de detección de mano en posición con ángulo



Después de verificar la detección de la mano en una posición con ángulo, se procede a probar el sistema con la mano parcialmente visible, evaluando su capacidad para detectar la mano incluso cuando no está completamente expuesta. En la Figura 53, se observa que a pesar de la presencia de un objeto delante de la mano, el sistema logra identificar correctamente los 21 puntos de referencia y delimitar la mano con el recuadro verde. Esto indica que el sistema es capaz de detectar la mano de manera precisa, incluso cuando parte de esta está obstruida por un objeto, demostrando su robustez en situaciones donde la visibilidad es limitada.

Figura 53

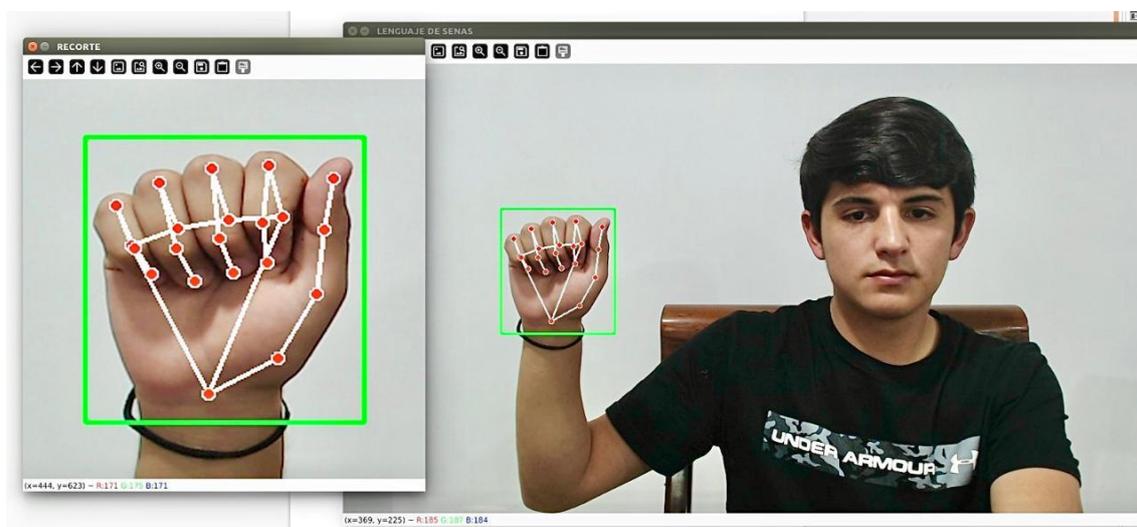
Pruebas de validación de detección de mano parcialmente visible



Luego de las pruebas con la mano parcialmente visible, se procede a evaluar la detección de la mano cuando esta está cerrada en puño. En la Figura 54, se puede observar cómo el sistema logra identificar correctamente los 21 puntos clave de la mano, incluso cuando los dedos están recogidos, y delimita la mano con el recuadro verde. Este resultado confirma que el sistema puede detectar la mano de manera precisa, incluso cuando esta adopta configuraciones más complejas como la de un puño cerrado, lo que demuestra su capacidad para adaptarse a diversas posturas de la mano.

Figura 54

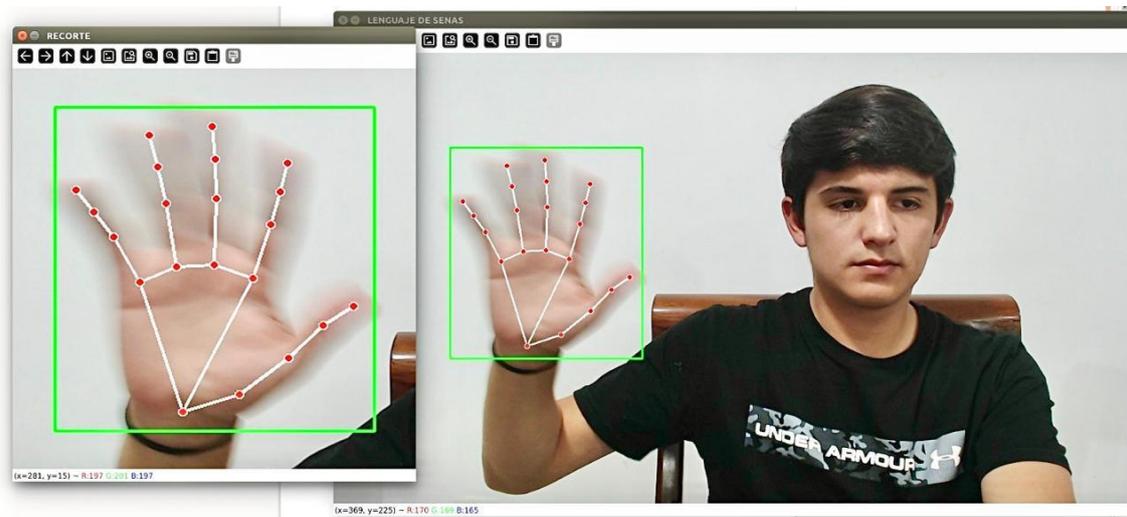
Pruebas de validación de detección de mano cerrada en puño



Después de evaluar la detección de la mano cuando está cerrada en puño, se procede a probar el rendimiento del sistema con la mano en movimiento. En la Figura 55, se puede observar que, a pesar del desplazamiento y la distorsión causada por el movimiento, el sistema logra detectar e identificar correctamente los 21 puntos de referencia de la mano, así como el cuadro delimitador. Este resultado confirma que el sistema es capaz de seguir y detectar la mano de manera precisa incluso cuando esta está en movimiento, lo que es fundamental para garantizar la efectividad del sistema en aplicaciones dinámicas.

Figura 55

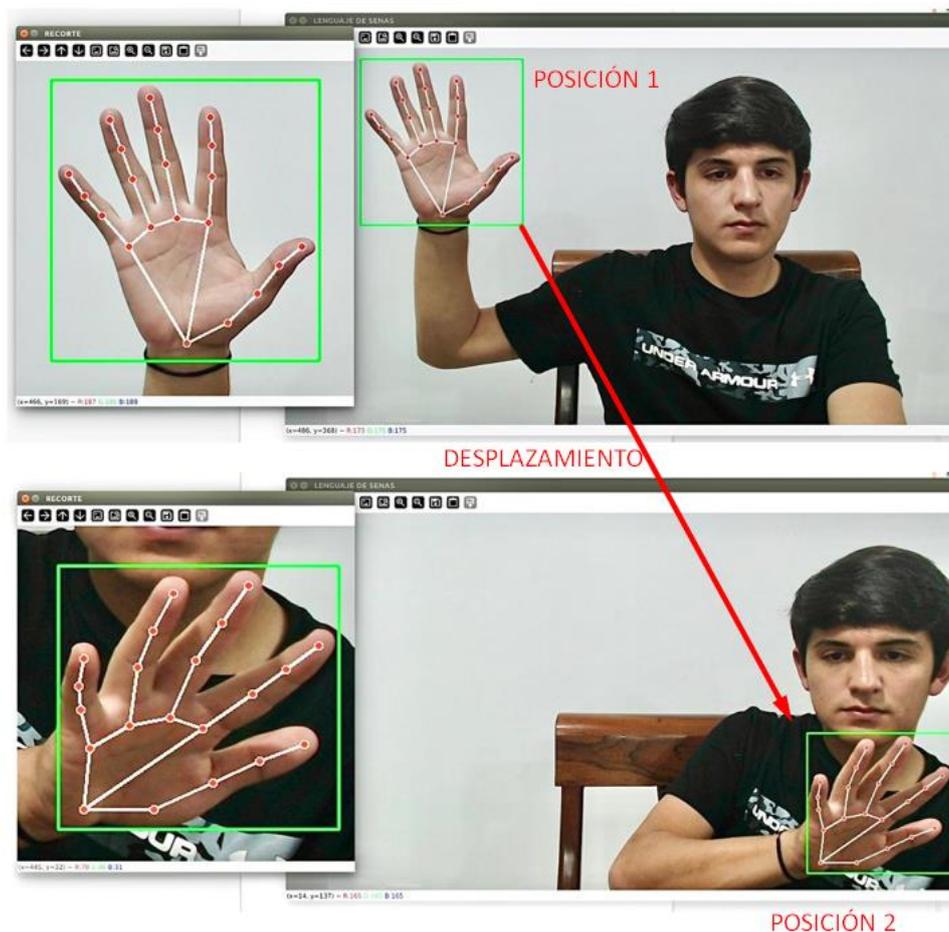
Pruebas de validación de detección de mano en movimiento



En la etapa final de las pruebas de validación de detección de mano, se llevó a cabo una prueba para verificar la capacidad del sistema para seguir y detectar la mano durante su movimiento y desplazamiento a lo largo del entorno capturado por la cámara. La Figura 56 ilustra esta prueba, en la cual la mano se desplaza desde la esquina superior izquierda (Posición 1) hasta la esquina inferior derecha (Posición 2). Durante este movimiento, el sistema logró realizar un seguimiento continuo y preciso de la mano, manteniendo su detección en todo momento. En la ventana “RECORTE”, se observa cómo la mano es detectada correctamente durante todo el desplazamiento, lo que confirma la efectividad y capacidad del sistema para seguir la mano en movimiento y asegurar su detección en todo el campo visual capturado por la cámara.

Figura 56

Pruebas de validación de seguimiento de la mano



En la Tabla 16, se presenta un resumen de los resultados obtenidos en las pruebas de validación de detección de mano bajo diversas condiciones y escenarios. A través de esta tabla, se puede apreciar que, en todos los casos evaluados, ya sea en un ambiente ideal, con poca iluminación, con un fondo con ruido visual o cuando el usuario se encuentra alejado de la cámara, el sistema fue capaz de detectar la mano de forma precisa. En cada caso, se visualizan tanto el cuadro delimitador como los puntos clave de referencia de la mano, lo que demuestra que el sistema es robusto y adaptable a diferentes condiciones de uso. Estos resultados confirman la capacidad del sistema para detectar y seguir la mano con precisión, independientemente de las variaciones en el entorno, garantizando su funcionamiento fiable en escenarios reales.

Además, en el ANEXO C se presentan las imágenes correspondientes a cada prueba de validación de detección de mano, abarcando todos los escenarios y condiciones evaluadas. Este anexo permite observar cómo el sistema responde ante diferentes entornos, proporcionando un respaldo visual que complementa y refuerza los resultados resumidos en la Tabla 16.

Tabla 16

Resultados de las pruebas de validación de detección de mano

Pruebas de validación de detección de mano (PM)				
Escenario	Condición	Cuadro delimitador visible (Si/No)	Puntos clave de la mano (Si/No)	Detección de la mano (Si/No)
Mano recta	Ambiente ideal	Si	Si	Si
	Ambiente oscuro	Si	Si	Si
	Fondo con ruido visual	Si	Si	Si
	Usuario alejado	Si	Si	Si
Mano en ángulo	Ambiente ideal	Si	Si	Si
	Ambiente oscuro	Si	Si	Si
	Fondo con ruido visual	Si	Si	Si
	Usuario alejado	Si	Si	Si
Mano parcialmente visible	Ambiente ideal	Si	Si	Si
	Ambiente oscuro	Si	Si	Si
	Fondo con ruido visual	Si	Si	Si
	Usuario alejado	Si	Si	Si

Mano cerrada en puño	Ambiente ideal	Si	Si	Si
	Ambiente oscuro	Si	Si	Si
	Fondo con ruido visual	Si	Si	Si
	Usuario alejado	Si	Si	Si
Mano en movimiento	Ambiente ideal	Si	Si	Si
	Ambiente oscuro	Si	Si	Si
	Fondo con ruido visual	Si	Si	Si
	Usuario alejado	Si	Si	Si
Seguimiento de la mano	Ambiente ideal	Si	Si	Si
	Ambiente oscuro	Si	Si	Si
	Fondo con ruido visual	Si	Si	Si
	Usuario alejado	Si	Si	Si

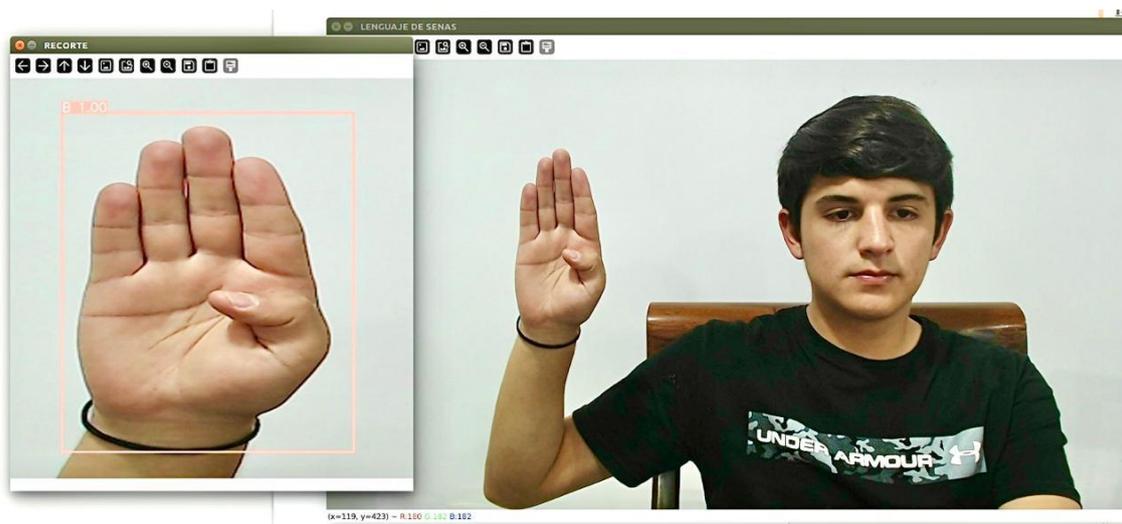
4.3.3 Pruebas de detección de letras del alfabeto ASL

En la presente sección, se realizarán las pruebas de detección de las letras del alfabeto en lenguaje de señas americano (ASL). Las pruebas consisten en ejecutar los 26 gestos correspondientes a cada letra del alfabeto ASL frente a la cámara del sistema, repitiendo cada gesto 10 veces para evaluar la consistencia del sistema en la detección de las letras. El objetivo es verificar si el sistema es capaz de identificar correctamente cada letra gestualizada en todas las repeticiones y, además, comprobar que el sintetizador de voz reproduce adecuadamente el sonido correspondiente a la letra detectada. Como ejemplo, en la Figura 57 se muestra la prueba realizada con la letra "B", donde se observa que el sistema la detecta con un 100% de coincidencia, lo que refleja un reconocimiento

preciso. Estas pruebas son esenciales para validar la eficacia del sistema en la detección y traducción de los gestos de las letras en tiempo real, garantizando su correcta funcionalidad tanto en la identificación visual como en la conversión de texto a voz.

Figura 57

Detección de letra B del alfabeto ASL



Los resultados obtenidos en la primera repetición de los 26 gestos del alfabeto ASL, correspondientes a las pruebas de detección de letras, se detallan en la Tabla 17, donde se evalúa la capacidad del sistema para identificar correctamente cada letra gestualizada y su posterior síntesis de voz. A partir de la tabla, se observa que la mayoría de las letras fueron detectadas con un alto porcentaje de coincidencia, superando el 91% en la mayoría de los casos. De las 26 letras totales, 24 fueron detectadas correctamente. Sin embargo, se identifican errores en las letras "H" y "Z", que fueron incorrectamente reconocidas como "Q". Estas detecciones incorrectas también resultaron en una síntesis de voz inexacta. Estos resultados indican que, aunque el sistema demuestra un alto nivel de precisión en general, existen ciertas letras que presentan similitudes gestuales que el modelo no distingue correctamente.

Tabla 17*Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 1*

Registro de detección de letras del alfabeto ASL – Repetición 1					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	97	Si	Si
2	B	B	100	Si	Si
3	C	C	98	Si	Si
4	D	D	94	Si	Si
5	E	E	94	Si	Si
6	F	F	91	Si	Si
7	G	G	92	Si	Si
8	H	Q	Detección incorrecta	No	No
9	I	I	98	Si	Si
10	J	J	92	Si	Si
11	K	K	95	Si	Si
12	L	L	97	Si	Si
13	M	M	96	Si	Si
14	N	N	92	Si	Si
15	O	O	98	Si	Si
16	P	P	94	Si	Si
17	Q	Q	97	Si	Si
18	R	R	93	Si	Si
19	S	S	94	Si	Si
20	T	T	92	Si	Si
21	U	U	99	Si	Si
22	V	V	96	Si	Si
23	W	W	95	Si	Si
24	X	X	95	Si	Si
25	Y	Y	98	Si	Si
26	Z	Q	Detección incorrecta	No	No

Después de realizar 10 repeticiones de las pruebas de detección de letras del alfabeto ASL, en la Tabla 18 se presentan los promedios del porcentaje de coincidencia de cada letra, junto con el porcentaje de detección correcta, la desviación estándar y el porcentaje de síntesis de voz correcta. Estos resultados proporcionan una visión clara

sobre la consistencia y precisión del sistema. Además, todos los resultados individuales de las 10 repeticiones están detallados en el ANEXO D, lo que proporciona un respaldo detallado de los datos reflejados en la Tabla 18.

Tabla 18

Aspectos evaluados en la detección de letras del alfabeto ASL tras 10 repeticiones

	Letra ASL	Promedio de porcentaje de coincidencia (%)	Porcentaje de detección correcta (%)	Desviación estándar	Porcentaje de síntesis de voz correcta (%)
1	A	97,90	100	0,88	100
2	B	99,40	100	0,97	100
3	C	97,22	90	0,97	90
4	D	94,56	90	1,33	90
5	E	92,30	100	1,42	100
6	F	92,00	90	1,58	90
7	G	94,10	100	1,10	100
8	H	95,57	70	1,13	70
9	I	98,10	100	1,29	100
10	J	92,9	80	2,03	80
11	K	95,78	90	1,20	90
12	L	98,1	100	0,74	100
13	M	95,22	90	1,39	90
14	N	95,11	90	1,90	90
15	O	98,30	100	0,95	100
16	P	93,90	100	2,18	100
17	Q	96,3	100	1,42	100
18	R	95,30	100	1,83	100
19	S	94,56	90	1,88	90
20	T	92,88	80	1,96	80
21	U	98,30	100	0,95	100
22	V	95,80	100	1,14	100
23	W	94,40	100	1,51	100
24	X	93,89	90	1,54	90
25	Y	97,10	100	1,29	100
26	Z	-	0	-	0
	PROMEDIO	95,56	90,38	1,38	90,38

Al analizar la Tabla 18, se puede observar que la mayoría de las letras presentan un porcentaje de coincidencia alto, con valores superiores al 92% y un promedio total de coincidencia del 95,56%, lo que refleja un rendimiento sólido del sistema. El porcentaje de detección correcta es también elevado en la mayoría de los casos, alcanzando el 100% para muchas letras, lo que demuestra la precisión del sistema en la identificación de gestos. En cuanto a la desviación estándar, se observa que, en general, es baja, lo que indica que el sistema mantiene resultados consistentes a lo largo de las repeticiones. Por otro lado, el porcentaje de síntesis de voz correcta es mayormente del 100%, lo que sugiere que cuando la detección de la letra es precisa, el sistema también reproduce correctamente el sonido correspondiente.

Sin embargo, la letra "Z" no fue reconocida en ninguna de las repeticiones. Esto se debe a que su gesto se realiza mediante un movimiento continuo, mientras que las otras letras se representan con posturas fijas. El modelo de visión artificial YOLOv8, que fue entrenado utilizando un dataset compuesto por imágenes estáticas, no está optimizado para procesar gestos en movimiento. Esta limitación explica la dificultad del sistema para detectar la letra "Z", ya que su reconocimiento requiere el análisis en tiempo real de un gesto dinámico. Por esta razón, no se ha calculado ni el promedio de porcentaje de coincidencia ni la desviación estándar para esta letra, ya que no se obtuvieron datos válidos para dichos cálculos.

A pesar de esta limitación, el sistema ha demostrado ser confiable en la detección de letras del alfabeto ASL, mostrando su eficacia en la mayoría de los gestos estáticos. Sin embargo, su rendimiento en la detección de gestos dinámicos podría mejorarse en futuras versiones, mediante la integración de modelos y equipos de hardware más avanzados que permitan la detección de movimientos en tiempo real, sin comprometer la eficiencia del sistema.

4.3.4 Pruebas de integración del sistema completo en funcionamiento

En esta sección, se llevarán a cabo las pruebas de integración del sistema completo en funcionamiento. El objetivo de estas pruebas es validar el rendimiento del traductor de lenguaje de señas basado en visión artificial, tanto con usuarios con discapacidad auditiva que dominan el lenguaje de señas, como con personas sin discapacidad auditiva que no tienen conocimiento de dicho lenguaje. La inclusión de ambos grupos permite evaluar las diferencias en el desempeño del sistema en escenarios con usuarios familiarizados o no con el lenguaje de señas.

Además, las pruebas se centrarán en evaluar métricas clave como la precisión de traducción, el tiempo de respuesta del sistema y su capacidad para adaptarse a distintos entornos, como fondo monocromático, fondo con ruido visual, condiciones de baja iluminación y escenarios donde el usuario se encuentra a una distancia considerable de la cámara. Este enfoque permitirá determinar la confiabilidad, efectividad y adaptabilidad del sistema en condiciones de uso reales, cubriendo una amplia variedad de situaciones y usuarios.

Para las pruebas de integración del sistema, se seleccionó un grupo de 14 personas, compuesto por 7 personas con discapacidad auditiva y 7 sin discapacidad auditiva. Todos los participantes fueron escogidos de manera aleatoria y dieron su consentimiento para formar parte de este proceso. Además, ambos grupos presentan diversidad en términos de edad y género, lo cual es fundamental para analizar cómo el sistema se adapta a diferentes características físicas y estilos de comunicación. Esta diversidad resulta crucial para garantizar que el traductor sea inclusivo y eficaz para una amplia gama de usuarios. La variabilidad en las personas que participan en estas pruebas proporciona una visión más integral del desempeño del sistema en situaciones reales y ayuda a identificar posibles áreas de mejora.

Para la documentación de las pruebas de integración del sistema completo en funcionamiento, cada una estará identificada de acuerdo con la Tabla 19. Esta tabla asigna un ID específico a cada entorno evaluado, en correspondencia con las pruebas descritas previamente en la Tabla 14, bajo la categoría PI (Pruebas de Integración del Sistema Completo en Funcionamiento).

Tabla 19

Tipos de entornos para las pruebas de integración

Pruebas de integración del sistema completo en funcionamiento (PI)		
ID	Entorno	Descripción
PI1	Fondo monocromático	El usuario realiza las letras del alfabeto ASL frente a un fondo sin patrones, de un solo color.
PI2	Fondo con ruido visual	El usuario ejecuta las letras del alfabeto ASL frente a un fondo compuesto por múltiples colores y patrones que generan complejidad visual para el sistema de visión artificial.
PI3	Ambiente oscuro	Las letras del alfabeto ASL se realizan en condiciones de iluminación baja o deficiente.
PI4	Usuario a distancia considerable de la cámara	El usuario realiza las letras del alfabeto ASL a una distancia de 2,50 metros de la cámara.

La Tabla 20 presenta las características de los grupos de participantes en las pruebas de integración. Este grupo está conformado por 7 personas con discapacidad auditiva que dominan el lenguaje de señas y 7 personas sin discapacidad auditiva que desconocen dicho lenguaje. La tabla refleja una diversidad en cuanto a edad y género, lo

cual permite un análisis más amplio y representativo del desempeño del sistema de traducción, abarcando distintos perfiles de usuarios y sus características.

Tabla 20

Características de los participantes en las pruebas de integración

Grupo	Persona	ID Persona	Edad (años)	Género
Con discapacidad auditiva que dominan el lenguaje de señas	Persona 1	PCD1	12	Femenino
	Persona 2	PCD2	51	Masculino
	Persona 3	PCD3	81	Masculino
	Persona 4	PCD4	52	Femenino
	Persona 5	PCD5	61	Masculino
	Persona 6	PCD6	26	Masculino
	Persona 7	PCD7	17	Femenino
Sin discapacidad auditiva que no tienen conocimiento del lenguaje de señas	Persona 8	PSD1	20	Femenino
	Persona 9	PSD2	17	Femenino
	Persona 10	PSD3	57	Masculino
	Persona 11	PSD4	52	Femenino
	Persona 12	PSD5	21	Masculino
	Persona 13	PSD6	82	Femenino
	Persona 14	PSD7	58	Masculino

4.3.4.1 Pruebas de integración con usuarios con discapacidad auditiva

En esta sección se presentan las pruebas de integración realizadas con usuarios con discapacidad auditiva que dominan el lenguaje de señas. El objetivo de estas pruebas es evaluar el desempeño del sistema de traducción utilizando dos métricas clave previamente mencionadas: la precisión de traducción y el tiempo de respuesta. Estas métricas se analizarán en los cuatro entornos previamente definidos (fondo monocromático, fondo con ruido visual, ambiente oscuro y usuario a distancia) con las siete personas del grupo de pruebas.

En primer lugar, se evaluará la precisión de traducción, que hace referencia al porcentaje de confianza o coincidencia proporcionado por el modelo YOLOv8 al identificar una letra en lenguaje de señas. Este porcentaje indica el nivel de certeza del modelo sobre si la letra detectada corresponde a la gestualizada por el usuario.

En la Tabla 21, se muestran los resultados obtenidos para la métrica de precisión de traducción, donde se reflejan los promedios de cada usuario en cada entorno. A partir de estos resultados, se calculan tanto el promedio de precisión para cada entorno como la desviación estándar, que mide la consistencia de los resultados en cada ambiente. Finalmente, se presenta un promedio global de precisión de traducción que abarca todos los entornos y personas, ofreciendo una visión integral del rendimiento del sistema bajo las condiciones evaluadas.

Adicionalmente, en el ANEXO E se presentan los resultados individuales obtenidos en las pruebas de integración del sistema con usuarios con discapacidad auditiva. Estos resultados detallados permiten observar el desempeño del sistema con cada participante de manera específica en cada entorno evaluado.

Tabla 21

Resultados de precisión de traducción para usuarios con discapacidad auditiva

Pruebas de integración con usuarios con discapacidad auditiva				
Persona	Precisión de traducción (%)			
	PI1	PI2	PI3	PI4
PCD1	96,88	96,04	95,43	93,14
PCD2	95,75	95,17	94,70	92,68
PCD3	95,30	94,82	94,13	92,71
PCD4	95,79	95,29	94,50	91,48

PCD5	96,96	96,43	95,86	93,05
PCD6	96,67	96,04	95,29	92,65
PCD7	96,50	96,13	95,46	92,59
PROMEDIO	96,26	95,70	95,05	92,61
DESVIACIÓN ESTÁNDAR	0,65	0,60	0,62	0,54
PROMEDIO TOTAL	94,91			

Al analizar los resultados de la Tabla 21, se observa que la precisión de traducción en todos los entornos evaluados es alta, con un promedio total de 94,91%. Los promedios por entorno muestran variaciones mínimas, destacándose PI1 (fondo monocromático) con el mayor promedio de precisión (96,26%), mientras que PI4 (usuario a distancia considerable de la cámara) presenta el promedio más bajo (92,61%). A pesar de este leve descenso en PI4, el sistema mantiene una alta precisión en dicho entorno, lo que demuestra su capacidad para adaptarse a distintos escenarios de uso. La baja desviación estándar en todos los casos indica que los resultados son consistentes. Mediante los resultados obtenidos se puede determinar que el sistema ha demostrado un rendimiento sólido en todos los entornos, con una alta precisión de traducción, incluso cuando el usuario se encuentra a una distancia considerable de la cámara.

Después de evaluar la precisión de traducción, se procede a analizar la métrica del tiempo de respuesta. Este indicador se refiere al tiempo que tarda el sistema en procesar e inferir una letra gestualizada por el usuario. El tiempo de respuesta se mide internamente a través del CMD, que muestra el valor en milisegundos (ms) para cada inferencia realizada. Esta métrica es clave, ya que permite evaluar la eficiencia del sistema en tiempo real.

En la Tabla 22 se muestran los resultados obtenidos para esta métrica, presentando los promedios de tiempo de respuesta para cada uno de los siete usuarios en los cuatro entornos previamente definidos. Al igual que con la precisión de traducción, se han calculado los promedios de tiempo de respuesta para cada entorno, así como la desviación estándar. Además, se ha calculado un promedio global de tiempo de respuesta que abarca todos los entornos y usuarios, proporcionando una visión integral de la eficiencia del sistema en los diferentes escenarios evaluados.

Tabla 22

Resultados de tiempo de respuesta para usuarios con discapacidad auditiva

Pruebas de integración con usuarios con discapacidad auditiva				
Persona	Tiempo de respuesta (ms)			
	PI1	PI2	PI3	PI4
PCD1	957,28	967,35	982,55	991,05
PCD2	956,38	965,79	980,52	990,89
PCD3	956,45	966,51	981,79	992,05
PCD4	955,12	966,5	979,93	991,06
PCD5	956,62	966,9	980,71	989,86
PCD6	955,88	965,99	980,5	991,57
PCD7	956,03	967,45	981,44	989,38
PROMEDIO	956,25	966,64	981,06	990,84
DESVIACIÓN ESTÁNDAR	0,67	0,63	0,90	0,93
PROMEDIO TOTAL	973,70			

Al analizar los resultados de la Tabla 22, se observa que el tiempo de respuesta promedio del sistema es de 973,70 ms, manteniendo tiempos relativamente uniformes en los diferentes entornos. El entorno PI1 (fondo monocromático) presenta el menor tiempo de respuesta promedio, con 956,25 ms, mientras que el entorno PI4 (usuario a distancia considerable de la cámara) muestra el mayor tiempo, con 990,84 ms. Este comportamiento es predecible, ya que una mayor distancia entre el usuario y la cámara puede generar ligeras demoras en el procesamiento visual del sistema.

En cuanto a la desviación estándar, esta se mantiene en valores bajos en todos los entornos, lo que indica que el sistema es consistente y fiable en sus tiempos de respuesta. Los resultados reflejan que el sistema responde de manera eficiente en condiciones de uso real, por lo que el sistema ha demostrado un desempeño estable y tiempos de respuesta adecuados en todos los entornos evaluados.

4.3.4.2 Pruebas de integración con usuarios sin discapacidad auditiva

En esta sección se presentan las pruebas de integración realizadas con usuarios sin discapacidad auditiva que no tienen conocimiento del lenguaje de señas. Al igual que en la sección anterior, el objetivo de estas pruebas es evaluar el desempeño del sistema de traducción, enfocándose en las dos métricas clave previamente mencionadas: la precisión de traducción y el tiempo de respuesta. Ambas métricas serán evaluadas en los cuatro entornos previamente definidos, lo que permite realizar una comparación integral del rendimiento del sistema en diferentes condiciones con usuarios que no están familiarizados con el lenguaje de señas.

En primer lugar, se evaluará la precisión de traducción. En la Tabla 23 se muestran los resultados obtenidos para esta métrica, reflejando los promedios de precisión de cada usuario en los distintos entornos. A partir de estos datos, se han calculado tanto el promedio de precisión por entorno como la desviación estándar, que mide la consistencia

de los resultados en cada ambiente. Finalmente, se presenta un promedio global de precisión de traducción, proporcionando una visión integral del rendimiento del sistema en las condiciones evaluadas.

Adicionalmente, en el ANEXO F se presentan los resultados individuales obtenidos en las pruebas de integración del sistema con usuarios sin discapacidad auditiva. Estos resultados detallados permiten observar el desempeño del sistema con cada participante de manera específica en cada entorno evaluado.

Tabla 23

Resultados de precisión de traducción para usuarios sin discapacidad auditiva

Pruebas de integración con usuarios sin discapacidad auditiva				
Persona	Precisión de traducción (%)			
	PI1	PI2	PI3	PI4
PSD1	91,05	90,84	87,82	86,62
PSD2	89,67	88,89	89,65	87,54
PSD3	89,91	88,90	88,26	85,07
PSD4	91,11	90,00	89,12	86,60
PSD5	91,30	90,26	89,41	87,17
PSD6	90,65	89,85	88,63	86,80
PSD7	91,50	90,50	87,60	87,71
PROMEDIO	90,74	89,89	88,64	86,79
DESVIACIÓN ESTÁNDAR	0,70	0,75	0,79	0,87
PROMEDIO TOTAL	89,02			

Al analizar los resultados de la Tabla 23, se observa que la precisión de traducción en los distintos entornos tiene un promedio total de 89,02%. El entorno PI1 (fondo monocromático) muestra el mayor promedio de precisión, con un 90,74%, mientras que el entorno PI4 (usuario a distancia considerable de la cámara) presenta el menor promedio, con un 86,79%. Las desviaciones estándar oscilan entre 0,70 y 0,87, lo que indica que los resultados son consistentes entre los usuarios en cada entorno. Mediante los resultados obtenidos se puede determinar que la precisión de traducción del sistema es buena, aunque varía ligeramente según el entorno, especialmente cuando el usuario se encuentra más lejos de la cámara, lo que demuestra la capacidad del sistema para identificar correctamente los gestos en diferentes condiciones.

Después de evaluar la precisión de traducción, se procede a analizar la métrica del tiempo de respuesta. En la Tabla 24 se presentan los resultados obtenidos para esta métrica, incluyendo los promedios de tiempo de respuesta de los siete usuarios en los cuatro entornos definidos. Asimismo, se han calculado los promedios por entorno y la desviación estándar. Finalmente, se incluye un promedio global de tiempo de respuesta que abarca todos los entornos y usuarios, proporcionando una visión integral del rendimiento del sistema.

Tabla 24

Resultados de tiempo de respuesta para usuarios sin discapacidad auditiva

Pruebas de integración sin usuarios sin discapacidad auditiva				
Persona	Tiempo de respuesta (ms)			
	PI1	PI2	PI3	PI4
PSD1	959,12	969,55	986,71	994,86
PSD2	958,01	970,73	986,75	996,26

PSD3	958,24	971,49	987,49	995,22
PSD4	960,29	970,35	985,62	995,56
PSD5	959,64	970,82	988,24	993,22
PSD6	958,65	971,42	986,97	995,41
PSD7	959,19	969,49	988,33	994,23
PROMEDIO	959,02	970,55	987,16	994,97
DESVIACIÓN ESTÁNDAR	0,80	0,81	0,95	0,99
PROMEDIO TOTAL	977,92			

Al analizar los resultados de la Tabla 24, se observa que el tiempo de respuesta promedio del sistema es de 977,92 ms. El entorno PI1 (fondo monocromático) presenta el menor tiempo de respuesta promedio, con 959,02 ms, mientras que el entorno PI4 (usuario a distancia considerable de la cámara) muestra el mayor tiempo, con 994,97 ms. Este comportamiento es comprensible, ya que una mayor distancia entre el usuario y la cámara puede generar pequeñas demoras en el procesamiento visual.

En cuanto a la desviación estándar, los valores son bajos, entre 0,80 y 0,99 ms, lo que indica que los tiempos de respuesta son consistentes en todos los entornos y entre los usuarios. Los resultados de la Tabla 24 demuestran que el sistema responde de manera eficiente en diferentes condiciones, manteniendo tiempos de respuesta adecuados y consistentes, lo que lo hace viable para un uso en tiempo real.

Tras culminar las pruebas de integración del sistema completo en funcionamiento, la Tabla 25 muestra una comparación de las métricas de precisión de traducción y tiempo de respuesta entre los dos grupos evaluados: personas con discapacidad auditiva que dominan el lenguaje de señas y personas sin discapacidad auditiva que no tienen

conocimiento de dicho lenguaje. Esta tabla permite analizar las diferencias en el rendimiento del sistema en función de la experiencia y familiaridad de los usuarios con el lenguaje de señas. Se presentan los promedios de cada métrica en los cuatro entornos evaluados, facilitando un análisis comparativo del desempeño del sistema bajo condiciones similares para ambos grupos de participantes.

Tabla 25

Comparación de precisión de traducción y tiempo de respuesta entre usuarios con y sin discapacidad auditiva en los cuatro entornos evaluados

Entorno	Precisión de traducción (%)		Tiempo de respuesta (ms)	
	Personas con discapacidad	Personas sin discapacidad	Personas con discapacidad	Personas sin discapacidad
	auditiva	auditiva	auditiva	auditiva
PI1	96,26	90,74	956,25	959,02
PI2	95,70	89,89	966,64	970,55
PI3	95,05	88,64	981,06	987,16
PI4	92,61	86,79	990,84	994,97
PROMEDIO TOTAL	94,91	89,02	973,70	977,92

Los resultados de la Tabla 25 demuestran que el sistema de traducción de lenguaje de señas tiene un mejor desempeño con usuarios con discapacidad auditiva que dominan el lenguaje de señas, en comparación con aquellos sin discapacidad que no lo conocen. La precisión de traducción fue más alta en el grupo con discapacidad auditiva, con un promedio de 94,91%, frente al 89,02% del otro grupo, lo que confirma que la experiencia y familiaridad de los usuarios con el lenguaje de señas contribuyen a un mejor desempeño del sistema. Ambos grupos tuvieron tiempos de respuesta promedio menores a 1000 ms, asegurando eficiencia en tiempo real; sin embargo, el grupo con discapacidad auditiva

registró un tiempo de respuesta ligeramente menor (973,70 ms), destacando cómo la práctica y conocimiento optimizan la interacción con el sistema.

4.4 Limitaciones del sistema

En esta sección se presentan las limitaciones identificadas durante el desarrollo, implementación y pruebas del sistema de traducción de lenguaje de señas basado en visión artificial para personas con discapacidad auditiva. Estas limitaciones destacan los factores que afectan el rendimiento y la operatividad del sistema en determinadas condiciones y pueden servir como base para futuras optimizaciones y mejoras.

4.4.1 Tamaño del objeto delante de la mano

La primera limitación identificada en el sistema de traducción de lenguaje de señas se refiere al tamaño del objeto que se encuentra delante de la mano para que la detección sea efectiva. El sistema puede reconocer la mano de manera precisa cuando un objeto de hasta 13 cm se interpone parcialmente entre la mano y la cámara. Sin embargo, si el objeto mide 14 cm o más, la detección falla y la mano deja de ser identificada. Esto ocurre porque al superar el umbral de 13 cm, la visibilidad de la mano se reduce lo suficiente como para que el sistema no pueda procesarla correctamente.

Esta limitación puede afectar el rendimiento del sistema en escenarios donde los usuarios interactúan con objetos o accesorios que bloquean parcialmente la mano, como herramientas o dispositivos. En entornos más complejos, donde los usuarios deben manipular objetos de cierto tamaño, el sistema podría no ser completamente confiable en la detección.

En la Figura 58, se muestra cómo el sistema detecta correctamente la mano del usuario cuando sostiene un objeto de 13 cm frente a ella. Esto evidencia que el sistema puede identificar la mano mientras el objeto que se interpone sea de 13 cm o menos. En

contraste, la Figura 59 ilustra que, al utilizar un objeto de 14 cm delante de la mano, el sistema no logra reconocerla. Esta comparación evidencia la limitación del sistema, que pierde su capacidad de detección cuando un objeto de ese tamaño obstruye parcialmente la visibilidad de la mano.

Figura 58

Detección de la mano con un objeto de 13 cm delante

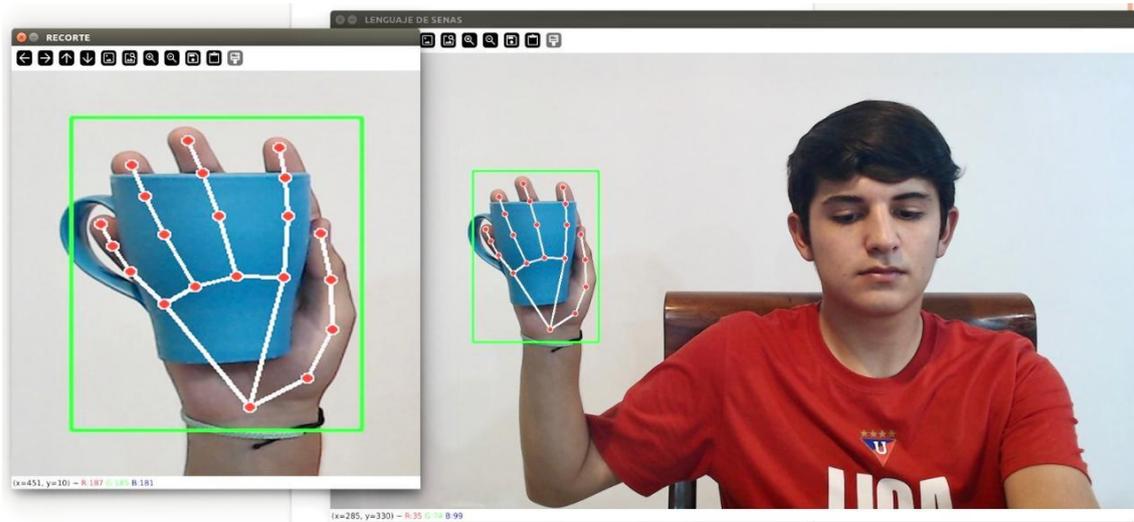
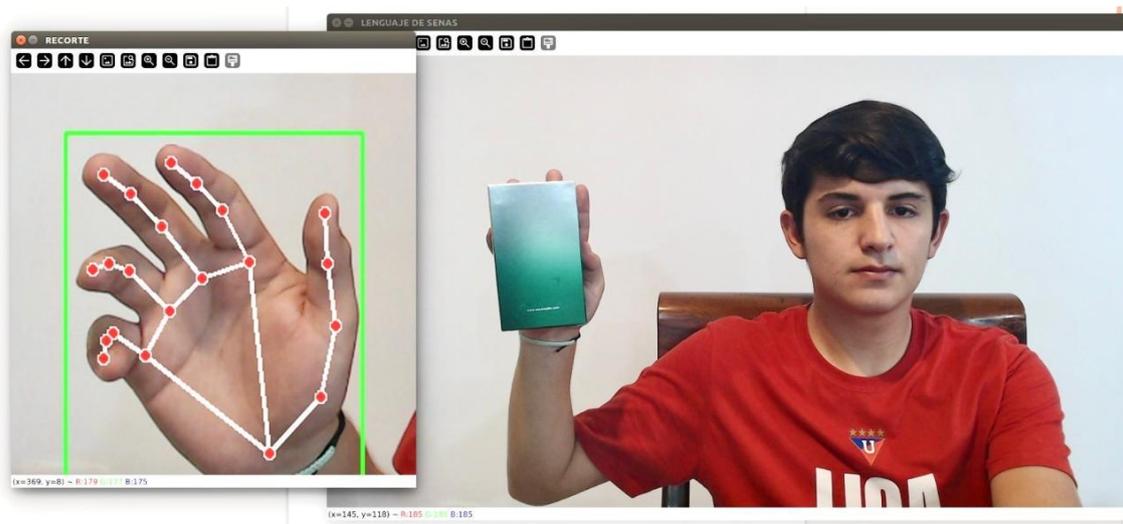


Figura 59

Falla en la detección de la mano con un objeto de 14 cm delante

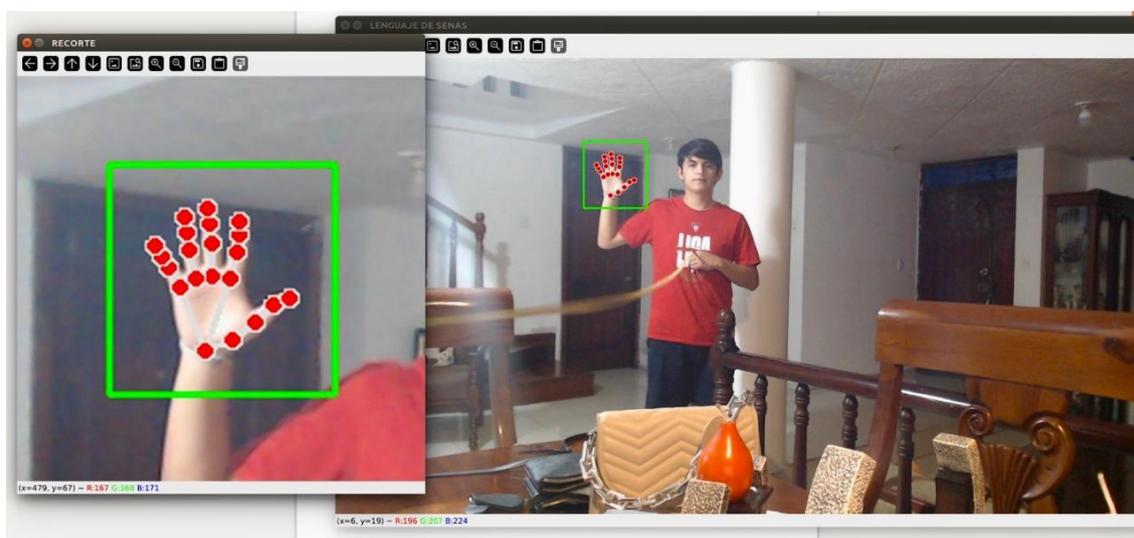
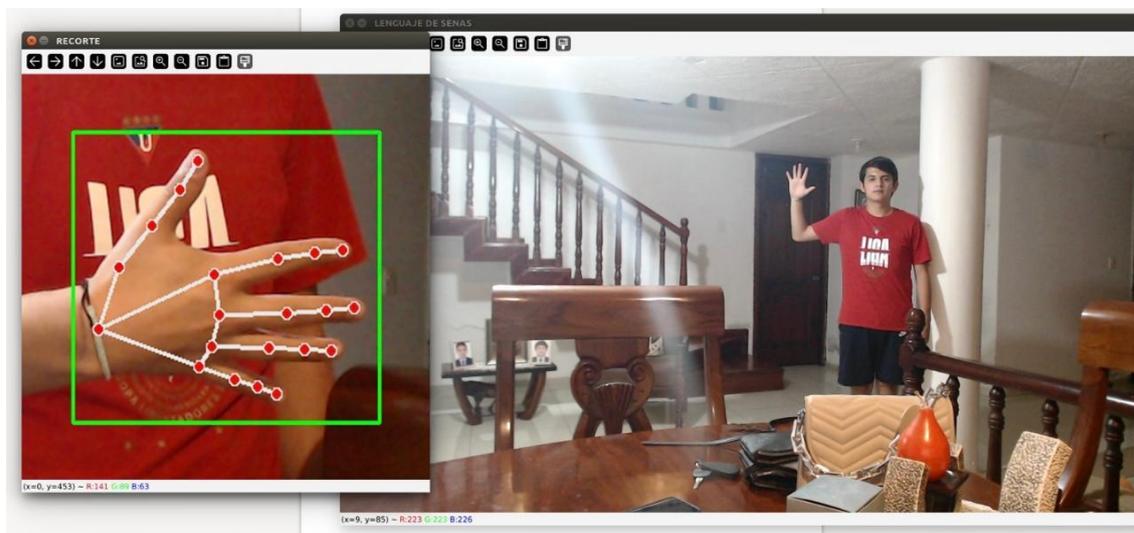


4.4.2 Distancia máxima de detección de la mano

La segunda limitación del sistema corresponde a la distancia máxima de detección de la mano. El sistema ha demostrado poder reconocer la mano del usuario hasta una distancia de 2,55 metros entre el usuario y la cámara. Sin embargo, cuando el usuario se aleja a 2,56 metros o más, la detección falla y la mano deja de ser reconocida. Esta limitación impone un rango de operación definido, lo que podría afectar la efectividad del sistema en entornos amplios o cuando el usuario se encuentra fuera de la distancia óptima.

Este aspecto es crucial para las aplicaciones prácticas del sistema, especialmente en escenarios donde el usuario pueda moverse o interactuar en un espacio amplio, como salones grandes o áreas abiertas. La restricción en la distancia máxima implica que, para mantener un rendimiento confiable, el sistema debe configurarse de manera que el usuario se mantenga dentro del rango efectivo de detección. Esto puede limitar la flexibilidad de uso y requerir ajustes en la ubicación de la cámara o en el espacio de operación.

En la Figura 60, se muestra cómo el sistema detecta correctamente la mano del usuario cuando este se encuentra a 2,55 metros de distancia de la cámara, demostrando que la detección es efectiva en ese rango. Sin embargo, en la Figura 61, se evidencia que, al aumentar la distancia a 2,56 metros, el sistema ya no logra identificar la mano del usuario. Esta comparación resalta la limitación del sistema en cuanto a la distancia máxima de detección, indicando que cualquier distancia superior a 2,55 metros impide que el sistema reconozca la mano de manera correcta.

Figura 60*Detección de la mano a una distancia de 2,55 metros***Figura 61***Falla en la detección de la mano a una distancia de 2,56 metros*

4.4.3 Detección de la letra "Z"

Una de las limitaciones del sistema es su incapacidad para detectar la letra "Z" del alfabeto ASL. A diferencia de las demás letras, que se representan mediante posturas fijas, la "Z" implica un movimiento continuo de la mano. El modelo de visión artificial YOLOv8, entrenado con un conjunto de datos basado en imágenes estáticas, no está

optimizado para procesar gestos en movimiento, lo que explica la dificultad del sistema para reconocer la "Z", ya que su reconocimiento implicaría la capacidad de analizar gestos dinámicos en tiempo real.

Sin embargo, es importante destacar que esta limitación afecta únicamente una letra del alfabeto, y el sistema sigue siendo efectivo en la detección de las demás letras con alta precisión. En aplicaciones prácticas, esta limitación puede abordarse con futuros desarrollos que incluyan modelos optimizados para movimientos, mejorando así la versatilidad del sistema sin afectar su rendimiento actual en la identificación de gestos estáticos.

4.5 Análisis Costo/Beneficio

La presente sección tiene como objetivo evaluar los costos asociados al desarrollo y mantenimiento del prototipo de traducción de lenguaje de señas en comparación con los beneficios que ofrece. Este análisis es esencial para determinar la viabilidad y el valor agregado del sistema, especialmente en cuanto a su potencial para mejorar la inclusión y accesibilidad de las personas con discapacidad auditiva. Evaluar los costos en relación con los beneficios permite comprender el impacto económico y social del prototipo, proporcionando una visión clara de su rentabilidad y sostenibilidad a largo plazo.

En la Tabla 26 se detallan los costos de hardware necesarios para el sistema de traducción de lenguaje de señas basado en visión artificial, destinado a personas con discapacidad auditiva. La tabla incluye una descripción de cada componente, la cantidad requerida, el costo unitario y el costo total. Los elementos enumerados, como la NVIDIA Jetson Nano (kit de desarrollo que incluye case y disipador térmico), la cámara Logitech c922 Pro, el monitor para visualizar en tiempo real la traducción, los parlantes para la reproducción del sintetizador de voz y el cable HDMI, son esenciales para el

funcionamiento efectivo del sistema. El costo total de estos componentes asciende a \$475 USD.

Tabla 26

Costos de hardware para el sistema de traducción de lenguaje de señas

Ítem	Descripción	Cantidad	Costo unitario (USD)	Costo Total (USD)
NVIDIA Jetson Nano	Kit de desarrollo que incluye la placa, el case y el disipador térmico, utilizado para ejecutar el modelo de detección de lenguaje de señas en tiempo real.	1	225	225
Cámara Logitech c922 Pro	Cámara HD para capturar imágenes precisas de los gestos en lenguaje de señas, necesaria para una detección confiable.	1	100	100
Monitor o pantalla digital	Dispositivo de visualización para mostrar en tiempo real la traducción del lenguaje de señas realizado por el usuario.	1	95	95
Parlantes o altavoces	Sistema de audio que permite la reproducción del sintetizador de voz, facilitando la retroalimentación auditiva de las letras detectadas.	1	40	40
Cable HDMI	Cable de conexión entre la Jetson Nano y el monitor para la visualización de los resultados del sistema.	1	15	15
TOTAL (USD)			475	475

En la Tabla 27 se detallan los costos de software necesarios para el sistema de traducción de lenguaje de señas basado en visión artificial para personas con discapacidad auditiva. Cada ítem, desde el IDE PyCharm y el lenguaje de programación Python hasta las librerías de visión artificial, la plataforma Google Colab, el dataset de imágenes y la CNN preentrenada de YOLOv8, es de acceso gratuito. Esto permite el desarrollo del sistema sin incurrir en gastos de licencia, promoviendo una implementación económica y accesible. En total, los costos de software ascienden a \$0 USD, lo cual facilita un desarrollo sostenible que mantiene la calidad y funcionalidad del sistema sin impactar el presupuesto.

Tabla 27

Costos de software para el sistema de traducción de lenguaje de señas

Ítem	Descripción	Costo Total (USD)
IDE PyCharm	Entorno de desarrollo integrado (IDE) utilizado para escribir, depurar y ejecutar el código del sistema.	0
Python	Lenguaje de programación de libre uso, que soporta librerías esenciales para visión artificial, como OpenCV, Numpy y MediaPipe.	0
Librerías	Conjunto de librerías necesarias para el funcionamiento del sistema. Las librerías utilizadas incluyen OpenCV, Imutils, Threading, MediaPipe, gTTS y OS para procesamiento de datos.	0
Google Colab	Plataforma de computación en la nube utilizada para el entrenamiento del modelo de visión artificial, ofrece recursos de procesamiento de alta capacidad,	0
Dataset de imágenes	Dataset "Real-Time-Sign-Language" utilizado en el entrenamiento del modelo de visión artificial. Contiene	0

imágenes de gestos de manos correspondientes a cada letra del alfabeto ASL		
CNN preentrenada de YOLOv8	Red neuronal preentrenada "yolov8n", permite el reconocimiento en tiempo real de los gestos de mano debido a su diseño optimizado para visión artificial.	0
TOTAL (USD)		0

En la Tabla 28 se presentan los costos de ingeniería asociados al diseño e implementación del sistema de traducción de lenguaje de señas basado en visión artificial para personas con discapacidad auditiva. La tabla incluye las actividades clave para el desarrollo del sistema, con una descripción breve de cada una, la cantidad de horas estimadas para su ejecución, el costo unitario por hora y el costo total de cada actividad. Las actividades contempladas abarcan el diseño del sistema, el entrenamiento del modelo y la programación e implementación del código.

Es relevante señalar que los valores presentados en el costo unitario de cada actividad o rol han sido determinados en base a las tarifas establecidas por el Ministerio del Trabajo del Ecuador para el salario mínimo legal de un ingeniero electrónico (Ministerio del Trabajo del Ecuador, 2021). Además, estos costos de ingeniería han sido cubiertos por el autor del proyecto, por lo que no se consideran dentro de la inversión económica que la empresa debe realizar.

Tabla 28

Costos de ingeniería para el sistema de traducción de lenguaje de señas

Actividad o rol	Descripción	Cantidad	Costo unitario (USD)	Costo Total (USD)
Diseño del sistema	Planificación y estructuración del sistema, incluyendo análisis de requerimientos y diseño de la arquitectura del sistema de detección.	40 horas	2,66	106,40
Entrenamiento del modelo	Configuración, entrenamiento y ajuste del modelo YOLOv8 en Google Colab, utilizando el dataset específico para lenguaje de señas.	30 horas	2,50	75
Programación e implementación	Desarrollo del código, integración de librerías de visión artificial (OpenCV, MediaPipe) y pruebas iniciales para asegurar el funcionamiento.	50 horas	2,66	133
TOTAL (USD)		120 horas	7,82	314,40

En la Tabla 29 se presentan los costos totales asociados al desarrollo del sistema de traducción de lenguaje de señas basado en visión artificial para personas con discapacidad auditiva. En esta tabla se realiza la sumatoria de los diferentes tipos de costos detallados previamente, incluyendo los costos de hardware, software e ingeniería. El costo total estimado asciende a \$789,40 USD.

Tabla 29*Costos totales para el sistema de traducción de lenguaje de señas*

Tipo de costo	Costo (USD)
Costos de hardware	475
Costos de software	0
Costos de ingeniería	314,40
TOTAL (USD)	789,40

El beneficio del sistema de traducción de lenguaje de señas basado en visión artificial se mide en función de su impacto en la vida de las personas a las que está dirigido, especialmente aquellas con discapacidad auditiva. En el ámbito social, promueve la inclusión en entornos familiares y comunitarios, facilitando una comunicación accesible entre personas con y sin conocimiento del lenguaje de señas. Esto permite la interacción sin necesidad de una persona que actúe como intérprete de lenguaje de señas, generando un impacto positivo en el ámbito familiar y en el entorno social.

Desde una perspectiva económica, el sistema presenta una alternativa rentable frente al costo de un intérprete de lengua de señas, cuyo servicio tiene un valor aproximado de 15 dólares por hora, según la Federación Nacional de Personas Sordas del Ecuador. Esto lo convierte en una solución accesible para personas o instituciones que requieren una herramienta de traducción constante, evitando gastos recurrentes en servicios de interpretación.

El prototipo también se destaca por su fácil ensamblaje y flexibilidad para integrarse en diversos entornos de uso variados, como hogares, aulas educativas y centros

comunitarios, permitiendo así su adaptación a distintas necesidades y condiciones. Además, aunque diseñado principalmente para personas con discapacidad auditiva, también puede ser utilizado por personas sin esta discapacidad como herramienta educativa. Esto abre oportunidades en el aprendizaje del lenguaje de señas, permitiendo a los usuarios practicar y mejorar su comprensión sin la necesidad de capacitaciones presenciales o extensas, que suelen durar entre 3 y 6 meses.

En conjunto, este sistema representa una innovación tecnológica que promueve la accesibilidad y la inclusión social, aportando además beneficios económicos y educativos. Su facilidad de implementación y bajo costo de operación lo convierten en una opción viable y sostenible, adecuada para hogares, instituciones educativas y organizaciones enfocadas en la inclusión.

4.6 Discusión

Este apartado tiene como objetivo analizar el aporte del sistema de traducción de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial en relación con proyectos similares. La revisión de otros trabajos permite contextualizar las características del presente proyecto y resaltar sus contribuciones y limitaciones en el ámbito de la traducción de lenguaje de señas, principalmente para personas con discapacidad auditiva.

El proyecto de Pinzón Bayona & Sanabria Orejuela (2021), *“Desarrollo de una aplicación móvil para traductor de lenguaje de señas mediante el uso de servicios web”*, propone una solución orientada al Lenguaje de Señas Colombiano (LSC) mediante una aplicación móvil que utiliza MediaPipe. Aunque ambos proyectos emplean esta librería, el prototipo de este estudio, basado en la NVIDIA Jetson Nano, abarca las 26 letras del

alfabeto ASL, mientras que el proyecto mencionado se limita a cinco señas, ampliando la funcionalidad y utilidad del sistema en contextos de comunicación más diversos.

Por otra parte, el proyecto “*Guante traductor de lenguaje de señas mexicano para personas sordomudas*” (Hernández et al., 2017) emplea sensores flex, acelerómetros y giroscopios en guantes para traducir el Lenguaje de Señas Mexicano (LSM). Aunque ambos proyectos buscan promover la inclusión social, el guante traductor depende de hardware específico y comunicación bluetooth, mientras que el sistema desarrollado en este proyecto se basa en visión artificial y CNNs, lo que facilita la interacción sin requerir dispositivos adicionales. Esta diferencia hace que el presente prototipo sea más accesible y menos dependiente de componentes externos, aunque el guante traductor permite reconocer frases mediante gestos complejos.

El proyecto de Guerrero (2018) “*Traductor de lenguaje de señas portátil por medio de reconocimiento de imágenes*”, utiliza una Raspberry Pi 3 para interpretar el ASL con formas estáticas de la mano. Sin embargo, el empleo de la Jetson Nano en el presente proyecto permite procesar en tiempo real, ofreciendo una traducción más eficiente. Además, el proyecto de Guerrero requiere ajustes de luminosidad antes de su uso, mientras que este sistema responde de manera confiable en entornos oscuros, aumentando su aplicabilidad en diversas condiciones.

Finalmente, el proyecto de Ortíz García (2021), “*Traductor de letras en lenguaje de señas con redes neuronales convolucionales*”, emplea el modelo ResNet50 y omite las letras "J" y "Z" debido a sus movimientos. Aunque el sistema de este proyecto también excluye la "Z" debido a las limitaciones de YOLOv8 en gestos dinámicos, incluye todas las letras estáticas, logrando una precisión de 96,26% en entornos de fondo monocromático, superando el 88% del proyecto de Ortiz.

En conjunto, el sistema desarrollado en esta investigación presenta aportes relevantes, como su capacidad para operar en múltiples entornos y realizar pruebas con usuarios con y sin discapacidad auditiva, lo que añade valor en el ámbito social y educativo. Sin embargo, algunos proyectos revisados presentan ventajas específicas, como el uso de guantes en la detección de frases y la precisión en entornos controlados.

CONCLUSIONES

El desarrollo del sistema de traducción de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial ha demostrado ser una herramienta eficaz para facilitar la comunicación entre personas con discapacidad auditiva que dominan el lenguaje de señas y aquellas que no la padecen y desconocen dicho lenguaje. Este proyecto no solo promueve la inclusión social al reducir barreras de comunicación, sino que también presenta una alternativa para el aprendizaje del lenguaje de señas, siendo útil tanto para personas con discapacidad auditiva como para aquellas interesadas en familiarizarse con esta forma de comunicación.

La selección del dataset “Real-Time-Sign-Language”, el cual cuenta con una variedad de gestos, poses y entornos, fue fundamental para alcanzar una alta precisión y adaptabilidad en el sistema. Este conjunto de datos, con un particionamiento recomendado de 80% para entrenamiento y 20% para validación, permitió entrenar el modelo YOLOv8 en un entorno robusto y representativo de situaciones reales, lo que contribuyó a su efectividad en diversas condiciones de uso.

Se determinó que el número óptimo de épocas para el entrenamiento del modelo de visión artificial es de 60, ya que entrenarlo por más de esta cantidad no proporciona mejoras significativas y puede llevar al sobrentrenamiento, mientras que entrenar con menos de 60 épocas resulta en un rendimiento insuficiente.

La arquitectura del prototipo, basada en la placa NVIDIA Jetson Nano, seleccionada por su capacidad para tareas de IA y visión artificial, proporciona el rendimiento necesario para la inferencia en tiempo real. El uso de librerías como OpenCV, MediaPipe y Google Text-to-Speech permite no solo la detección de gestos en

tiempo real, sino también la síntesis de voz de las letras detectadas. Además, la implementación de hilos de procesamiento permite ejecutar tareas de forma simultánea, optimizando la captura de video, la detección de gestos y la reproducción de audio, lo cual mejora el rendimiento del sistema.

Las pruebas realizadas en diversas condiciones y escenarios confirman la capacidad del sistema de traducción de lenguaje de señas para detectar y seguir de la mano, independientemente de las variaciones en el entorno. Además, las pruebas demostraron una alta precisión en la detección y traducción de las letras del alfabeto ASL, alcanzando un promedio de precisión de traducción del 95,56%. Las pruebas de integración con usuarios con y sin discapacidad auditiva mostraron que el sistema presenta un mejor desempeño con usuarios familiarizados con el lenguaje de señas, alcanzando un promedio de precisión de traducción de 94,91% frente al 89,02% del otro grupo. Ambos grupos presentaron tiempos de respuesta promedio menores a 1000 ms, confirmando la eficiencia del sistema para el uso en tiempo real.

A pesar del óptimo rendimiento del sistema, se identificaron ciertas limitaciones. La detección la mano se ve afectada cuando objetos de más de 13 cm interfieren en su visualización, lo que limita su uso en entornos donde los usuarios interactúan con objetos de mayor tamaño. Además, el sistema tiene una limitación de detección correcta de la mano hasta una distancia de 2,55 metros entre el usuario y la cámara. Sin embargo, cuando el usuario se aleja a 2,56 metros o más, la detección falla y la mano deja de ser reconocida, lo cual restringe su funcionamiento en espacios amplios. Finalmente, debido a que el modelo YOLOv8 fue entrenado con imágenes estáticas, el sistema no puede detectar la letra "Z", ya que su gesto implica un movimiento continuo de la mano en tiempo real.

RECOMENDACIONES

Los resultados de las pruebas realizadas evidencian una mayor precisión de traducción y menores tiempos de respuesta cuando la inferencia se realiza en un entorno con fondo monocromático. Para maximizar el desempeño del sistema, se recomienda implementar el prototipo en este tipo de entorno, minimizando las distracciones visuales que podrían interferir con la precisión de la detección.

Para mejorar la versatilidad del sistema, se sugiere que este proyecto sirva como base para investigar la inclusión de modelos que optimicen la detección de gestos en movimiento, como el correspondiente a la letra "Z". Esto requerirá el uso de dispositivos con mayores recursos computacionales, como placas embebidas avanzadas, que puedan soportar modelos de detección dinámica sin comprometer el rendimiento del sistema.

Se recomienda configurar el sistema para que la traducción de lenguaje de señas inicie automáticamente al encender la placa embebida. Esto eliminará la necesidad de intervención manual para compilar y ejecutar el código desde un entorno de desarrollo, mejorando la experiencia del usuario al simplificar el proceso de uso y garantizar un acceso inmediato a la funcionalidad del sistema.

YOLOv8 ofrece diferentes CNNs preentrenadas, pero dada la demanda computacional de la inferencia en tiempo real, se sugiere utilizar la CNN "yolov8n.pt". Esta configuración presenta una carga de recursos menor en comparación con otras opciones, manteniendo un balance entre precisión y eficiencia en sistemas embebidos como la NVIDIA Jetson Nano.

En el proceso de entrenamiento del modelo, se recomienda utilizar los recursos ofrecidos por Google Colab, como la GPU virtual NVIDIA Tesla T4. Estos recursos

permiten reducir considerablemente el tiempo de entrenamiento, representando una alternativa eficiente en comparación con equipos físicos que requieren mayor tiempo y potencia.

Dado que la resolución de 640x640 píxeles es un requerimiento para YOLOv8, se recomienda redimensionar la imagen a esta resolución. Además, para minimizar la pérdida de información en la reducción de tamaño, se sugiere implementar la interpolación cúbica, la cual mejora la calidad y precisión de la imagen sin una pérdida significativa de detalles en el proceso de inferencia.

Al analizar los resultados de las pruebas de funcionamiento del sistema, la desviación estándar proporciona información valiosa sobre la consistencia de los resultados. Este análisis permite evaluar la confiabilidad del rendimiento bajo condiciones variables, optimizando su precisión y robustez en diferentes entornos.

Al seleccionar una placa embebida adecuada para la implementación de modelos de visión artificial, se recomienda comparar múltiples opciones en función de las necesidades computacionales y de procesamiento. Esto permitirá seleccionar la placa más adecuada para los objetivos del proyecto, optimizando el rendimiento del sistema y evitando limitaciones de hardware.

REFERENCIAS

- Aguilar Alvarado, J. V., & Campoverde Molina, M. A. (2020). Clasificación de frutas basadas en redes neuronales convolucionales. *Polo del Conocimiento: Revista científico - profesional*, ISSN-e 2550-682X, Vol. 5, No. 1, 2020, págs. 3-22, 5(1), 3–22. <https://doi.org/10.23857/pc.v5i01.1210>
- Albanie, S., “ Ul Varol, G., Momeni, L., Bull, H., Afouras, T., Chowdhury, H., Fox, N., Woll, B., Cooper, R., Mcparland, A., & Zisserman, A. (2021). *BBC-Oxford British Sign Language Dataset*. <https://arxiv.org/abs/2111.03635v1>
- Alexandra Cevallos-Sánchez, G., Enriqueta Rodríguez-Camacho, M., & Rene Astudillo-Orellana, W. (2017). *Ciencias Sociales*. 3(2), 615–633. <https://doi.org/10.23857/dom.cien.pocaip.2017.3.2.esp.615-633>
- Alvear, V. (2017). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *Enfoque UTE*, 8(1), 244–256. <https://doi.org/10.29019/ENFOQUEUTE.V8N1.121>
- Amaya, S. (2018). Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM). *Lámpsakos (revista descontinuada)*, 15(15), 43–50. <https://doi.org/10.21501/21454086.1702>
- Andrade Carrera, H., Sinche Maita, S., & Hidalgo Lascano, P. (2021). Modelo para detectar el uso correcto de mascarillas en tiempo real utilizando redes neuronales convolucionales. *Revista de Investigación en Tecnologías de la Información: RITI*, ISSN-e 2387-0893, Vol. 9, No. Extra 17, 2021 (Ejemplar dedicado a: Especial Taller Andino de Comunicaciones Inalámbricas y sus Aplicaciones), págs. 111-120, 9(17), 111–120. <https://doi.org/10.36825/RITI.09.17.011>

- Artola, Á. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python*.
- Badillo, F. L., Hernández, C. A. R., Narváez, B. M., & Trillos, Y. E. A. (2021). Redes neuronales convolucionales: un modelo de Deep Learning en imágenes diagnósticas. Revisión de tema. *Revista colombiana de radiología*, 32(3), 5591–5599. <https://doi.org/10.53903/01212095.161>
- Benítez, P., Giraldo, J., & Rodríguez, S. (2021). Kit didáctico para el aprendizaje del lenguaje de señas ecuatoriano. *InGenio Journal: La revista de la Facultad de Ciencias de la Ingeniería de la UTEQ*, ISSN-e 2697-3642, Vol. 4, No. 1, 2021, págs. 1-10, 4(1), 1–10. <https://doi.org/10.18779/ingenio.v4i1.362>
- Borella, B. (2022). *INTRODUCTION TO COMPUTER VISION. PROCESSES AND APPLICATIONS*.
- Caballero, E. (2017). *Aplicación Práctica de la Visión Artificial para el Reconocimiento de Rostros en una Imagen, Utilizando Redes Neuronales y Algoritmos de Reconocimiento de Objetos de la Biblioteca OPENCV - hdl:11349/6104*. <https://repository.udistrital.edu.co/handle/11349/6104>
- Carrión, C. (2020). *REDES CONVOLUCIONALES*.
- Coanqui Apaza, F. Y. (2022). *Aplicación de inteligencia y visión artificial para detección y reconocimiento de manos*. <https://repositorio.upeu.edu.pe/handle/20.500.12840/6132>
- CONADIS. (2012). *Ley Orgánica de Discapacidades*.

Constitución de la República del Ecuador. (2015). CONSTITUCION DE LA REPUBLICA DEL ECUADOR. En *Registro Oficial* (Vol. 449, Número 20).
www.lexis.com.ec

Delgado Olivera, L. de la C., Díaz Alonso, L. M., Delgado Olivera, L. de la C., & Díaz Alonso, L. M. (2021). Modelos de Desarrollo de Software. *Revista Cubana de Ciencias Informáticas*, 15(1), 37–51.
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992021000100037&lng=es&nrm=iso&tlng=es

Duque, M. (2021). *Sistema de detección y conteo de manos en un racimo de bananos mediante visión artificial*. Universidad EIA.
<https://repository.eia.edu.co/handle/11190/3432>

Durán Suárez, J., Del, A., Torres, R., & Suárez, D. (2021). *Redes Neuronales Convolucionales en R Reconocimiento de caracteres escritos a mano* *Redes Neuronales Convolucionales en R Reconocimiento de caracteres escritos a mano* *Redes Neuronales Convolucionales en R*.

Estupiñán, R. J., Yelandi Leyva Vázquez, M., Javier Peñafiel Palacios, A., & El Assafiri Ojeda, Y. (2021). *Inteligencia artificial y propiedad intelectual*.

Fresno, P. (2020). *Implementación de un sistema software de visión artificial para la detección de objetos en movimiento*. <https://riunet.upv.es/handle/10251/151678>

García, J. C. (2017). *LA DISCAPACIDAD AUDITIVA. PRINCIPALES MODELOS Y AYUDAS TÉCNICAS PARA LA INTERVENCIÓN*. (*The hearing impaired. Top models and technical assistance for intervention*). <http://riai.jimdo.com/>

Guerrero, J. (2018). *TRADUCTOR DE LENGUAJE DE SEÑAS PORTATIL POR MEDIO DE RECONOCIMIENTO DE IMÁGENES.*

Gutiérrez Leguizamón, J. J., Plazas López, J. A., Suárez Barón, M. J., & González Sanabria, J. S. (2022). Reconocimiento de lengua de señas colombiana mediante redes neuronales convolucionales y captura de movimiento. *Tecnura*, 26(74), 70–86. <https://doi.org/10.14483/22487638.19213>

Hernández, Á., Díaz Collazo, G., Alberto Trujillo Miceli, F., Ángel Zepeda Hernández, J., Moreno Rincón, R., & Ronay López Estrada, F. (2017). Guante traductor de lenguaje de señas mexicano para personas sordomudas. *Revista Tecnología Digital*, 7(1), 75–86. www.revistatecnologiadigital.com

Horna, A. N., & Manzano Ramos, E. (2021). *Artificial vision system based on convolutional neural networks for the selection of blueberries according to export standards.* <https://doi.org/10.24265/campus.2021.v26n31.01>

Kakani, V., Nguyen, V. H., Kumar, B. P., Kim, H., & Pasupuleti, V. R. (2020). A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research*, 2, 100033. <https://doi.org/10.1016/J.JAFR.2020.100033>

L.-Dellamary, L. E. (2019). Gestualidad y lengua en la lengua de señas mexicana. *Lingüística Mexicana. Nueva Época*, 1(1), 141–166. https://linguisticamexicana-aml.colmex.mx/index.php/Linguistica_mexicana/article/view/300

Massiris, M., Delrieux, C., & Fernández, J. Á. (2020). *Detección de equipos de protección personal mediante red neuronal convolucional YOLO.* 1022–1029. <https://doi.org/10.17979/SPUDC.9788497497565.1022>

- Mera-Jiménez, L., & Ochoa-Gómez, J. F. (2021). Redes neuronales convolucionales para la clasificación de componentes independientes de rs-fMRI. *TecnoLógicas*, 24(50), undefined-undefined. <https://doi.org/10.22430/22565337.1626>
- Ministerio del Trabajo del Ecuador. (2021). *Remuneración mensual por puesto*. <https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2016/04/literal-c-Remuneracion-mensual-por-puesto.-MARZO.pdf>
- ONU. (2021). *Día Internacional de las Lenguas de Señas*. 1(1), 141–166. https://linguisticamexicana-ama.colmex.mx/index.php/Linguistica_mexicana/article/view/300
- Ortíz García, C. D. (2021). *Traductor de letras en lenguaje de señas con redes neuronales convolucionales*.
- Ortiz-Farfán, N., & Camargo-Mendoza, J. E. (2020). Modelo computacional para reconocimiento de lenguaje de señas en un contexto colombiano. *TecnoLógicas*, 23(48), 197–232. <https://doi.org/10.22430/22565337.1585>
- Pachón Espinel, D. L. (2019). *Prototipo de sistema automatizado con visión artificial para la selección de empaques de plástico, vidrio y lata en el proceso de reciclaje*. <https://repositorio.ucundinamarca.edu.co/handle/20.500.12558/2497>
- Peña, A., Rodríguez, X., & Fresquet, M. (2021). *Sign language. Its importance in deaf education*.
- Pérez, M., Andrea Cavanzo Nisso, G., & Villavisán Buitrago, F. (2021). Centro de Investigación de la Universidad Distrital Francisco José de Caldas. *Visión electrónica*, 12(1), 97–101. <https://doi.org/10.14483/22484728.15087>

- Pinzón Bayona, G., & Sanabria Orejuela, Y. (2021). *DESARROLLO DE UNA APLICACIÓN MÓVIL PARA TRADUCTOR DE LENGUAJE DE SEÑAS MEDIANTE EL USO DE SERVICIOS WEB.*
- Reza, H., Joze, V., Redmond, M., & Koller, O. (2018). MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language. *30th British Machine Vision Conference 2019, BMVC 2019*. <https://arxiv.org/abs/1812.01053v2>
- Rouhiainen, L. (2018). *INTELIGENCIA ARTIFICIAL, 101 COSAS QUE DEBES SABER HOY SOBRE NUESTRO FUTURO INTELIGENCIA ARTIFICIAL.*
www.planetadelibros.com
- Tinajero, J. (2020a). *Sistema de visión artificial para clasificación de latas de pintura por color considerando el espacio de color RGB.*
- Tinajero, J. (2020b). Sistema de visión artificial para clasificación de latas de pintura por color considerando el espacio de color RGB Artificial vision system for the classification of tin the paint by color considering the RGB color space Contenido. *ISSN, 41.*
- Ureta-Andrade, J. J., & Donoso-Cedeño, M. M. (2022). Lengua de Señas Ecuatoriana (Lsec) como Herramienta de Inclusión en Educación Especializada. *Abril-Junio*, 8(2), 251–261. <https://doi.org/10.23857/dc.v8i1.2643>
- Varela, J., Huerta, C., & Tello, O. (2017). APUNTES PARA UNA TEORÍA DE LA LENGUA DE SEÑAS. *CONDUCTUAL*, 5(2), 75–98.
<https://doi.org/10.59792/SFGT8992>
- Varela, N. (2021). *Evaluación de la plataforma Nvidia Jetson Nano para aplicaciones de visión artificial.* <https://repositorio.uam.es/handle/10486/698381>

ANEXOS

ANEXO A: Revisión de trabajos relacionados

Objetivo:

- Extraer los requerimientos de software y hardware de los trabajos relacionados a visión artificial aplicada al reconocimiento de gestos con la mano.

En el presente documento se realiza un resumen respecto a la revisión de trabajos relacionados al tema del actual proyecto, de esta forma se llevará a cabo la identificación de requerimientos necesarios para el diseño y la implementación del traductor de lenguaje de señas para personas con discapacidad auditiva basado en visión artificial.

1. Trabajo relacionado 1:

Ficha bibliográfica 1

Autor: Alarcón Nelson.

Editorial: Universidad de Guayaquil.

Título: “Traductor de Lenguaje de señas basado en Visión artificial para personas con discapacidad auditiva”.

Año: 2022.

Ciudad: Guayaquil.

Resumen del contenido:

- **Placa utilizada:**

El desarrollo del prototipo fue realizado en una NVIDIA Jetson Nano, la cual cuenta con un beneficio significativo al disponer de una tarjeta gráfica, de esta forma este microordenador es capaz de realizar múltiples tareas destinadas a visión artificial.

- **Programación: Python 3.8**

La utilización de Python para el desarrollo de software de visión artificial ofrece numerosas ventajas, ya que Python es compatible con múltiples plataformas y su versión 3.8 es altamente compatible con una variedad de funciones y librerías específicamente diseñadas para este tipo de aplicaciones.

- **Dataset**

El dataset ha sido creado previamente e incluye las 26 letras del Alfabeto Americano ASL. Cada letra está representada por 500 imágenes destinadas tanto para su validación como para su entrenamiento, lo que suma un total de 13,000 imágenes en el conjunto de datos completo.

- **Estandarización de las imágenes**

Las imágenes se estandarizaron con un tamaño de 300x300 píxeles de alto y ancho, además de que es necesario de que deben ser transformadas de RGB a BGR debido a como las bibliotecas utilizadas manejan los canales de color para garantizar la interoperabilidad y el correcto funcionamiento de las operaciones realizadas.

2. Trabajo relacionado 2:

Ficha bibliográfica 2

Autores: Pigou Lionel, Dieleman Snder, **Editorial:** Ghent University.

Kindermans, Pieter, Schrauwen,

Bnejamin.

Título: “Sign Language Recognition Using Convolutional Neural Networks”.

Año: 2015.

Ciudad: Ghent.

Resumen del contenido:

- **Preprocesamiento**

En la etapa de procesamiento las imágenes toman un tamaño de 640x640 píxeles, además de que se reducen los umbrales para reducir el ruido en los mapas de profundidad.

- **Segmentación**

El método de segmentación temporal consiste en predecir los fotogramas iniciales y finales de cada gesto en las muestras, implementando la técnica de Sliding Windows, o Ventanas Corredoras, donde cada intervalo de 32 frames se evalúan con el modelo entrenado.

- **Entrenamiento**

El entrenamiento del modelo de visión artificial fue realizado en la plataforma virtual Google Colab, debido a que proporciona acceso a recursos computacionales como GPU y TPU, los cual es ampliamente beneficioso ya que proporciona un entrenamiento a mayor velocidad.

3. Trabajo relacionado 3:

Ficha bibliográfica 3

Autor: Luis Sivisapa.

Editorial: Universidad de las Fuerzas Armadas.

Título: “Visión Artificial aplicada para el reconocimiento del lenguaje de señas”.

Año: 2014

Ciudad: Loja

Resumen del contenido:

- **Librería de visión artificial**

La librería predominante empleada en visión artificial fue OpenCV, seleccionada por su amplia documentación y la diversidad de funciones específicamente diseñadas para este campo. Además, es compatible con Python, el lenguaje de programación utilizado en el proyecto.

- **Distribución de Sistema Operativo**

Los sistemas operativos basados en distribuciones Linux presentan numerosas ventajas en comparación con otros sistemas operativos. Estas incluyen la disponibilidad de controladores de software de código abierto y un menor consumo de recursos para llevar a cabo las operaciones del sistema operativo.

- **Utilización de computador**

Se emplea un computador COMPAQ Presario CQ60, la cual cuenta con un procesador Inter Core 2 Duo CPU T5800, una memoria RAM de 2 Gb, un disco duro de 50 Gb. Con las presentes características de recursos computacionales no se presentó ningún problema durante el desarrollo del sistema de visión artificial.

4. Trabajo relacionado 4:

Ficha bibliográfica 4

Autor: Lucía Cheung, Carlos Medina. **Editorial:** UTP-Ridda2.

Título: “Implementación y Análisis de un Detector de Manos Basado en Visión Artificial”.

Año: 2013. **Ciudad:** Ciudad de Panamá.

Resumen del contenido:

- **Partición de Dataset**

Se requiere para examinar la capacidad de generalización del detector de manos, por lo tanto, las muestras se dividen en dos conjuntos: el conjunto de entrenamiento y el conjunto de validación. El conjunto de entrenamiento se utiliza para que el algoritmo de aprendizaje automático ajuste los parámetros del modelo, mientras que el conjunto de prueba evalúa la capacidad del modelo para generalizar con muestras completamente nuevas.

- **Optimización**

Se implementaron dos estrategias para disminuir el tiempo de procesamiento. La primera consistió en emplear una tabla de búsqueda para almacenar los factores de normalización, los cuales son compartidos por todos los intervalos de los histogramas de un bloque determinado. La segunda estrategia implicó la paralelización del código.

- **Detección**

El proceso de detección desarrollado examina individualmente cada cuadro de video. Cada cuadro, que tiene una resolución de 640x640 píxeles, es explorado y analizado en su totalidad para identificar la presencia de manos.

5. Trabajo relacionado 5:

Ficha bibliográfica 5

Autor: Valeria Andrade.

Editorial: Universidad Nacional de Chimborazo.

Título: “Diseño e implementación de un sistema traductor de lengua de señas mediante inteligencia artificial para personas con discapacidad auditiva”.

Año: 2022.

Ciudad: Riobamba.

Resumen del contenido:

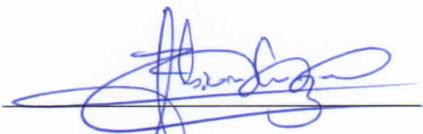
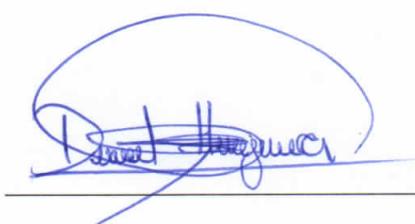
- **IDE de programación**

Se emplea el IDE PyCharm, especialmente diseñado para la programación en Python. Este entorno ofrece un asistente inteligente que proporciona soluciones rápidas para corregir errores durante el proceso de desarrollo.

- **Herramientas para el entrenamiento**

En el proceso de entrenamiento de la red neuronal, se utiliza Tensorflow para la creación y ejecución del entrenamiento de los datos normalizados, junto con Keras para la gestión de datos. Tensorflow facilita la manipulación de los datos almacenados en el clasificador, que incluye un total de 27 gestos de la mano previamente entrenados.

Firma de autorización

<p>Ing. Fabián Geovanny Cuzme Rodríguez</p>  <p>Director</p>	<p>Sr. David Santiago Hinojosa González</p>  <p>Estudiante</p>
---	--

ANEXO B: Análisis del número óptimo de épocas en el entrenamiento del modelo de visión artificial

Objetivo:

- Determinar el número óptimo de épocas para el proceso de entrenamiento del modelo de visión artificial, evaluando distintas cantidades de épocas para obtener un modelo eficiente y preciso.

En el presente documento se presenta un análisis para determinar el número óptimo de épocas para el proceso de entrenamiento del modelo de visión artificial, en el cual se lleva a cabo el entrenamiento con diferentes cantidades de épocas: 10, 30, 45, 60, 75, 90 y 110, para determinar con cuál se obtiene un modelo eficiente y preciso.

Estas cantidades para el análisis fueron seleccionadas considerando un rango de 15 a 20 épocas entre cada una, ya que este intervalo ofrece un equilibrio adecuado, siendo lo suficientemente amplio para captar tendencias significativas en la evolución del modelo sin omitir fases cruciales del aprendizaje, y no tan estrecho como para hacer un análisis redundante. Este rango permite identificar puntos de inflexión donde el modelo podría estar sobreajustándose o estabilizando su capacidad de predicción.

El análisis se realiza utilizando las gráficas generadas al finalizar el proceso de entrenamiento para cada una de las cantidades de épocas mencionadas anteriormente. A continuación, se realiza una descripción de cada una de estas gráficas:

- **Matriz de confusión:** Muestra la cantidad de predicciones correctas e incorrectas para cada clase, en donde las filas representan las etiquetas reales y las columnas representan las etiquetas que son predichas.

- **Curva de F1-Confianza:** Muestra la relación entre la confianza en las predicciones y la puntuación F1, la cual es una medida del balance entre precisión y recall (Capacidad del modelo para identificar las muestras verdaderas positivas).
- **Curva de Precisión-Confianza:** Muestra como varía la precisión del modelo de visión artificial con diferentes niveles de confianza en las predicciones.
- **Curva de Precisión-Recall:** Muestra la relación entre la precisión y el recall a diferentes umbrales de confianza.

El primer número de épocas con el que se realizó el entrenamiento del modelo de visión artificial es de 10 épocas. A continuación, en la Tabla 1, se realiza el análisis de cada una de las gráficas descritas anteriormente para determinar si las 10 épocas realizadas durante en entrenamiento del modelo de visión artificial son las adecuadas para obtener un modelo eficiente.

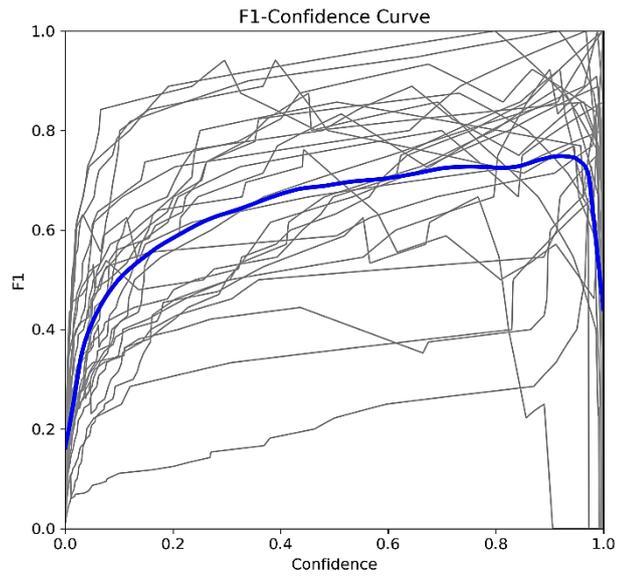
Tabla 1

Análisis de gráficas generadas para un entrenamiento de 10 épocas.

Análisis de entrenamiento con 10 épocas		
Tipo de gráfica	Análisis	Gráfica
Matriz de confusión	En la gráfica de matriz de confusión se visualiza una cantidad significativa de errores de clasificación, determinando que el modelo necesita más entrenamiento para mejorar su capacidad de traducción.	<p>Confusion Matrix</p> <p>Predicted</p> <p>True</p>

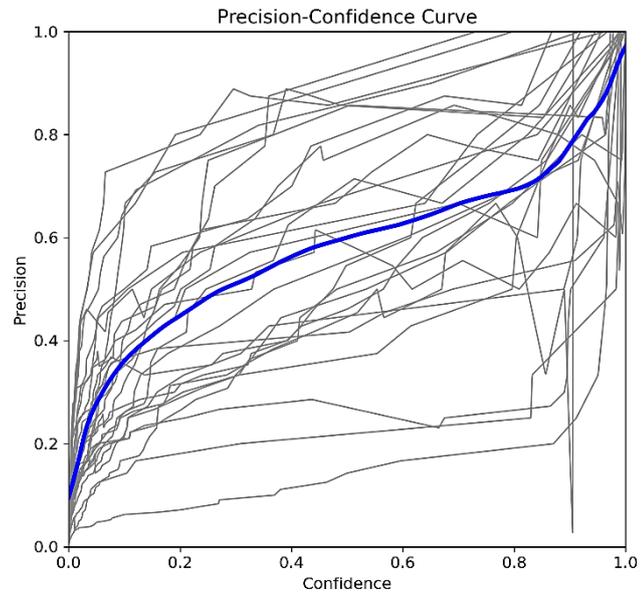
**Curva de
F1-
Confianza**

La puntuación F1 es relativamente baja, en donde se puede observar que a medida que aumenta la confianza, F1 no muestra una mejora considerable.



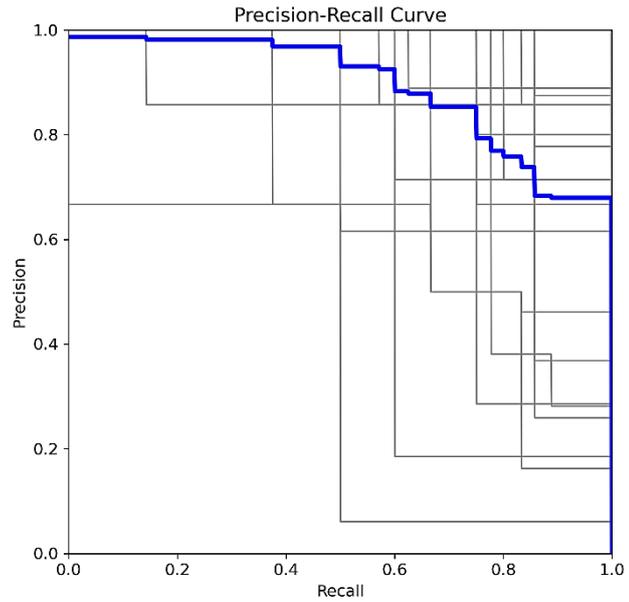
**Curva de
Precisión-
Confianza**

En la presente gráfica es posible visualizar una fluctuación en la precisión, lo cual indica que el modelo no es confiable.



Curva de Precisión-Recall

En la presente gráfica se muestra un área bajo la curva baja, con lo que se determina que el modelo tiene un desempeño insuficiente.



Posteriormente se realizó el entrenamiento del modelo de visión artificial con 30 épocas. A continuación, en la Tabla 2, se realiza el análisis de cada una de las gráficas para determinar si las 30 épocas realizadas durante en entrenamiento del modelo de visión artificial son las adecuadas para obtener un modelo eficiente.

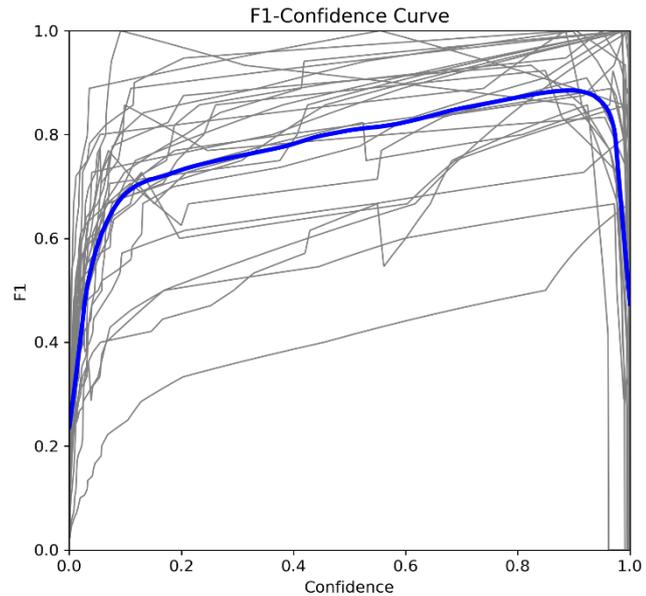
Tabla 2

Análisis de gráficas generadas para un entrenamiento de 30 épocas.

Análisis de entrenamiento con 30 épocas		
Tipo de gráfica	Análisis	Gráfica
Matriz de confusión	En la presente gráfica es posible visualizar una mejora en las predicciones correctas en comparación con 10 épocas, sin embargo, aún existe una cantidad considerable de errores de clasificación.	<p>The confusion matrix shows the relationship between predicted and true classes. The y-axis is labeled 'Predicted' and the x-axis is labeled 'True'. Both axes include letters A through Z and a 'background' category. The diagonal elements, representing correct classifications, are generally higher than the off-diagonal elements, indicating some improvement in performance compared to 10 epochs, though misclassifications are still present.</p>

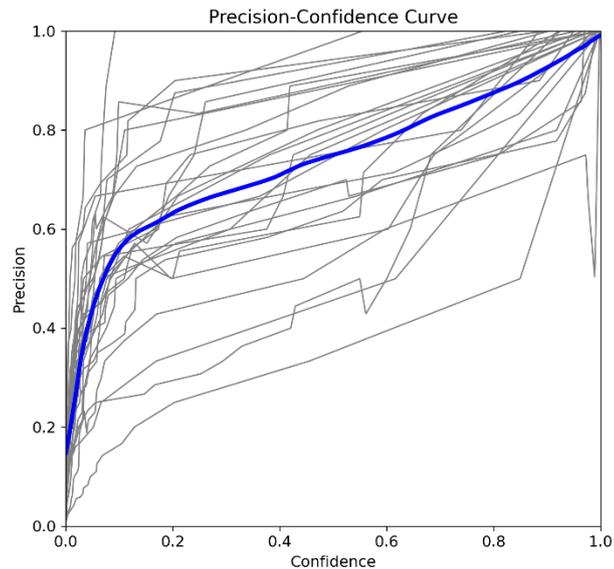
Curva de F1-Confianza

Es posible visualizar una mejora en la puntuación F1 en comparación con 10 épocas, sin embargo, se busca tener mejores resultados.



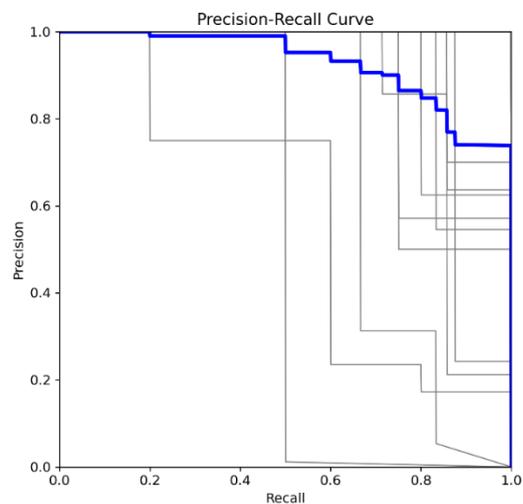
Curva de Precisión-Confianza

En la presente gráfica la precisión muestra una tendencia creciente con el aumento de la confianza, sin embargo, sigue siendo inestable, indicando que aún no ha alcanzado un nivel de confianza estable.



Curva de Precisión-Recall

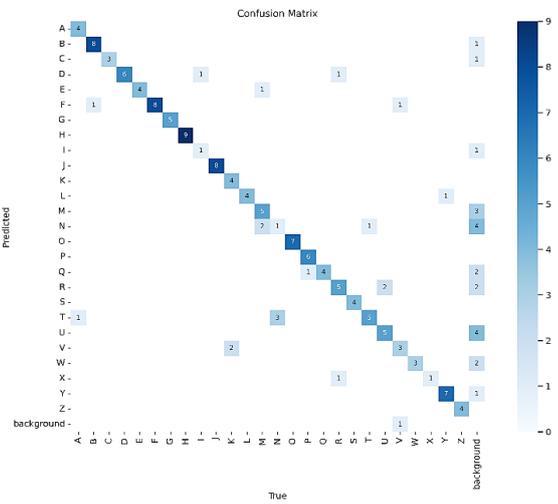
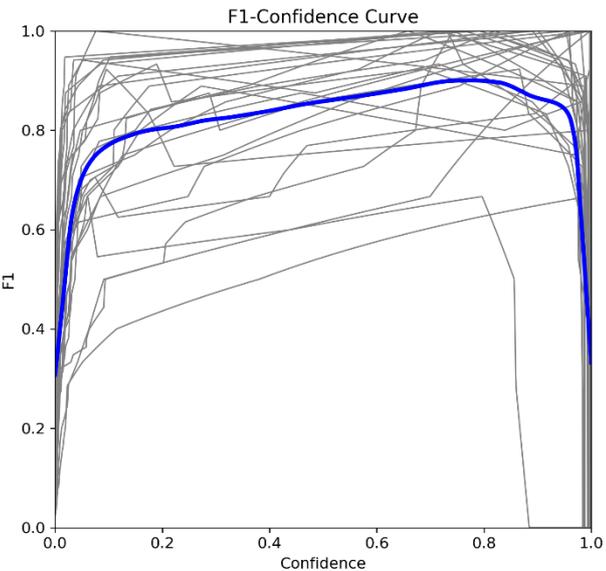
En la presente gráfica se observa que el área bajo la curva ha mejorado en comparación con 10 épocas, determinando que el modelo está progresando en su capacidad de equilibrar precisión y recall.



Posteriormente se realizó el entrenamiento del modelo de visión artificial con 45 épocas. A continuación, en la Tabla 3, se realiza el análisis de cada una de las gráficas para determinar si las 45 épocas realizadas durante en entrenamiento del modelo de visión artificial son las adecuadas para obtener un modelo eficiente.

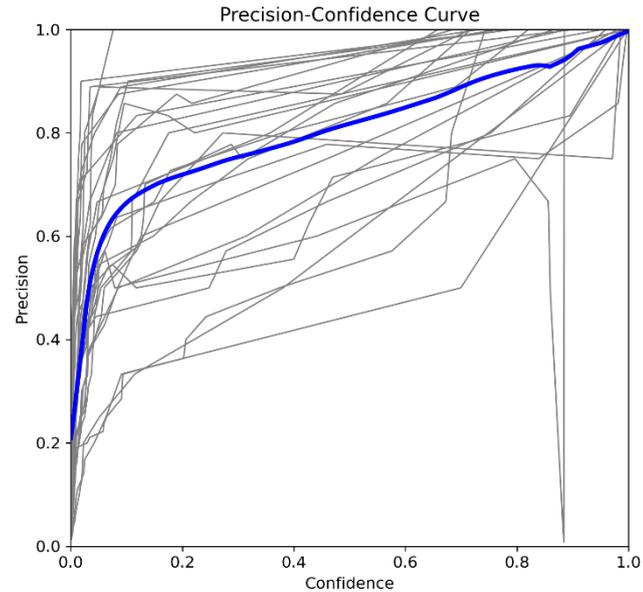
Tabla 3

Análisis de gráficas generadas para un entrenamiento de 45 épocas.

Análisis de entrenamiento con 45 épocas		
Tipo de gráfica	Análisis	Gráfica
Matriz de confusión	En la presente gráfica es posible visualizar que las predicciones correctas aumentan significativamente y los errores de clasificación disminuyen, indicando que se está consiguiendo un rendimiento óptimo.	 <p>The confusion matrix shows a strong diagonal, indicating that the model is correctly classifying most instances. The axes are labeled 'Predicted' and 'True', with categories A through Z and 'background'. A color scale on the right ranges from 0 to 9.</p>
Curva de F1-Confianza	En la presente gráfica se observa una mejora considerable de la puntuación F1, determinando que modelo está logrando un balance efectivo en la detección de verdaderos positivos y descartar falsos negativos.	 <p>The F1-Confidence Curve plot shows the relationship between F1 score (y-axis, 0.0 to 1.0) and Confidence (x-axis, 0.0 to 1.0). Multiple thin lines represent individual runs, and a thick blue line indicates the overall performance, showing a high F1 score across a range of confidence levels.</p>

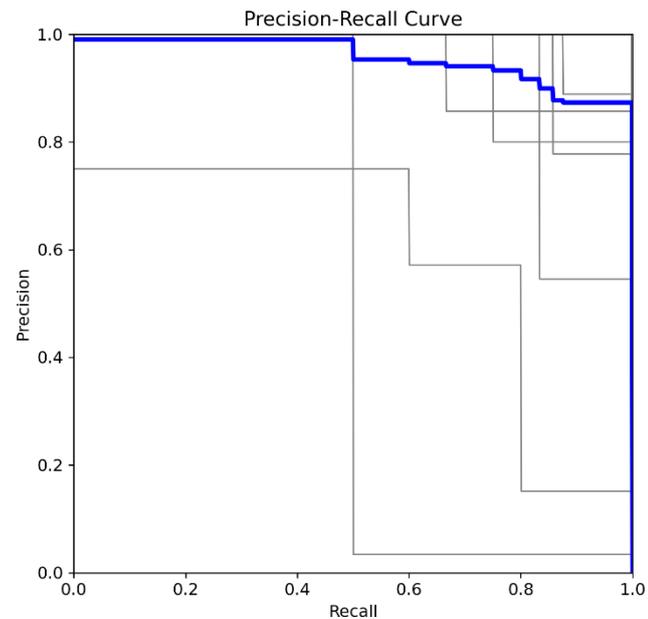
Curva de Precisión-Confianza

En la presente gráfica es posible visualizar que la precisión es más alta y estable en comparación que en las épocas anteriores, indicando que el modelo cuenta con un nivel consistente en sus predicciones.



Curva de Precisión-Recall

En la presente gráfica es posible observar que el área bajo la curva ha mejorado en comparación con las épocas anteriores, es decir, esta es más alta, determinando que el modelo ha mejorado su capacidad de mantener un buen equilibrio entre precisión y recall.



Posteriormente se realizó el entrenamiento del modelo de visión artificial con 60 épocas. A continuación, en la Tabla 4, se realiza el análisis de cada una de las gráficas para determinar si las 60 épocas realizadas durante en entrenamiento del modelo de visión artificial son las adecuadas para obtener un modelo eficiente.

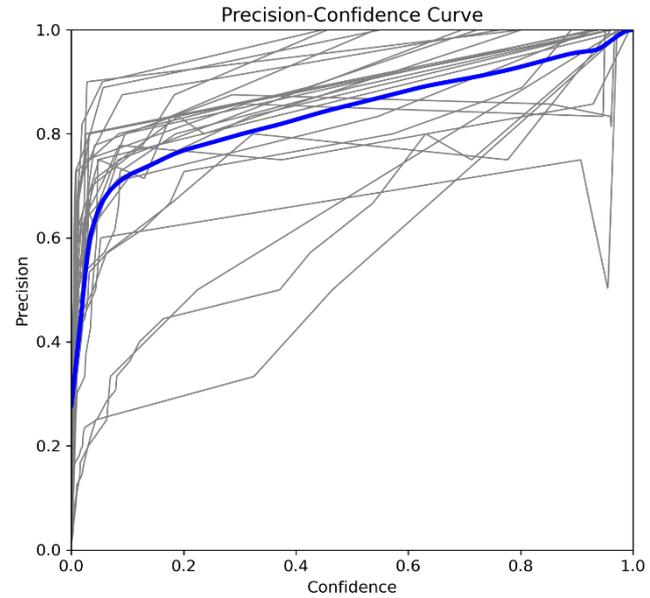
Tabla 4

Análisis de gráficas generadas para un entrenamiento de 60 épocas.

Análisis de entrenamiento con 60 épocas		
Tipo de gráfica	Análisis	Gráfica
Matriz de confusión	<p>En la presente gráfica se puede visualizar que las predicciones correctas son altas y los errores de clasificación son bajos, indicando que el modelo tiene una eficiente comprensión de las características de cada clase.</p>	<p>The Confusion Matrix heatmap displays the relationship between predicted and true classes. The x-axis is labeled 'True' and the y-axis is labeled 'Predicted', both ranging from 'A' to 'Z' and 'background'. The diagonal elements, representing correct classifications, are the most prominent, with values ranging from 1 to 9. A color scale on the right indicates the count of instances, ranging from 0 (lightest blue) to 9 (darkest blue).</p>
Curva de F1-Confianza	<p>En la presente gráfica es posible observar que la puntuación F1 es alta y estable en niveles de confianza moderados, indicando que el modelo mantiene un equilibrio óptimo entre precisión y recall a través de diferentes niveles de confianza, con lo que se determina que cuenta con un ajuste adecuado.</p>	<p>The F1-Confidence Curve plot shows the F1 score on the y-axis (ranging from 0.0 to 1.0) against Confidence on the x-axis (ranging from 0.0 to 1.0). Multiple grey lines represent individual runs, while a thick blue line represents the average performance. The average F1 score is consistently high, peaking around 0.9 at a confidence level of approximately 0.6-0.8, indicating a strong and stable model performance across different confidence thresholds.</p>

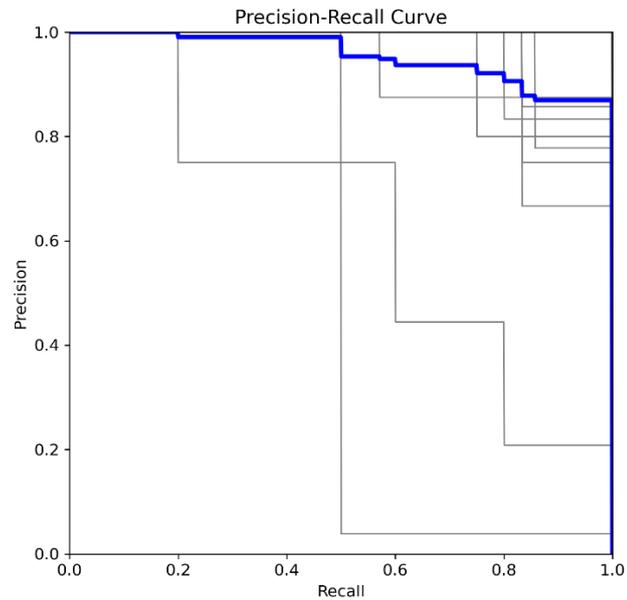
Curva de Precisión-Confianza

En la presente gráfica es posible observar que la precisión es alta y se mantiene estable en diferentes niveles de confianza, determinando que el modelo es confiable y eficiente.



Curva de Precisión-Recall

En la presente gráfica se puede tener que se cuenta con un área bajo la curva alta, determinando que el modelo ha alcanzado un rendimiento óptimo en términos de equilibrio entre precisión y recall.



Posteriormente se realizó el entrenamiento del modelo de visión artificial con 75 épocas. A continuación, en la Tabla 5, se realiza el análisis de cada una de las gráficas para determinar si las 75 épocas realizadas durante en entrenamiento del modelo de visión artificial son las adecuadas para obtener un modelo eficiente.

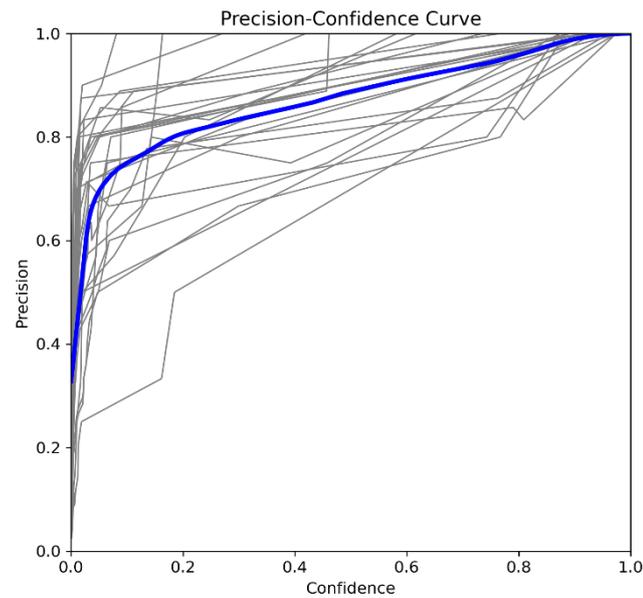
Tabla 5

Análisis de gráficas generadas para un entrenamiento de 75 épocas.

Análisis de entrenamiento con 75 épocas		
Tipo de gráfica	Análisis	Gráfica
Matriz de confusión	<p>En la presente gráfica es posible observar que se cuenta con altas predicciones correctas y errores de clasificación bajos, similar con 60 épocas, determinando que la falta de mejora significativa indica que el modelo ya ha alcanzado su punto óptimo de rendimiento.</p>	<p>The figure is a Confusion Matrix heatmap. The x-axis is labeled 'True' and the y-axis is labeled 'Predicted'. Both axes list classes from A to Z, plus a 'background' class. The diagonal elements, representing correct classifications, are the most prominent, with values ranging from 1 to 8. A color scale on the right indicates the count for each cell, ranging from 0 (lightest blue) to 8 (darkest blue).</p>
Curva de F1-Confianza	<p>En la presente gráfica es posible observar que la puntuación F1 es alta, pero similar a la obtenida con 60 épocas, determinando que más entrenamiento no mejora de forma significativa su balance entre precisión y recall.</p>	<p>The figure is an F1-Confidence Curve plot. The x-axis is labeled 'Confidence' and ranges from 0.0 to 1.0. The y-axis is labeled 'F1' and ranges from 0.0 to 1.0. The plot shows multiple thin grey lines representing individual runs, and a thick blue line representing the overall performance. The blue line starts at a high F1 score (around 0.8) at low confidence and remains high (above 0.8) until confidence reaches approximately 0.9, where it drops sharply to 0.0 at confidence 1.0.</p>

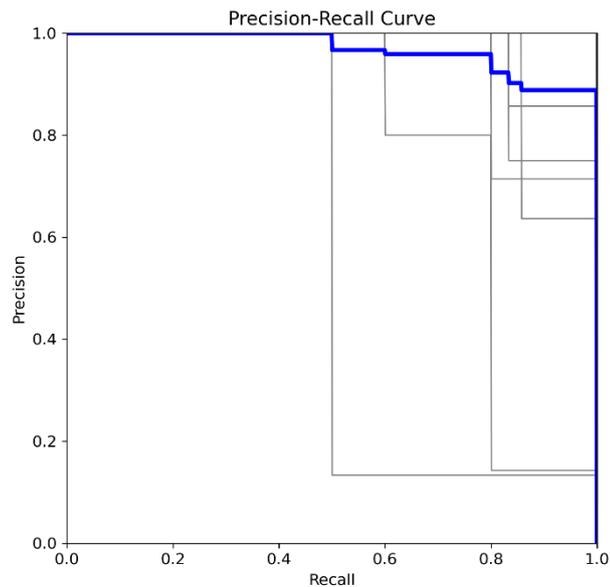
Curva de Precisión-Confianza

En la presente gráfica es posible observar que la precisión sigue siendo alta y estable, por lo que su falta de mejora significativa indica que más épocas no añadirán valor adicional en el presente parámetro.



Curva de Precisión-Recall

En la presente gráfica es posible observar que el área bajo la curva se mantiene alto, pero similar a 60 épocas, confirmando que el modelo no se beneficia de entrenamiento adicional.



Posteriormente, se realizó el entrenamiento del modelo de visión artificial con 90 épocas, mismo que se presenta en la Tabla 6, con la cual se podrá analizar si se presentan mejoras significativas en comparación a las cantidades de épocas presentadas anteriormente.

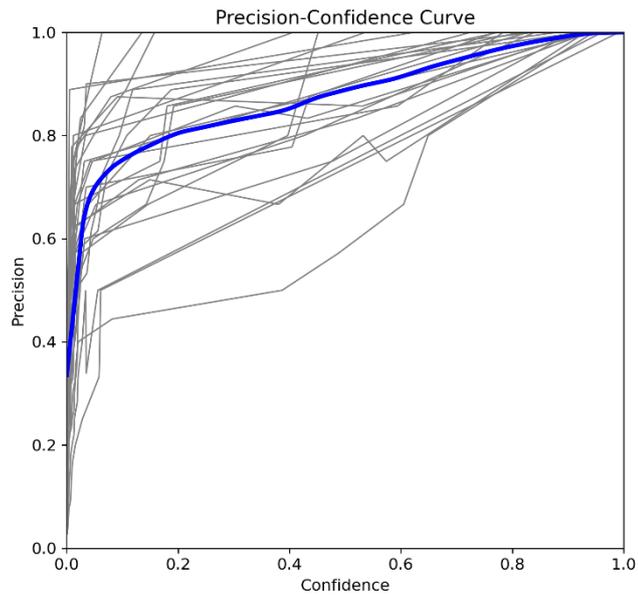
Tabla 6

Análisis de gráficas generadas para un entrenamiento de 90 épocas.

Análisis de entrenamiento con 90 épocas		
Tipo de gráfica	Análisis	Gráfica
Matriz de confusión	<p>En la presente gráfica es posible observar que no se presenta una mejora significativa en las predicciones correctas comparando con 75 y 60 épocas, indicando que el modelo ha alcanzado su límite de rendimiento y continuar el entrenamiento podría llevar al sobrentrenamiento.</p>	<p>The Confusion Matrix heatmap displays the relationship between predicted and actual classes. The x-axis represents the actual classes (A-Z, background) and the y-axis represents the predicted classes (A-Z, background). The color scale ranges from 0 (light blue) to 9 (dark blue). The diagonal elements, representing correct classifications, are the most prominent, with values ranging from 1 to 9. There are several off-diagonal elements, indicating misclassifications, with values ranging from 1 to 7.</p>
Curva de F1-Confianza	<p>De igual forma, en la presente gráfica, se observa que la puntuación F1 es alta pero no mejora significativamente y podría estar ajustándose demasiado a los datos de entrenamiento, reduciendo su capacidad.</p>	<p>The F1-Confidence Curve plot shows the F1 score on the y-axis (ranging from 0.0 to 1.0) against the Confidence level on the x-axis (ranging from 0.0 to 1.0). Multiple thin grey lines represent individual runs, while a thick blue line represents the average performance. The average F1 score is consistently high, starting around 0.8 at low confidence and peaking near 0.95 at a confidence level of approximately 0.8. The curve shows a slight dip at the very end of the confidence scale (near 1.0).</p>

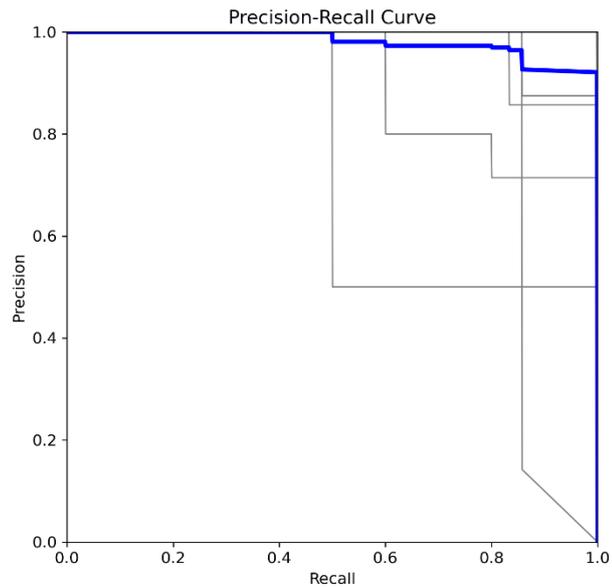
Curva de Precisión-Confianza

En la presente gráfica es posible observar que la precisión se mantiene alta, pero sin mejora significativa, determinando que la posible disminución de la precisión en altos niveles de confianza son indicativos de sobrentrenamiento.



Curva de Precisión-Recall

En la presente gráfica es posible observar que el área bajo la curva es alta, pero no muestra mejora en comparación con 60 y 75 épocas, confirmando que el modelo ha alcanzado su capacidad máxima de rendimiento y más entrenamiento es innecesario.



Finalmente, se realizó el entrenamiento del modelo de visión artificial con 110 épocas, mismo que se presenta en la Tabla 7.

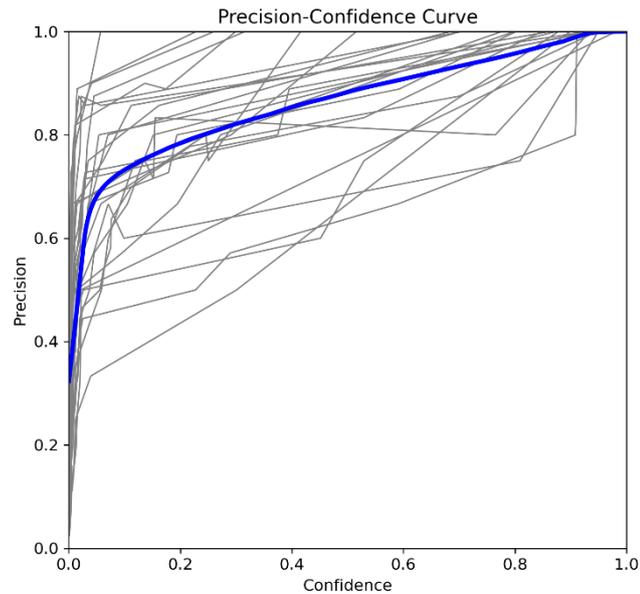
Tabla 7

Análisis de gráficas generadas para un entrenamiento de 110 épocas.

Análisis de entrenamiento con 110 épocas		
Tipo de gráfica	Análisis	Gráfica
Matriz de confusión	<p>En la presente gráfica, al comparar la matriz de confusión obtenida con 110 épocas con la de 90 épocas, se observa un incremento en las predicciones incorrectas. Este aumento indica que el modelo comienza a mostrar signos de sobrentrenamiento.</p>	<p>The confusion matrix shows the relationship between predicted and true classes. The x-axis is labeled 'True' and the y-axis is labeled 'Predicted'. Both axes list classes A through Z and a 'background' category. The color scale on the right ranges from 0 (lightest) to 9 (darkest). The diagonal elements, representing correct classifications, are the most prominent, with values ranging from 1 to 9. There are also several off-diagonal elements, indicating misclassifications, with values ranging from 1 to 4.</p>
Curva de F1-Confianza	<p>En la presente gráfica, la curva F1-Confianza es más baja en comparación con la obtenida con 90 épocas, lo que sugiere que el balance entre precisión y recall se ha deteriorado, con lo cual se comprueban los indicios de sobrentrenamiento, ya que no mejora su capacidad para identificar correctamente las muestras verdaderas positivas.</p>	<p>The F1-Confidence Curve plot shows the relationship between F1 score (y-axis, 0.0 to 1.0) and Confidence (x-axis, 0.0 to 1.0). The plot contains many thin grey lines representing individual runs or trials. A thick blue line highlights the overall trend, showing a peak in F1 score around 0.85 confidence, followed by a sharp decline as confidence approaches 1.0.</p>

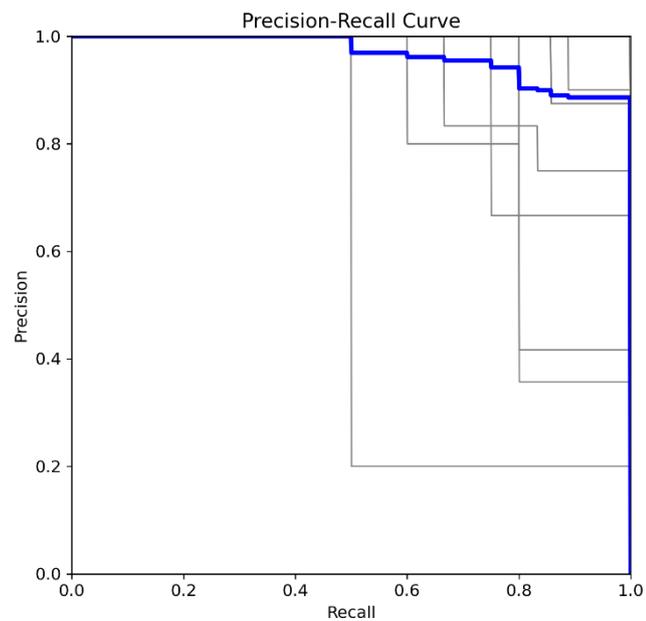
Curva de Precisión-Confianza

En la presente gráfica, se nota una disminución en la precisión general en comparación con 90 épocas, especialmente en los niveles de confianza más altos.



Curva de Precisión-Recall

En la presente gráfica, también se muestra un área más baja bajo la curva en comparación con los entrenamientos de 90 y 75 épocas, indicando que el modelo ha perdido su capacidad para mantener un equilibrio adecuado entre precisión y recall, lo que es un indicativo de sobrentrenamiento.



A través del análisis realizado mediante las gráficas correspondientes a las diferentes épocas de entrenamiento, se puede concluir que el número óptimo de épocas para entrenar el modelo de visión artificial con la CNN preentrenada de YOLOv8 es 60 épocas.

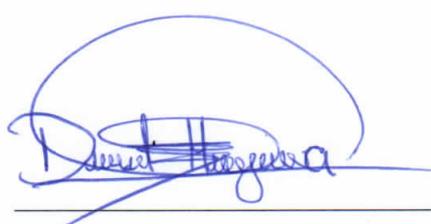
La selección de 60 épocas como el número óptimo se basa en un equilibrio entre eficiencia, rendimiento y la estabilidad del modelo. Aunque se observan mejoras con 75

y 90 épocas, estas son mínimas en comparación con las ganancias significativas que ocurren entre 30, 45 y 60 épocas. A partir de las 60 épocas, la curva de mejora se estabiliza, lo que sugiere que el modelo ya ha capturado la mayoría de los patrones relevantes. Continuar el entrenamiento más allá de este punto podría aumentar el riesgo de sobreajuste, donde el modelo se adapta demasiado a los datos específicos del conjunto de entrenamiento, reduciendo su capacidad de generalización, es decir, su habilidad para aplicar lo aprendido a nuevos datos no vistos previamente.

El análisis con 110 épocas muestra incluso una disminución en el rendimiento, evidenciando signos de sobrentrenamiento, donde el modelo comienza a perder precisión en nuevas predicciones. Además, entrenar con más de 60 épocas implica un mayor consumo de recursos y tiempo, sin un retorno justificado en términos de rendimiento.

Este número de épocas proporciona un modelo eficiente y preciso, con altos niveles de precisión y estabilidad en las predicciones, sin indicios de sobrentrenamiento. Entrenar el modelo por más de 60 épocas no proporciona mejoras significativas adicionales y podría llevar al sobrentrenamiento, mientras que entrenar con menos de 60 épocas resultaría en un rendimiento insuficiente.

Firma de autorización

<p>Ing. Fabián Geovanny Cuzme Rodríguez</p>  <p>Director</p>	<p>Sr. David Santiago Hinojosa González</p>  <p>Estudiante</p>
---	--

ANEXO C: Imágenes de las pruebas de validación de detección de mano en diferencias escenarios y condiciones

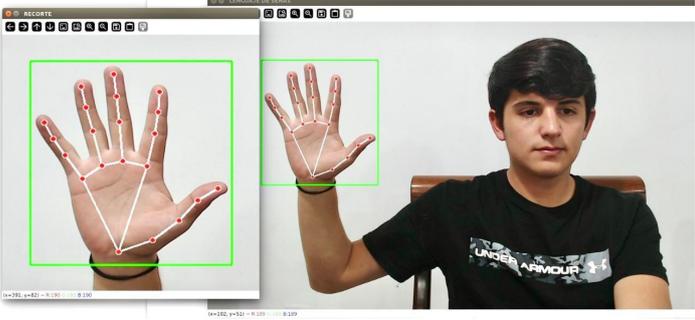
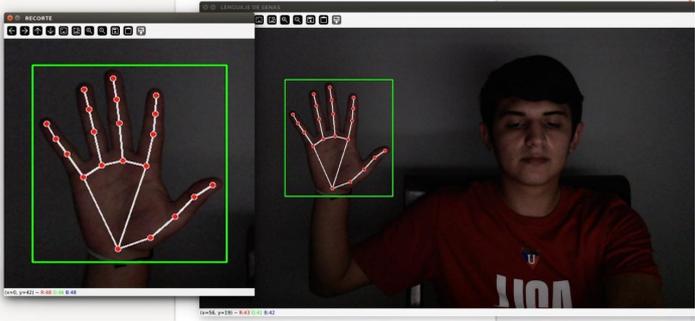
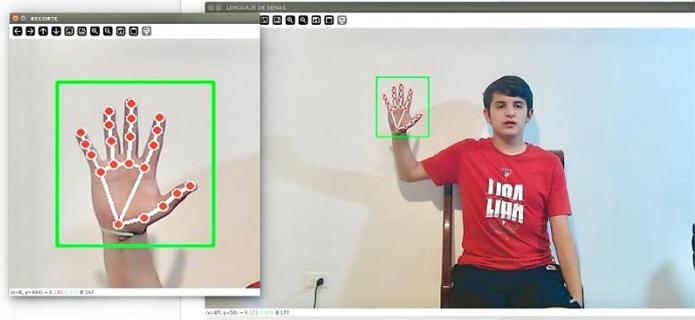
Objetivo:

- Proporcionar un respaldo visual que complemente los resultados obtenidos en las pruebas de validación de detección de mano, a través de la presentación de imágenes capturadas en diversos escenarios y condiciones evaluadas.

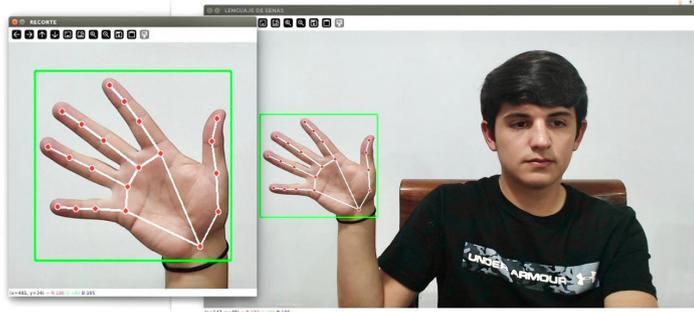
En el presente documento se presentan las imágenes capturadas durante las pruebas de validación de detección de mano, abarcando todos los escenarios evaluados: mano recta, mano en ángulo, mano parcialmente visible, mano cerrada en puño, mano en movimiento y seguimiento de mano. Además, se incluyen las diferentes condiciones en las que se realizaron las pruebas, tales como: ambiente ideal, ambiente oscuro, fondo con ruido visual y el usuario alejado de la cámara. Las imágenes obtenidas se presentan en la siguiente Tabla 1 de este documento, proporcionando un respaldo visual detallado de los resultados obtenidos en las pruebas.

Tabla 1

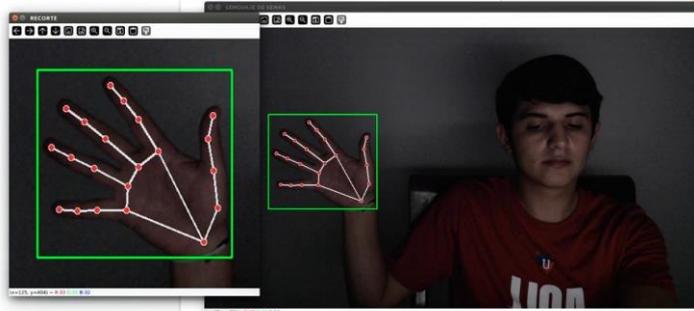
Pruebas de validación de detección de mano en diferentes escenarios y condiciones.

Escenario	Condición	Imagen
	Ambiente ideal	
	Ambiente oscuro	
Mano recta	Fondo con ruido visual	
	Usuario alejado	

Ambiente
ideal



Ambiente
oscuro

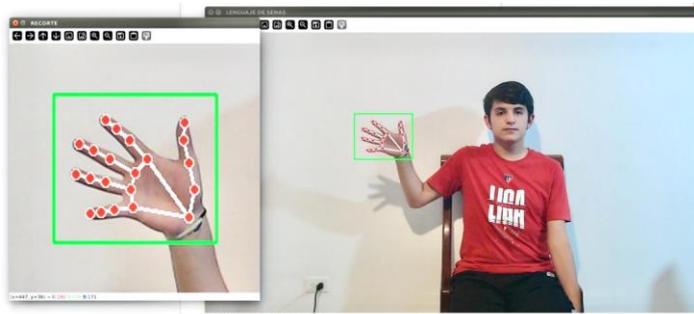


Mano en
ángulo

Fondo con
ruido visual

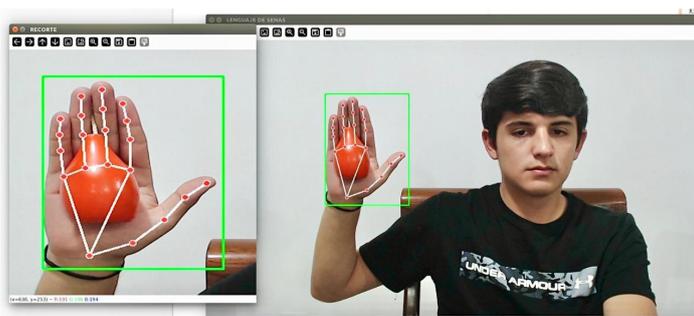


Usuario
alejado

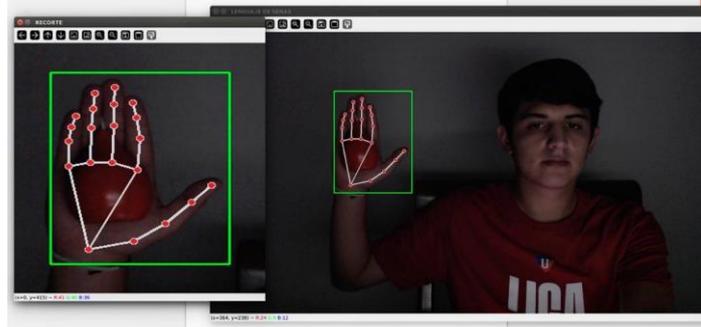


Mano
parcialmente
visible

Ambiente
ideal



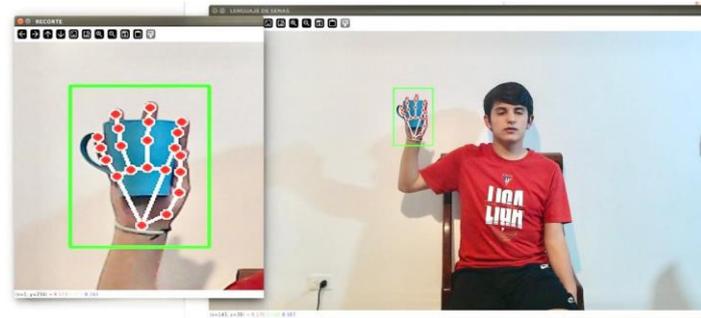
Ambiente
oscuro



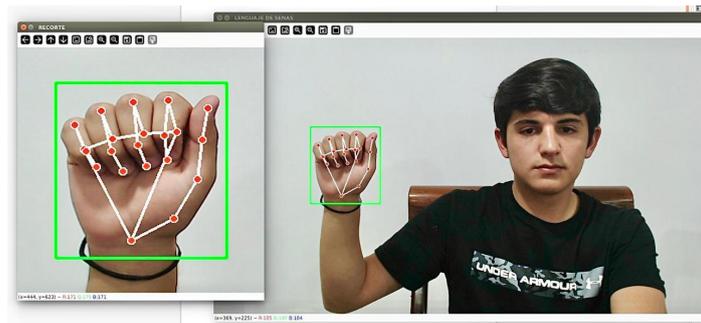
Fondo con
ruido visual



Usuario
alejado

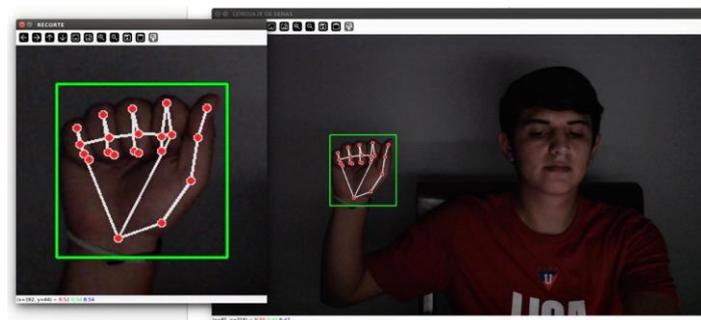


Ambiente
ideal



**Mano
cerrada en
puño**

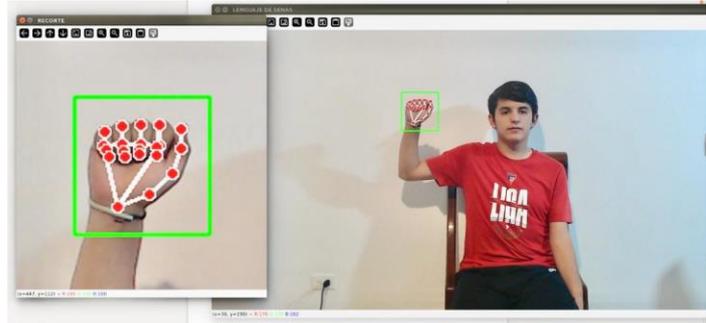
Ambiente
oscuro



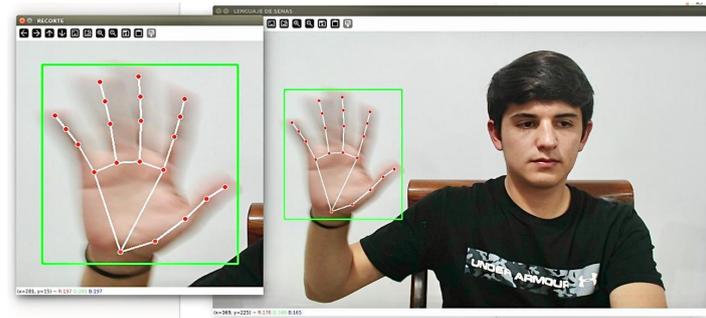
Fondo con
ruido visual



Usuario
alejado

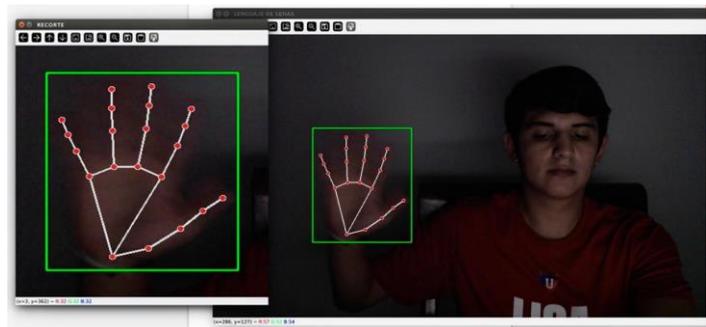


Ambiente
ideal

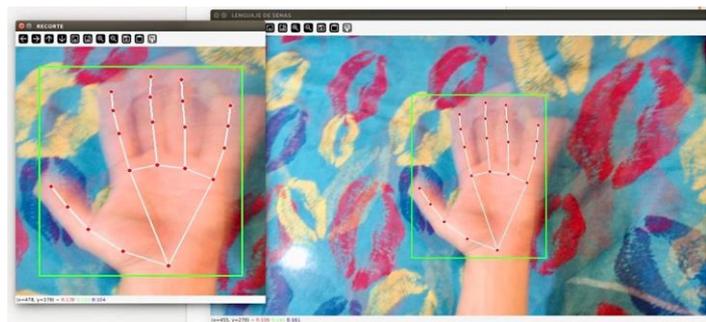


**Mano en
movimiento**

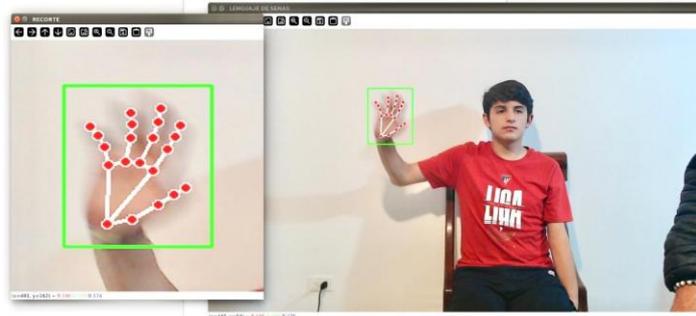
Ambiente
oscuro



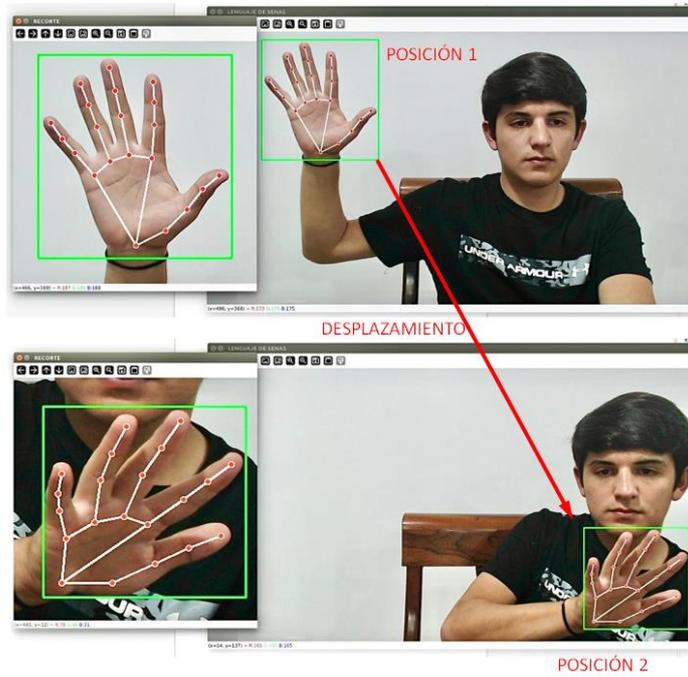
Fondo con
ruido visual



Usuario alejado

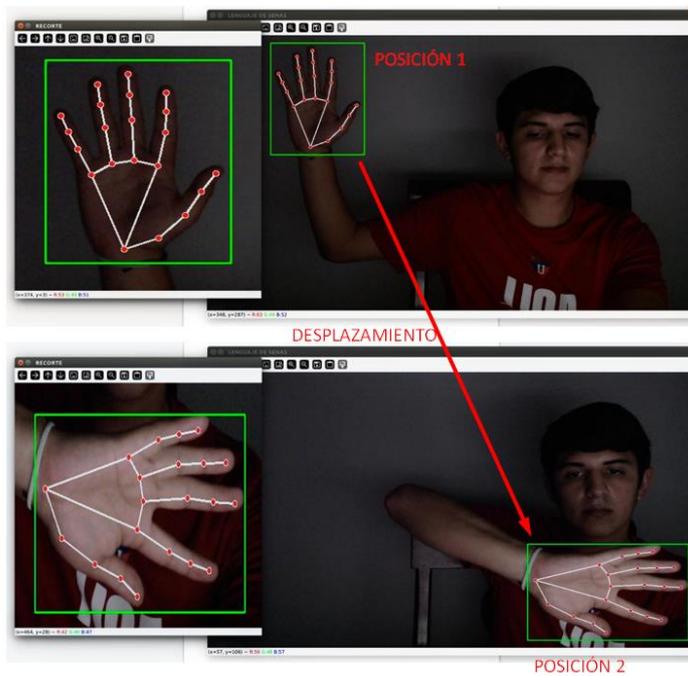


Ambiente ideal



Seguimiento de la mano

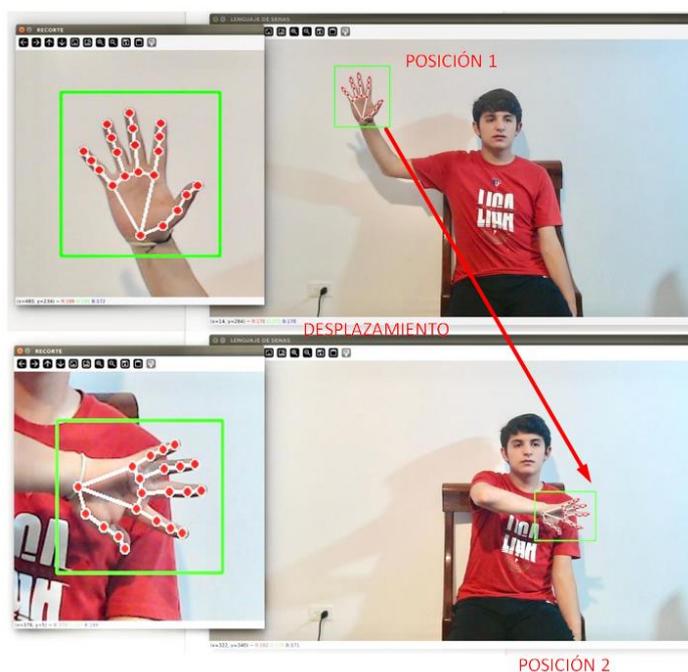
Ambiente oscuro



Fondo con
ruido visual



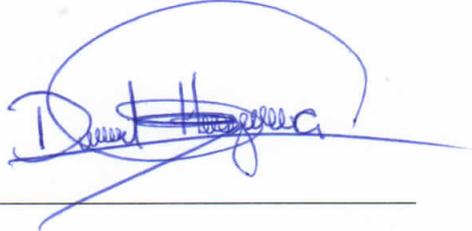
Usuario
alejado



Tras finalizar las pruebas de validación de detección de mano en diferentes condiciones y escenarios, se concluye que el sistema ha demostrado su efectividad en todos los casos evaluados. Las imágenes presentadas en la Tabla 1 de este anexo muestran que el sistema pudo detectar la mano tanto en entornos ideales, como en condiciones de poca iluminación, con fondo con ruido visual y cuando el usuario estaba alejado de la

cámara. En cada prueba, se observan claramente el cuadro delimitador y los puntos clave de referencia de la mano, lo que confirma la solidez y adaptabilidad del sistema a diversas condiciones de uso, asegurando su funcionamiento confiable.

Firma de autorización

<p data-bbox="245 654 775 689">Ing. Fabián Geovanny Cuzme Rodríguez</p>  <p data-bbox="456 987 564 1016">Director</p>	<p data-bbox="847 654 1347 689">Sr. David Santiago Hinojosa González</p>  <p data-bbox="1027 987 1166 1016">Estudiante</p>
--	--

ANEXO D: Resultados detallados de las pruebas de detección de letras del alfabeto

ASL

Objetivo:

- Presentar de manera detallada los resultados obtenidos en las 10 repeticiones de las pruebas de detección de letras del alfabeto ASL, proporcionando un respaldo que permite conocer los resultados individuales de cada repetición.

En el presente documento se presentan los resultados obtenidos tras finalizar las pruebas de detección de letras del alfabeto en lenguaje de señas americano (ASL). Estas pruebas consisten en ejecutar los 26 gestos correspondientes a cada letra del alfabeto ASL frente a la cámara del sistema, repitiendo cada gesto 10 veces para evaluar la consistencia del sistema en la detección de las letras. A partir de estas 10 repeticiones, se han calculado y analizado los promedios del porcentaje de coincidencia de cada letra, así como el porcentaje de detección correcta, proporcionando una evaluación detallada del rendimiento del sistema en estas pruebas.

En la Tabla 1 del presente Anexo D se muestran los resultados obtenidos en la primera repetición de los 26 gestos del alfabeto ASL. La mayoría de las letras fueron detectadas con un porcentaje de coincidencia superior al 91%, con una media del 95,29%. De las 26 letras, 24 fueron correctamente identificadas. No obstante, se detectaron errores en las letras "H" y "Z", que fueron incorrectamente reconocidas como "Q", lo que también generó una síntesis de voz inexacta.

Tabla 1

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 1.

Registro de detección de letras del alfabeto ASL – Repetición 1					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	97	Si	Si
2	B	B	100	Si	Si
3	C	C	98	Si	Si
4	D	D	94	Si	Si
5	E	E	94	Si	Si
6	F	F	91	Si	Si
7	G	G	92	Si	Si
8	H	Q	Detección incorrecta	No	No
9	I	I	98	Si	Si
10	J	J	92	Si	Si
11	K	K	95	Si	Si
12	L	L	97	Si	Si
13	M	M	96	Si	Si
14	N	N	92	Si	Si
15	O	O	98	Si	Si
16	P	P	94	Si	Si
17	Q	Q	97	Si	Si
18	R	R	93	Si	Si
19	S	S	94	Si	Si
20	T	T	92	Si	Si
21	U	U	99	Si	Si
22	V	V	96	Si	Si
23	W	W	95	Si	Si
24	X	X	95	Si	Si
25	Y	Y	98	Si	Si
26	Z	Q	Detección incorrecta	No	No

Tras la primera repetición, en la Tabla 2 del presente anexo se presentan los resultados obtenidos en la segunda repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia

superior al 90%, alcanzando una media de 95,48%. De las 26 letras, 23 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "J", "T" y "Z", que fueron incorrectamente reconocidas como "I", "M" y "D", respectivamente, lo que también resultó en una síntesis de voz incorrecta en esos casos.

Tabla 2

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 2.

Registro de detección de letras del alfabeto ASL – Repetición 2					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	98	Si	Si
2	B	B	99	Si	Si
3	C	C	98	Si	Si
4	D	D	93	Si	Si
5	E	E	95	Si	Si
6	F	F	90	Si	Si
7	G	G	94	Si	Si
8	H	H	94	Si	Si
9	I	I	99	Si	Si
10	J	I	Detección incorrecta	No	No
11	K	K	96	Si	Si
12	L	L	98	Si	Si
13	M	M	96	Si	Si
14	N	N	93	Si	Si
15	O	O	97	Si	Si
16	P	P	95	Si	Si
17	Q	Q	98	Si	Si
18	R	R	93	Si	Si
19	S	S	92	Si	Si
20	T	M	Detección incorrecta	No	No
21	U	U	100	Si	Si
22	V	V	96	Si	Si
23	W	W	94	Si	Si
24	X	X	91	Si	Si
25	Y	Y	97	Si	Si

26	Z	D	Detección incorrecta	No	No
----	---	---	----------------------	----	----

Tras la segunda repetición, en la Tabla 3 del presente anexo se presentan los resultados obtenidos en la tercera repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia superior al 89%, alcanzando una media de 95,33%. De las 26 letras, 24 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "N" y "Z", que fueron incorrectamente reconocidas como "S" y "D", respectivamente, lo que también resultó en una síntesis de voz inexacta en esos casos.

Tabla 3

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 3.

Registro de detección de letras del alfabeto ASL – Repetición 3					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	99	Si	Si
2	B	B	100	Si	Si
3	C	C	97	Si	Si
4	D	D	95	Si	Si
5	E	E	92	Si	Si
6	F	F	92	Si	Si
7	G	G	96	Si	Si
8	H	H	95	Si	Si
9	I	I	98	Si	Si
10	J	J	89	Si	Si
11	K	K	97	Si	Si
12	L	L	99	Si	Si
13	M	M	94	Si	Si
14	N	S	Detección incorrecta	No	No
15	O	O	98	Si	Si
16	P	P	94	Si	Si
17	Q	Q	97	Si	Si
18	R	R	94	Si	Si

19	S	S	95	Si	Si
20	T	T	93	Si	Si
21	U	U	97	Si	Si
22	V	V	94	Si	Si
23	W	W	93	Si	Si
24	X	X	95	Si	Si
25	Y	Y	95	Si	Si
26	Z	D	Detección incorrecta	No	No

Tras la tercera repetición, en la Tabla 4 del presente anexo se presentan los resultados obtenidos en la cuarta repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia superior al 89%, alcanzando una media de 95,50%. De las 26 letras, 24 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "D" y "Z", que fueron incorrectamente reconocidas como "X" e "I", respectivamente, lo que también resultó en una síntesis de voz inexacta en esos casos.

Tabla 4

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 4.

Registro de detección de letras del alfabeto ASL – Repetición 4					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	97	Si	Si
2	B	B	100	Si	Si
3	C	C	98	Si	Si
4	D	X	Detección incorrecta	No	No
5	E	E	91	Si	Si
6	F	F	95	Si	Si
7	G	G	94	Si	Si
8	H	H	95	Si	Si
9	I	I	99	Si	Si
10	J	J	93	Si	Si
11	K	K	96	Si	Si

12	L	L	98	Si	Si
13	M	M	97	Si	Si
14	N	N	94	Si	Si
15	O	O	99	Si	Si
16	P	P	90	Si	Si
17	Q	Q	96	Si	Si
18	R	R	95	Si	Si
19	S	S	92	Si	Si
20	T	T	89	Si	Si
21	U	U	99	Si	Si
22	V	V	97	Si	Si
23	W	W	97	Si	Si
24	X	X	94	Si	Si
25	Y	Y	97	Si	Si
26	Z	I	Detección incorrecta	No	No

Tras la cuarta repetición, en la Tabla 5 del presente anexo se presentan los resultados obtenidos en la quinta repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia superior al 92%, alcanzando una media de 95,54%. De las 26 letras, 24 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "H" y "Z", que fueron incorrectamente reconocidas como "P" y "Y", respectivamente, lo que también resultó en una síntesis de voz inexacta en esos casos.

Tabla 5

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 5.

Registro de detección de letras del alfabeto ASL – Repetición 5					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	98	Si	Si
2	B	B	99	Si	Si
3	C	C	97	Si	Si
4	D	D	94	Si	Si
5	E	E	93	Si	Si

6	F	F	92	Si	Si
7	G	G	94	Si	Si
8	H	P	Detección incorrecta	No	No
9	I	I	96	Si	Si
10	J	J	96	Si	Si
11	K	K	94	Si	Si
12	L	L	98	Si	Si
13	M	M	93	Si	Si
14	N	N	95	Si	Si
15	O	O	100	Si	Si
16	P	P	94	Si	Si
17	Q	Q	95	Si	Si
18	R	R	98	Si	Si
19	S	S	94	Si	Si
20	T	T	93	Si	Si
21	U	U	98	Si	Si
22	V	V	95	Si	Si
23	W	W	96	Si	Si
24	X	X	96	Si	Si
25	Y	Y	95	Si	Si
26	Z	Y	Detección incorrecta	No	No

Tras la quinta repetición, en la Tabla 6 del presente anexo se presentan los resultados obtenidos en la sexta repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia superior al 91%, alcanzando una media de 96,30%. De las 26 letras, 23 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "F", "S" y "Z", que fueron incorrectamente reconocidas como "P", "A" y "Q", respectivamente, lo que también resultó en una síntesis de voz inexacta en esos casos.

Tabla 6

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 6.

Registro de detección de letras del alfabeto ASL – Repetición 6					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	99	Si	Si
2	B	B	100	Si	Si
3	C	C	98	Si	Si
4	D	D	94	Si	Si
5	E	E	91	Si	Si
6	F	P	Detección incorrecta	No	No
7	G	G	93	Si	Si
8	H	H	97	Si	Si
9	I	I	99	Si	Si
10	J	J	94	Si	Si
11	K	K	96	Si	Si
12	L	L	99	Si	Si
13	M	M	97	Si	Si
14	N	N	97	Si	Si
15	O	O	99	Si	Si
16	P	P	96	Si	Si
17	Q	Q	97	Si	Si
18	R	R	95	Si	Si
19	S	A	Detección incorrecta	No	No
20	T	T	95	Si	Si
21	U	U	98	Si	Si
22	V	V	96	Si	Si
23	W	W	93	Si	Si
24	X	X	94	Si	Si
25	Y	Y	98	Si	Si
26	Z	Q	Detección incorrecta	No	No

Tras la sexta repetición, en la Tabla 7 del presente anexo se presentan los resultados obtenidos en la séptima repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia

superior al 90%, alcanzando una media de 95,30%. De las 26 letras, 23 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "C", "M" y "Z", que fueron incorrectamente reconocidas como "O", "N" y "D", respectivamente, lo que también resultó en una síntesis de voz inexacta en esos casos.

Tabla 7

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 7.

Registro de detección de letras del alfabeto ASL – Repetición 7					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	97	Si	Si
2	B	B	97	Si	Si
3	C	O	Detección incorrecta	No	No
4	D	D	96	Si	Si
5	E	E	91	Si	Si
6	F	F	93	Si	Si
7	G	G	94	Si	Si
8	H	H	95	Si	Si
9	I	I	100	Si	Si
10	J	J	93	Si	Si
11	K	K	95	Si	Si
12	L	L	97	Si	Si
13	M	N	Detección incorrecta	No	No
14	N	N	98	Si	Si
15	O	O	97	Si	Si
16	P	P	90	Si	Si
17	Q	Q	93	Si	Si
18	R	R	98	Si	Si
19	S	S	96	Si	Si
20	T	T	94	Si	Si
21	U	U	97	Si	Si
22	V	V	98	Si	Si
23	W	W	92	Si	Si
24	X	X	94	Si	Si
25	Y	Y	97	Si	Si

26	Z	D	Detección incorrecta	No	No
----	---	---	----------------------	----	----

Tras la séptima repetición, en la Tabla 8 del presente anexo se presentan los resultados obtenidos en la octava repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia superior al 90%, alcanzando una media de 95,92%. De las 26 letras, 24 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "T" y "Z", que fueron incorrectamente reconocidas como "M" y "P", respectivamente, lo que también resultó en una síntesis de voz inexacta en esos casos.

Tabla 8

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 8.

Registro de detección de letras del alfabeto ASL – Repetición 8					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	97	Si	Si
2	B	B	100	Si	Si
3	C	C	97	Si	Si
4	D	D	95	Si	Si
5	E	E	93	Si	Si
6	F	F	90	Si	Si
7	G	G	95	Si	Si
8	H	H	97	Si	Si
9	I	I	98	Si	Si
10	J	J	94	Si	Si
11	K	K	98	Si	Si
12	L	L	98	Si	Si
13	M	M	94	Si	Si
14	N	N	96	Si	Si
15	O	O	98	Si	Si
16	P	P	95	Si	Si
17	Q	Q	97	Si	Si
18	R	R	97	Si	Si
19	S	S	95	Si	Si

20	T	M	Detección incorrecta	No	No
21	U	U	98	Si	Si
22	V	V	95	Si	Si
23	W	W	94	Si	Si
24	X	X	92	Si	Si
25	Y	Y	99	Si	Si
26	Z	P	Detección incorrecta	No	No

Tras la octava repetición, en la Tabla 9 del presente anexo se presentan los resultados obtenidos en la novena repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia superior al 92%, alcanzando una media de 95,87%. De las 26 letras, 23 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "H", "J" y "Z", que fueron incorrectamente reconocidas como "G", "I" y "D", respectivamente, lo que también resultó en una síntesis de voz inexacta en esos casos.

Tabla 9

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 9.

Registro de detección de letras del alfabeto ASL – Repetición 9					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	98	Si	Si
2	B	B	99	Si	Si
3	C	C	97	Si	Si
4	D	D	93	Si	Si
5	E	E	92	Si	Si
6	F	F	92	Si	Si
7	G	G	94	Si	Si
8	H	G	Detección incorrecta	No	No
9	I	I	98	Si	Si
10	J	I	Detección incorrecta	No	No

11	K	K	95	Si	Si
12	L	L	99	Si	Si
13	M	M	95	Si	Si
14	N	N	95	Si	Si
15	O	O	99	Si	Si
16	P	P	96	Si	Si
17	Q	Q	96	Si	Si
18	R	R	95	Si	Si
19	S	S	95	Si	Si
20	T	T	95	Si	Si
21	U	U	99	Si	Si
22	V	V	96	Si	Si
23	W	W	95	Si	Si
24	X	X	94	Si	Si
25	Y	Y	98	Si	Si
26	Z	D	Detección incorrecta	No	No

Finalmente, en la Tabla 10 del presente anexo se presentan los resultados obtenidos en la décima repetición de los 26 gestos del alfabeto ASL. En esta ocasión, la mayoría de las letras fueron detectadas con un porcentaje de coincidencia superior al 91%, alcanzando una media de 95,78%. De las 26 letras, 23 fueron correctamente identificadas. Sin embargo, se produjeron errores en las letras "K", "X" y "Z", que fueron incorrectamente reconocidas como "P", "D" y "I", respectivamente, lo que también resultó en una síntesis de voz incorrecta en esos casos.

Tabla 10

Resultados de las pruebas de detección de letras del alfabeto ASL en repetición 10.

Registro de detección de letras del alfabeto ASL – Repetición 10					
	Letra ASL	Letra detectada por el sistema	Porcentaje de coincidencia (%)	Detección correcta (Si/No)	Síntesis de voz correcta (Si/No)
1	A	A	99	Si	Si
2	B	B	100	Si	Si
3	C	C	95	Si	Si
4	D	D	97	Si	Si

5	E	E	91	Si	Si
6	F	F	93	Si	Si
7	G	G	95	Si	Si
8	H	H	96	Si	Si
9	I	I	96	Si	Si
10	J	J	92	Si	Si
11	K	P	Detección incorrecta	No	No
12	L	L	98	Si	Si
13	M	M	95	Si	Si
14	N	N	96	Si	Si
15	O	O	98	Si	Si
16	P	P	95	Si	Si
17	Q	Q	97	Si	Si
18	R	R	95	Si	Si
19	S	S	98	Si	Si
20	T	T	92	Si	Si
21	U	U	98	Si	Si
22	V	V	95	Si	Si
23	W	W	95	Si	Si
24	X	D	Detección incorrecta	No	No
25	Y	Y	97	Si	Si
26	Z	I	Detección incorrecta	No	No

Tras finalizar las 10 repeticiones de los 26 gestos del alfabeto ASL, se concluye que el sistema de detección ha demostrado un alto nivel de precisión y consistencia, alcanzando una media final del 95,56% en el porcentaje de coincidencia al promediar los resultados de todas las repeticiones. Sin embargo, se observó que la letra "Z" no fue reconocida en ninguna de las repeticiones, ya que este gesto se realiza con un movimiento continuo, mientras que las demás letras se representan con posturas fijas. El modelo de visión artificial YOLOv8, entrenado con un dataset basado en imágenes estáticas, no está diseñado para procesar movimientos, lo que explica la dificultad en la detección de la letra "Z".

A pesar de esta limitación, el sistema ha demostrado ser confiable para el reconocimiento de la mayoría de las letras, y el sintetizador de voz ha reproducido correctamente los sonidos correspondientes a las letras detectadas. Estos resultados confirman la eficacia del sistema para la mayoría de los gestos estáticos, aunque su rendimiento en gestos dinámicos podría mejorarse en futuras versiones, incorporando modelos y equipos de hardware más avanzados que permitan detectar movimientos en tiempo real sin sacrificar la eficiencia.

Firma de autorización

<p>Ing. Fabián Geovanny Cuzme Rodríguez</p>  <p>Director</p>	<p>Sr. David Santiago Hinojosa González</p>  <p>Estudiante</p>
---	--

ANEXO E: Resultados individuales de las pruebas de integración con personas con discapacidad auditiva

Objetivo:

- Presentar los resultados individuales obtenidos en las pruebas de integración del sistema con usuarios con discapacidad auditiva, detallando las métricas de precisión de traducción y tiempo de respuesta en los cuatro entornos evaluados: fondo monocromático, fondo con ruido visual, ambiente oscuro y usuario a distancia considerable de la cámara, para cada participante.

En el presente documento se exponen los resultados individuales obtenidos en las pruebas de integración del sistema realizadas con cada usuario con discapacidad auditiva en los cuatro entornos evaluados. Para facilitar la comprensión y el análisis detallado, este anexo se organiza por entorno, mostrando los resultados específicos de cada participante en los diferentes escenarios: fondo monocromático, fondo con ruido visual, ambiente oscuro y usuario a distancia considerable de la cámara.

Se evaluará la precisión de traducción para cada usuario. Esta métrica hace referencia al porcentaje de confianza o coincidencia proporcionado por el modelo YOLOv8 al identificar una letra en lenguaje de señas. Este porcentaje indica el nivel de certeza del modelo sobre si la letra detectada corresponde a la gestualizada por el usuario.

Adicionalmente, se analizará la métrica del tiempo de respuesta para cada usuario. Este indicador se refiere al tiempo que tarda el sistema en procesar e inferir una letra gestualizada por el usuario. El tiempo de respuesta se mide internamente a través del CMD, que muestra el valor en milisegundos (ms) para cada inferencia realizada. Esta métrica es clave, ya que permite evaluar la eficiencia del sistema en tiempo real.

Fondo monocromático

En este apartado se presentan los resultados de las pruebas realizadas por las personas con discapacidad auditiva en un entorno de fondo monocromático. Este tipo de fondo se caracteriza por ser de un color uniforme, sin patrones ni distracciones visuales, lo que permite evaluar el desempeño del sistema en condiciones visuales simples y controladas. Durante estas pruebas, el participante ejecutará las 26 letras del alfabeto ASL, registrándose si cada letra es detectada correctamente, así como el porcentaje de precisión de traducción y el tiempo de respuesta.

A continuación, se presentan los resultados individuales de las pruebas realizadas en un entorno de fondo monocromático para cada una de las siete personas con discapacidad auditiva. Desde la Tabla 1 hasta la Tabla 7, se detallan los resultados obtenidos para cada participante, permitiendo conocer el desempeño del sistema de manera específica para cada usuario en condiciones visuales controladas.

Tabla 1

Resultados de la persona con discapacidad auditiva 1 en fondo monocromático.

Fondo monocromático					
Persona con discapacidad auditiva 1					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		98	929,9
2	B	X		100	912,7
3	C	X		98	985,4
4	D	X		94	948,6
5	E	X		95	936,5
6	F	X		94	942,4
7	G	X		92	1005,9
8	H		X	-	-
9	I	X		98	964,2
10	J	X		96	917,4
11	K	X		95	992,7
12	L	X		98	935,9

13	M	X		100	949,5
14	N	X		97	968,7
15	O	X		100	999,7
16	P	X		98	1031,4
17	Q	X		98	902
18	R	X		98	954,5
19	S	X		95	944,3
20	T	X		94	982,1
21	U	X		99	982,4
22	V	X		99	975,5
23	W	X		95	905,7
24	X	X		96	909,3
25	Y	X		98	998,1
26	Z		X	-	-

Tabla 2

Resultados de la persona con discapacidad auditiva 2 en fondo monocromático.

Fondo monocromático					
Persona con discapacidad auditiva 2					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		99	954,9
2	B	X		100	933,4
3	C	X		97	998,5
4	D	X		91	932,1
5	E	X		96	1006,8
6	F	X		93	991,3
7	G	X		92	934,4
8	H	X		90	922,3
9	I	X		95	901,6
10	J	X		93	933,2
11	K		X	-	-
12	L	X		97	955,7
13	M	X		96	933,1
14	N	X		95	981,2
15	O	X		98	987,3
16	P	X		97	985,9
17	Q	X		94	924,6
18	R	X		99	986,8
19	S	X		94	951,9
20	T	X		96	913,2
21	U	X		97	979,5
22	V	X		98	960,7

23	W	X		97	933,4
24	X	X		95	981,3
25	Y	X		99	970
26	Z		X	-	-

Tabla 3

Resultados de la persona con discapacidad auditiva 3 en fondo monocromático.

Fondo monocromático					
Persona con discapacidad auditiva 3					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		89	1012,5
2	B	X		100	973,2
3	C	X		94	998,5
4	D	X		89	960,8
5	E	X		97	921,8
6	F	X		91	916,3
7	G	X		92	967,9
8	H		X	-	-
9	I	X		96	925,5
10	J	X		92	994,2
11	K		X	-	-
12	L	X		99	923,3
13	M	X		98	948,2
14	N	X		97	937,1
15	O	X		99	902,8
16	P	X		94	952,9
17	Q	X		95	961,9
18	R	X		97	974,5
19	S	X		96	982,6
20	T	X		93	917,4
21	U	X		98	911,3
22	V	X		98	983,3
23	W	X		97	988,7
24	X	X		93	979,9
25	Y	X		98	963,8
26	Z		X	-	-

Tabla 4

Resultados de la persona con discapacidad auditiva 4 en fondo monocromático.

Fondo monocromático					
Persona con discapacidad auditiva 4					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	939,8
2	B	X		98	947,1
3	C	X		97	985,2
4	D	X		92	917,1
5	E	X		95	970,7
6	F	X		94	974,5
7	G	X		93	963,4
8	H	X		91	935,2
9	I	X		98	942,3
10	J	X		94	1046,8
11	K		X	-	-
12	L	X		98	984,8
13	M	X		97	922
14	N	X		97	973,1
15	O	X		98	901,8
16	P	X		95	999,7
17	Q	X		94	985,5
18	R	X		98	974,2
19	S	X		97	936,3
20	T	X		92	902,8
21	U	X		99	953,6
22	V	X		99	939,4
23	W	X		97	915,5
24	X	X		92	944,9
25	Y	X		99	967,1
26	Z		X	-	-

Tabla 5

Resultados de la persona con discapacidad auditiva 5 en fondo monocromático.

Fondo monocromático					
Persona con discapacidad auditiva 5					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		98	912,4
2	B	X		100	967,7

3	C	X		98	950,1
4	D	X		94	979,1
5	E	X		95	934,5
6	F		X	-	-
7	G	X		95	996,8
8	H	X		95	940,3
9	I	X		99	971,9
10	J	X		98	952,2
11	K	X		93	948,6
12	L	X		99	954,7
13	M	X		98	903,6
14	N	X		97	989,1
15	O	X		99	947,4
16	P	X		96	911,2
17	Q	X		97	1034,5
18	R	X		98	930
19	S	X		96	999,7
20	T		X	-	-
21	U	X		98	928,3
22	V	X		99	977,7
23	W	X		95	900,4
24	X	X		94	998,2
25	Y	X		99	973,9
26	Z		X	-	-

Tabla 6

Resultados de la persona con discapacidad auditiva 6 en fondo monocromático.

Fondo monocromático					
Persona con discapacidad auditiva 6					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		99	903,4
2	B	X		99	943,8
3	C	X		99	940,1
4	D	X		95	937,7
5	E	X		97	951,3
6	F	X		93	992,9
7	G	X		95	1017,6
8	H		X	-	-
9	I	X		98	994,8
10	J	X		97	988,4
11	K	X		96	940,4
12	L	X		100	901,9

13	M	X		99	947,2
14	N	X		99	984,1
15	O	X		98	916,7
16	P	X		94	954
17	Q	X		98	985,5
18	R	X		95	977,3
19	S	X		97	982,6
20	T	X		92	998,1
21	U	X		97	934,7
22	V	X		98	903,4
23	W	X		94	949,7
24	X	X		91	920
25	Y	X		100	975,5
26	Z		X	-	-

Tabla 7

Resultados de la persona con discapacidad auditiva 7 en fondo monocromático.

Fondo monocromático					
Persona con discapacidad auditiva 7					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		97	932,1
2	B	X		100	967,6
3	C	X		98	901,4
4	D	X		93	932,3
5	E	X		95	975,2
6	F	X		93	984,3
7	G	X		94	954,9
8	H	X		92	955,6
9	I	X		99	911,3
10	J	X		98	937,7
11	K		X	-	-
12	L	X		99	930
13	M	X		98	964,8
14	N	X		97	912,6
15	O	X		99	948,4
16	P	X		95	944,3
17	Q	X		96	971,5
18	R	X		98	996,8
19	S	X		97	987,4
20	T	X		94	948,1
21	U	X		99	969,7
22	V	X		100	920,2

23	W	X	97	1047,4
24	X	X	90	951,3
25	Y	X	98	999,7
26	Z	X	-	-

Tras completar las pruebas de integración del sistema en el entorno de fondo monocromático con los siete usuarios con discapacidad auditiva, se obtuvo un promedio de precisión de traducción de 96,26% y un tiempo de respuesta promedio de 956,25 ms. Además, se calculó el porcentaje de detección correcta, que mide la efectividad del sistema para identificar cada letra del alfabeto ASL correctamente tras ser gestualizada por los usuarios. Este porcentaje se obtiene evaluando cuántas veces cada letra fue correctamente detectada en las siete pruebas realizadas (una por cada usuario).

En promedio, el sistema alcanzó un 91,21% de detección correcta en este entorno, demostrando una alta precisión en la identificación de letras, incluso con la variabilidad entre usuarios. La Tabla 8 muestra en detalle la cantidad de detecciones correctas para cada letra y su respectivo porcentaje de detección tras ser gestualizadas por los siete participantes.

Tabla 8

Detecciones correctas y porcentaje de detección correcta en fondo monocromático.

Fondo monocromático			
	Letra ASL	Detecciones correctas	Porcentaje de detección correcta (%)
1	A	7	100
2	B	7	100
3	C	7	100
4	D	7	100
5	E	7	100
6	F	6	85,71
7	G	7	100

8	H	4	57,14
9	I	7	100
10	J	7	100
11	K	3	42,86
12	L	7	100
13	M	7	100
14	N	7	100
15	O	7	100
16	P	7	100
17	Q	7	100
18	R	7	100
19	S	7	100
20	T	6	85,71
21	U	7	100
22	V	7	100
23	W	7	100
24	X	7	100
25	Y	7	100
26	Z	0	0
Promedio			91,21

Fondo con ruido visual

En este apartado se presentan los resultados de las pruebas realizadas por las personas con discapacidad auditiva en un entorno de fondo con ruido visual. Este tipo de fondo se caracteriza por contener múltiples colores y patrones, lo que añade complejidad visual y simula condiciones menos controladas, permitiendo evaluar el desempeño del sistema en situaciones con distracciones visuales. Durante estas pruebas, cada participante ejecutará las 26 letras del alfabeto ASL, registrándose si cada letra es detectada correctamente, así como el porcentaje de precisión de traducción y el tiempo de respuesta.

A continuación, se presentan los resultados individuales obtenidos en las pruebas realizadas en el entorno de fondo con ruido visual para cada una de las siete personas con discapacidad auditiva. Desde la Tabla 9 hasta la Tabla 15, se detallan los resultados de

cada participante, permitiendo observar el desempeño del sistema de manera específica para cada usuario en un entorno con mayor complejidad visual.

Tabla 9

Resultados de la persona con discapacidad auditiva 1 en fondo con ruido visual.

Fondo con ruido visual					
Persona con discapacidad auditiva 1					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		97	933,4
2	B	X		100	923,5
3	C	X		98	992,7
4	D	X		92	953,4
5	E	X		95	961,2
6	F	X		91	949,9
7	G	X		92	1013,9
8	H		X	-	-
9	I	X		97	978
10	J	X		95	921,5
11	K	X		94	1009,7
12	L	X		97	946,2
13	M	X		98	958,8
14	N	X		97	971,9
15	O	X		99	1006,3
16	P	X		98	1047,5
17	Q	X		97	907,8
18	R	X		96	962,5
19	S		X	-	-
20	T	X		93	994,2
21	U	X		98	999,6
22	V	X		99	982,4
23	W	X		94	913,4
24	X	X		94	917,7
25	Y	X		98	1003,5
26	Z		X	-	-

Tabla 10

Resultados de la persona con discapacidad auditiva 2 en fondo con ruido visual.

Fondo con ruido visual					
Persona con discapacidad auditiva 2					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		99	959,1
2	B	X		99	938
3	C	X		96	1002,1
4	D	X		91	937,7
5	E	X		96	1018,4
6	F	X		93	999,9
7	G	X		91	945,1
8	H	X		90	928,9
9	I	X		94	977,2
10	J	X		93	935,8
11	K		X	-	-
12	L	X		96	964,3
13	M	X		96	917,7
14	N	X		95	993,8
15	O	X		97	992,5
16	P	X		97	981,5
17	Q	X		92	933,2
18	R	X		97	992,4
19	S	X		94	995,5
20	T	X		95	914,8
21	U	X		97	983,3
22	V	X		97	977,3
23	W	X		96	939
24	X	X		95	987,9
25	Y	X		98	963,6
26	Z		X	-	-

Tabla 11

Resultados de la persona con discapacidad auditiva 3 en fondo con ruido visual.

Fondo con ruido visual					
Persona con discapacidad auditiva 3					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		89	1021,1
2	B	X		99	929,8

3	C	X		94	1013,1
4	D	X		89	998,4
5	E	X		96	953,4
6	F	X		90	922,9
7	G	X		92	976,5
8	H		X	-	-
9	I	X		94	936
10	J	X		91	1009,8
11	K		X	-	-
12	L	X		98	938,9
13	M	X		98	953,8
14	N	X		95	944,7
15	O	X		100	915,4
16	P	X		94	967,5
17	Q	X		94	973,6
18	R	X		96	944,1
19	S	X		95	999,2
20	T		X	-	-
21	U	X		98	920,9
22	V	X		98	986,9
23	W	X		96	995,3
24	X	X		93	990,5
25	Y	X		97	971,4
26	Z		X	-	-

Tabla 12

Resultados de la persona con discapacidad auditiva 4 en fondo con ruido visual.

Fondo con ruido visual					
Persona con discapacidad auditiva 4					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	945,4
2	B	X		97	959,7
3	C	X		97	992,8
4	D	X		92	973,7
5	E	X		95	975,3
6	F	X		94	984,1
7	G	X		93	972
8	H	X		91	948,8
9	I	X		97	955,9
10	J	X		94	1057,4
11	K		X	-	-
12	L	X		96	998,4

13	M	X		95	930,6
14	N	X		97	977,7
15	O	X		97	950,4
16	P	X		95	1010,1
17	Q	X		94	996,1
18	R	X		98	984,8
19	S	X		96	913,9
20	T	X		92	914,4
21	U	X		98	968,2
22	V	X		99	946
23	W	X		97	924,1
24	X	X		92	940,7
25	Y	X		97	975,6
26	Z		X	-	-

Tabla 13

Resultados de la persona con discapacidad auditiva 5 en fondo con ruido visual.

Fondo con ruido visual					
Persona con discapacidad auditiva 5					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		97	916
2	B	X		99	973,4
3	C	X		98	963,7
4	D	X		94	995,7
5	E	X		95	938,6
6	F		X	-	-
7	G	X		95	1018,4
8	H	X		94	948,9
9	I	X		98	977,5
10	J	X		98	963,8
11	K	X		93	952,2
12	L	X		97	971,4
13	M	X		98	914,9
14	N	X		97	996,7
15	O	X		98	959
16	P	X		96	923,8
17	Q	X		97	1049,1
18	R	X		98	952,6
19	S	X		96	1003,3
20	T		X	-	-
21	U	X		97	938,9
22	V	X		96	984,3

23	W	X		95	906
24	X	X		95	1005,1
25	Y	X		97	985,5
26	Z		X	-	-

Tabla 14

Resultados de la persona con discapacidad auditiva 6 en fondo con ruido visual.

Fondo con ruido visual					
Persona con discapacidad auditiva 6					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		98	907,5
2	B	X		99	955,4
3	C	X		97	949,7
4	D		X	-	-
5	E	X		97	960,9
6	F	X		92	999,5
7	G	X		95	1023,2
8	H		X	-	-
9	I	X		98	1008,4
10	J	X		95	995
11	K	X		96	927
12	L	X		99	901,5
13	M	X		99	954,8
14	N	X		97	993,7
15	O	X		98	946,3
16	P	X		93	963,5
17	Q	X		98	992,4
18	R	X		95	988,9
19	S	X		94	995,2
20	T	X		92	1018,7
21	U	X		97	945,3
22	V	X		97	911,9
23	W	X		94	953,3
24	X	X		91	944,6
25	Y	X		98	981,1
26	Z		X	-	-

Tabla 15

Resultados de la persona con discapacidad auditiva 7 en fondo con ruido visual.

Fondo con ruido visual					
Persona con discapacidad auditiva 7					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		96	945,7
2	B	X		99	974,2
3	C	X		98	931
4	D	X		93	939,8
5	E	X		95	982
6	F	X		92	996,9
7	G	X		94	964,5
8	H	X		92	961,2
9	I	X		98	929,8
10	J	X		97	944,3
11	K		X	-	-
12	L	X		99	937,3
13	M	X		98	976,4
14	N	X		97	923,2
15	O	X		98	995,1
16	P	X		95	959,8
17	Q	X		96	984,1
18	R	X		98	1012,6
19	S	X		96	995,4
20	T	X		94	954,2
21	U	X		99	976,3
22	V	X		99	933,8
23	W	X		97	1018
24	X	X		90	967,8
25	Y	X		97	1015,4
26	Z		X	-	-

Tras completar las pruebas de integración del sistema en el entorno de fondo con ruido visual con los siete usuarios con discapacidad auditiva, se obtuvo un promedio de precisión de traducción de 95,70% y un tiempo de respuesta promedio de 966,64 ms. Además, se calculó el porcentaje de detección correcta, con un promedio del 89,56% en este entorno, lo cual demuestra una alta precisión en la identificación de letras, pese a la complejidad visual y la variabilidad entre usuarios.

La Tabla 16 muestra en detalle la cantidad de detecciones correctas para cada letra y su respectivo porcentaje de detección tras ser gestualizadas por los siete participantes.

Tabla 16

Detecciones correctas y porcentaje de detección correcta en fondo con ruido visual.

Fondo con ruido visual			
	Letra ASL	Detecciones correctas	Porcentaje de detección correcta (%)
1	A	7	100
2	B	7	100
3	C	7	100
4	D	6	85,71
5	E	7	100
6	F	6	85,71
7	G	7	100
8	H	4	57,14
9	I	7	100
10	J	7	100
11	K	3	42,86
12	L	7	100
13	M	7	100
14	N	7	100
15	O	7	100
16	P	7	100
17	Q	7	100
18	R	7	100
19	S	6	85,71
20	T	5	71,43
21	U	7	100
22	V	7	100
23	W	7	100
24	X	7	100
25	Y	7	100
26	Z	0	0
	Promedio		89,56

Ambiente oscuro

En este apartado se presentan los resultados de las pruebas realizadas por personas con discapacidad auditiva en un entorno de ambiente oscuro. Este tipo de entorno se caracteriza por tener condiciones de iluminación reducida, lo cual representa un desafío adicional para el sistema, ya que requiere adaptarse a niveles bajos de luz para reconocer correctamente los gestos en lenguaje de señas. Durante las pruebas en ambiente oscuro, cada participante ejecutará las 26 letras del alfabeto ASL, registrándose si cada letra es detectada correctamente, así como el porcentaje de precisión de traducción y el tiempo de respuesta correspondiente.

A continuación, se presentan los resultados individuales de las pruebas realizadas en este entorno para cada una de las siete personas con discapacidad auditiva. Desde la Tabla 17 hasta la Tabla 23, se detallan los resultados obtenidos para cada participante, permitiendo conocer el desempeño específico del sistema en condiciones de iluminación limitada.

Tabla 17

Resultados de la persona con discapacidad auditiva 1 en ambiente oscuro.

Ambiente oscuro					
Persona con discapacidad auditiva 1					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		96	959
2	B	X		98	951,1
3	C	X		97	1027,7
4	D	X		92	949,1
5	E	X		95	981,3
6	F	X		90	963,5
7	G	X		92	984,5
8	H		X	-	-
9	I	X		95	971,6
10	J	X		95	957,9
11	K		X	-	-

12	L	X		95	966,8
13	M	X		98	987,4
14	N	X		97	994,5
15	O	X		98	1087,9
16	P	X		98	1125,1
17	Q	X		96	941,3
18	R	X		96	994,1
19	S	X		94	993,9
20	T	X		93	938,2
21	U	X		97	1003,2
22	V	X		99	976
23	W	X		93	925
24	X	X		94	927,4
25	Y	X		97	992,2
26	Z		X	-	-

Tabla 18

Resultados de la persona con discapacidad auditiva 2 en ambiente oscuro.

Ambiente oscuro					
Persona con discapacidad auditiva 2					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		99	992,5
2	B	X		98	948,6
3	C	X		96	1023,7
4	D	X		90	955,3
5	E	X		96	1033,1
6	F	X		93	1002,4
7	G	X		91	953,4
8	H	X		90	937,5
9	I	X		94	991,8
10	J	X		92	993,4
11	K		X	-	-
12	L	X		96	979,9
13	M	X		95	924,4
14	N	X		95	998,4
15	O	X		97	998,6
16	P		X	-	-
17	Q	X		92	983,8
18	R	X		96	1018
19	S	X		94	955,1
20	T	X		95	973,4
21	U	X		97	985,6

22	V	X		94	986,5
23	W	X		96	944,8
24	X	X		95	996,5
25	Y	X		97	975,3
26	Z		X	-	-

Tabla 19

Resultados de la persona con discapacidad auditiva 3 en ambiente oscuro.

Ambiente oscuro					
Persona con discapacidad auditiva 3					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		88	1031,2
2	B	X		97	982,4
3	C	X		94	1025,7
4	D	X		89	1026,1
5	E	X		94	977,1
6	F	X		90	949,4
7	G	X		92	986,1
8	H		X	-	-
9	I	X		94	942,2
10	J	X		91	1016,4
11	K		X	-	-
12	L	X		98	924,5
13	M	X		95	968,4
14	N	X		95	953,4
15	O	X		98	923
16	P	X		94	979,1
17	Q	X		94	996,2
18	R	X		96	1008,4
19	S	X		95	1013,8
20	T	X		93	938,5
21	U	X		95	938,4
22	V	X		98	1007,5
23	W	X		96	1010,3
24	X	X		93	999,1
25	Y	X		96	984
26	Z		X	-	-

Tabla 20

Resultados de la persona con discapacidad auditiva 4 en ambiente oscuro.

Ambiente oscuro					
Persona con discapacidad auditiva 4					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	955,3
2	B	X		95	972,5
3	C	X		97	1003,5
4	D	X		92	944,4
5	E	X		93	987,9
6	F	X		94	998,3
7	G	X		93	991,6
8	H	X		91	990,4
9	I	X		94	972,5
10	J	X		94	1079
11	K		X	-	-
12	L	X		96	1020,4
13	M	X		94	958,2
14	N	X		95	986,6
15	O	X		97	913,1
16	P	X		93	1023,5
17	Q	X		94	1008,7
18	R	X		98	996,4
19	S	X		95	925,5
20	T	X		92	928
21	U	X		97	979,6
22	V	X		97	962,2
23	W	X		97	976,4
24	X	X		91	953,2
25	Y	X		95	991,1
26	Z		X	-	-

Tabla 21

Resultados de la persona con discapacidad auditiva 5 en ambiente oscuro.

Ambiente oscuro					
Persona con discapacidad auditiva 5					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		96	921,6
2	B	X		98	983

3	C	X		98	973,4
4	D	X		94	1018,4
5	E	X		94	959,2
6	F		X	-	-
7	G		X	-	-
8	H	X		94	964,5
9	I	X		98	987,1
10	J	X		98	975,4
11	K	X		93	966,8
12	L	X		97	993,1
13	M	X		95	923,6
14	N	X		97	1084,4
15	O	X		97	982,3
16	P	X		96	950,4
17	Q	X		96	1093,2
18	R	X		98	959,2
19	S	X		94	1004,9
20	T		X	-	-
21	U	X		97	943,5
22	V	X		95	991,9
23	W	X		95	915,8
24	X	X		95	1018,5
25	Y	X		94	965,5
26	Z		X	-	-

Tabla 22

Resultados de la persona con discapacidad auditiva 6 en ambiente oscuro.

Ambiente oscuro					
Persona con discapacidad auditiva 6					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		98	973,1
2	B	X		99	965
3	C	X		95	953,9
4	D	X		93	961,9
5	E	X		97	974,5
6	F	X		91	1012,4
7	G	X		95	1045,8
8	H		X	-	-
9	I	X		98	1016
10	J	X		95	1003,5
11	K	X		95	938,5
12	L	X		99	928,1

13	M	X	97	972,4
14	N	X	94	999,3
15	O	X	97	942,2
16	P	X	93	971,4
17	Q	X	97	999,1
18	R	X	95	993,5
19	S	X	94	1018,9
20	T	X	91	1030,4
21	U	X	97	962,4
22	V	X	95	927,5
23	W	X	94	983,8
24	X	X	91	958,2
25	Y	X	97	1000,1
26	Z		X	-

Tabla 23

Resultados de la persona con discapacidad auditiva 7 en ambiente oscuro.

Ambiente oscuro					
Persona con discapacidad auditiva 7					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	952,7
2	B	X		99	992,1
3	C	X		98	944,6
4	D	X		92	940,4
5	E	X		95	993,3
6	F	X		92	1023,9
7	G	X		93	976,1
8	H	X		92	983,8
9	I	X		95	935,4
10	J	X		97	970,4
11	K		X	-	-
12	L	X		99	963,1
13	M	X		96	979,5
14	N	X		97	930,6
15	O	X		97	1000,5
16	P	X		95	984,2
17	Q	X		96	973,3
18	R	X		98	1027,2
19	S	X		96	1054,3
20	T	X		92	963,8
21	U	X		99	981,9
22	V	X		96	974,8

23	W	X		97	1059,5
24	X	X		90	927,2
25	Y	X		95	1022
26	Z		X	-	-

Tras completar las pruebas de integración del sistema en el entorno de ambiente oscuro con los siete usuarios con discapacidad auditiva, se obtuvo un promedio de precisión de traducción de 95,05% y un tiempo de respuesta promedio de 981,06 ms. Además, se calculó el porcentaje de detección correcta, alcanzando un promedio de 89,01% en este entorno, lo cual demuestra una alta precisión en la identificación de letras, incluso en condiciones de iluminación reducida y con la variabilidad entre usuarios.

La Tabla 24 muestra en detalle la cantidad de detecciones correctas para cada letra y su respectivo porcentaje de detección tras ser gestualizadas por los siete participantes.

Tabla 24

Detecciones correctas y porcentaje de detección correcta en ambiente oscuro.

Ambiente oscuro			
	Letra ASL	Detecciones correctas	Porcentaje de detección correcta (%)
1	A	7	100
2	B	7	100
3	C	7	100
4	D	7	100
5	E	7	100
6	F	6	85,71
7	G	6	85,71
8	H	4	57,14
9	I	7	100
10	J	7	100
11	K	2	28,57
12	L	7	100
13	M	7	100
14	N	7	100

15	O	7	100
16	P	6	85,71
17	Q	7	100
18	R	6	85,71
19	S	7	100
20	T	6	85,71
21	U	7	100
22	V	7	100
23	W	7	100
24	X	7	100
25	Y	7	100
26	Z	0	0
Promedio			89,01

Usuario a distancia considerable de la cámara

En este apartado se presentan los resultados de las pruebas realizadas por las personas con discapacidad auditiva en un entorno donde el usuario se encuentra a una distancia considerable de la cámara, específicamente a 2,50 metros. Este entorno permite evaluar el desempeño del sistema en condiciones de mayor alejamiento, simulando escenarios en los que el usuario no interactúa de cerca con el dispositivo de captura. Durante estas pruebas, cada participante ejecutó las 26 letras del alfabeto ASL, registrándose si cada letra fue detectada correctamente, así como el porcentaje de precisión de traducción y el tiempo de respuesta.

A continuación, se presentan los resultados individuales de las pruebas realizadas en este entorno para cada una de las siete personas con discapacidad auditiva. Desde la Tabla 25 hasta la Tabla 31, se detallan los resultados obtenidos para cada participante, permitiendo analizar el desempeño del sistema en condiciones de mayor distancia de captura.

Tabla 25

Resultados de la persona con discapacidad auditiva 1 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona con discapacidad auditiva 1					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	980,4
2	B	X		98	969,4
3	C	X		94	994,7
4	D	X		88	995,4
5	E	X		94	1001,9
6	F		X	-	-
7	G	X		90	996,1
8	H		X	-	-
9	I	X		94	1002,2
10	J	X		92	970,5
11	K	X		94	1088,1
12	L	X		95	990,5
13	M	X		95	999
14	N	X		94	1054,3
15	O	X		98	1097,9
16	P		X	-	-
17	Q	X		93	951,2
18	R	X		90	944,7
19	S	X		94	1006,5
20	T	X		89	960,6
21	U	X		91	1014,8
22	V	X		95	968,3
23	W	X		93	932,6
24	X	X		91	948
25	Y	X		93	935,9
26	Z		X	-	-

Tabla 26

Resultados de la persona con discapacidad auditiva 2 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona con discapacidad auditiva 2					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	972,1
2	B	X		96	998,2
3	C	X		92	1029,3
4	D	X		86	958,3
5	E	X		92	1036,2
6	F	X		88	1007
7	G	X		89	956,3
8	H	X		87	1001,4
9	I	X		92	966,7
10	J		X	-	-
11	K		X	-	-
12	L	X		96	993,5
13	M	X		95	986,7
14	N	X		92	959,3
15	O	X		94	1003,2
16	P	X		95	993,5
17	Q	X		92	1007,4
18	R	X		96	1018,6
19	S		X	-	-
20	T	X		95	1003
21	U	X		97	991,2
22	V	X		92	936,3
23	W	X		91	957,4
24	X	X		93	1015,1
25	Y	X		95	1008,9
26	Z		X	-	-

Tabla 27

Resultados de la persona con discapacidad auditiva 3 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona con discapacidad auditiva 3					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		88	1041,8
2	B	X		92	1013,1
3	C	X		90	1034,4
4	D		X	-	-
5	E	X		94	985,6
6	F	X		89	953
7	G	X		92	991,6
8	H		X	-	-
9	I	X		94	995,7
10	J	X		91	1039,1
11	K		X	-	-
12	L	X		95	953,9
13	M	X		95	974
14	N	X		89	956,9
15	O	X		95	984,5
16	P	X		94	994,2
17	Q	X		91	1010,6
18	R	X		96	997,5
19	S	X		94	1023,1
20	T	X		90	943,7
21	U	X		93	940,5
22	V	X		97	1009,8
23	W	X		93	996,7
24	X		X	-	-
25	Y	X		95	993,3
26	Z		X	-	-

Tabla 28

Resultados de la persona con discapacidad auditiva 4 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona con discapacidad auditiva 4					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		92	962,3
2	B	X		90	984,1
3	C	X		96	1210,1
4	D	X		88	953
5	E	X		93	995,4
6	F	X		91	1005,8
7	G	X		93	997,2
8	H	X		90	967,6
9	I	X		94	988,1
10	J	X		91	1081,6
11	K		X		-
12	L	X		96	1009,4
13	M	X		89	963,9
14	N	X		88	991,2
15	O	X		94	924,5
16	P	X		90	1030,1
17	Q	X		91	1012,4
18	R		X		-
19	S	X		93	938,1
20	T	X		88	977,5
21	U	X		91	930,2
22	V	X		90	954,9
23	W	X		95	955,4
24	X	X		88	973,9
25	Y	X		93	987,6
26	Z		X		-

Tabla 29

Resultados de la persona con discapacidad auditiva 5 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona con discapacidad auditiva 5					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		92	927,5
2	B	X		95	991,4
3	C	X		94	970
4	D	X		94	1041,5
5	E	X		90	963,4
6	F		X	-	-
7	G	X		89	1037,6
8	H	X		94	971,1
9	I	X		95	999,4
10	J	X		91	981,6
11	K		X	-	-
12	L	X		95	998,1
13	M	X		91	935,2
14	N	X		90	1095,1
15	O	X		96	993,8
16	P	X		94	957,6
17	Q		X	-	-
18	R	X		96	977,7
19	S	X		90	1024,5
20	T		X	-	-
21	U	X		94	969,1
22	V	X		93	1001,5
23	W	X		95	929,4
24	X	X		94	1023,2
25	Y	X		92	998,4
26	Z		X	-	-

Tabla 30

Resultados de la persona con discapacidad auditiva 6 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona con discapacidad auditiva 6					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	928,5
2	B	X		96	977,4
3	C	X		95	998,5
4	D	X		93	970,5
5	E	X		93	983,1
6	F	X		90	1024
7	G		X	-	-
8	H		X	-	-
9	I	X		97	1023,9
10	J	X		88	1007,1
11	K	X		89	948,5
12	L	X		95	975
13	M	X		93	994,4
14	N	X		94	1026,7
15	O	X		95	988,4
16	P	X		89	980,3
17	Q	X		93	1005,7
18	R	X		92	999,4
19	S	X		94	1027,4
20	T	X		88	1047,9
21	U	X		90	962,4
22	V	X		93	985,2
23	W	X		94	971,4
24	X	X		91	986,7
25	Y	X		95	993,7
26	Z		X	-	-

Tabla 31

Resultados de la persona con discapacidad auditiva 7 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona con discapacidad auditiva 7					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		93	952,7
2	B	X		95	992,1
3	C	X		94	944,6
4	D		X	-	-
5	E	X		90	993,3
6	F	X		88	1023,9
7	G	X		87	976,1
8	H	X		90	983,8
9	I	X		95	975,4
10	J	X		96	970,4
11	K		X	-	-
12	L	X		97	973,1
13	M	X		89	979,5
14	N	X		88	984,6
15	O	X		95	1000,5
16	P	X		95	984,2
17	Q	X		94	993,3
18	R	X		96	1027,2
19	S	X		96	1014,3
20	T	X		92	963,8
21	U		X	-	-
22	V	X		96	954,8
23	W	X		92	1059,5
24	X	X		88	997,2
25	Y	X		91	1022
26	Z		X	-	-

Tras completar las pruebas de integración del sistema en el entorno donde el usuario se encuentra a una distancia considerable de la cámara (2,50 metros) con los siete usuarios con discapacidad auditiva, se obtuvo un promedio de precisión de traducción de 92,61% y un tiempo de respuesta promedio de 990,84 ms. Además, se calculó el

porcentaje de detección correcta, con un promedio de 84,62% en este entorno, lo cual refleja una alta precisión en la identificación de letras, a pesar del desafío adicional que representa la mayor distancia de captura y la variabilidad entre usuarios.

La Tabla 32 muestra en detalle la cantidad de detecciones correctas para cada letra y su respectivo porcentaje de detección tras ser gestualizadas por los siete participantes en estas condiciones de prueba.

Tabla 32

Detecciones correctas y porcentaje de detección correcta en ambiente oscuro

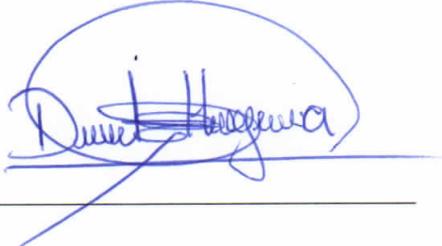
Usuario a distancia considerable			
	Letra ASL	Detecciones correctas	Porcentaje de detección correcta (%)
1	A	7	100
2	B	7	100
3	C	7	100
4	D	5	71
5	E	7	100
6	F	5	71,43
7	G	6	85,71
8	H	4	57,14
9	I	7	100
10	J	6	85,71
11	K	2	28,57
12	L	7	100
13	M	7	100
14	N	7	100
15	O	7	100
16	P	6	85,71
17	Q	6	85,71
18	R	6	85,71
19	S	6	86
20	T	6	85,71
21	U	6	85,71
22	V	7	100

23	W	7	100
24	X	6	85,71
25	Y	7	100
26	Z	0	0
Promedio			84,62

Los resultados obtenidos en el presente documento indican que el sistema de traducción de lenguaje de señas mantiene un alto desempeño en los cuatro entornos evaluados con usuarios con discapacidad auditiva, presentando variaciones en las métricas de precisión de traducción, tiempo de respuesta y porcentaje de detección correcta.

La precisión de traducción se mantuvo alta, superando el 92% incluso en condiciones menos favorables, como cuando el usuario se encontraba a mayor distancia de la cámara. El tiempo de respuesta, aunque ligeramente superior en dicho entorno, promedió menos de 1000 ms, asegurando una capacidad de respuesta eficiente en tiempo real. El porcentaje de detección correcta presentó una mayor variabilidad entre entornos, siendo menor cuando el usuario estaba a distancia y mayor en el fondo monocromático. Esto sugiere que entornos más complejos pueden influir en la detección, aunque el sistema ha demostrado ser confiable y adaptable en diversas condiciones de uso.

Firma de autorización

<p>Ing. Fabián Geovanny Cuzme Rodríguez</p>  <p>Director</p>	<p>Sr. David Santiago Hinojosa González</p>  <p>Estudiante</p>
---	--

ANEXO F: Resultados individuales de las pruebas de integración con personas sin discapacidad auditiva

Objetivo:

- Presentar los resultados individuales obtenidos en las pruebas de integración del sistema con usuarios sin discapacidad auditiva, detallando las métricas de precisión de traducción y tiempo de respuesta en los cuatro entornos evaluados: fondo monocromático, fondo con ruido visual, ambiente oscuro y usuario a distancia considerable de la cámara, para cada participante.

En el presente documento se exponen los resultados individuales obtenidos en las pruebas de integración del sistema realizadas con cada usuario sin discapacidad auditiva en los cuatro entornos evaluados. Para facilitar la comprensión y el análisis detallado, este anexo se organiza por entorno, mostrando los resultados específicos de cada participante en los diferentes escenarios: fondo monocromático, fondo con ruido visual, ambiente oscuro y usuario a distancia considerable de la cámara.

Se evaluará la precisión de traducción para cada usuario. Esta métrica hace referencia al porcentaje de confianza o coincidencia proporcionado por el modelo YOLOv8 al identificar una letra en lenguaje de señas. Este porcentaje indica el nivel de certeza del modelo sobre si la letra detectada corresponde a la gestualizada por el usuario.

Adicionalmente, se analizará la métrica del tiempo de respuesta para cada usuario. Este indicador se refiere al tiempo que tarda el sistema en procesar e inferir una letra gestualizada por el usuario. El tiempo de respuesta se mide internamente a través del CMD, que muestra el valor en milisegundos (ms) para cada inferencia realizada. Esta métrica es clave, ya que permite evaluar la eficiencia del sistema en tiempo real.

Fondo monocromático

En este apartado se presentan los resultados de las pruebas realizadas por las personas sin discapacidad auditiva en un entorno de fondo monocromático. Este tipo de fondo se caracteriza por ser de un color uniforme, sin patrones ni distracciones visuales, lo que permite evaluar el desempeño del sistema en condiciones visuales simples y controladas. Durante estas pruebas, el participante ejecutará las 26 letras del alfabeto ASL, registrándose si cada letra es detectada correctamente, así como el porcentaje de precisión de traducción y el tiempo de respuesta.

A continuación, se presentan los resultados individuales de las pruebas realizadas en un entorno de fondo monocromático para cada una de las siete personas sin discapacidad auditiva. Desde la Tabla 1 hasta la Tabla 7, se detallan los resultados obtenidos para cada participante, permitiendo conocer el desempeño del sistema de manera específica para cada usuario en condiciones visuales controladas.

Tabla 1

Resultados de la persona sin discapacidad auditiva 1 en fondo monocromático.

Fondo monocromático					
Persona sin discapacidad auditiva 1					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		96	931,2
2	B	X		97	985,7
3	C	X		88	954,9
4	D		X	-	-
5	E	X		90	927,2
6	F	X		87	954
7	G		X	-	-
8	H	X		91	948,5
9	I	X		95	971,3
10	J		X	-	-
11	K	X		91	959,6
12	L	X		94	915,1

13	M	X		84	950,8
14	N	X		86	996,9
15	O	X		91	987,4
16	P		X	-	-
17	Q	X		93	961,7
18	R	X		91	960
19	S	X		89	941,5
20	T		X	-	-
21	U	X		95	984,2
22	V	X		97	966,7
23	W	X		87	913,5
24	X	X		85	980,2
25	Y	X		94	991,9
26	Z		X	-	-

Tabla 2

Resultados de la persona sin discapacidad auditiva 2 en fondo monocromático.

Fondo monocromático					
Persona sin discapacidad auditiva 2					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		91	948,1
2	B	X		94	953,9
3	C	X		90	989,4
4	D	X		81	985,3
5	E	X		83	990,5
6	F		X	-	-
7	G		X	-	-
8	H		X	-	-
9	I	X		90	907
10	J	X		88	939,8
11	K	X		93	974,5
12	L	X		96	953,4
13	M		X	-	-
14	N	X		84	977,6
15	O	X		95	993,7
16	P		X	-	-
17	Q	X		91	923,7
18	R		X	-	-
19	S	X		85	958
20	T	X		86	917,6
21	U	X		93	983,8
22	V	X		92	967,1

23	W	X		89	947,1
24	X		X	-	-
25	Y	X		93	933,7
26	Z		X	-	-

Tabla 3

Resultados de la persona sin discapacidad auditiva 3 en fondo monocromático.

Persona sin discapacidad auditiva 3					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		97	985,4
2	B	X		90	963,1
3	C	X		81	995,9
4	D	X		86	987,6
5	E		X	-	-
6	F	X		85	917,3
7	G	X		84	970,3
8	H		X	-	-
9	I	X		92	927
10	J	X		88	982,8
11	K	X		93	946,5
12	L	X		94	923,3
13	M	X		88	952,5
14	N	X		85	911,9
15	O	X		94	900,8
16	P	X		96	989,4
17	Q	X		91	981,6
18	R		X	-	-
19	S	X		86	999,2
20	T	X		85	924
21	U	X		95	957,9
22	V	X		94	977,2
23	W	X		92	940,3
24	X	X		89	991,9
25	Y	X		93	955,3
26	Z		X	-	-

Tabla 4

Resultados de la persona sin discapacidad auditiva 4 en fondo monocromático.

Fondo monocromático					
Persona sin discapacidad auditiva 4					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	945,2
2	B	X		99	993
3	C	X		93	977,2
4	D		X	-	-
5	E	X		89	988,2
6	F		X	-	-
7	G	X		90	952,7
8	H	X		91	974,7
9	I		X	-	-
10	J		X	-	-
11	K	X		88	924,1
12	L	X		95	985,2
13	M	X		86	982,5
14	N		X	-	-
15	O	X		91	929,3
16	P	X		87	998,1
17	Q	X		92	980,7
18	R	X		84	977
19	S	X		90	941,9
20	T		X	-	-
21	U	X		95	959,5
22	V	X		93	940,2
23	W	X		94	917,1
24	X	X		89	930,4
25	Y	X		91	948,5
26	Z		X	-	-

Tabla 5

Resultados de la persona sin discapacidad auditiva 5 en fondo monocromático.

Fondo monocromático					
Persona sin discapacidad auditiva 5					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		98	970,7
2	B	X		100	977,1

3	C	X		95	942,5
4	D	X		88	991,8
5	E	X		89	930,8
6	F	X		89	967,2
7	G		X	-	-
8	H	X		90	945,7
9	I	X		93	952,3
10	J		X	-	-
11	K	X		90	998,7
12	L	X		96	953,1
13	M	X		87	905,4
14	N	X		89	996,8
15	O	X		97	937,4
16	P		X	-	-
17	Q		X	-	-
18	R	X		90	939,5
19	S	X		93	981,9
20	T	X		83	922,6
21	U		X	-	-
22	V	X		87	957,3
23	W	X		90	948,8
24	X	X		86	995
25	Y	X		96	978,2
26	Z		X	-	-

Tabla 6

Resultados de la persona sin discapacidad auditiva 6 en fondo monocromático.

Fondo monocromático					
Persona sin discapacidad auditiva 6					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	917
2	B	X		98	945,2
3	C		X	-	-
4	D	X		86	953,2
5	E	X		88	948,7
6	F	X		91	982,1
7	G	X		90	1023,4
8	H		X	-	-
9	I	X		94	995,1
10	J	X		89	991,4
11	K	X		89	925,5
12	L	X		98	937,2

13	M		X	-	-
14	N		X	-	-
15	O	X		94	929,3
16	P	X		85	948,3
17	Q	X		91	994,8
18	R	X		87	981
19	S	X		90	982,9
20	T		X	-	-
21	U	X		92	931,4
22	V	X		93	929,5
23	W	X		86	958,2
24	X	X		84	921,8
25	Y	X		93	977
26	Z		X	-	-

Tabla 7

Resultados de la persona sin discapacidad auditiva 7 en fondo monocromático.

Fondo monocromático					
Persona sin discapacidad auditiva 7					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	947,9
2	B	X		97	970,2
3	C	X		92	926,8
4	D	X		84	948,4
5	E	X		87	962,9
6	F	X		87	976,9
7	G	X		89	954,4
8	H	X		91	957,6
9	I		X	-	-
10	J		X	-	-
11	K	X		93	973,5
12	L		X	-	-
13	M	X		90	966,7
14	N	X		92	913,7
15	O	X		97	990,2
16	P	X		91	928,9
17	Q		X	-	-
18	R	X		88	1000,2
19	S		X	-	-
20	T		X	-	-
21	U	X		96	979,6
22	V	X		95	923,5

23	W		X	-	-
24	X	X		90	941,7
25	Y	X		93	1002,4
26	Z		X	-	-

Tras completar las pruebas de integración del sistema en el entorno de fondo monocromático con los siete usuarios sin discapacidad auditiva, se obtuvo un promedio de precisión de traducción de 90,74% y un tiempo de respuesta promedio de 959,02 ms. Además, se calculó el porcentaje de detección correcta, que mide la efectividad del sistema para identificar cada letra del alfabeto ASL correctamente tras ser gestualizada por los usuarios. Este porcentaje se obtiene evaluando cuántas veces cada letra fue correctamente detectada en las siete pruebas realizadas (una por cada usuario).

En promedio, el sistema alcanzó un 75,27% de detección correcta en este entorno. La Tabla 8 muestra en detalle la cantidad de detecciones correctas para cada letra y su respectivo porcentaje de detección tras ser gestualizadas por los siete participantes.

Tabla 8

Detecciones correctas y porcentaje de detección correcta en fondo monocromático.

Fondo monocromático			
	Letra ASL	Detecciones correctas	Porcentaje de detección correcta (%)
1	A	7	100
2	B	7	100
3	C	6	85,71
4	D	5	71,43
5	E	6	85,71
6	F	5	71,43
7	G	4	57,14
8	H	4	57,14
9	I	5	71,43
10	J	3	42,86
11	K	7	100

12	L	6	85,71
13	M	5	71,43
14	N	5	71,43
15	O	7	100
16	P	4	57,14
17	Q	5	71,43
18	R	5	71,43
19	S	6	85,71
20	T	3	42,86
21	U	6	85,71
22	V	7	100
23	W	6	85,71
24	X	6	85,71
25	Y	7	100
26	Z	0	0
	Promedio		75,27

Fondo con ruido visual

En este apartado se presentan los resultados de las pruebas realizadas por las personas sin discapacidad auditiva en un entorno de fondo con ruido visual. Este tipo de fondo se caracteriza por contener múltiples colores y patrones, lo que añade complejidad visual y simula condiciones menos controladas, permitiendo evaluar el desempeño del sistema en situaciones con distracciones visuales. Durante estas pruebas, cada participante ejecutará las 26 letras del alfabeto ASL, registrándose si cada letra es detectada correctamente, así como el porcentaje de precisión de traducción y el tiempo de respuesta.

A continuación, se presentan los resultados individuales obtenidos en las pruebas realizadas en el entorno de fondo con ruido visual para cada una de las siete personas sin discapacidad auditiva. Desde la Tabla 9 hasta la Tabla 15, se detallan los resultados de cada participante, permitiendo observar el desempeño del sistema de manera específica para cada usuario en un entorno con mayor complejidad visual.

Tabla 9

Resultados de la persona sin discapacidad auditiva 1 en fondo con ruido visual.

Fondo con ruido visual					
Persona sin discapacidad auditiva 1					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	943,2
2	B	X		97	986,8
3	C	X		89	963,7
4	D		X	-	-
5	E	X		88	985,1
6	F		X	-	-
7	G	X		90	973,9
8	H	X	X	-	-
9	I	X		93	980,9
10	J		X	-	-
11	K	X		90	970,1
12	L	X		93	924
13	M	X		85	956,5
14	N	X		85	979,2
15	O	X		96	991,8
16	P		X	-	-
17	Q	X		92	968,4
18	R	X		89	973,9
19	S	X		85	979,4
20	T	X		89	941,8
21	U	X		95	987,1
22	V	X		96	977,3
23	W	X	X	-	-
24	X	X		87	948,4
25	Y	X		92	990
26	Z		X	-	-

Tabla 10

Resultados de la persona sin discapacidad auditiva 2 en fondo con ruido visual.

Fondo con ruido visual					
Persona sin discapacidad auditiva 2					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		90	951,1
2	B	X		94	963,4

3	C		X	-	-
4	D	X		83	948,9
5	E	X		81	998,5
6	F		X	-	-
7	G	X		90	974
8	H		X	-	-
9	I	X		91	999,7
10	J	X		84	984,5
11	K	X		88	988,3
12	L	X		94	957,9
13	M	X		87	964,2
14	N		X	-	-
15	O	X		94	1007,4
16	P		X	-	-
17	Q	X		91	942,4
18	R		X	-	-
19	S	X		85	975,1
20	T	X		85	945,6
21	U	X		93	973,4
22	V		X	-	-
23	W	X		90	959,4
24	X	X		86	982,4
25	Y	X		94	957
26	Z		X	-	-

Tabla 11

Resultados de la persona sin discapacidad auditiva 3 en fondo con ruido visual.

Fondo con ruido visual					
Persona sin discapacidad auditiva 3					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	993,4
2	B	X		90	966,7
3	C	X		82	1025,4
4	D		X	-	-
5	E	X		88	1003,9
6	F	X		83	975,2
7	G		X	-	-
8	H		X	-	-
9	I	X		91	989,5
10	J	X		87	954,1
11	K		X	-	-
12	L	X		92	928

13	M	X		86	962,8
14	N	X		85	923,5
15	O	X		93	954,3
16	P	X		94	949,4
17	Q	X		89	1002,6
18	R		X	-	-
19	S	X		84	1038
20	T	X		82	942,9
21	U	X		95	965,4
22	V	X		91	977,6
23	W	X		90	988,1
24	X	X		88	971,5
25	Y	X		94	917,5
26	Z		X	-	-

Tabla 12

Resultados de la persona sin discapacidad auditiva 4 en fondo con ruido visual.

Fondo con ruido visual					
Persona sin discapacidad auditiva 4					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		93	987,1
2	B	X		95	924,3
3	C	X		93	974,8
4	D		X	-	-
5	E		X	-	-
6	F		X	-	-
7	G	X		88	971,5
8	H	X		90	996,5
9	I		X	-	-
10	J		X	-	-
11	K	X		88	978,9
12	L	X		93	999
13	M	X		88	947,4
14	N		X	-	-
15	O	X		90	932,5
16	P	X		85	1010,8
17	Q	X		93	985,1
18	R		X	-	-
19	S	X		88	971,4
20	T	X		86	991,6
21	U	X		92	977
22	V	X		94	949,4

23	W		X	-	-
24	X	X		84	944,6
25	Y	X		90	954
26	Z		X	-	-

Tabla 13

Resultados de la persona sin discapacidad auditiva 5 en fondo con ruido visual.

Fondo con ruido visual					
Persona sin discapacidad auditiva 5					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	977,5
2	B	X		97	975,8
3	C	X		94	960,8
4	D		X	-	-
5	E	X		90	989,1
6	F		X	-	-
7	G	X		87	952,4
8	H	X		89	977,2
9	I		X	-	-
10	J		X	-	-
11	K	X		90	940
12	L	X		95	983,5
13	M	X		87	923,8
14	N	X		87	1001,8
15	O	X		96	954,9
16	P		X	-	-
17	Q	X		91	988,2
18	R	X		88	977,1
19	S	X		93	990,5
20	T	X		81	941,8
21	U		X	-	-
22	V	X		86	959,3
23	W	X		90	948,9
24	X	X		85	1024,9
25	Y	X		94	978,1
26	Z		X	-	-

Tabla 14

Resultados de la persona sin discapacidad auditiva 6 en fondo con ruido visual.

Fondo con ruido visual					
Persona sin discapacidad auditiva 6					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	941,3
2	B	X		97	985,8
3	C	X		95	938,4
4	D	X		87	953,4
5	E	X		87	949,1
6	F		X	-	-
7	G	X		85	1017,2
8	H		X	-	-
9	I	X		92	1010,9
10	J	X		89	1001,4
11	K	X		88	935,1
12	L	X		97	935,8
13	M		X	-	-
14	N	X		92	940,2
15	O		X	-	-
16	P	X		86	955
17	Q	X		88	1008,4
18	R	X		87	999,6
19	S	X		91	991,2
20	T	X		89	977,5
21	U	X		92	959,2
22	V		X	-	-
23	W	X		85	999,2
24	X	X		84	941,7
25	Y	X		91	988
26	Z		X	-	-

Tabla 15

Resultados de la persona sin discapacidad auditiva 7 en fondo con ruido visual.

Fondo con ruido visual					
Persona sin discapacidad auditiva 7					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	952,9
2	B	X		95	947,7

3	C		X	-	-
4	D	X		86	948,8
5	E	X		85	970,4
6	F	X		87	985,1
7	G		X	-	-
8	H	X		90	979,3
9	I	X		90	984,5
10	J		X	-	-
11	K	X		91	997,6
12	L		X	-	-
13	M	X		85	983
14	N		X	-	-
15	O	X		95	974,3
16	P	X		90	952,5
17	Q	X		94	974,3
18	R	X		84	1018,9
19	S		X	-	-
20	T		X	-	-
21	U	X		92	983,6
22	V	X		95	923,9
23	W		X	-	-
24	X		X	-	-
25	Y	X		94	935,1
26	Z		X	-	-

Tras completar las pruebas de integración del sistema en el entorno de fondo con ruido visual con los siete usuarios sin discapacidad auditiva, se obtuvo un promedio de precisión de traducción de 89,89% y un tiempo de respuesta promedio de 970,55 ms. Además, se calculó el porcentaje de detección correcta, con un promedio del 70,88% en este entorno.

La Tabla 16 muestra en detalle la cantidad de detecciones correctas para cada letra y su respectivo porcentaje de detección tras ser gestualizadas por los siete participantes.

Tabla 16

Detecciones correctas y porcentaje de detección correcta en fondo con ruido visual.

Fondo con ruido visual			
	Letra ASL	Detecciones correctas	Porcentaje de detección correcta (%)
1	A	7	100,00
2	B	7	100,00
3	C	5	71,43
4	D	3	42,86
5	E	6	85,71
6	F	2	28,57
7	G	5	71,43
8	H	3	42,86
9	I	5	71,43
10	J	3	42,86
11	K	6	85,71
12	L	6	85,71
13	M	6	85,71
14	N	4	57,14
15	O	6	85,71
16	P	4	57,14
17	Q	7	100,00
18	R	3	42,86
19	S	7	100,00
20	T	6	85,71
21	U	6	85,71
22	V	5	71,43
23	W	4	57,14
24	X	6	85,71
25	Y	7	100,00
26	Z	0	0,00
	Promedio		70,88

Ambiente oscuro

En este apartado se presentan los resultados de las pruebas realizadas por personas sin discapacidad auditiva en un entorno de ambiente oscuro. Este tipo de entorno se caracteriza por tener condiciones de iluminación reducida, lo cual representa un desafío

adicional para el sistema, ya que requiere adaptarse a niveles bajos de luz para reconocer correctamente los gestos en lenguaje de señas. Durante las pruebas en ambiente oscuro, cada participante ejecutará las 26 letras del alfabeto ASL, registrándose si cada letra es detectada correctamente, así como el porcentaje de precisión de traducción y el tiempo de respuesta correspondiente.

A continuación, se presentan los resultados individuales de las pruebas realizadas en este entorno para cada una de las siete personas sin discapacidad auditiva. Desde la Tabla 17 hasta la Tabla 23, se detallan los resultados obtenidos para cada participante, permitiendo conocer el desempeño específico del sistema en condiciones de iluminación limitada.

Tabla 17

Resultados de la persona sin discapacidad auditiva 1 en ambiente oscuro.

Ambiente oscuro					
Persona sin discapacidad auditiva 1					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		93	974,1
2	B	X		93	951,8
3	C	X		85	999,4
4	D		X	-	-
5	E	X		85	987,6
6	F		X	-	-
7	G		X	-	-
8	H	X		84	1017,5
9	I	X		84	994,5
10	J		X	-	-
11	K	X		87	994,1
12	L	X		91	962,9
13	M	X		83	984,4
14	N	X		85	979,1
15	O	X		91	1009,5
16	P	X		87	983,5
17	Q	X		85	984
18	R		X	-	-

19	S	X		86	970,8
20	T		X	-	-
21	U	X		91	1013,4
22	V	X		92	992,1
23	W		X	-	-
24	X	X		88	957,1
25	Y	X		90	1005
26	Z		X	-	-

Tabla 18

Resultados de la persona sin discapacidad auditiva 2 en ambiente oscuro.

Ambiente oscuro					
Persona sin discapacidad auditiva 2					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		93	983,1
2	B		X	-	-
3	C	X		97	995,4
4	D	X		86	991,8
5	E	X		87	1008,9
6	F	X		90	984,2
7	G	X		90	996
8	H		X	-	-
9	I	X		91	956,7
10	J		X	-	-
11	K	X		88	1010,3
12	L	X		90	994,3
13	M	X		94	972
14	N		X	-	-
15	O	X		93	1024,1
16	P		X	-	-
17	Q	X		88	978,4
18	R		X	-	-
19	S	X		87	995,8
20	T	X		80	958,7
21	U	X		91	1000,7
22	V		X	-	-
23	W	X		87	981,2
24	X		X	-	-
25	Y	X		92	943,1
26	Z		X	-	-

Tabla 19*Resultados de la persona sin discapacidad auditiva 3 en ambiente oscuro.*

Ambiente oscuro					
Persona sin discapacidad auditiva 3					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		93	997,5
2	B	X		90	1014,9
3	C	X		83	1017,4
4	D		X	-	-
5	E	X		85	981,4
6	F		X	-	-
7	G		X	-	-
8	H	X		94	953,1
9	I	X		91	994,1
10	J	X		85	994,7
11	K		X	-	-
12	L	X		90	986,5
13	M	X		87	953,8
14	N	X		83	933,5
15	O	X		90	978
16	P		X	-	-
17	Q	X		94	1000,2
18	R	X		88	983,7
19	S	X		84	1015,4
20	T	X		82	992,9
21	U		X	-	-
22	V	X		90	1004,3
23	W	X		87	977,6
24	X	X		88	1003,5
25	Y	X		93	979,9
26	Z		X	-	-

Tabla 20*Resultados de la persona sin discapacidad auditiva 4 en ambiente oscuro.*

Ambiente oscuro				
Persona sin discapacidad auditiva 4				

	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		97	993,1
2	B	X		94	971,9
3	C	X		93	993,5
4	D		X	-	-
5	E	X		91	942,4
6	F		X	-	-
7	G	X		89	1002,4
8	H	X		87	999,5
9	I		X	-	-
10	J		X	-	-
11	K	X		86	998,8
12	L	X		91	1005
13	M		X	-	-
14	N		X	-	-
15	O	X		97	962,7
16	P	X		80	1023,1
17	Q	X		95	994,1
18	R	X		86	983,8
19	S	X		83	977
20	T		X	-	-
21	U	X		94	994,3
22	V	X		84	953,1
23	W		X	-	-
24	X	X		82	968,9
25	Y	X		86	992
26	Z		X	-	-

Tabla 21

Resultados de la persona sin discapacidad auditiva 5 en ambiente oscuro.

Ambiente oscuro					
Persona sin discapacidad auditiva 5					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A		X	-	-
2	B	X		98	1022,8
3	C	X		94	994,8
4	D		X	-	-
5	E	X		89	978,4
6	F		X	-	-
7	G	X		87	1032,1

8	H	X		92	983,7
9	I	X		90	991,3
10	J		X	-	-
11	K	X		87	971
12	L	X		99	988,3
13	M	X		85	931
14	N	X		86	1033,7
15	O		X	-	-
16	P	X		91	994,5
17	Q		X	-	-
18	R	X		85	974,3
19	S	X		91	1000,9
20	T	X		82	945,8
21	U		X	-	-
22	V	X		93	982,2
23	W	X		84	977,2
24	X		X	-	-
25	Y	X		87	998,1
26	Z		X	-	-

Tabla 22

Resultados de la persona sin discapacidad auditiva 6 en ambiente oscuro.

Ambiente oscuro					
Persona sin discapacidad auditiva 6					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	971,5
2	B	X		99	970,8
3	C	X		96	942,8
4	D		X	-	-
5	E		X	-	-
6	F	X		84	999,1
7	G	X		80	1002,7
8	H	X		89	1001,6
9	I	X		83	1013,2
10	J		X	-	-
11	K	X		93	981,1
12	L	X		91	953
13	M	X		92	986,7
14	N		X	-	-
15	O	X		94	1025
16	P	X		82	958,3
17	Q	X		87	1031,1

18	R		X	-	-
19	S	X		87	993,9
20	T	X		89	984,4
21	U	X		90	977,7
22	V		X	-	-
23	W	X		86	995,9
24	X	X		81	947,2
25	Y	X		87	1016,4
26	Z		X	-	-

Tabla 23

Resultados de la persona sin discapacidad auditiva 7 en ambiente oscuro.

Ambiente oscuro					
Persona sin discapacidad auditiva 7					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	935,9
2	B	X		96	1002,4
3	C		X	-	-
4	D		X	-	-
5	E	X		80	991,7
6	F		X	-	-
7	G	X		84	982,6
8	H	X		88	982,1
9	I	X		88	1007,2
10	J		X	-	-
11	K	X		90	921,1
12	L	X		91	997,5
13	M		X	-	-
14	N		X	-	-
15	O	X		85	1028
16	P	X		84	999,5
17	Q		X	-	-
18	R	X		82	1019,1
19	S	X		82	1000,3
20	T		X	-	-
21	U	X		93	995,8
22	V	X		86	932,5
23	W		X	-	-
24	X		X	-	-
25	Y	X		90	1029,2
26	Z		X	-	-

Tras completar las pruebas de integración del sistema en el entorno de ambiente oscuro con los siete usuarios sin discapacidad auditiva, se obtuvo un promedio de precisión de traducción de 88,64% y un tiempo de respuesta promedio de 987,16 ms. Además, se calculó el porcentaje de detección correcta, alcanzando un promedio de 67,03% en este entorno.

La Tabla 24 muestra en detalle la cantidad de detecciones correctas para cada letra y su respectivo porcentaje de detección tras ser gestualizadas por los siete participantes.

Tabla 24

Detecciones correctas y porcentaje de detección correcta en ambiente oscuro.

Ambiente oscuro			
	Letra ASL	Detecciones correctas	Porcentaje de detección correcta (%)
1	A	6	85,71
2	B	6	85,71
3	C	6	85,71
4	D	1	14,29
5	E	6	85,71
6	F	2	28,57
7	G	5	71,43
8	H	6	85,71
9	I	6	85,71
10	J	1	14,29
11	K	6	85,71
12	L	7	100,00
13	M	5	71,43
14	N	3	42,86
15	O	6	85,71
16	P	5	71,43
17	Q	5	71,43
18	R	4	57,14
19	S	7	100,00
20	T	4	57,14
21	U	5	71,43

22	V	5	71,43
23	W	4	57,14
24	X	4	57,14
25	Y	7	100,00
26	Z	0	0,00
Promedio			67,03

Usuario a distancia considerable de la cámara

En este apartado se presentan los resultados de las pruebas realizadas por las personas sin discapacidad auditiva en un entorno donde el usuario se encuentra a una distancia considerable de la cámara, específicamente a 2,50 metros. Este entorno permite evaluar el desempeño del sistema en condiciones de mayor alejamiento, simulando escenarios en los que el usuario no interactúa de cerca con el dispositivo de captura. Durante estas pruebas, cada participante ejecutó las 26 letras del alfabeto ASL, registrándose si cada letra fue detectada correctamente, así como el porcentaje de precisión de traducción y el tiempo de respuesta.

A continuación, se presentan los resultados individuales de las pruebas realizadas en este entorno para cada una de las siete personas sin discapacidad auditiva. Desde la Tabla 25 hasta la Tabla 31, se detallan los resultados obtenidos para cada participante, permitiendo analizar el desempeño del sistema en condiciones de mayor distancia de captura.

Tabla 25

Resultados de la persona sin discapacidad auditiva 1 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona sin discapacidad auditiva 1					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A		X	-	-
2	B	X		94	991,1
3	C		X	-	-
4	D	X		82	945,3
5	E	X		83	1007,9
6	F		X	-	-
7	G		X	-	-
8	H	X		88	1032,9
9	I	X		85	1004,5
10	J		X	-	-
11	K	X		89	1014,3
12	L	X		87	951,1
13	M		X	-	-
14	N		X	-	-
15	O	X		88	1030,9
16	P	X		90	993,4
17	Q		X	-	-
18	R	X		84	951,7
19	S	X		80	980,6
20	T		X	-	-
21	U		X	-	-
22	V	X		89	1008,2
23	W		X	-	-
24	X		X	-	-
25	Y	X		87	1021,3
26	Z		X	-	-

Tabla 26

Resultados de la persona sin discapacidad auditiva 2 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona sin discapacidad auditiva 2					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		94	1020,1
2	B	X		96	992,6
3	C		X	-	-
4	D		X	-	-
5	E	X		84	1028,2
6	F		X	-	-
7	G	X		82	1001,3
8	H		X	-	-
9	I	X		90	994,7
10	J		X	-	-
11	K		X	-	-
12	L	X		87	1003,9
13	M		X	-	-
14	N	X		83	938,5
15	O	X		81	1029
16	P		X	-	-
17	Q	X		84	951,4
18	R		X	-	-
19	S	X		86	999,9
20	T		X	-	-
21	U	X		95	997,7
22	V	X		91	1005
23	W		X	-	-
24	X		X	-	-
25	Y	X		85	989,1
26	Z		X	-	-

Tabla 27

Resultados de la persona sin discapacidad auditiva 3 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona sin discapacidad auditiva 3					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		91	1028,4
2	B		X	-	-
3	C	X		87	1020,1
4	D		X	-	-
5	E	X		80	1007,2
6	F		X	-	-
7	G	X		84	941,7
8	H		X	-	-
9	I	X		85	994,1
10	J	X		83	1019
11	K		X	-	-
12	L	X		93	938,2
13	M	X		85	973,8
14	N		X	-	-
15	O		X	-	-
16	P	X		83	988,6
17	Q	X		87	1023,2
18	R		X	-	-
19	S	X		85	1017,2
20	T	X		81	997,2
21	U		X	-	-
22	V	X		86	1011,3
23	W		X	-	-
24	X	X		81	973,1
25	Y		X	-	-
26	Z		X	-	-

Tabla 28

Resultados de la persona sin discapacidad auditiva 4 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona sin discapacidad auditiva 4					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		95	975,1
2	B	X		96	989,5
3	C	X		91	1009,3
4	D		X	-	-
5	E		X	-	-
6	F		X	-	-
7	G	X		81	1032,7
8	H	X		83	999,9
9	I		X	-	-
10	J		X	-	-
11	K	X		85	952,4
12	L	X		92	1023,8
13	M		X	-	-
14	N		X	-	-
15	O		X	-	-
16	P	X		81	1033,1
17	Q	X		84	974,6
18	R		X	-	-
19	S	X		89	972,3
20	T	X		82	1002,9
21	U	X		83	1024,9
22	V	X		88	943,6
23	W		X	-	-
24	X	X		81	1009,9
25	Y	X		88	989,4
26	Z		X	-	-

Tabla 29

Resultados de la persona sin discapacidad auditiva 5 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona sin discapacidad auditiva 5					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		97	1030,8
2	B		X	-	-
3	C	X		91	984,1
4	D		X	-	-
5	E		X	-	-
6	F	X		86	1029,5
7	G	X		81	1013,4
8	H		X	-	-
9	I	X		93	923,5
10	J		X	-	-
11	K		X	-	-
12	L	X		88	994,3
13	M	X		84	947,9
14	N		X	-	-
15	O	X		90	998
16	P		X	-	-
17	Q		X	-	-
18	R		X	-	-
19	S	X		83	1003,9
20	T		X	-	-
21	U	X		88	987,7
22	V	X		82	991,3
23	W	X		83	1014,2
24	X		X	-	-
25	Y		X	-	-
26	Z		X	-	-

Tabla 30

Resultados de la persona sin discapacidad auditiva 6 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona sin discapacidad auditiva 6					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A	X		98	926,4
2	B	X		84	999,5
3	C		X	-	-
4	D	X		89	1028,9
5	E		X	-	-
6	F	X		85	1009,9
7	G	X		80	1000,5
8	H		X	-	-
9	I	X		81	1023,2
10	J	X		89	971,6
11	K		X	-	-
12	L	X		95	941,8
13	M		X	-	-
14	N	X		91	994,3
15	O	X		91	1008,9
16	P		X	-	-
17	Q		X	-	-
18	R		X	-	-
19	S	X		83	1035,2
20	T	X		85	974,6
21	U	X		85	1007,7
22	V		X	-	-
23	W		X	-	-
24	X	X		80	989,2
25	Y	X		86	1019,5
26	Z		X	-	-

Tabla 31

Resultados de la persona sin discapacidad auditiva 7 a distancia considerable de la cámara.

Usuario lejos de la cámara					
Persona sin discapacidad auditiva 7					
	Letra ASL	Detección correcta	Detección incorrecta	Precisión de traducción (%)	Tiempo de respuesta (ms)
1	A		X	-	-
2	B	X		97	1007,4
3	C		X	-	-
4	D		X	-	-
5	E	X		81	997,7
6	F	X		83	994,5
7	G		X	-	-
8	H	X		90	1014,6
9	I	X		90	1023,1
10	J		X	-	-
11	K	X		87	923,9
12	L	X		92	999,2
13	M	X		83	976
14	N		X	-	-
15	O	X		91	1039,4
16	P		X	-	-
17	Q		X	-	-
18	R	X		80	1029
19	S	X		82	951,2
20	T	X		85	998,4
21	U	X		92	1029,3
22	V	X		95	935,5
23	W		X	-	-
24	X		X	-	-
25	Y		X	-	-
26	Z		X	-	-

Tras completar las pruebas de integración del sistema en el entorno donde el usuario se encuentra a una distancia considerable de la cámara (2,50 metros) con los siete usuarios sin discapacidad auditiva, se obtuvo un promedio de precisión de traducción de

86,79% y un tiempo de respuesta promedio de 994,97 ms. Además, se calculó el porcentaje de detección correcta, con un promedio de 51,10% en este entorno.

La Tabla 32 muestra en detalle la cantidad de detecciones correctas para cada letra y su respectivo porcentaje de detección tras ser gestualizadas por los siete participantes en estas condiciones de prueba.

Tabla 32

Detecciones correctas y porcentaje de detección correcta en ambiente oscuro

Usuario a distancia considerable			
	Letra ASL	Detecciones correctas	Porcentaje de detección correcta (%)
1	A	5	71,43
2	B	5	71,43
3	C	3	42,86
4	D	2	28,57
5	E	4	57,14
6	F	3	42,86
7	G	2	28,57
8	H	3	42,86
9	I	6	85,71
10	J	2	28,57
11	K	3	42,86
12	L	7	100,00
13	M	3	42,86
14	N	2	28,57
15	O	5	71,43
16	P	3	42,86
17	Q	3	42,86
18	R	2	28,57
19	S	7	100,00
20	T	4	57,14
21	U	5	71,43
22	V	6	85,71
23	W	1	14,29
24	X	3	42,86

25	Y	4	57,14
26	Z	0	0,00
Promedio			51,10

Los resultados obtenidos en el presente documento indican que el sistema de traducción de lenguaje de señas mantiene un rendimiento aceptable en los cuatro entornos evaluados con usuarios sin discapacidad auditiva, aunque se observan diferencias en comparación con usuarios que sí dominan el lenguaje de señas. La precisión de traducción superó el 86% incluso en condiciones menos favorables, como cuando el usuario se encontraba a mayor distancia de la cámara. El tiempo de respuesta promedió 977,92 ms, lo cual asegura un procesamiento eficiente en tiempo real. El porcentaje de detección correcta fue más variable, reflejando la falta de familiaridad de los usuarios sin experiencia en lenguaje de señas, con un incremento de errores, especialmente en el entorno donde el usuario se encontraba a una distancia considerable de la cámara. Estos resultados sugieren que, aunque el sistema se desempeña de forma estable, su precisión en la detección puede verse afectada por la práctica y conocimiento del usuario.

Firma de autorización

<p>Ing. Fabián Geovanny Cuzme Rodríguez</p>  <p>Director</p>	<p>Sr. David Santiago Hinojosa González</p>  <p>Estudiante</p>
---	--