UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE

INGENIERIAS EN CIENCIAS APLICADAS CARRERA:

INGENIERÍA MECATRÓNICA

INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR,

MODALIDAD DE PROYECTO DE INVESTIGACIÓN

TEMA:

"Prototipo para la localización de secciones y productos por percha para personas con discapacidad visual.""

Trabajo de titulación previo a la obtención del título de Ingeniería Mecatrónica

Línea de investigación: Prototipos Industriales

Autor: Christian Damián Borja Cadena

Director: Fernando Vinicio Valencia Aguirre

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1002987699		
APELLIDOS Y NOMBRES:	BORJA CADENA CHRISTIAN DAMIAN		
DIRECCIÓN:	VICTOR GOMEZJURADO 1-15		
EMAIL:	cdborjac@utn.edu.ec		
TELÉFONO FIJO:	062658131	**TELÉFONO MÓVIL:**	+593983727582

DATOS DE LA OBRA	
TÍTULO:	PROTOTIPO PARA LA LOCALIZACIÓN DE SECCIONES Y PRODUCTOS POR PERCHA PARA PERSONAS CON DISCAPACIDAD VISUAL
AUTOR (ES):	BORJA CADENA CHRISTIAN DAMIAN
FECHA: DD/MM/AAAA	12/03/2023
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☒ **PREGRADO** ☐ **POSGRADO**
TITULO POR EL QUE OPTA:	INGENIERO MECATRÓNICO
ASESOR /DIRECTOR:	MSc. Fernando Vinicio Valencia Aguirre

2. CONSTANCIAS

El autor (es) manifiesta (n) 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 (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 12 días del mes de marzo de 2023.

EL AUTOR:

Firma.....

Nombre: Christian Damian Borja Cadena

CONSTANCIAS

El autor (es) manifiesta (n) 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 (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume(n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 12 días, del mes de marzo de 2023.

EL AUTOR:

Firma.....

Nombre: Christian Damian Ibarra Cadena

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTERGRACIÓN CURRICULAR

Ibarra, 14 de marzo de 2023

MSc. Fernando Vinicio Valencia Aguirre

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

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

(f) 

……

Msc. Fernando Valencia

C.C.: 1003188669

APROBACIÓN DEL COMITÉ CALIFICADOR

El Tribunal Examinador del trabajo de titulación “Prototipo para la localización de secciones y productos por percha para personas con discapacidad visual” elaborado por Christian Damian Borja Cadena, previo a la obtención del título de Ingeniero Mecatrónico, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

(f):…….

Msc. Fernando Valencia

C.C.:1003188669

(f):…….

PhD. David Ojeda

C.C.:

REGISTRO BIBLIOGRÁFICO

DEDICATORIA

Este trabajo de investigación se la dedico a toda mi familia y amigos, que me acompañaron en el transcurso de esta nueva meta.

Damian

AGRADECIMIENTOS

Agradezco a todos quienes me acompañaron en este largo camino.

Damián

RESUMEN EJECUTIVO

El presente trabajo responde a las exigencias como sociedad que permiten la inclusión, en este caso, de personas con discapacidad visual, facilitando su independencia en actividades cotidianas como es realizar compras dentro de un supermercado, con el uso de la Inteligencia Artificial para reconocer secciones de productos en percha. Con el uso de modelos de entrenamiento de IA, se realizan diferentes estudios con la finalidad de obtener un modelo que optimice el sistema embebido a usar y satisfaga las necesidades de los usuarios no videntes. La utilidad del dispositivo en el usuario se evidencia de manera inmediata, reconociendo un símbolo previamente diseñado y notificando al usuario que ingresó a un sector específico.

Palabras clave: Visión artificial, detección de objetos, discapacidad visual, Deep learning, machine learning.

ABSTRACT

The present work responds to the demands as a society that allows the inclusion, in this case, of people with visual disabilities, facilitating their independence in their daily activities such as making purchases in a supermarket, with the use of Artificial Intelligence to recognize sections of products on hanger. With the use of AI training models, different studies are conducted in order to obtain a model that optimizes the embedded system to be used and satisfies the needs of blind users. The utility of the device to the user is immediately evident, recognizing a previously designed symbol and notifying the user that they have entered a specific sector.

Keywords: artificial vision, object detection, visual disability, deep learning, machine learning.

ÍNDICE

IDENTIFICACIÓN DE LA OBRA 2
AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD 3
REGISTRO BIBLIOGRÁFICO 7
DEDICATORIA 8
AGRADECIMIENTOS 9
RESUMEN EJECUTIVO 10
ABSTRACT 11
ÍNDICE 12
ÍNDICE DE FIGURAS 15
ÍNDICE DE TABLAS 19
INTRODUCCIÓN 21
DESCRIPCIÓN DEL PROBLEMA 21
OBJETIVOS 22
Objetivo general 22
Objetivos específicos 22
JUSTIFICACIÓN 22
ALCANCE 24
1. CAPÍTULO I 25
MARCO REFERENCIAL 25
1.1. DISCAPACIDAD VISUAL 25
1.1.1. Autoconcepto y discapacidad visual 25
1.1.2. Orientación y movilidad en las personas con discapacidad visual 26
1.1.3. Dispositivos de movilidad 27

1.2. ESTADO DEL ARTE 31
1.3. CONCEPTOS BÁSICOS DEL PROCESAMIENTO DE IMÁGENES 33
1.3.1. Píxeles en imágenes 34
1.3.2. Ubicación de píxeles: sistema de coordenadas de imagen 34
1.3.3. Valores de píxeles 35
1.3.4. Open CV 36
1.4. INTELIGENCIA ARTIFICIAL 40
1.4.1. Subcampos de la IA 40
1.4.2. Ingeniería y Sistema Experto (Engineering and Expert System) 42
1.4.3. Models of Brain and Evolution 42
1.5. MECANISMOS DE DETECCIÓN E IDENTIFICACIÓN DE OBJETOS 43
1.5.1. Visión Artificial 44
1.5.2. Detección de objetos de aprendizaje profundo posterior 44
1.5.3. Machine learning 48
1.5.4. Deep Learning 51
1.5.5. Modelos de detección de objetos 53
1.5.6. Evaluación del rendimiento de un modelo de aprendizaje automático ... 56
2. CAPÍTULO II 62
MARCO METODOLÓGICO 62
2.1. TIPO DE INVESTIGACIÓN 62
2.2. NIVEL DE INVESTIGACIÓN 62
2.3. METODOLOGÍA 63
2.3.1. Fase 1: Adquisición de requerimientos 63
2.3.2. Fase 2: Diseño del Sistema 68
2.3.3. Fase 3: Diseño del algoritmo de visión artificial 71

2.3.4. Fase 4: Construcción e implementación del prototipo 78
3. CAPÍTULO III 87
RESULTADOS 87
3.1. ESPECIFICACIONES DEL SISTEMA 87
3.1.1. Especificaciones de Técnicas: 87
3.2. FUNCIONAMIENTO 88
3.2.1. Fase 5: Validación y pruebas de funcionamiento. 88
3.3. ANÁLISIS Y RESULTADOS 98
4. CAPÍTULO IV 100
CONCLUSIONES Y RECOMENDACIONES 100
4.1. CONCLUSIONES 100
4.2. RECOMENDACIONES 101
BIBLIOGRAFÍA 103
ANEXOS 111
ANEXO1: GUION DE ENTREVISTA 111
ANEXO 2: ENTREVISTA 113
ANEXO 3: FLUJOGRAMA 115
ANEXO 4: DIAGRAMA DE BLOQUES 116
ANEXO 5: ALGORITMO 118

ÍNDICE DE FIGURAS

Figura 1. *Bastón blanco* 28

Figura 2. *Bastón Verde* 28

Figura 3. Canes amaestrados como guias 29

Figura 4. *Dispositivo Sonic Pathfinder* 29

Figura 5. *Cinturón strap* 30

Figura 6. *Banda Sunu para no videntes* 30

Figura 7. Dispositivo de ultrasonido para movilidad 31

Figura 8. Guante guía para no videntes 32

Figura 9. Prototipo de traje para la orientación y movilidad de no videntes. 32

Figura 10. Chaleco y cinturón para orientación y movilidad de personas con discapacidad visual. 33

Figura 11. Imagen construida por mas de un millo 34

Figura 12. Imagen que pone en referencia el sistemas de coordenadas con su punto de origen 34

Figura 13. Grafica representativa de valores de cada pixel 35

Figura 14. Relación RGB de una imagen. 36

Figura 15 Numpy funcionalidades 37

Figura 16. Pandas funcionalidades 37

Figura 17. Imutils ejemplo de cambio de colores 38

Figura 18. Imagen distribuida en matrices 38

Figura 19. Neurona 44

Figura 20 Representación red neuronal artificial 45

Figura 21. Tipos de aprendizaje 49

Figura 22. Diagrama de bloques de MobileNet-SSD Arquitectura 54

Figura 23: Dos conceptos de detección de objetos arquitectónicos 55

Figura 24. Curva de ROC con 5 ejemplos 60

Figura 25. Diagrama de Bloques 69

Figura 26. Flujo grama del Proceso General 70

Figura 27. Fotografía de productos del supermercado 71

Figura 28 *Símbolos Bliss para el reconocimiento de los percheros* 72

Figura 29 *Interfaz camera-capture* 73

Figura 30 *Primer entrenamiento* 75

Figura 31 *Segundo entrenamiento* 76

Figura 32 *Clasificación live camera* 77

Figura 33 *Descarga JetPAck SDK* 79

Figura 34 *Flashear imagen JetPack* 79

Figura 35 *Terminal Ubuntu* 80

Figura 36 *Versiones Python OpenCV* 80

Figura 37 *Repositorio Github* 81

Figura 38 Model Downloader 81

Figura 39 *Docker* 82

Figura 40 *Partición memoria SWAP* 83

Figura 41 *Información SWAP* 83

Figura 42 *SWAP modificada* 84

Figura 43 *SWAP modificada* 84

Figura 44. Modulo Wemos 18650 V3 85

Figura 45. Diagrama de conexiones del sistema 86

Figura 46. Carcasa del sistema embebido 86

Figura 47: Curva de ROC- Objeto: Lácteos 89

Figura 48. Detección de estante de lácteos 90

Figura 49: Curva de ROC- Objeto: Carnes 91

Figura 50. Detección de estante de carnes 92

Figura 51: Curva de ROC- Objeto: Vegetales: 93

Figura 52. Detección de estante de Vegetales 94

Figura 53: Curva de ROC- Objeto: Bebidas 95

Figura 54. Detección de estante de Bebidas 96

Figura 55: Curva de ROC- Objeto: Frutas 97

Figura 56. Detección de estante de frutas 98

ÍNDICE DE TABLAS

Tabla 1. Análisis de rendimiento de los tres modelos. 56

Tabla 2: Matriz de confusión 58

Tabla 3. *Fórmulas para el cálculo de parámetros obtenidos* 58

Tabla 4: Requerimientos de usuario 65

Tabla 5. Requerimientos Operacionales 66

Tabla 6 . Requerimientos del sistema 66

Tabla 7. Requerimientos de Arquitectura 67

Tabla 8. Requerimientos del Sistema 68

Tabla 9. Tabla comparativa de selección de tarjeta embebida 78

Tabla 10: Matriz de confusión – objeto: Lácteos 88

Tabla 11: Parámetros del algoritmo – objeto: Lácteos 88

Tabla 12: Matriz de confusión – objeto: Carnes 90

Tabla 13: Parámetros del algoritmo – objeto: Carnes 90

Tabla 14: Matriz de confusión – objeto: Vegetales: 92

Tabla 15: Parámetros del algoritmo – objeto: Vegetales: 92

Tabla 16: Matriz de confusión – objeto: Bebidas..... 94

Tabla 17: Parámetros del algoritmo – objeto: Bebidas 94

Tabla 18: Matriz de confusión – objeto: Frutas 96

Tabla 19: Parámetros del algoritmo – objeto: Frutas 96

Tabla 20. Tabla general de resultados 98

INTRODUCCIÓN

Descripción del problema

En Ecuador existen 54,662 personas con discapacidad visual mayor al 30% de grado de discapacidad según el CONADIS, esta institución pública toma en cuenta si esta es superior al 30%. Según Las estadísticas nacionales el 95% de las personas con discapacidad visual son mayores de edad, y presentan una cierta autonomía para realizar sus actividades diarias (CONADIS, 2021).

Una de las dificultades que se presentan en su día a día es el de reconocer un producto en un micro mercado al momento de ir de compras. En ocasiones necesitan de otra persona que les ayude con estas actividades cotidianas, lo cual impide que tengan la autonomía que ellos quisieran, esta dependencia puede llegar a afectar la autoestima de las personas con discapacidad (Solano, 2021). El bastón blanco presenta una ayuda a las personas no videntes y a un bajo costo, sin embargo, este dispositivo, con el paso del tiempo puede verse obsoleto y con limitaciones.

La visión artificial por computadora es una disciplina en creciente auge con múltiples aplicaciones, como inspección automática, reconocimiento de objetos, mediciones y robótica. Con los avances de la tecnología, se busca facilitar y mejorar la calidad de vida de las personas con discapacidad visual, usando dispositivos que utilicen las nuevas tecnologías. La visión por computador como sustituto de la visión humana, es una herramienta fundamental en la implementación de dispositivos de apoyo a personas con discapacidad visual. La visión artificial, a diferencia de las tecnologías que incorporan dispositivos electrónicos con sensores,

permite una interpretación del entorno, ofreciendo un mayor grado de representación de la realidad a partir de mayor complejidad en el procesamiento de la información.

Objetivos

Objetivo general

Diseñar un prototipo para la asistencia en el reconocimiento de productos de consumo masivo en un supermercado para personas con discapacidad visual con el uso de inteligencia artificial y etiquetas con símbolo.

Objetivos específicos

- Identificar los requerimientos para la propuesta mediante la revisión de la bibliografía y el estudio de dispositivos existentes en el mercado.
- Diseñar un algoritmo con el uso de inteligencia artificial capaz de reconocer símbolos que representan los productos de la canasta básica.
- Construir un prototipo portátil que pueda ser usado por personas con discapacidad visual para el reconocimiento de productos de consumo masivo usando símbolos en las etiquetas de un micro mercado.
- Validar los resultados de las pruebas del prototipo desarrollado en un grupo de personas con discapacidad visual.

Justificación

La presente investigación se enfoca un estudio de herramientas tecnológicas, visión artificial, que permitan reconocer productos de consumo masivo en un supermercado, para que puedan ser usadas por personas con discapacidad visual, ya sea parcial o completa, ya que este

segmento de personas presenta dificultades al momento de realizar compras y depende de otra persona para poder realizarlas de manera correcta. Con el uso de la inteligencia artificial se pretende entrenar un algoritmo que sea capaz de reconocer diferentes productos y la información obtenida del producto sea transmitida por audio al usuario. El entrenamiento con el tiempo puede ser optimizado y el algoritmo será capaz de reconocer de mejor manera los productos, esto incluye velocidad de procesamiento y mayor cantidad de objetos a reconocer. Beneficios que tendrán los no videntes. El usuario mejorará su estado emocional y psicológico al mejorar su autonomía y no depender de otra persona para realizar las compras. El dispositivo forma parte de un mercado inclusivo, que es un mercado que requiere atención, presenta necesidades y que por la limitante de no ser un mercado con un beneficio económico representativo no termina de ser atendido.

Alcance

Este trabajo de grado tiene la finalidad de construir un prototipo para reconocer productos de consumo masivo, por medio de un algoritmo de visión artificial, mediante el entrenamiento de una red neuronal. Los productos serán reconocidos mediante símbolos previamente establecidos e implementados en las señaléticas de un micro mercado. El prototipo podrá ser usado por personas mayores de 18 años que tengan una discapacidad visual superior al 30%. La etiqueta será reconocida, y por medio de un dispositivo de salida de audio el usuario podrá escuchar que tipo de producto tiene al frente. Las pruebas serán realizadas en personas con discapacidad visual superior al 60% y en un rango de edades entre 20 y 40 años, en un micro mercado de la ciudad.

CAPÍTULO I

Marco Referencial

En este capítulo se investiga y analiza los conceptos previos a la construcción del algoritmo de visión artificial, que nos ayudara a identificar los requerimientos para la propuesta mediante la revisión de la bibliografía y el estudio de dispositivos existentes en el mercado.

1.1. Discapacidad visual

La discapacidad es un concepto que evoluciona y forma parte de la condición humana, la definición de discapacidad ha tenido cambios con el paso de los años. Actualmente se asume el modelo dinámico multidimensional de la Clasificación Internacional del Funcionamiento, de la salud y discapacidad. Este modelo presenta un modelo de diversidad funcional en las actividades y define a la discapacidad, desde el punto de vista relacional, como un término que abarca deficiencias, limitaciones de actividad y restricciones en la participación. En el caso de la discapacidad visual, se refiere a una deficiencia visual que limita la actividad de ver, ya sea parcial o total, y restringe la participación en la orientación de la persona (Jiménez y otros, 2002).

1.1.1. Autoconcepto y discapacidad visual

Se entiende por autoconcepto un sistema de creencias y opiniones aprendidas por cada persona durante su crecimiento, estas opiniones son consideradas verdaderas en cuanto a su existencia personal se refiere. El autoconcepto influye directamente en el desarrollo social y sicológico en la niñez y adolescencia de cada persona; la consecución de metas, crecimiento personal y evolución en la sociedad se basan en el autoconcepto que cada persona tiene. Los

cimientos del autoconcepto se forman durante los primeros años, etapa en la que el niño empieza a interactuar con su entorno mediante la comprensión, exploración y experiencia. (Brodercick & Blewitt, 2006) Investigadores descubrieron, en un experimento realizado a 60 adolescentes no videntes, que el 94% de la muestra tienen un autoconcepto de bajo a moderado. (Halder & Datta, 2011). Este bajo autoconcepto por parte de los adolescentes puede llegar a generar actitudes negativas, comportamientos pesimistas y principalmente se originan en la dependencia para realizar actividades diarias, que una persona vidente puede realizar sin tener esta dependencia.

1.1.2. Orientación y movilidad en las personas con discapacidad visual

El desarrollo de habilidades para desempeñar actividades cotidianas como caminar por la calle, tomar un transporte, desplazarse al lugar de trabajo o al sitio de vivienda implica para las personas con discapacidad visual adquirir técnicas de orientación y movilidad que les permita ejercer su autonomía e independencia.

Además de la utilización del bastón, que de por sí ya brinda seguridad a quien lo lleva, se requiere de otras habilidades que van desde el reconocimiento del entorno por donde ha de movilizarse, hasta la interacción con otras personas.

Sin embargo, cuando las circunstancias obligan a transitar por sitios desconocidos, el bastón termina convirtiéndose en una ayuda imprescindible y complementaria de su movilidad.

Los recorridos que se realizan frecuentemente, como ir de la casa al sitio de estudio o regresar a la casa desde el lugar de trabajo, se memorizan como si fuese un mapa mental, de manera que las personas con discapacidad visual ya no necesitan de la ayuda de otras personas para orientarse.

En el mapa mental que se memoriza se van fijando puntos de referencia como el olor de la panadería que queda en la esquina, un paso nivel en la calle, el sonido de una fuente de agua o el color fuerte en la fachada de un edificio, este último referente para las personas con baja visión (Lozada, 2018).

1.1.3. Dispositivos de movilidad

Existe una variedad de aparatos de orientación y movilidad para los individuos que padecen de incapacidades de la vista. Esta variedad se le divide en dispositivos de baja y alta tecnología.

Baja Tecnología:

Hay varios medios de baja tecnología, ya que estas ameritan únicamente de un manejo básico de modalidad como, por ejemplo:

Bastón Blanco. El bastón blanco (Figura 1) es una herramienta usada por las personas con ceguera parcial o total que permite al usuario desplazarse con mayor facilidad. El bastón debe llevarse con el brazo ligeramente doblado, cerca del cuerpo y centrado por la línea media (puede tomarse como referencia el ombligo). La mano debe sujetar el bastón con el dedo índice prolongado a lo largo en la parte plana del mango y los dedos restantes sujetando el bastón. Si el bastón no se centra, la persona tiende a caminar torcida. El extremo bajo del bastón ayuda a la persona a revisar la zona por donde se desplaza la persona, detectando irregularidades que pueda presentar el piso por dónde camina y de esta manera evitar accidentes, el cuerpo del bastón protege el tren inferior del usuario (Juarez, 2014).

Figura 1. *Bastón blanco*

Fuente: (AFADACS, 2020)

Bastón verde. Es una herramienta de movilidad como el anterior, la diferencia es que este tipo de bastón lo utilizan personas de baja visión, elaborado en aluminio, plegable en cuatro cañas, con un mango de caucho ilustrado en la Figura 2 (Silva, 2019).

Figura 2. *Bastón Verde*

Fuente: (Silva, 2019)

Perro Guía. Este can está entrenado para reconocer y evitar obstáculos, tanto estáticos como en movimiento, a nivel del suelo o en altura (Figura 3). Este "perro lazarillo" está capacitado también para desobedecer una orden del usuario cuando su ejecución implique un

peligro para su integridad física, debido a una circunstancia que no haya advertido, como un vehículo que se aproxima en un cruce de calle (ONCE, 2018).

Figura 3. Canes amaestrados como guias

Fuente: (ONCE, 2018)

Alta tecnología:

Existen varios tipos de dispositivos de alta tecnología, entre ellos los mas renombrados son:

Sonic Pathfinder: El Sonic Pathfinder advierte por adelantado al usuario si existen objetos en su camino, por medio de ocho tonos musicales diferentes el aparato advierte al usuario la distancia y la posición del objeto que detecta en el camino, este dispositivo se lo coloca en la cabeza como se puede ver en la Figura 4 (Young, 2016).

Figura 4. *Dispositivo Sonic Pathfinder*

Fuente: (Young, 2016)

Strap (Figura 5***):*** detecta objetos que están a más distancia que un bastón largo, el usuario escucha tonos que indican la distancia a la que se encuentra el objeto, el Strap no debe usarse como la única ayuda para transportarse. (Silva, 2019).

Figura 5. *Cinturón strap*

Fuente: (Pérez y Jiménez., 2023)

Sunu Band: La Sunu Band puede ser usado como ayuda secundaria en conjunto con un bastón largo o un perro guía, cuando el obstáculo está dentro del área de la Sunu Band, éste vibra o emite un sonido (ORIENTATECH, 2023).

Figura 6. *Banda Sunu para no videntes*

Fuente: (ORIENTATECH, 2023)

1.2. Estado del Arte

Existen varios trabajos de investigación en el campo de la orientación y movilidad de las personas no videntes, como los siguientes:

Dispositivo de ultrasonido para movilidad. El dispositivo presenta una forma de vara similar al bastón blanco, la diferencia radica en el uso de un sistema electrónico capaz de detectar obstáculos sin necesidad de golpearlos. Posee sensores que miden la distancia de los objetos, y alertan al usuario por medio de vibraciones, este sistema es ergonómico como se ve en la Figura 7. (Vargas, 2015)

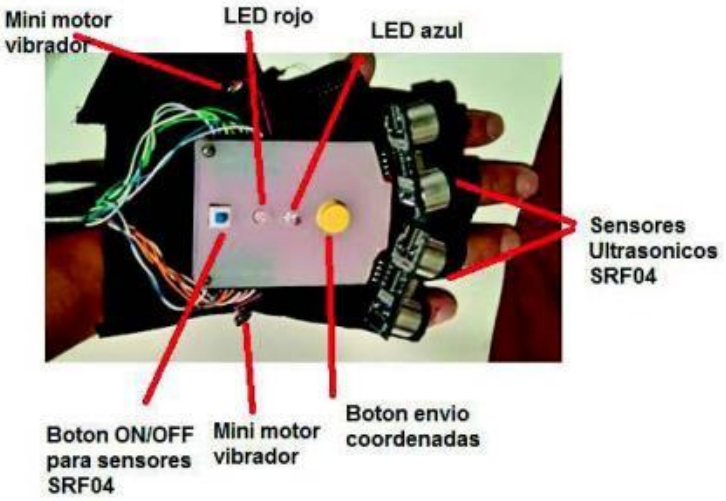
Figura 7. Dispositivo de ultrasonido para movilidad

Fuente: (Vargas, 2015)

Guante con Orientación y localización. Un prototipo que se basa en sensores ultrasónicos integrados en el dispositivo como se ve en la Figura 8,que son usados para detectar

y medir la distancia a la que se encuentra un obstáculo durante el desplazamiento de la persona. Mediante vibraciones alerta al usuario la distancia a la que se encuentra un objeto. Posee un sistema de localización en base a la tecnología GSM/GPS, a través del cual el usuario puede enviar su ubicación a algún familiar o persona de confianza en caso de presentarse alguna emergencia o eventualidad. La geolocalización ayuda a asegurarse que la persona no vidente se encuentra en lugares seguros (Azaña, 2017)

Figura 8. Guante guía para no videntes



Fuente: (Azaña, 2017)

Traje tecnológico de no videntes. Es un traje especial que alerta al usuario (Figura 9), por medio de vibraciones, objetos cercanos por medio del uso de sensores, estos sensores fueron distribuidos en partes estáticas y estratégicas del cuerpo humano, para que la recepción de señales por parte de los sensores sea más fácil y se pueda detectar obstáculos por debajo y por encima de la cintura. La cantidad de sensores preliminares colocados se distribuyeron, dos en los hombros, dos sensores de proximidad en la cintura, uno por cada mano, y uno en cada pie. (Condo, 2013)

Figura 9. Prototipo de traje para la orientación y movilidad de no videntes.

Fuente: (Condo, 2013)

Chalecos y cinturones. Los chalecos y cinturones (Figura 10) brindan instrucciones de navegación por medio de vibraciones que alertan al usuario sobre obstáculos. La frecuencia y patrón de las vibraciones varían respecto a la distancia que se encuentra un obstáculo, son diferentes instrucciones que se interpretan como pueden ser detenerse, avanzar, o si el usuario puede aumentar o disminuir la velocidad de desplazamiento (Ortega & Sanchez, 2016).

Figura 10. Chaleco y cinturón para orientación y movilidad de personas con discapacidad visual.

Fuete: (Ortega y Sanchez, 2016)

1.3. Conceptos básicos del procesamiento de imágenes

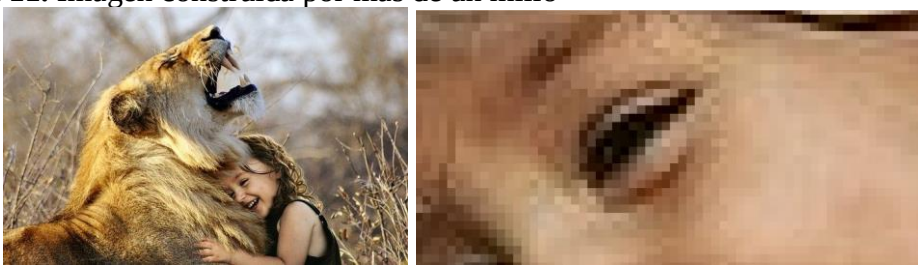
EL PROCESAMIENTO de imágenes tiene como objetivo mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar, una imagen

puede haber sido generada de muchas maneras, por ejemplo, fotográficamente, o electrónicamente, por medio de monitores de televisión, para tener en claro este concepto es necesario ver diferentes fundamentos que ayudan a complementar el conocimiento.

1.3.1. Píxeles en imágenes

Las imágenes están formadas por píxeles Figura 11. Los píxeles se pueden considerar como objetos pequeños de forma cuadrada que, cuando se combinan, forman una imagen. Sirven como pequeños bloques de construcción de cualquier imagen (Hafsa et al., 2020).

Figura 11. Imagen construida por mas de un millo



Fuente: (Hafsa et al., 2020)

1.3.2. Ubicación de píxeles: sistema de coordenadas de imagen

Se hace referencia a un píxel específico utilizando su ubicación en la imagen, cada imagen tiene un sistema de coordenadas específico y el estándar es que la esquina superior izquierda de una imagen actúa como origen, (0,0), como se ilustra en la Figura 12 (Hafsa et al., 2020).

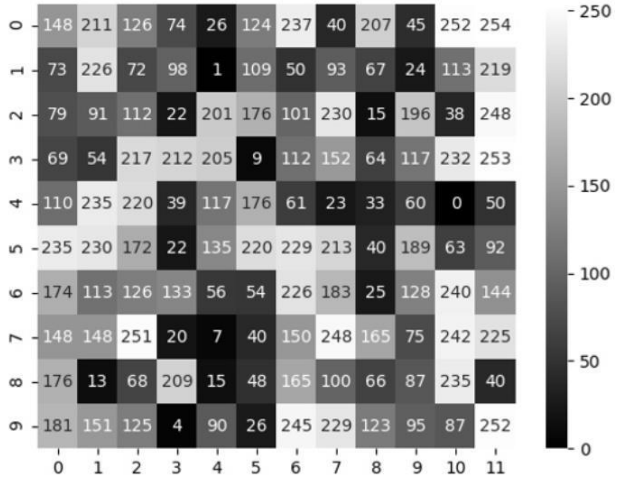
Figura 12. Imagen que pone en referencia el sistemas de coordenadas con su punto de origen

Fuente: (Hafsa et al., 2020)

1.3.3. Valores de píxeles

En el caso más simple de las imágenes binarias, el valor del píxel es un número de 1 bit que indica el primer plano o el fondo. Para representar imágenes en color, se deben especificar componentes separados de rojo, verde y azul para cada píxel (asumiendo un espacio de color RGB), así el valor del píxel es en realidad un vector de tres números como se muestra en la Figura 13 (GTI, 2020).

Figura 13. Grafica representativa de valores de cada pixel

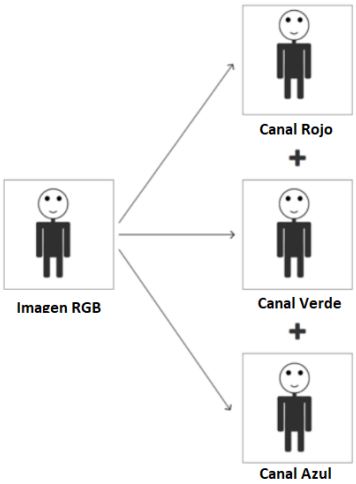


Fuente: (Hafsa et al., 2020)

A menudo los tres componentes diferentes se almacenan como tres imágenes separadas en escala de grises conocidas como planos de color, uno para cada uno de los colores rojo, verde

y azul, que tienen que ser recombinados cuando se muestran o procesan como se corelaciona en la Figura 14.

Figura 14. Relación RGB de una imagen.



Fuente: (Hafsa et al., 2020)

La escala de grises o la intensidad de los componentes de color de cada píxel puede no se almacenarse explícitamente.

A menudo, lo que se almacena para cada píxel es un índice en un mapa de colores en el que se puede buscar la intensidad real o los colores.

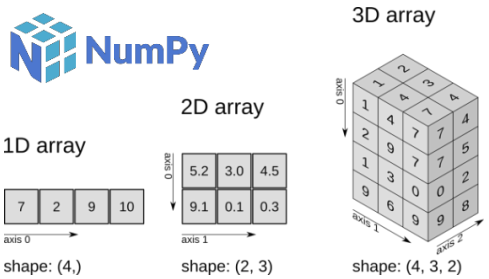
1.3.4. Open CV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Lo que lo diferencia de otras bibliotecas de visión por computadora es que es rápido y fácil de usar, brinda soporte para bibliotecas como QT y OpenGL y, lo que es más importante, brinda herramientas de dispositivos de hardware Intel. Estas potentes características/ventajas hacen de OpenCV una buena opción para comprender e implementar diferentes conceptos de visión artificial.

Librerías para el manejo de OpenCV

Numpy. Es una biblioteca de Python que se especializa en computación y análisis de datos, especialmente para grandes conjuntos de datos. La ventaja de Numpy(Figura 15) sobre las listas integradas de Python es que la operación de matrices es más rápida (hasta 50 veces más rápida) lo cual lo hace eficiente para manejar matrices y vectores grandes (NUMPY, 2023).

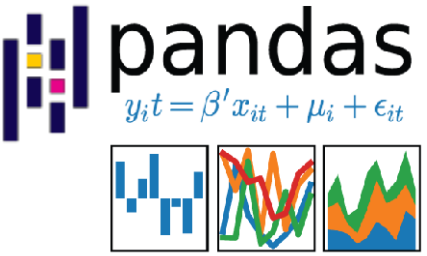
Figura 15 Numpy funcionalidades



Fuente: (NUMPY, 2023)

Pandas. Es una biblioteca Python esencial para el procesamiento y análisis de datos (Figura 16). Proporciona estructuras de datos y funciones para trabajar con tablas de números y series temporales, como Excel en Python. Es un software gratuito distribuido bajo la licencia BSD.1 El nombre proviene de datos de panel, un término econométrico para datos que combinan escalas de tiempo y secciones transversales (Pandas.org, 2022).

Figura 16. Pandas funcionalidades



Fuente: (Pandas.org, 2022)

Imutils. Esta librería da muchas funciones útiles para simplificar operaciones gráficas importantes como transformación, rotación, tamaño, esqueletización, visualización de imágenes de Matplotlib, forma de contorno, reconocimiento de límites y más con OpenCV y Python 2.7 y Python 3 como se puede observar nen un ejemplo de cambio de matices en la Figura 17 (PyImageSearch/imutils, 2023).

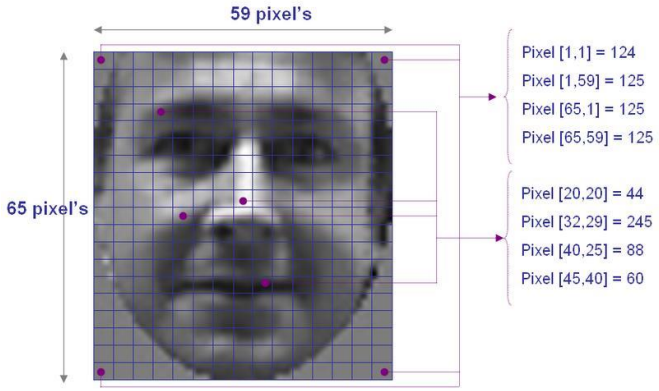
Figura 17. Imutils ejemplo de cambio de colores

Fuente: (PyImageSearch/imutils, 2023)

Imágenes en OpenCV

OpenCV tiene su propia clase para representar imágenes: cv::Mat. Viene la parte "Mat" del término matriz. Ahora, esto no debería ser una sorpresa ya que las imágenes son nada más que matrices como se puede ver en la Figura 18 (Hafsa et al., 2020).

Figura 18. Imagen distribuida en matrices



Fuente: (Hafsa et al., 2020)

Toda imagen tiene tres atributos específicos a sus dimensiones: ancho, alto y número de canales. también se conoce que cada canal de una imagen es una colección de valores de píxeles que se encuentran entre 0 y 255.

Observe cómo el canal de una imagen comienza a parecerse a una matriz 2D. Entonces, una imagen se convierte en una colección de matrices 2D apiladas una encima de la otra.

Como resumen rápido, al usar OpenCV en Python, las imágenes se representan como matrices NumPy. NumPy es un módulo de Python comúnmente utilizado para el cálculo numérico.

Es por eso que una imagen RGB (que tiene tres canales) se verá como tres matrices 2D NumPy apiladas una encima de la otra (Hafsa et al., 2020).

Acceso y manipulación de píxeles

Se usa OpenCV para leer y procesar una imagen, no obstante puede dividir y fusionar los canales de una imagen, por ende puede acceder y manipular píxeles, junto con los componentes básicos de una imagen.

OpenCV accede y manipula píxeles en función, usando las ubicaciones en la sección con el sistema de coordenadas de una imagen. También se tiene en cuenta que las imágenes en OpenCV en Python se representan como matrices NumPy. Es por eso que el problema de acceder a los píxeles se convierte en el problema general de acceder a los elementos de una matriz NumPy (Hafsa et al., 2020).

1.4. Inteligencia artificial

La IA comúnmente se refiere al campo de la ciencia destinado a desarrollar máquinas con las mismas capacidades de un ser humano, tales como lógica, razonamiento, planificación, aprendizaje y percepción. Este tipo de inteligencia no se presenta únicamente en máquinas o robots, también es usada en diversas aplicaciones y sistema (Perez y otros, 2018).

Actualmente las tecnologías IA son usadas en aviación, conducción, en lo que se conoce como vehículos autónomos, publicidad online, medicina, para el reconocimiento temprano de enfermedades y reconocimiento facial. Sin embargo, las actuales tecnologías son todavía limitadas para muchas aplicaciones. El sentido común es una capacidad del ser humano que la IA no puede interpretar con los conocimientos o datos adquiridos, en caso de que la decisión que deba tomar dependa de si está “bien” o “mal”, la IA no será una herramienta de utilidad, lo cual representa una limitación importante a considerar para el desarrollo de esta tecnología, que contrario o lo que se piensa, todavía tiene un camino largo por desarrollar para ser aceptada por toda la comunidad, especialmente en el ámbito ético.

1.4.1. Subcampos de la IA

La IA consiste en numerosos subcampos, el propósito de los sistemas IA evoluciona y existen tipos de sistemas de IA, que usan diversas técnicas, como las siguientes:

Procesamiento de voz (Speech Processing)

El procesamiento de voz comenzó en los años 70, cuando DARPA (Defense Advanced Research Projects Agency) fundó un consorcio de laboratorios líderes en el área de procesamiento de voz, el cual tenía como objetivo crear un sistema completamente funcional de reconocimiento de voz con un amplio vocabulario. (Perez y otros, 2018). El procesamiento de voz comienza con el análisis de características, que extrae un conjunto de parámetros que caracterizan las propiedades espectrales de los diversos sonidos del habla, filtra la señal obtenida y combina la información con un modelo acústico, un vocabulario establecido y un modelo de lenguaje para encontrar la secuencia óptima de palabra. Un ejemplo de este subtipo es la herramienta “Siri” del sistema operativo iOS (Rabiner & Juang, 2006).

Lenguaje de procesamiento natural (Natural Language Processing)

Lenguaje de procesamiento natural, hace referencia al área de especialización en ciencias de la computación que se encarga de analizar o derivar información útil proveniente del lenguaje humano, también conocido como lenguaje natural. Técnicas como tokenización que consiste sustituir un elemento de datos sensibles por un equivalente no sensible, un ejemplo es el pago móvil a través de una aplicación móvil con una tarjeta de crédito; clasificación de palabras, detección de oraciones, se usan en procesamientos de lenguaje natural de alto nivel. Como ejemplo una palabra puede tener significados diferentes dependiendo del contexto. Búsqueda de palabras, traducciones, reconocimiento de entidades nombradas e incluso el reconocimiento de voz, son algunos ejemplos de NLP. (Gollapudi, 2019)

Planning

Específicamente la planificación robótica se relaciona con escoger la secuencia correcta de acciones para alcanzar un objetivo o cumplir con una tera determinada, para esto se requiere tener una representación clara y eficiente del problema. La planificación no debe únicamente cumplir con la tares especifica asignada, esta planificación debe ser óptima en tiempo y espacio (Chowdhary, 2020).

1.4.2. Ingeniería y Sistema Experto (Engineering and Expert System)

El problema fundamental de AI es la representación del conocimiento, como representarlo. Varios esquemas de representación se clasifican en dos categorías: representación **simbólica,** que es proposicional y basada en la lógica de primer orden, y la representación **gráfica,** como las redes semánticas, estructuras de datos, ontologías y desentendencias conceptuales. (Chowdhary, 2020)

Fuzzy Systems

El objetivo de los sistemas difusos o *soft-computing* es aprovechar la tolerancia de imprecisión e incertidumbre, para alcanzar manejabilidad, robustez y aplicaciones de bajo costo. (Chowdhary, 2020). Estos sistemas pueden ser integrados con otras técnicas.

1.4.3. Models of Brain and Evolution

Los modelos de cerebro humano y su evolución se relacionan con dos aproximaciones con la inteligencia artificial. La primera es que la psicología del humano opera a un nivel simbólico, como lo hacen algunos lenguajes de programación de inteligencia artificial, que

trabajan en términos matemáticos de alto nivel, procesamiento de datos que corresponde a una visión macroscópica de la inteligencia artificial.

La segunda aproximación es operar como la evolución del ser humano, con redes neurales artificiales y algoritmos basados en la genética de bajo nivel del concepto de la vida. No son modelos que le dan vida a una máquina o a un proceso, se basan en el concepto de evolución y el aprendizaje que tiene el ser humano para resolver problemas, las redes neurales aprenden por si mismas desde patrones. Este aprendizaje es usado para la clasificación, predicción o control de aplicaciones que presentan soluciones, en la que cada solución es un programa de computador y engloba la inteligencia artificial (Chowdhary, 2020).

Existen diferentes tipos de sistemas de Inteligencia Artificial basados en su propósito principal. Cada tipo de sistema representa un avance en esta tecnología para desarrollar sistemas más inteligentes.

1.5. Mecanismos de detección e identificación de objetos

El reconocimiento de objetos se define como la capacidad de los programas informáticos para identificar ciertos objetos en videos, imágenes o archivos PDF en función de ciertos grupos o categorías. Estos grupos pueden hacer referencia a caras, matrículas, personas, pantallas, papeles, etc. Esto se basa en la idea de que cada clase de objeto tiene su propio conjunto de propiedades o características.

Es necesario aprender los conceptos arraigados a la detección para comprender el mecanismo y poder redefinir las propiedades del mismo.

1.5.1. Visión Artificial

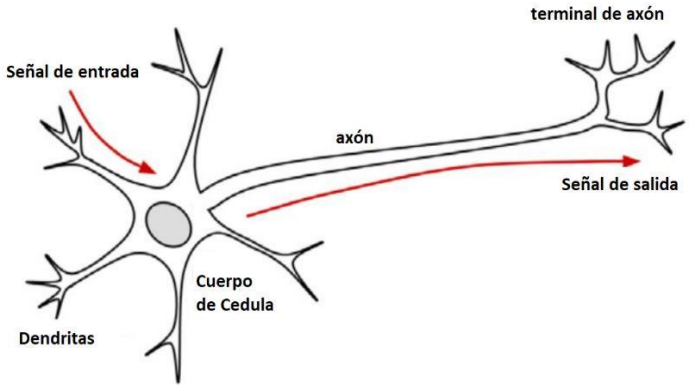
Es la rama de la inteligencia artificial que se encarga de tratar las imágenes, en el cual se intenta que el computador imite el funcionamiento de los ojos humanos. La visión artificial o visión por computador es una técnica que usa la clasificación de objetos, localización y detección para identificar el tipo de objeto que se muestra en una imagen, la localización de objetos y ambos, identificación y localización a la vez (Esteva, y otros, 2021). Se define como el conjunto de métodos usados por un computador para adquirir, analizar y procesar imágenes y transformarlas en datos numéricos que el computador pueda usar (Morena, Molina, & Garay, 2019).

1.5.2. Detección de objetos de aprendizaje profundo posterior

Red Neuronal Artificial

Modela la relación entre un conjunto de señales de entrada y una señal de salida, similar a la actividad cerebral humana. En una neurona, las celdas llamadas dendritas reciben las señales de entrada, estas se ponderan de acuerdo a su frecuencia e importancia. Mientras las señales de entrada se acumulan, estas alcanzan un umbral en el cual la célula dispara una señal de salida que es transmitida por medio del axón, como se muestra en la Figura 19.

Figura 19. Neurona

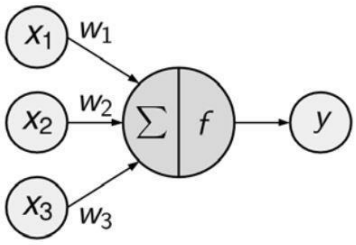


Fuente: (Lantz, 2013)

Una red neuronal biológica está compuesta por neuronas, estas reciben las señales, posteriormente emiten señales y así se establece una red neuronal biológica. Similar a esta red, el Deep Learning usa una red neuronal artificial que recibe señales, interpreta mediante las capas que poseen diferentes características propias de cada elemento a interpretar por la red.

En la Figura 20, se muestra un diagrama en el cual se representa la relación entre las señales recibidas, variables ***x*** y la señal de salida, variable ***y.*** Similar a la neurona biológica, cada señal es ponderada, variable ***w,*** que representa la importancia de la señal***;*** las señales de entrada se suman en el cuerpo de la célula y es procesada por la función de activación, que viene definida por ***f.***

Figura 20 Representación red neuronal artificial



Nota. Adaptado de (Lantz, 2013)

Una red neuronal está compuesta por los siguientes elementos:

- Capas: Estructura fundamental de una red neuronal. Es un módulo de procesamiento de datos que toma las entradas de uno o más tensores y las salidas de uno o más tensores (Chollet, 2018).

- Modelos: red de capas, la más común es una red que tiene una entrada y una salida, sin embargo, existen más modelos que definen una serie de operaciones de los tensores, dependiendo el modelo. El objetivo de seleccionar el modelo correcto es encontrar el valor de las ponderaciones, ***w,*** para optimizar el aprendizaje.

- Función de perdida: durante el aprendizaje la función de perdida representa la medida de éxito del aprendizaje, comparando con la estructura de datos esperada. Esta cantidad debe ser mínima y la función de perdida ajusta los parámetros durante el entrenamiento.

- Optimizadores: Determina como se debe actualizar la red neuronal, basándose en la función de perdida.

RCNN.

Conocidas por sus siglas en inglés, CNN, (Convolutional Neuronal Networks), son un tipo de red neuronal artificial con aprendizaje supervisado, con la diferencia que la arquitectura de una red convolucional asume que los datos de entrada que recibe son imágenes, lo cual permite codificar ciertas propiedades en la arquitectura y de esta manera hacerla más eficiente reduciendo la cantidad de parámetros de la red (Stanford Vision and Learning Lab, 2021).

Una red convolucional está compuesta por capas, de las que se extraen características, que alternan capas de reducción y capas de percepción multicapa, que son capas de conexión total. (López, 2021)

Estructura de una Red Neuronal Convolucional

Se utilizan tres principales tipos de capas para construir arquitecturas de convolución, la capa convolucional, capa de agrupación y la capa totalmente conectada.

- **Capa Convolucional:** esta capa calculará las salidas de las neuronas que están conectadas a regiones locales de entrada. Cada capa calcula el producto entre los pesos o ponderaciones y una región a la que están conectadas en el volumen de entrada. Como ejemplo, se tiene una imagen de entrada de 40x40 pixeles, y es una imagen RGB. Se decide aplicar 10 filtros. El volumen de salida resultante sería [40x40x10].

- **Capa de agrupación:** la capa reduce la resolución de la imagen en sus dimensiones espaciales, ancho y alto de la imagen. Dando como resultado en el volumen [20x20x10] únicamente el alto y ancho se modifican.

- **Capa totalmente conectada:** calcula las puntuaciones por cada clase ponderando los resultados y presenta un resultado final, cada nodo de esta capa estará conectado a los números del volumen previo.

Fast RCNN

Es un método de red convolucional basada en regiones rápidas (Fast R-CNN) para la detección de objetos. Fast R-CNN se basa en trabajos anteriores para clasificar de manera eficiente las propuestas de objetos utilizando redes convolucionales profundas. En comparación con trabajos anteriores, Fast R-CNN emplea varias innovaciones para mejorar la velocidad de entrenamiento y prueba al tiempo que aumenta la precisión de detección. Fast R-CNN entrena la red VGG16 muy profunda 9 veces más rápido que R-CNN, es 213 veces más rápido en el

momento de la prueba y logra un mAP más alto en PASCAL VOC 2012. En comparación con SPPnet, Fast R-CNN entrena VGG16 3 veces más rápido, prueba 10 veces más rápido y es más preciso. Fast R-CNN se implementa en Python y C++ (Girshick, 2015).

Mask RCNN

Presenta un marco conceptualmente simple, flexible y general para la segmentación de instancias de objetos. Nuestro enfoque detecta de manera eficiente los objetos en una imagen y, al mismo tiempo, genera una máscara de segmentación de alta calidad para cada instancia. El método, llamado Mask R-CNN, amplía Faster R-CNN al agregar una rama para predecir una máscara de objeto en paralelo con la rama existente para el reconocimiento del cuadro delimitador. Mask R-CNN es fácil de entrenar y agrega solo una pequeña sobrecarga a Faster R-CNN, que se ejecuta a 5 fps. Además, Mask R-CNN es fácil de generalizar a otras tareas, por ejemplo, permitiéndonos estimar poses humanas en el mismo marco. Se muestra los mejores resultados en las tres pistas del conjunto de desafíos de COCO, incluida la segmentación de instancias, la detección de objetos de cuadro delimitador y la detección de puntos clave de personas. Sin campanas ni silbatos, Mask R-CNN supera a todos los existentes, entradas de un solo modelo en cada tarea, incluidos los ganadores del desafío COCO 2016. Se espera que nuestro enfoque simple y efectivo sirva como base sólida y ayude a facilitar futuras investigaciones en el reconocimiento a nivel de instancia (Kaiming et al., 2016).

1.5.3. Machine learning

El aprendizaje es una de las actividades más importantes de los seres vivos, permite la adaptación al ambiente y tener una mejor calidad de vida. El aprendizaje requiere transformaciones de ideas y estructuras de información que son procesadas en la mente humana

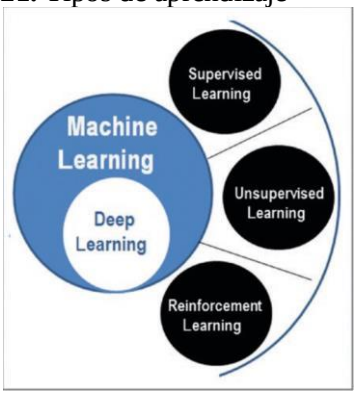
para la toma de decisiones de acuerdo con el aprendizaje previo (Chowdhary, 2020). El término Machine Learning fue inicialmente acuñado en 1959 por Arthur Samuel (López, 2021). Es una manera de construir inteligencia dentro de una máquina, así de esta manera sea capaz de aprender a través del tiempo y realizar sus actividades de mejor manera basándose en sus experiencias previas. (Gollapudi, 2019) El propósito principal de Machine Learning es introducir algoritmos que reciban datos de entrada, aplicar un análisis computacional con el fin de predecir los valores de salida con un rango aceptable de precisión, identificar patrones y tendencias dentro de los datos recibidos, y finalmente aprender de experiencias previas (Endelman y otros, 2018).

Tipos de aprendizaje automático

Los algoritmos son los motores que ejecutan el aprendizaje automático. En la actualidad, se utilizan comúnmente dos tipos principales de algoritmos de aprendizaje automático:

Aprendizaje supervisado y aprendizaje no supervisado no obstante se añade otro tipo de aprendizaje reforzado (OCI, 2023).

Figura 21. Tipos de aprendizaje



Nota. Adaptado de: (Chowdhary, 2020)

Aprendizaje supervisado

Requiere la intervención humana, y esta es la principal diferencia con los otros tipos de aprendizaje. Trabaja en base a una expectativa conocida por lo que requiere una predicción previamente realizada de forma que pronostique a que clase pertenecen los ejemplos dados y los clasifica en esas categorías (Chowdhary, 2020). Un aprendizaje supervisado utiliza machine Learning para inducir un clasificador de un dataset previamente etiquetado. Un clasificador típicamente aprende con la ayuda de un set de entrenamiento que contiene ejemplos de las predicciones que deberá realizar, identificando el elemento con su respectiva etiqueta.

Aprendizaje no supervisado

No requiere la intervención humana, en este método el sistema debe descubrir la forma de clasificar los datos. El sistema se encarga de analizar características, patrones, relaciones o comportamientos entre los datos para agruparlos y clasificarlos. Un ejemplo para comprender de mejor manera el aprendizaje no supervisado es el trabajo que realizan los científicos, ellos no tienen un profesor que les enseñe lo que necesitan aprender, formulan hipótesis para explicar lo que observan durante sus investigaciones, y prueban sus hipótesis con experimentos diseñados por ellos mismos, sin la ayuda de un agente externo (Chowdhary, 2020).

Aprendizaje por refuerzo

Los algoritmos aprenden de la experiencia. Cada vez que el algoritmo acierta en sus predicciones, se le otorga un “refuerzo positivo”, de manera similar a la que se premia a nuestra mascota por su buen comportamiento o por aprender a sentarse cuando se le ordena sentarse (Chowdhary, 2020).

Definiciones

Etiquetas. Es el valor referido que se trata de predecir, este permite a los analistas incomunicar variables en los grupos de datos.

Atributos. Se la denomina una variable de entrada en la regresión lineal simple. Este conlleva a utilizar distintos tipos de aprendizajes, en lo cual se puede denominar que entre mas atributos tenga el aprendizaje mas sofisticado llega a hacer su aprendizaje.

Ejemplos. Es considerada una instancia en particular que se da el dato de exclusión de variables, estas se divides en un ejemplo etiquetado que es el que incluye tantos atributos como la etiqueta, y los ejemplos sin etiquetas las cuales contiene atributos, pero sin etiquetas.

Modelos. Este se define la relación entre los atributos y etiquetas, la cual tiene dos fases que son: Entrenamiento lo cual crea o aprende el modelo y la inferencia que aplica el modelo entrenado a ejemplos sin etiqueta.

Regresión frente a clasificación. La regresión predice valores continuos y la clasificación predice valores discretos. (Health Big Data, 2022).

1.5.4. Deep Learning

El aprendizaje profundo o Deep Learning se basa en la ciencia del cerebro, en como se conectan las neuronas, identifican patrones y clasificar datos. Es un subconjunto del Machine Learning que se caracteriza por la utilización de redes neuronales profundas, (López, 2021) en

muchas capas, que se usan para resolver problemas de alta complejidad que involucran grandes cantidades de datos.

Tensores

Todos los sistemas de Machine Learning, usan tensores como su unidad básica de estructura de datos. Un tensor es un contenedor de datos, generalmente numéricos. (Chollet, 2018). Para el manejo de estructura de datos se usan elementos como vectores, matrices 2D y 3D, que son tensores. Un tensor viene definido por sus tres atributos:

- Dimensión del tensor: al referirnos a un tensor se llaman *ejes.* Un tensor 2D que representa una matriz tiene dos ejes, en el caso de la matriz se refiere a las filas y columnas

- Forma: es una tupla que describe cuantas dimensiones tiene el tensor a lo largo de cada eje.

- Tipo de datos: como su nombre lo indica representa el tipo de datos con el que trabaja, generalmente datos numéricos. En las librerías que se importan para trabajar con machine Learning no existen tensores de cadenas de caracteres ya que una cadena de caracteres tiene una longitud variable, lo que impide el uso dentro de un tensor.

Pytorch

Es una librería para Python que facilita la construcción de arquitecturas de aprendizaje profundo. Provee un núcleo de estructura de datos, el tensor, que optimiza las operaciones matemáticas. Usado en contextos profesionales, se considera como la librería fundamental

dentro del Deep Learning que se debería conocer y aprender. Similar a la librería NumPy, que trabajo con arreglos y matrices, Pytorch trabaja de manera más optima, con una aceleración de GPU y calculo automático de gradientes, lo que lo hace adecuado para calcular el mínimo de una función, método de gradiente descendente, y calcula los datos del paso de una regresión a partir de una expresión futura (Stevens & Antiga, 2019)

Keras

Keras tiene una arquitectura simple. Es más legible y conciso. Tensorflow, por otro lado, no es muy fácil de usar a pesar de que proporciona Keras como un marco que facilita el trabajo. PyTorch tiene una arquitectura compleja y la legibilidad es menor en comparación con Keras.

1.5.5. Modelos de detección de objetos

Mobilenet SSDV2

Un modelo de PyTorch muy rápido y fácil de usar que logra resultados de última generación (o casi de última generación).

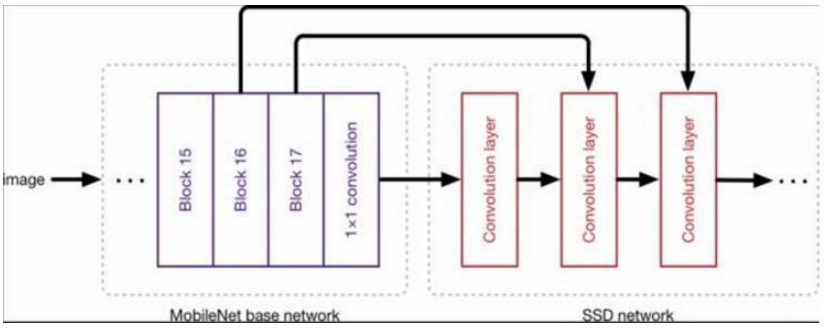
La tendencia general observada es que los modelos de visión por computadora son cada vez más profundos y complejos para lograr una mayor precisión, este es un modelo diseñado específicamente para aplicaciones móviles e integradas que requieren alta velocidad.

Su primera versión (MobileNetV1) tenía una convolución separable en profundidad, lo que redujo el tamaño del modelo y el costo de complejidad de la red a un nivel decente, para que se pudiera utilizar en aplicaciones de bajo procesamiento, MobileNet es un modelo que ofrece una velocidad decente y su único inconveniente es su precisión, además SSD demostró

ser muy útil para el modelo, ya que obtuvo los medios para mejorar su precisión mientras mantenía la velocidad de los modelos (Yu-Chen et al., 2020).

El algoritmo SSD se diseñó de tal manera que pudiera integrarse con varias redes como la arquitectura YOLO, Mobilenet y VGG, esta arquitectura se la puede apreciar en la Figura 22. Diagrama de bloques de MobileNet-SSD Arquitectura

Figura 22. Diagrama de bloques de MobileNet-SSD Arquitectura



Fuente: (Varadharajan et al., 2021)

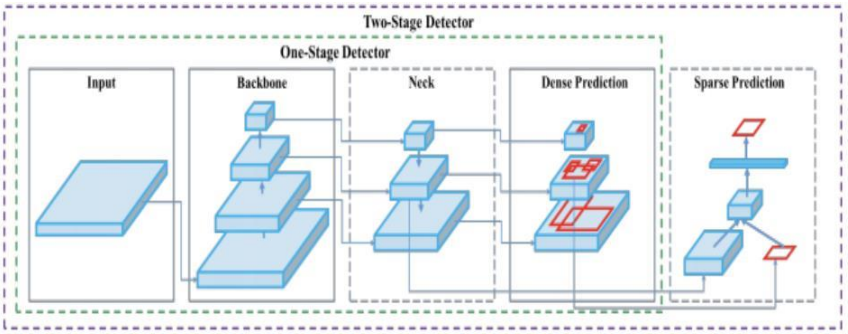
YOLO V5

Esta arquitectura proporciona buenos resultados en tiempo real en computación limitada. Este diseñado para ejecutarse en tiempo real (30 cuadros por segundo) incluso en dispositivos móviles, YOLO ha estado dominando su campo durante mucho tiempo y hubo un gran avance en mayo de 2020.

Al no ser el autor convencional de la serie YOLO, esta nueva versión fue recibida con cierta controversia, pero saltándola, el modelo v5 ha mostrado un aumento sustancial en el rendimiento de sus predecesores; junto con el desarrollo de YOLO en 2016, muchos algoritmos de detección de objetos con diferentes enfoques también han logrado logros notables. (Jocher et al.)

Estos avances han formulado dos conceptos de detección de objetos arquitectónicos: detector de una etapa y detector de dos etapas.

Figura 23: Dos conceptos de detección de objetos arquitectónicos



Fuente: (Varadharajan et al., 2021)

Mobilenet SSDV2 vs YOLO V5

En el trabajo de investigación de Varadharajan et al., 2021 se llevo a cabo un experimento donde los tres modelos que son Mobilenet SSDV2, Yolo V3 y Yolo V5 después de la finalización del proceso de formación se realizó una comparación. Los modelos se implementaron utilizando tres dispositivos, a saber, Jetson Nano, Nvidia GTX 1660 Ti y Nvidia Tesla T4. Esto se ha hecho para determinar su desempeño en una unidad de procesamiento de nivel alto, medio y bajo. Ahora, pasando a los factores que deben tenerse en cuenta para distinguir los tres modelos, hay dos parámetros tan influyentes que determinarían el modelo que sería adecuado para casos de uso específicos. Estos son la precisión promedio (mAP) y la capacidad de procesamiento de la modelo medida a través de cuadros por segundo (fps) del video procesado resultante.

A menudo, al validar el rendimiento de un modelo, la precisión se prioriza sobre la velocidad y se supone que las GPU de alto calibre están disponibles en abundancia y, por lo tanto, la velocidad factor será tolerable. Mientras que para la implementación en tiempo real, la

velocidad es un factor igualmente crucial y tales modelos son comúnmente utilizado en aparatos que poseen relativamente bajo Capacidades de procesamiento para aplicaciones rutinarias como el uno presente Para que una detección se considere en tiempo real, la el valor generalmente aceptable de fps es 15 (Varadharajan et al., 2021).

Tabla 1. Análisis de rendimiento de los tres modelos.

MODELO	mAP	FPS		
	(%)	Tesla T4	1660Ti	Jetson Nano
YOLO V3	54.3	80	21	8
YOLO v5s	37.3	100	28	15
MobileNet-SSD V2	33.7	94	26	15

El rendimiento de cada uno de los modelos entrenados se proporciona en la Tabla 1. Teniendo en cuenta los puntos mencionados anteriormente, se puede inferir que YOLOv5s es el modelo que mejor se adapta a situaciones en tiempo real con valores óptimos tanto de precisión como de fps.

1.5.6. Evaluación del rendimiento de un modelo de aprendizaje automático

Par poder validar un modelo de detección de objetos que opera con RCNN, se opta por el uso de la matriz de confusión y la curva de ROC que nos darán porcentaje de atributos que ayudaran a un análisis para determinar de déficit del algoritmo.

Matriz de confusión y parámetros de un algoritmo de predicción y detección

Para la validación de un algoritmo de visión artificial en el área de detección de objetos es necesario, calcular los distintos parámetros de predicción y detección, como:

- **Exactitud (ACC). -** representa el porcentaje de predicciones correctas frente al total.

- **Precisión (PPV). -** se refiere a lo cerca que está el resultado de una predicción del valor verdadero.

- **Sensibilidad (TPR). -** Es la proporción entre los casos positivos bien clasificados por el modelo, respecto al total de positivos.

- **Especificidad (SPC). -** Es la proporción entre los casos negativos bien clasificados por el modelo, respecto al total de negativos.

- **Valor predictivo negativo (NPV). -** Probabilidad de un resultado negativo en una prueba verdaderamente no esté afectada.

- **Tasa de falsos positivos (FPR). -** Número de casos que la prueba declara positivos y que en realidad son negativos.

- **Tasa de descubrimiento falso (FDR).–** Es la proporción de pruebas que bajo la hipótesis nula siendo verdaderas dentro del conjunto de pruebas rechazadas.

- **Tasa de falsos negativos (FNR). -** Proporción de casos positivos que la prueba detecta como negativo.

- **Puntuación F1. -** Este parámetro hace más fácil el poder comparar el rendimiento combinado de la precisión y la exhaustividad entre varias soluciones.

Para el cálculo de estos parámetros y a la vez dar una validación y análisis del modelo es necesario utilizar la matriz de confusión (Tabla 2) que es una herramienta que permite

visualizar el desempeño de un algoritmo que se emplea en aprendizaje supervisado (Nighania, 2018).

Tabla 2: Matriz de confusión

Matriz de confusión		Estimado por el modelo	
		Negativo (N)	Positivo (P)
Real	Negativo	TN	FP
	Positivo	FN	TP

No obstante, se tomó en cuenta el número de objetos que el dispositivo debe detectar, lo cual llevo en hacer una matriz individual por objeto, con el fin de ver el rendimiento individual.

La Matriz de confusión está formada por los siguientes elementos que son:

- Verdaderos positivos (TP). - Predichos positivos y realmente positivos.
- Falsos Positivos (FP). - Predichos positivos y en realidad son negativos.
- Negativos verdaderos (NT). - Negativos previstos y realmente negativos.
- Falsos negativos (FN). - Predicho negativo y en realidad es positivo.

Estos elementos ayudaran a obtener los porcentajes de los parámetros como se observa en la Tabla 3.

Tabla 3. *Fórmulas para el cálculo de parámetros obtenidos*

Medida	Ecuación
Sensibilidad	$\square P\square = \square e\square\square i\square i\square i\square\square = \frac{\square P}{\square P + FN}$
Especificidad	$\text{Especificidad} = \frac{\square N}{\square N + FP}$
Precisión	$\text{PPV} = P\square eci\square ió\square = \frac{\square P}{\square P + FP}$
Valor predictivo negativo	$\text{NPV} = \frac{\square N}{\square N + FN} * N$
Tasa de Falsos Positivos	$\text{FPR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}$
Tasa de descubrimiento falso	$F\square\square = \frac{FP}{(FP + \square P)}$
Tasa de falsos negativos	$FN\square = \frac{FN}{(FN + \square P)}$
Exactitud	$A\square\square = \frac{\square P + \square N}{\square P + FP + \square N + FN}$
Puntuación F1	$F1 = \frac{2}{\frac{1}{p\square eci\square ió\square} + \frac{1}{\square ec\square pe\square ació\square}}$

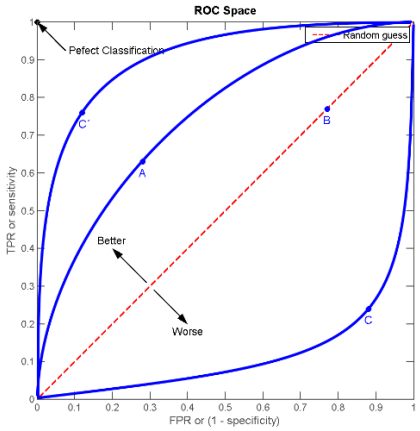
Curva ROC (Características operativas del receptor)

Todos los parámetros anteriormente mencionados, se relacionan para poder graficar la curva de ROC que nos permitirá ver cuánto se varía el umbral de discriminación. Este esencialmente se utiliza para evaluar el rendimiento de los algoritmos de clasificación binaria, es decir, en dos estados como, está el objeto o no lo está.

Los parámetros de esta gráfica se obtienen calculando la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR), con lo cual se obtiene un solo clasificador en una variedad de umbrales (Melillanca, 2018).

Para interpretar la curva de ROC es necesario interpretar algunos ejemplos, los cuales se puede ver en la Figura 24. Curva de ROC con 5 ejemplos

Figura 24. Curva de ROC con 5 ejemplos



Fuente: (Melillanca, 2018).

En la imagen, aparecen cuatro puntos que corresponden a distintos modelos. Para comprender el espacio en la Curva ROC se analizará cada uno de ellos.

- A, tiene TVP = 0,63, TFP = 0,28 y Precisión = 0,68.
- B, tiene TVP = 0,77, TFP = 0,77 y Precisión = 0,50.
- C, tiene TVP = 0,24, TFP = 0,88 y Precisión = 0,18.
- C', tiene TVP = 0,76, TFP = 0,12 y Precisión = 0,82.

La curva a lo largo del gráfico corresponde a la línea de estimación aleatoria (diagonal), y sin modelo ni clasificación aleatoria, se tiene un 50 % de conjetura correcta (clasificación binaria). La posición (1.0) en el diagrama muestra una clasificación perfecta porque tiene la máxima sensibilidad, es decir, 100 % de tasa de verdaderos positivos y 0 % de tasa de falsos positivos (también es 100 % de especificidad porque TVP = 1 - especificidad). Esto significa que en ese punto se tiene un modelo perfecto, y cuanto más se acerque nuestro modelo a ese punto, mejor funcionará. El resultado de B cae dentro de una conjetura aleatoria; es 50%

preciso. El modelo C es el peor de los tres con puntuaciones muy bajas. El modelo A funciona mejor que los modelos aleatorios, pero no es un modelo admirable. El modelo dado C\' es el modelo de mejor rendimiento, se debe observar su métrica y su posición en el gráfico (Melillanca, 2018).

CAPÍTULO II

Marco Metodológico

En el siguiente capítulo se describe el proceso metodológico con el cual se diseñó el algoritmo de visión artificial para el reconocimiento de símbolos que representan los productos de la canasta básica, además de la construcción e implementación de un prototipo portátil que pueda ser usado por personas con discapacidad visual para el reconocimiento de productos de consumo masivo de un micro mercado.

2.1. Tipo de investigación

En este trabajo de investigación se utiliza dos tipos de investigación que son la de campo y la aplicada.

Referente a la investigación de campo es recurrente involucrarse en el terreno, ya que amerita la recopilación de datos de fuentes primarias que dan al propósito especifico, con ello viendo la realidad del entorno de las personas no videntes y poder crear estrategias viables que lleguen a la solución del problema; Así mismo en la investigación aplicativa o empírica, busca la aplicación o utilización de estos conocimientos para la solución del problema en concreto, viendo la fiabilidad de un prototipo que pueda ser probado y optimizado

2.2. Nivel de investigación

El nivel de este proyecto es aplicativo ya que planea resolver el problema, resaltando la parte de innovación técnica y tecnológica, para mejorar la calidad de vida de las personas no videntes.

2.3. Metodología

La metodología que se empleó en este proyecto es el método cascado en la que se basa en seguir las secuencias de fases y nunca avanza hasta que haya completado la fase anterior, aplicando a la vez un enfoque ingenieril donde se tomo los datos de requerimientos y los aplica para crear o diseñar la solución de ingeniería a este problema

2.3.1. Fase 1: Adquisición de requerimientos

Actividad 1.- Estudio bibliográfico:

Se realiza el respectivo estudio bibliográfico estipulado en el capítulo anterior, en este se estudió los conceptos propicios a la inteligencia artificial, visión artificial, machine learning y Deep learning.

Actividad 2.- Entrevistas con especialistas y usuarios

En esta investigación se procede a utilizar la entrevista como técnica de recolección de información, la cual es una interacción entre una o varias personas las cuales responden a debidas preguntas establecidas.

Entrevista

La entrevista se realizó en la ciudad de Ibarra, en la provincia de Imbabura, en la Asociación de No videntes de Imbabura (ANVI) ubicada en la calle Juan Hernández 1-56 y Raúl Montalvo, esta entrevista documenta un guion junto con un documento contextual de toda la entrevista ubicada en el Anexo 2.

Actualmente ANVI cuenta con más de 140 miembros anexados de todos los alrededores de la provincia de Imbabura y de todos los rangos etarios a los cuales se les brinda atención, capacitación y formación para su inclusión en la sociedad.

ANVI está representada por la Srta presidenta Noemi Trejo la cual dio acogida a este proyecto aportando con la experiencia de los diferentes miembros, en esta entrevista referida en el ANEXO 2, dio a conocer la realidad en la que viven las personas no videntes.

Específicamente con respecto a la habitualidad de la movilidad y orientación , dijo que las personas no videntes determinan la orientación y movilidad, dependiendo de la edad en la que se adquiriere la discapacidad, ya que si es de nacimiento, desde niños se va adquiriendo el desarrollo prematuro de los demás sentidos, obteniendo en si el manejo del espacio en el que se encuentra, al contrario cuando las personas de edad avanzada adquieren esta discapacidad, es un poco más difícil adaptarse, ya que los sentidos están acostumbrados a manejarse con la visión, y eso que independientemente depende del porcentaje de la discapacidad visual.

Determino también que tienen diversas herramientas para su movilidad, como el bastón, el cual ocupa tres colores diferentes de acuerdo a la gravedad de su discapacidad, y que la orientación se da de acuerdo a la experiencia que vayan adquiriendo, de acuerdo a las veces que transcurran en los diversos sitios, en los cuales agudizan los sentidos, teniendo en contexto sonidos o olores principales de cada lugar, para determinar en donde están.

El problema habitual al que se enfrentan es prácticamente en los diversos supermercados, en los cuales el problema no es la movilidad ni la orientación, si no mas bien en la identificación de algunos objetos, que ellos desean comprar, ya que ellos buscan su autonomía y un lugar dentro de la sociedad; luego argumento que si existen diversos equipos

de los cuales han experimentado, pero por lo general se experimenta mucho estrés en los diversos avisos audibles, ya que tornan en ser repetitivos y redundantes, en los cuales a veces dan la respuesta equivocada del objeto, ya que los productos contienen diversas texturas y contornos similares.

Actividad 3.- Adquisición de requerimientos

Con la adquisición de la información recolectada bibliográficamente como en la entrevista, se recopilo diferentes tipos de requerimientos como:

- Requerimientos de Usuario (RU)
- Requerimientos Operacionales (RO)
- Requerimientos del Sistema (RS)
- Requerimientos de arquitectura (RA)

Requerimientos de Usuario

Los requisitos de usuario (RU o URS en inglés) son lo que la organización espera del sistema para satisfacer sus necesidades de quien la opere en gestión.

En la Tabla 4, se tiene los respectivos requerimientos de la información recopilada.

Tabla 4: Requerimientos de usuario

RU					
	REQUERIMIENTOS DE USUARIO				
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
RU 1	Fácil de llevar en un carro de compras.		X		

RU 2	Reconocimiento de los percheros para identificar 5 grupos de productos.	X
RU 3	Fácil de Usar.	X
RU 4	Respuesta Audible entendible	X
RU 5	Que no genere Estrés	X

Requerimientos Operacionales

Esto se refiere a los requisitos de aplicación que los usuarios tienen para el negocio, donde las condiciones organizativas y técnicas influyen en dónde funcionará el sistema observado en la Tabla 5.

Tabla 5. Requerimientos Operacionales

RO					
	REQUERIMIENTOS DE OPERACIONALES				
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
RO 1	Debe ser portátil	X			
RO 2	El algoritmo debe de operar a menos de 30 FPS.	X			
RO 3	Debe de operar en Software libre	X			
RO 4	Modelo con menos 2.00 de perdida	X			
RO 5	Precisión y exactitud superior al 70%.	X			

Requerimientos del Sistema

Los requisitos/requisitos del sistema (RS) describen los servicios que el sistema debe proporcionar y las restricciones asociadas con su funcionamiento como se muestra en la Tabla 6.

Tabla 6 . Requerimientos del sistema

RS			
	REQUERIMIENTOS DEL SISTEMA		
NÚMERO	REQUERIMIENTOS	PRIORIDAD	RELACIÓN

		ALTA	MEDIA	BAJA
RS 1	Alto rendimiento	X		
RS 2	Funcionalidad optima.	X		
RS 3	Utilizar tarjeta embebida especializada en visión Artificial	X		
RS 4	Utiliza CUDA	X		
RS 5	Tarta embebida mayor de 2GB RAM	X		
RS 6	Flexibilidad en procesos		X	
RS 7	Los datos deben de ser obtenidos rápidamente	X		
RS 8	El proceso debe rendir a largo plazo y no saturarse.		X	
RS 9	Se toma datos de manera automática		X	
RS 10	Tendrá un estado de internación, mientras tenga los tiempos de espera		X	

Requerimientos de Arquitectura

Los requisitos de arquitectura contienen un conjunto de declaraciones que describen lo que se debe considerar al diseñar o implementar una arquitectura al implementarla como se ilustra en la Tabla 7.

Tabla 7. Requerimientos de Arquitectura

.

REQUERIMIENTOS DEL ARQUITECTURA					
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
RA 1	Gestión de proyectos por hilos.		X		
RA 2	Procesamiento previo de los datos	X			
RA 3	Entrada como matriz unidimensional con tensores.	X			
RA 4	Entrada con diagrama de Graph	X			
RA 5	Debe tener un sistema operativo en base a Linux		X		

RA 6	Base de tecnología nvidea		X
RA 7	Tarjeta embebida que contenga GPU para Graficas	X	
RA 8	Librerías especializadas para detección de objetos	X	
RA 9	Canalizaciones de datos altamente escalables para cargar datos		X
RA 10	Herramientas para validar y transformar conjuntos de datos grandes		X

2.3.2. Fase 2: Diseño del Sistema

Actividad 4.- Selección de librería de inteligencia artificial para detección de objetos.

En esta actividad se designará un cuadro comparativo que nos llegará a la selección del tipo de algoritmo para nuestro modelo de detección de Objetos, este cuadro de la Tabla 8 comparativa lleva consigo los distintos requerimientos para la aprobación del algoritmo.

Tabla 8. Requerimientos del Sistema

Librería Machine Learning	**REQUERIMIENTOS**					**VALORACIÓN TOTAL**
	RS1	RS2	RS3	RA3	RA8	
Keras	0	0	1	1	1	3
PyTorch	0	1	0	1	1	3
TensorFlow	1	1	0	1	1	4
	Cumple = 1					
	No Cumple = 0					

En conclusión, se opta por la selección de la librería tensor Flow ya que esta librería es capaz de manejar gran conjunto de datos, y por ello maneja un alto rendimiento lo cual ayuda a obtener una funcionalidad optima, y esta se especializa en la detección de objetos por lo que Tensor Flow es un marco de código abierto que facilita la construcción y el entrenamiento y

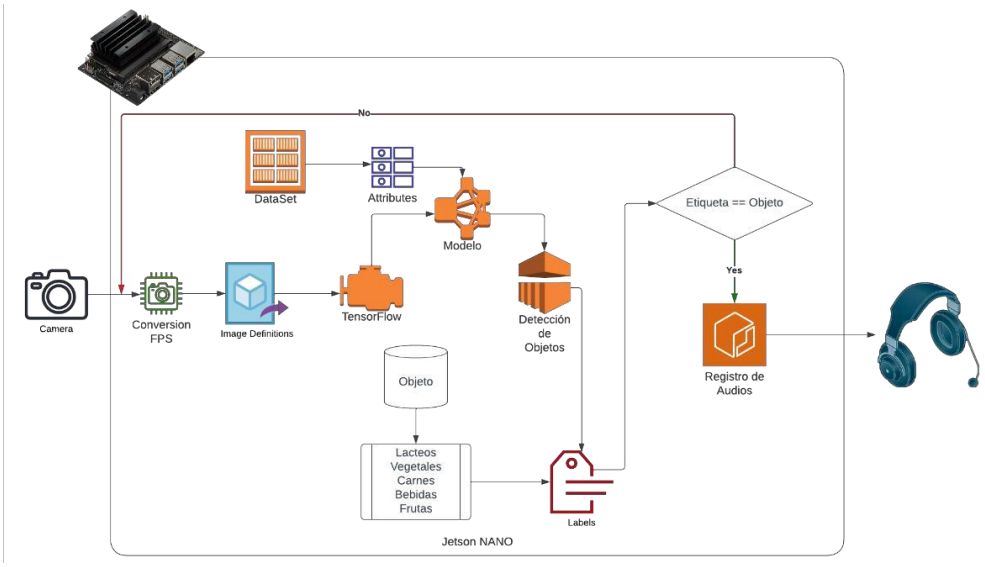
una implementación de modelos de detección de objetos teniendo consigo una API especializada en ello.

Teniendo en claro el tipo de librería a utilizar, se debe utilizar un modelo de detección de objetos llamado “Mobilenet SDDV2”, por lo que este tiene una arquitectura que proporciona buenos resultados en tiempo real en computación limitada. Está diseñado para ejecutarse en tiempo real (30 cuadros por segundo) incluso en dispositivos móviles para un marco de TensorFlow.

Actividad 5.- Diagramas de bloques

En esta actividad se destaca el diagrama de bloques que no es más que una representación del funcionamiento interno del sistema, definiendo su organización; en la Figura 25, destaca el funcionamiento desde la obtención de la imagen de la cámara, proseguida por su digitalización en la tarjeta embebida Jetson, en la cual se define los atributos con procesamiento de imágenes, luego se entrega al motor del algoritmo de tensorflow, el cual ayudara a identificar los atributos semejantes obtenidos por el modelo desarrollado, a continuación de ello se realizar una comparación para definir si dentro de la imagen se encuentra los objeto, designados por etiquetas, si estas coinciden entre si se denomina la existencia del objeto en el frame, enviando una aprobación al registro de audios para que pueda enviarse a la salida de audio de la tarjeta embebida.

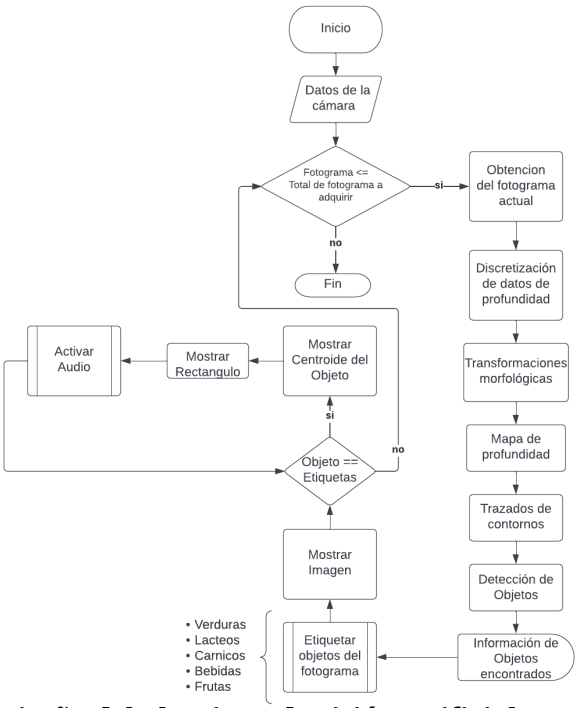
Figura 25. Diagrama de Bloques



Actividad 6.-flujograma

En esta actividad se tomó en cuenta los procesos y subprocesos que se deben ejecutar, para obtener el funcionamiento pleno del dispositivo, en la Figura 26, se puede observar el proceso general, desde la bineralización de los datos tomados en cámara, por lo cual determina si es un video o no, quiere decir que si no es menor o igual a los fotogramas determinados no puede seguir el proceso, esto suele generarse cuando no hay una cámara conectada y simplemente sale del proceso general, caso contrario se obtiene el fotograma, se discretiza la profundidad de la imagen, realizando diferentes transformaciones morfológicas, obteniendo el trazado de los contornos, con los cuales se puede distinguir el objeto, obteniendo la información generada de la red neuronal, obteniendo los atributos y diferenciarlos para obtener el objeto, continuando el proceso grafica la detección y diferencia que objeto de las etiquetas es, para proceder con el audio de la autoidentificación.

Figura 26. Flujo grama del Proceso General



2.3.3. Fase 3: Diseño del algoritmo de visión artificial

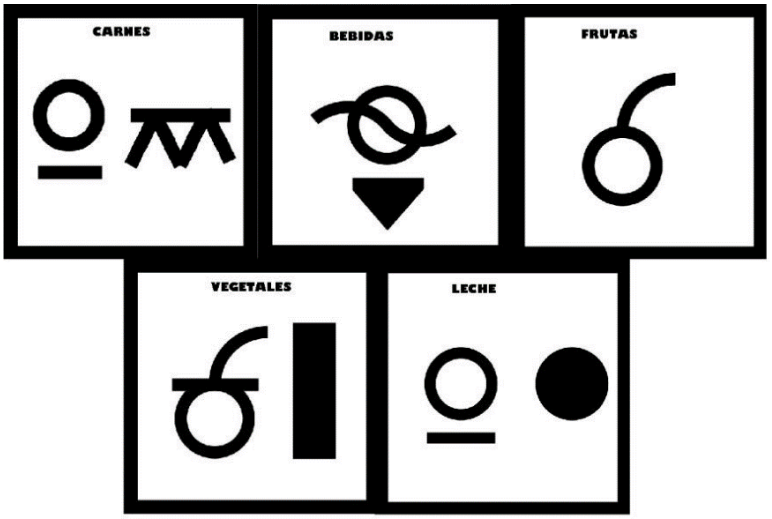
Actividad 7.- Preparación de la data set

Dentro del trabajo de campo se tomó a consideración que el reconocer diversos productos con un algoritmo de visión artificial, sería una tarea muy complicada, ya que diversos productos son semejantes por sus diversas morfologías, texturas, colores entre otros atributos que puedan señalarse de acuerdo con el empaque como se puede ver en la **Figura 30** ***Primer entrenamiento***

Figura 27. Fotografía de productos del supermercado

Por ende se opta utilizar una estrategia de identificación, como son los símbolos por Bliss, creados por Karl Blitz con el objeto de desarrollar un sistema de comunicación que sirviese como complemento o sustituto del habla para niños pre-lectores (McDonald, 1985). Los símbolos Bliss pretenden ser un lenguaje universal pictográfico que pueda ser leído en todas las lenguas. Se seleccionaron 5 símbolos (Figura 28) que representan algunos de los alimentos más importantes, bebidas, carnes, frutas, leche, vegetales.

Figura 28 *Símbolos Bliss para el reconocimiento de los percheros*



Crear Dataset

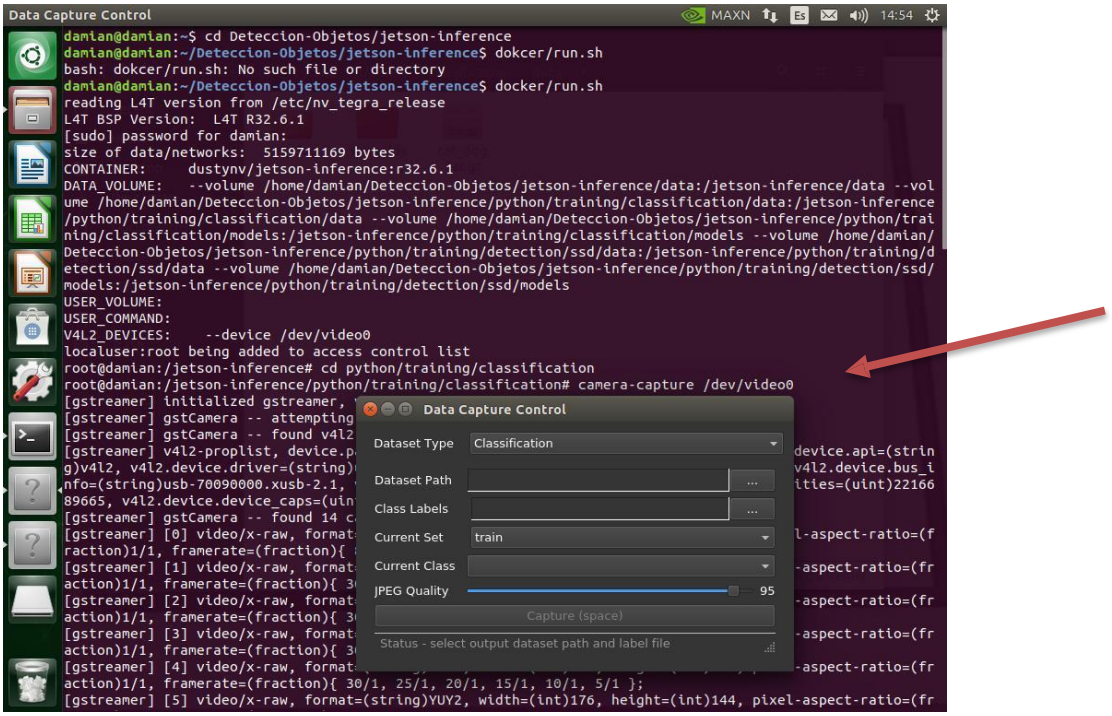
Para la creación del dataset se procede a usar la herramienta camera-capture, propia de NVIDIA, como se muestra en la Figura 29.

En terminal, dentro del contenedor, en el directorio **python/training/classification**, se ejecuta el comando **camera-capture /dev/video0,** se usa /dev/video0 en caso de usar una cámara web; en caso de usar una cámara propia de Raspberry se usará csi://0. Se crea una

carpeta dentro del directorio donde se almacenarán las imágenes de entrenamiento, prueba y valoración.

Dentro de esta carpeta se crea un archivo .txt donde consten los nombres de las clases a clasificar. En la opción Dataset Path se selecciona la carpeta creada y en la opción Class Labels se selecciona el archivo .txt creado previamente. Esto desarrolla los subdirectorios propios del entrenamiento que se realizará (Figura 29 *Interfaz camera-capture*).

Figura 29 *Interfaz camera-capture*



Se comienza la creación del dataset, tomando imágenes de cada uno de los símbolos que serán reconocidos, en la carpeta train y seleccionando la etiqueta correspondiente, se toma la mayor cantidad de imágenes posible, y se debe designar un porcentaje claro en 3 carpetas designadas llamadas:

- Train: Esta carpeta es la muestra de datos utilizada para ajustar el modelo.

- Validation: esta es una muestra de datos utilizada para evaluar objetivamente el ajuste del modelo al conjunto de datos de entrenamiento al ajustar los hiperparámetros del modelo. Las estimaciones se vuelven más sesgadas cuando las habilidades en el conjunto de datos de validación se incluyen en el ajuste del modelo.

- Test: la muestra de datos utilizada para proporcionar una evaluación imparcial de un ajuste de modelo final en el conjunto de datos de entrenamiento.

En este trabajo se distribuye 2500 imágenes para el data set de la siguiente manera, el 60% al train, el 20% al Validation y el 20% al Test. Hay que tomar en cuenta que las imágenes no deben de ser repetidas, caso contrario aumenta la probabilidad de tener diferentes errores como un overbooking que usualmente es cuando se tiene una perdida optimizada, pero tiene un pésimo desempeño al momento de la validación.

Las proporciones destinadas se hace de acuerdo a las que se necesita para la mayor parte de muestras guiadas al entrenamiento, ya que a mayor número de muestras mayor es el índice de aprendizaje, y menor el índice de perdida.

Una vez designadas las categorías se procede a obtener los archivos globales de cada carpeta en formato .CSV(Valores separadas por comas), en el que se encuentra básicamente un comprimido de las coordenadas de las imágenes preseleccionadas

Actividad 9.- Entrenamiento

Una vez creado el dataset, se procede a entrenar el modelo ejecutando el algoritmo train.py que se encuentra en el directorio /python/training/classification dentro del contenedor, que es un modelo previamente entrenado por NVIDIA. Desde consola, se ejecuta el siguiente comando.

"python3 train.py –model-dir=models/[nombre del directorio donde se almacena el nuevo modelo entrenado, dentro de la carpeta models] -- batch-size=8 --workers=1 --epochs=35 data/[nombre de la carpeta donde se almacenaron las imágenes con camera-capture]"

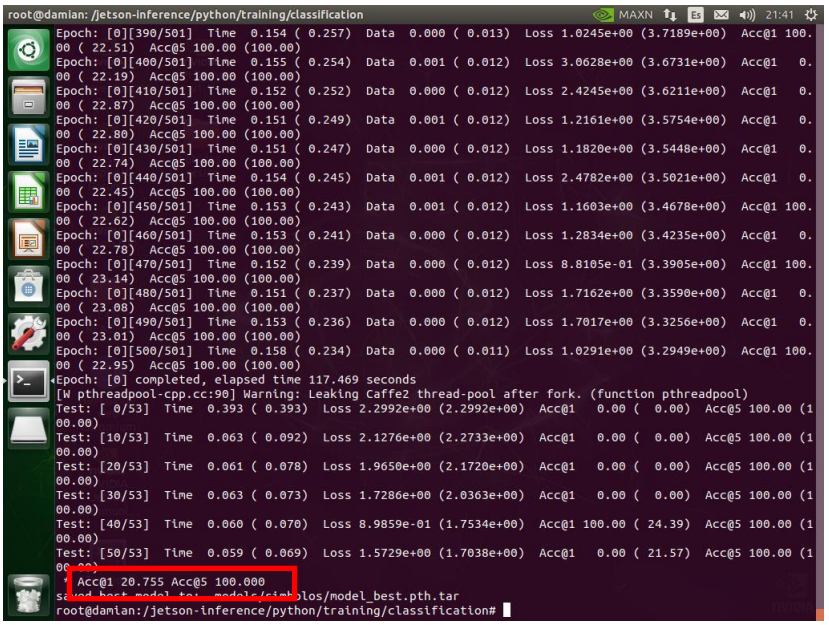
- **batch-size**: establece cuantas imágenes procesa a la vez durante el entrenamiento, por defecto son 8 pero para disminuir el trabajo realizado por la Jetson Nano, se puede colocar un número inferior.

- **workers:** número de cargadores de datos, por defecto son 2.

- **epochs:** es el número de iteraciones que realizará al momento de entrenar el modelo. Una iteración es un paso por todas las imágenes. Por defecto son 30 iteraciones, con el cual se tiene una precisión aproximada del 80% sin necesidad de forzar demasiado los recursos.

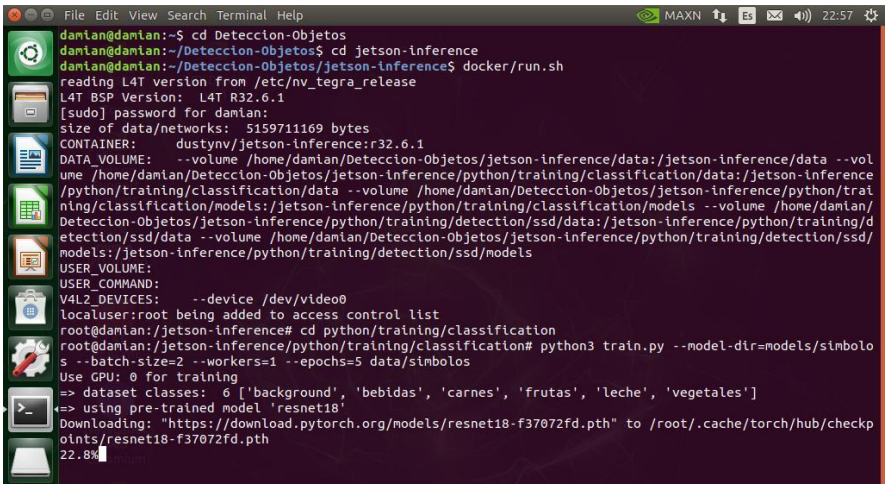
En la Figura 30 se ve el final del primer entrenamiento, se obtiene una precisión del 20.755%. Una precisión aceptable, considerando batch-size=1, workers=1 y epochs=1.

Figura 30 ***Primer entrenamiento***



Se procede a realizar un segundo entrenamiento, cambiando los parámetros de entrenamiento, batch-size=2, workers=1 y epochs=5, como se muestra en la Figura 31. Obteniendo una precisión del 50%, aceptable en un tiempo corto de entrenamiento.

Figura 31 ***Segundo entrenamiento***



Después de terminar el entrenamiento, se procede a exportar el modelo entrenado de PyTorch al formato ONNX, que es un modelo de formato abierto que soporta diversos entornos

de trabajo, frameworks, propios del Deep Learning, y simplifica la transferencia de modelos. Pytorch incluye un algoritmo para exportar modelos de Pytorch a ONNX, para esto se ejecuta el script onnx_export.py, con la siguiente línea de comandos.

"python3 onnx_export.py –model-dir=models/[nombre del directorio donde se almacena el nuevo modelo entrenado]"

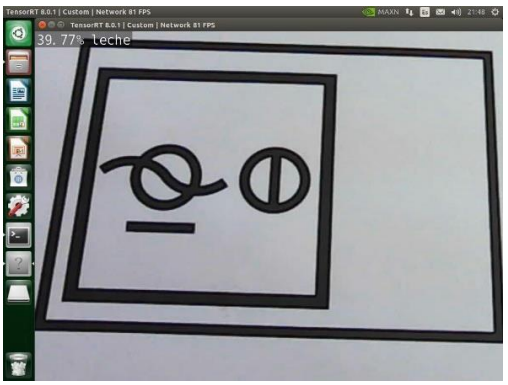
Esto genera un modelo llamado "ModelsMSDDv2" dentro de la carpeta 'models'.

Una vez exportado el modelo, se procede a probar el modelo con el método imageNet, que usa la cámara para procesar las imágenes y clasificar según las etiquetas, usando la cámara. En consola se ejecuta la siguiente línea de comandos:

imagenet.py –model=models/[nombre del directorio donde se almacena el nuevo modelo entrenado]/resnet18.onnx –labels=data/[nombre del directorio donde se almacena el nuevo modelo entrenado]/[nombre archivo txt con las etiquetas].txt –input_blob=input_0 –output_blob=output_0 /dev/video0

Se muestra en la Figura 32 la detección y clasificación de un símbolo con un 39.77% de precisión.

Figura 32 ***Clasificación live camera***



2.3.4. Fase 4: Construcción e implementación del prototipo

Actividad 10.- Selección de tarjeta embebida

Para la selección de la tarjeta embebida es necesario relacionar los requerimientos que ameritan, por ello se recrea una matriz comparativa para la selección de este sistema, la cual se eligió al cumplir la mayor parte de los requerimientos como se puede ilustrar en la siguiente Tabla 9.

Tabla 9. Tabla comparativa de selección de tarjeta embebida

Tarjeta Embebida	**REQUERIMIENTOS**								**VALORACIÓN TOTAL**
	RU2	RU4	RO2	RS4	RS5	RS8	RA5	RA10	
Raspberry Pi B4	1	1	0	0	1	0	1	0	4
Jetson NANO	1	1	1	1	1	1	1	1	8
									Cumple = 1
									No Cumple = 0

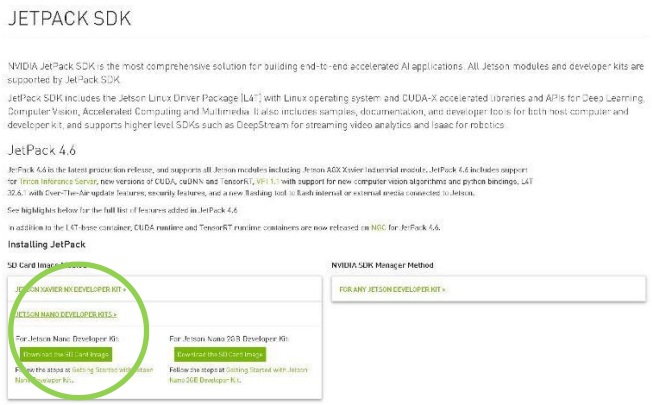
La tarjeta seleccionada es la Jetson NANO, a pesar de que ciertos elementos son similares hay ciertas diferencias que hace de la Jetson NANO la mejor elección, Ya que ambos tienen un procesador ARM, 4GB RAM y una serie de conexiones periféricas, pero la mayor diferencia entre los dos es que NVIDIA Jetson Nano incluye una GPU (unidad de procesamiento de gráficos) más potente y de mayor rendimiento, mientras que Raspberry Pi 4 tiene un procesador multimedia VideoCore de bajo consumo.

Actividad 11.- Instalación de dependencias en el hardware

En esta actividad se puede ver que no hay que intervenir el hardware directamente, ya que los periféricos que se conectan normalmente en los puertos USB de3.0, no obstante se tiene que realizar la instalación del sistema operativo, y de complementos que ayudaran a optimizar el objetivo del dispositivo

Para preparar la Jetson Nano se descarga el JetPack SDK de la página oficial de NVIDIA, como se muestra en la Figura 33.

Figura 33 ***Descarga JetPAck SDK***



Nota. Descargar Jetson Nano Developer Kit de acuerdo con la memoria RAM de la Jetson Nano que se tenga.

Para poder instalar el sistema Operativo dentro de la microSD y que el ordenador pueda correrlo, es necesario *flashear* la imagen en la microSD. Para esto se usa una herramienta de uso libre, BalenaEtcher.

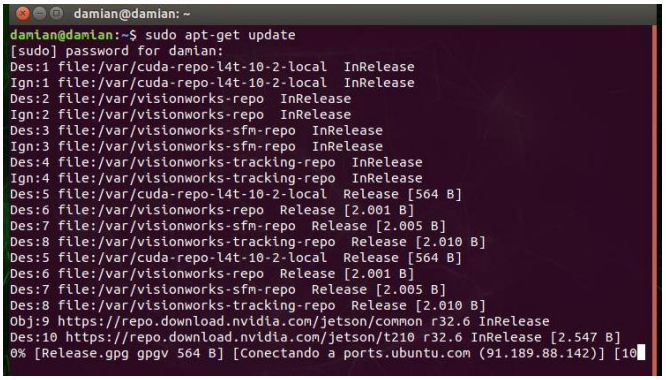
Una vez descargado la imagen, se abre el software BalenaEtcher, se presiona en la imagen desde la ubicación donde se descargó, se selecciona la microSD como target y para terminar el botón ‘flash’. Luego de un momento la microSD está lista para trabajar dentro de la Jetson Nano. Se muestra la interfaz de BalenaEtcher en la Figura 34

Figura 34 ***Flashear imagen JetPack***

Una vez lista la imagen flasheada en la microSD, se inserta en ranura de la JetsonNano y conecta al ordenador, se muestra el asistente de instalación del sistema operativo Ubuntu, que es un asistente fácil e intuitivo. Para acceder a una conexión a internet, la Jetson Nano tiene un puerto para cable Ethernet, caso contrario se puede adquirir las antenas propias de la Jetson Nano para acceder a Internet via Wifi.

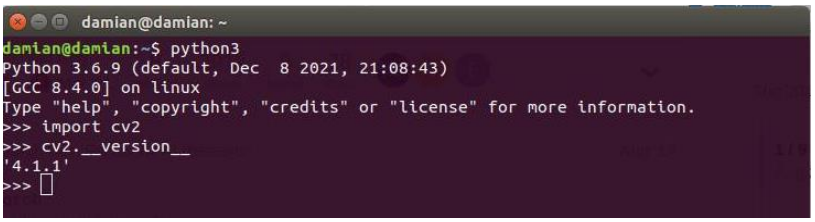
Una vez instalado el Sistema Operativo, se ejecuta los comandos **sudo apt-get update** y **sudo apt-get upgrade**, como se muestra en la Figura 35 *Terminal Ubuntu* para actualizar todos los parches de NVIDIA que pueden no estar instalados en la imagen descargado en el JetPack. Se procede a la Revisación de la versión de Python ejecutando el comando **python3** dentro del terminal de comandos.

Figura 35 *Terminal Ubuntu*



Una vez dentro del entorno de Python se revisa, importa la librería cv2, Open CV, y se verifica su versión como se muestra en la Figura 36 *Versiones Python OpenCV*.

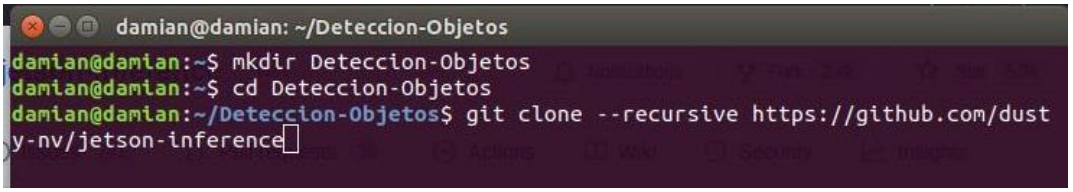
Figura 36 *Versiones Python OpenCV*



Docker Image

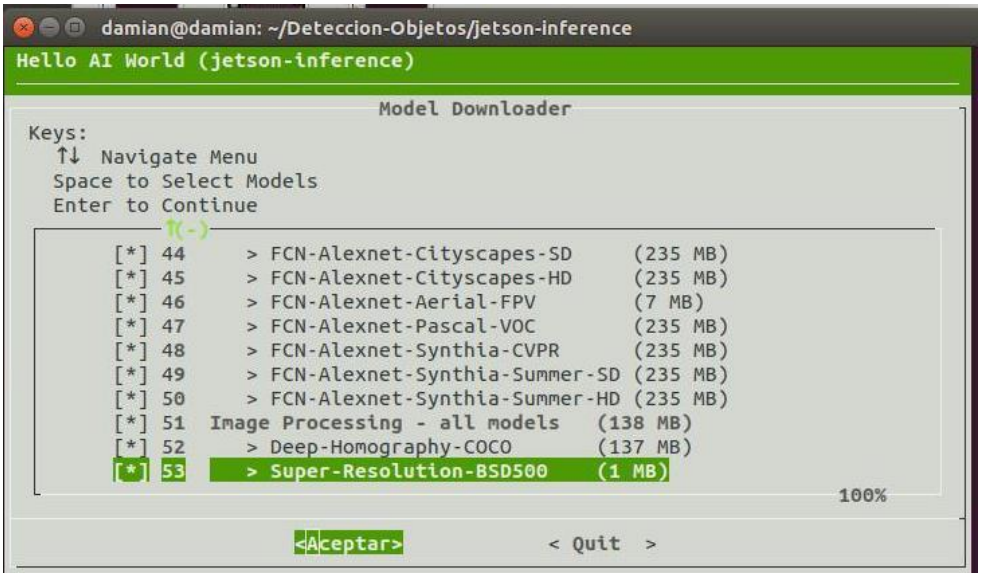
Como herramienta para crear el contenedor, o Docker, se clona un repositorio de Github, propio de NVIDIA como se muestra en la Figura 37.

Figura 37 ***Repositorio Github***



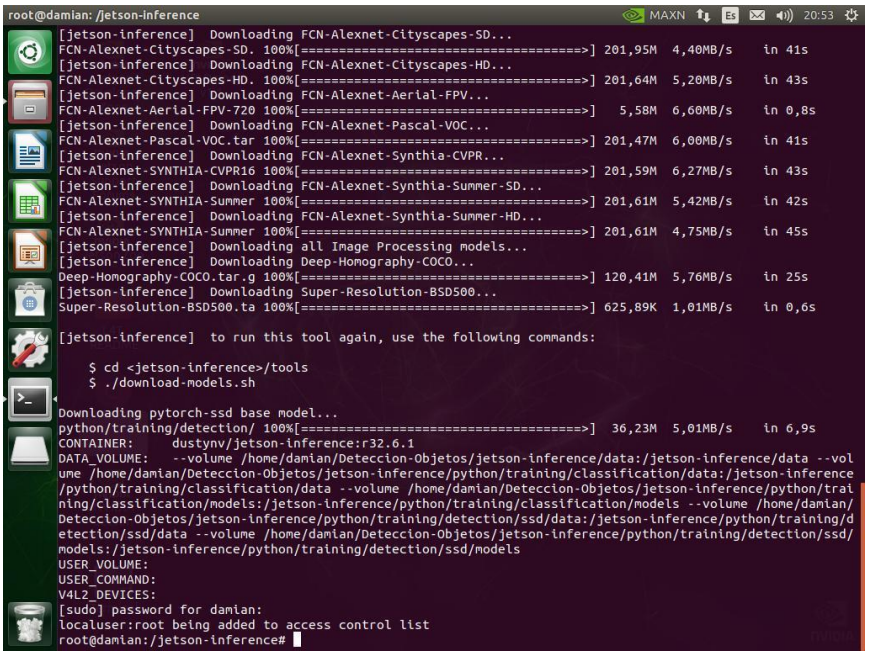
Una vez clonado el repositorio se accede a la carpeta **'jetson-inference'** y se ejecuta el comando '**./docker/run.sh'** Una vez ejecutado el comando se despliega el Model Downloader, para descargar algunos modelos de detección y entrenamiento de imágenes, como se muestra en la Figura 38 , al terminar de descargar los paquetes, que están dentro del contenedor.

Figura 38 Model Downloader



Esto lo hace para no correr el algoritmo directamente en el sistema operativo, se ejecuta en una especie de máquina virtual pequeña, que es el contenedor o Docker. En la Figura 39 se muestra que está dentro del Docker, la palabra root inicia la línea de comandos.

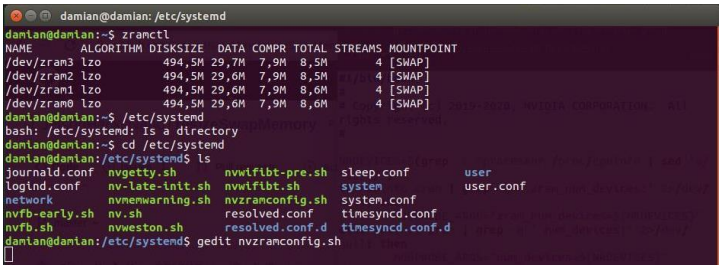
Figura 39 ***Docker***



Optimizador Memoria SWAP

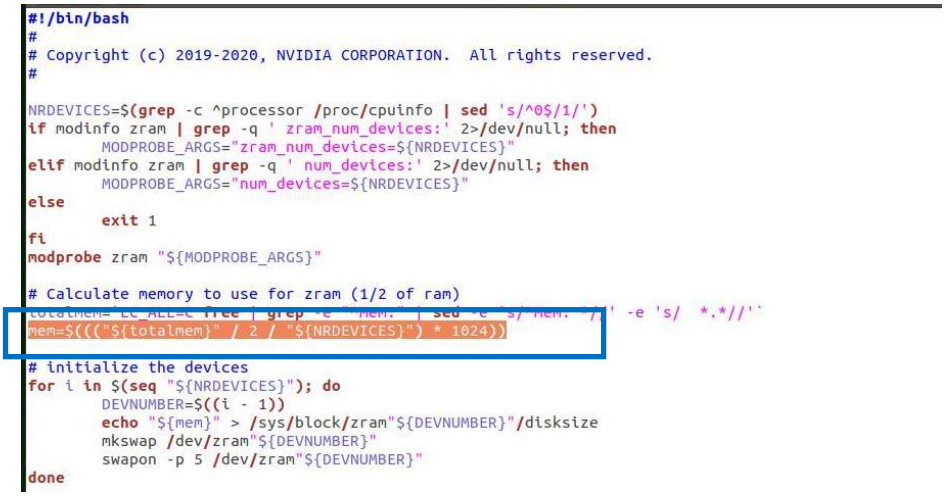
La Jetson Nano tiene una memoria RAM de 4 Gb, sin embargo, es capaz de realizar un aprendizaje profundo de pequeños datasets, ampliando su memoria SWAP (Figura 40).

Figura 40 ***Partición memoria SWAP***



En Linux, la función del espacio SWAP () se describe como un espacio de intercambio, cuando la memoria RAM del sistema no es suficiente, una parte de esta es liberada y usada por el disco duro temporalmente, reduciendo el uso de la RAM (Qiang, 2018).

Figura 41 ***Información SWAP***

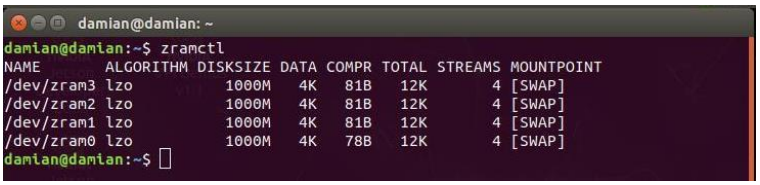


Para esto se revisa la partición de la Jetson Nano, con el comando **zramctl**. como se muestra en la Figura 40, tiene 500 Mb por cada núcleo, es decir 2 Gb destinado a procesar. Para realizar un entrenamiento, se aumenta la memoria SWAP a 4 Gb.

Figura 42 ***SWAP modificada***

Como se muestra en la Figura 41 en la línea 18 se detalla la memoria a usar, se cambia el valor como se muestra en la Figura 42 y en la Figura 43 se muestra la memoria SWAP modificada por consola, lista para comenzar el entrenamiento.

Figura 43 ***SWAP modificada***



Actividad 12. Selección de fuente de alimentación

Teniendo en cuenta que el consumo de la Jetson es básicamente de 5 Watts junto con un voltaje de 5Voltios, tendríamos la corriente consumida de 1 Amperio, no obstante, al momento de hacer la ejecución se activan procesos administrativos de CUDA(Compute Unified Device Architecture), que básicamente es una plataforma de computación paralela en otras

palabras una API, que genera un consumo máximo de 3,84 Amperios o 3840 mA, para tener una fuente no fija se realizó el siguiente calculo:

$Ih = \frac{□i□ * I□a□}{□} = 14400□Ah$	**(1)**

$$Ih = \frac{3h * 3840□A}{0.8} = 14400□Ah$$

En la en la ecuación (1), tenemos lim como el limite de tiempo que debe durar la batería, Imax como el consumo del sistema y C como la capacidad de 0.8. Teniendo como resultado de la operación 14,4 Ah.

De acuerdo con el consumo total anteriormente mencionado se escogió dentro del mercado nacional el módulo USB Wemos 18650 V3 (Figura 44) con batería de 3.7v por 8 Amperios, que tiene 3 tipos de salida, la salida USB de 5 voltios 2 Amperios, la de 3.3 voltios 2 amperios y 5voltios 8 Amperios.

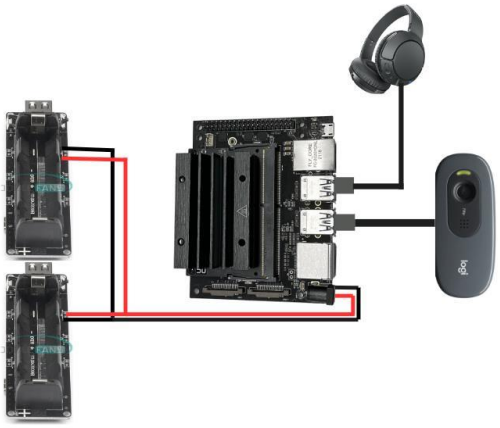
Figura 44. Modulo Wemos 18650 V3

En vista del resultado se solicito 2 de estos modulares para ocuparlos paralelamente, teniendo una carga de 5 Voltios 16 Amperios.

Actividad 13. Conexiones con tarjeta embebida

Las conexiones son netamente sencillas por ende son directas a la Jetson NANO como se muestra en la Figura 45. Diagrama de conexiones del sistema

Figura 45. Diagrama de conexiones del sistema



Actividad 14. Carcasa del equipo

Se empleo una carcasa de material PETG, que es resistente a distintos impacto y fuerzas mecánicas, hecho básicamente para armazones resistentes, este tiene su armazón con un espacio para el disipador y la respectiva ventilación.

Figura 46. Carcasa del sistema embebido

CAPÍTULO III

Resultados

3.1. Especificaciones del sistema

Para llegar al prototipo final, se hizo una investigación de campo, abordando la vida diaria de las personas no videntes, lo cual llevo a utilizar diversas estrategias tecnológicas, para llevar a cabo el objetivo principal, con ello llevo al equipo a tener las siguientes especificaciones:

3.1.1. Especificaciones de Técnicas:

- GPU: NVIDIA Maxwell con 128 núcleos
- Procesador: ARM A57 de cuatro núcleos a 1,43 GHz
- Memoria: LPDDR4 de 64 bits 4 GB 25,6 GB / s
- Codificador de video : 4 K @ 30 | 2x 1080p @ 60 | 4x 1080p @ 30 | 9x 720p @ 30 (H.264/H.265)
- USB : x4 USB 3.0, USB 2.0 Micro-B
- Alimentación: Micro-USB 5V 2A o DC 5V 4A
- Botones de encendido, recuperación forzada, reinicio Pruebas y validación
- Modelo de entrenamiento Mobilenet SSDv2 con una pérdida de 0.98778.
- Precisión del algoritmo del 50%.
- Sistema embebido escalable, abierto a diversas actualizaciones.
- Carcasa resistente de material PETG.
- Cámara de 720p/30fps.
- Campo visual de 55º.
- Corrección Lumínica automática RIGHT LIGHT 2.

- Detección de 5 símbolos: Perchero de verduras, carnes, lácteos, frutas y bebidas.

3.2. Funcionamiento

3.2.1. Fase 5: Validación y pruebas de funcionamiento.

Validación del algoritmo

Para poder evaluar el algoritmo se debe utilizar la matriz de confusión la cual ayuda a obtener los parámetros característicos de un algoritmo en base a porcentajes, ayudando a determinar os elementos principales del algoritmo.

Dichos elementos son conseguidos al tener una muestra de imágenes positivas y negativas del objeto a evaluar, en otras palabras, una cantidad de imágenes con el objeto y otra cantidad de imágenes sin el objeto, en este caso se tomó 501 imágenes positivas y 501 imágenes negativas por objeto, procediendo a los siguientes resultados:

Objeto 1: Lácteos

- *Matriz de confusión*

Tabla 10: Matriz de confusión – objeto: Lácteos

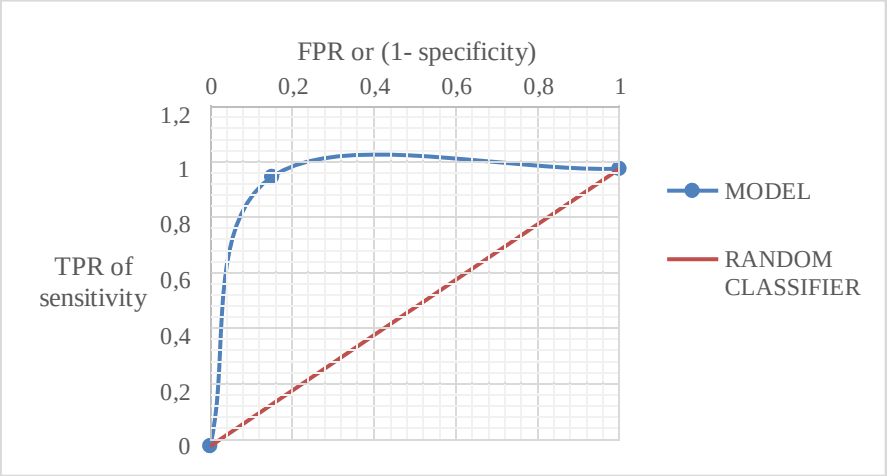
Matriz de confusión		Estimado por el modelo	
		Negativo (N)	Positivo (P)
Real	Negativo	414	12
	Positivo	87	489

Tabla 11: Parámetros del algoritmo – objeto: Lácteos

Medida	Valor	Ecuación
Sensibilidad	0.9718	$\square P\square = \square e\square\square i\square i\square i\square\square = \frac{\square P}{\square P + FN}$
Especificidad	0.8490	Especificidad $= \frac{\square N}{\square N + FP}$
Precisión	0.9012	PPV $= P\square eci\square ió\square = \frac{\square P}{\square P + FP}$
Valor predictivo negativo	0.1779	NPV $= \frac{\square N}{\square N + FN} * N$
Tasa de Falsos Positivos	0.1510	FPR $= \frac{\text{FP}}{(\text{FP} + \text{TN})}$
Tasa de descubrimiento falso	0.1736	$F\square\square = \frac{FP}{(FP + \square P)}$
Tasa de falsos negativos	0.0281	$FN\square = \frac{FN}{(FN + \square P)}$
Exactitud	0.9012	$A\square\square = \frac{\square P + \square N}{\square P + FP + \square N + FN}$
Puntuación F1	0.8932	$F1 = \frac{2}{\frac{1}{p\square eci\square ió\square} + \frac{1}{\square ec\square pe\square aci ó\square}}$

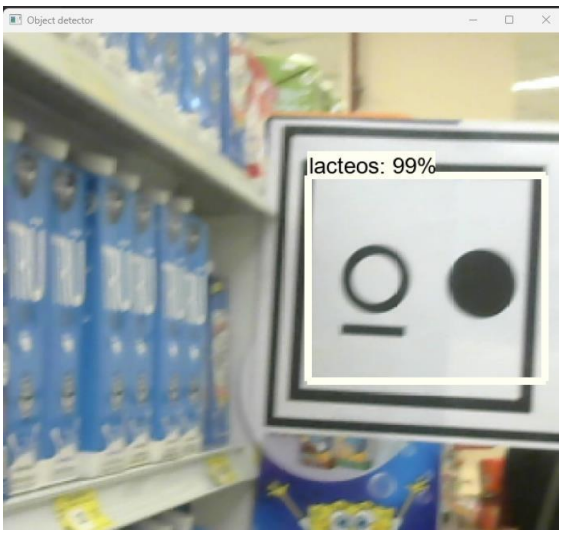
- ***Curva ROC***

Figura 47: Curva de ROC- Objeto: Lácteos



- **Funcionamiento - Lácteos:**

Figura 48. Detección de estante de lácteos



Objeto 2: Carnes

- ***Matriz de confusión***

Tabla 12: Matriz de confusión – objeto: Carnes

Matriz de confusión		Estimado por el modelo	
		Negativo (N)	Positivo (P)
Real	Negativo	433	73
	Positivo	68	428

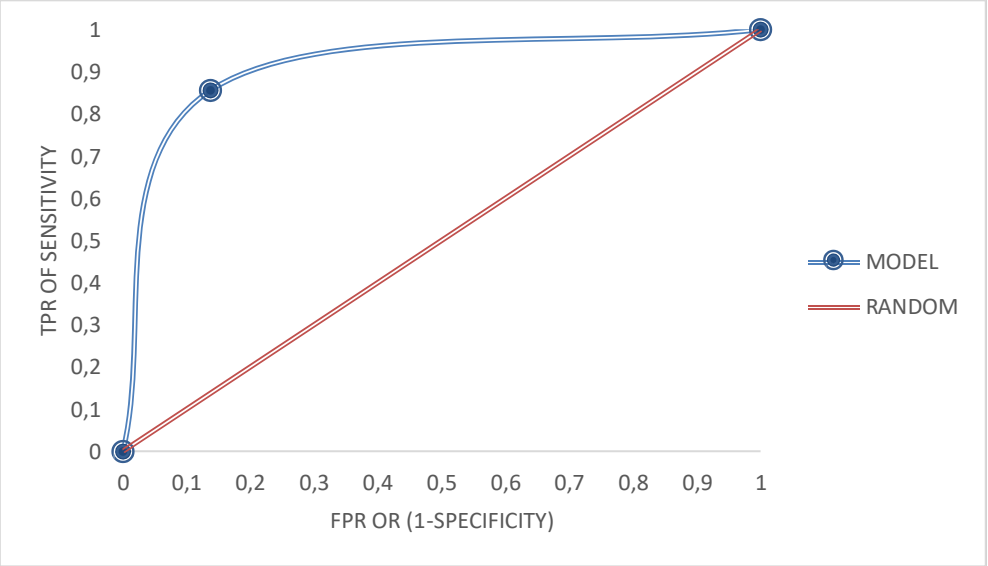
Tabla 13: Parámetros del algoritmo – objeto: Carnes

Medida	Valor	Ecuación
Sensibilidad	0.8642	$\square P\square = \square e\square\square i\square i\square i\square\square = \frac{\square P}{\square P + FN}$

Medida	Valor	Ecuación
Especificidad	0.8629	$\text{Especificidad} = \frac{\square N}{\square N + FP}$
Precisión	0.8592	$\text{PPV} = P\square eci\square ió\square = \frac{\square P}{\square P + FP}$
Valor predictivo negativo	0.1754	$\text{NPV} = \frac{\square N}{\square N + FN} * N$
Tasa de Falsos Positivos	0.137	$\text{FPR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}$
Tasa de descubrimiento falso	0.0457	$F\square\square = \frac{FP}{(FP + \square P)}$
Tasa de falsos negativos	0.0120	$FN\square = \frac{FN}{(FN + \square P)}$
Exactitud	0.8593	$A\square\square = \frac{\square P + \square N}{\square P + FP + \square N + FN}$
Puntuación F1	0.8600	$F1 = \frac{2}{\frac{1}{p\square eci\square ió\square} + \frac{1}{\square ec\square pe\square aci\acute{o}\square}}$

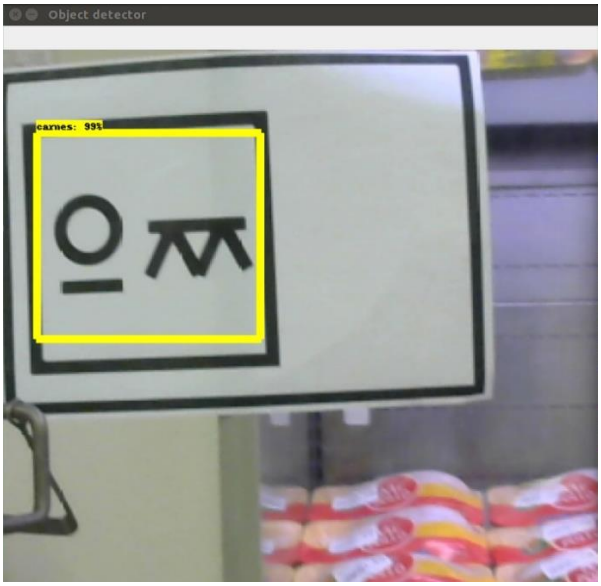
- ***Curva ROC***

Figura 49: Curva de ROC- Objeto: Carnes



- Funcionamiento – Carnes:

Figura 50. Detección de estante de carnes



Objeto 3: Vegetales

- ***Matriz de confusión***

Tabla 14: Matriz de confusión – objeto: Vegetales:

Matriz de confusión		Estimado por el modelo	
		Negativo (N)	Positivo (P)
Real	Negativo	424	18
	Positivo	77	483

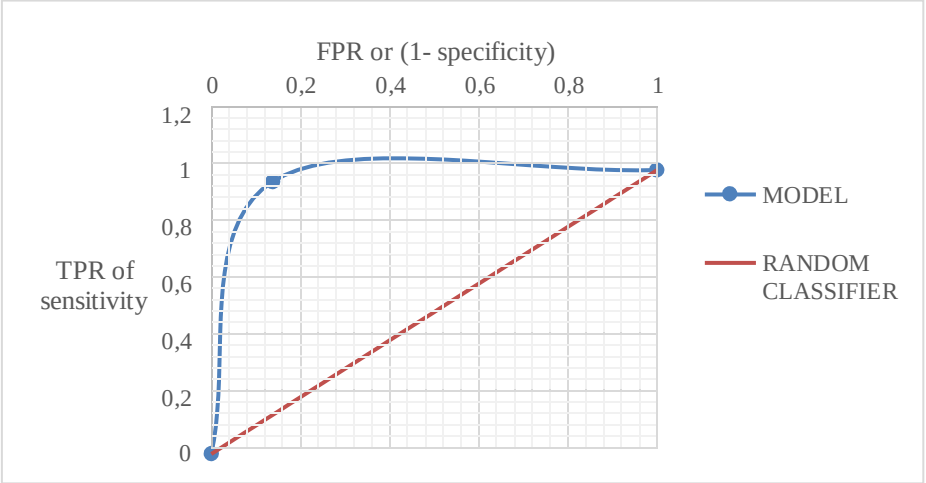
Tabla 15: Parámetros del algoritmo – objeto: Vegetales:

Medida	Valor	Ecuación
Sensibilidad	0.9593	$\square P\square = \square e\square\square i\square i\square i\square\square = \frac{\square P}{\square P + FN}$

Medida	Valor	Ecuación
Especificidad	0.8625	$\text{Especificidad} = \frac{\square N}{\square N + FP}$
Precisión	0.8463	$\text{PPV} = P\square eci\square ió\square = \frac{\square P}{\square P + FP}$
Valor predictivo negativo	0.1312	$\text{NPV} = \frac{\square N}{\square N + FN} * N$
Tasa de Falsos Positivos	0.1375	$\text{FPR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}$
Tasa de descubrimiento falso	0.1451	$F\square\square = \frac{FP}{(FP + \square P)}$
Tasa de falsos negativos	0.1244	$FN\square = \frac{FN}{(FN + \square P)}$
Exactitud	0.9052	$A\square\square = \frac{\square P + \square N}{\square P + FP + \square N + FN}$
Puntuación F1	0.8993	$F1 = \frac{2}{\frac{1}{p\square eci\square ió\square} + \frac{1}{\square ec\square pe\square ació\square}}$

- ***Curva ROC***

Figura 51: Curva de ROC- Objeto: Vegetales:



- Funcionamiento – Vegetales:

Figura 52. Detección de estante de Vegetales

Objeto 4: Bebidas

- ***Matriz de confusión***

Tabla 16: Matriz de confusión – objeto: Bebidas

Matriz de confusión		Estimado por el modelo	
		Negativo (N)	Positivo (P)
Real	Negativo	434	31
	Positivo	67	495

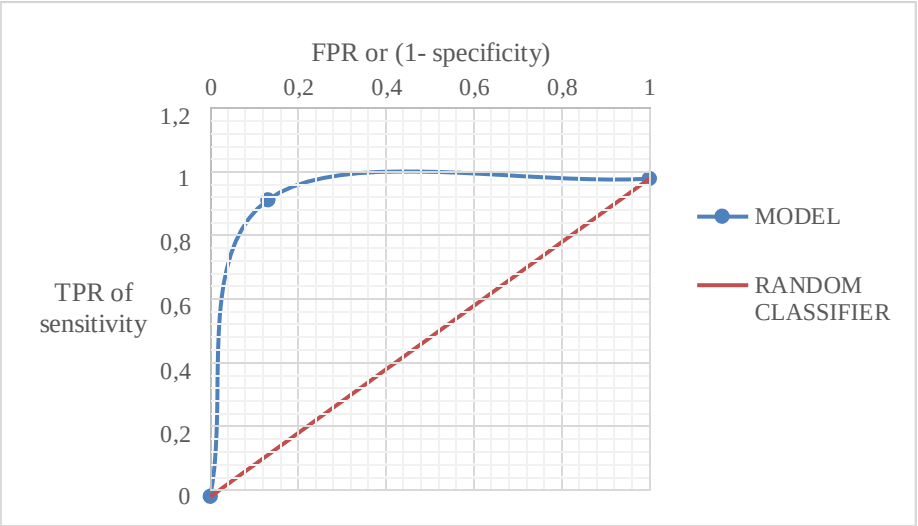
Tabla 17: Parámetros del algoritmo – objeto: Bebidas

Medida	Valor	Ecuación
Sensibilidad	0.9333	$□P□ = □e□□i□i□i□□ = \frac{□P}{□P + FN}$

Medida	Valor	Ecuación
Especificidad	0.8808	$\text{Especificidad} = \frac{\square N}{\square N + FP}$
Precisión	0.8662	$\text{PPV} = P\square eci\square ió\square = \frac{\square P}{\square P + FP}$
Valor predictivo negativo	0.6514	$\text{NPV} = \frac{\square N}{\square N + FN} * N$
Tasa de Falsos Positivos	0.1192	$\text{FPR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}$
Tasa de descubrimiento falso	0.0014	$F\square\square = \frac{FP}{(FP + \square P)}$
Tasa de falsos negativos	0.9333	$FN\square = \frac{FN}{(FN + \square P)}$
Exactitud	0.9046	$A\square\square = \frac{\square P + \square N}{\square P + FP + \square N + FN}$
Puntuación F1	0.8985	$F1 = \frac{2}{\frac{1}{p\square eci\square ió\square} + \frac{1}{\square ec\square pe\square aci\acute{o}\square}}$

- ***Curva ROC: Bebidas***

Figura 53: Curva de ROC- Objeto: Bebidas



- **Funcionamiento - Bebidas:**

Figura 54. Detección de estante de Bebidas

Objeto 5: Frutas

- ***Matriz de confusión***

Tabla 18: Matriz de confusión – objeto: Frutas

Matriz de confusión		Estimado por el modelo	
		Negativo (N)	Positivo (P)
Real	Negativo	495	6
	Positivo	6	495

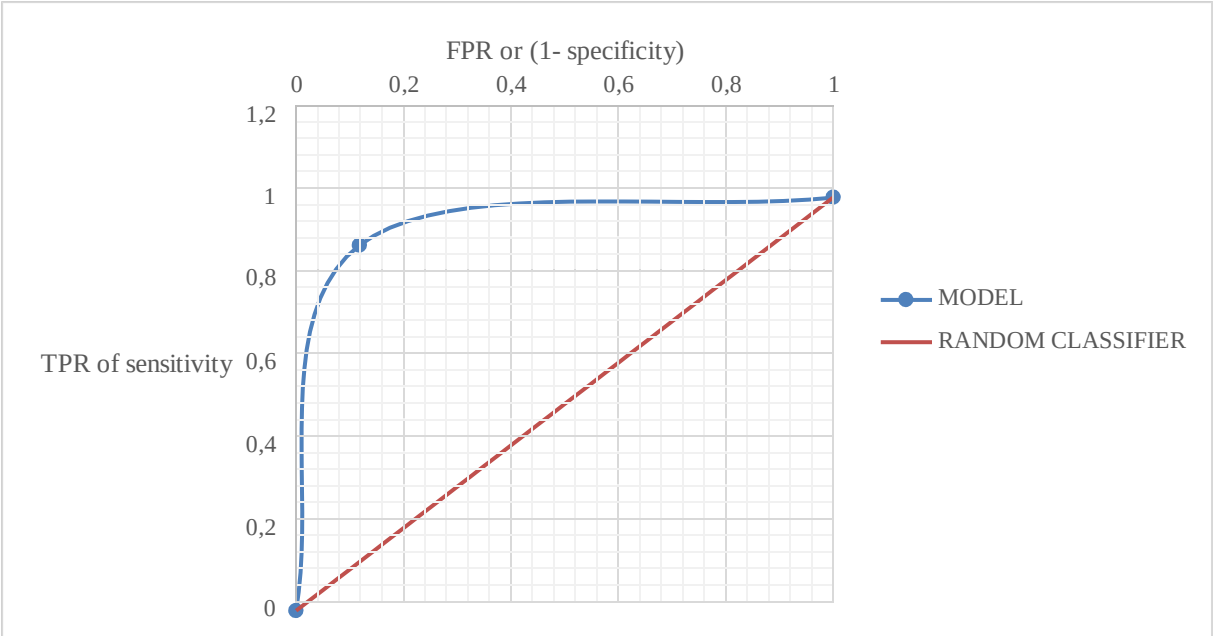
Tabla 19: Parámetros del algoritmo – objeto: Frutas

Medida	Valor	Ecuación
Sensibilidad	0.9880	$\square P\square = \square e\square\square i\square i\square i\square\square = \frac{\square P}{\square P + FN}$

Medida	Valor	Ecuación
Especificidad	0.8825	$\text{Especificidad} = \frac{\square N}{\square N + FP}$
Precisión	0.8822	$\text{PPV} = P\square eci\square ió\square = \frac{\square P}{\square P + FP}$
Valor predictivo negativo	0.884	$\text{NPV} = \frac{\square N}{\square N + FN} * N$
Tasa de Falsos Positivos	0.1175	$\text{FPR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}$
Tasa de descubrimiento falso	0.1891	$F\square\square = \frac{FP}{(FP + \square P)}$
Tasa de falsos negativos	0.1231	$FN\square = \frac{FN}{(FN + \square P)}$
Exactitud	0.8832	$A\square\square = \frac{\square P + \square N}{\square P + FP + \square N + FN}$
Puntuación F1	0.8831	$F1 = \frac{2}{\frac{1}{p\square eci\square ió\square} + \frac{1}{\square ec\square pe\square ació\square}}$

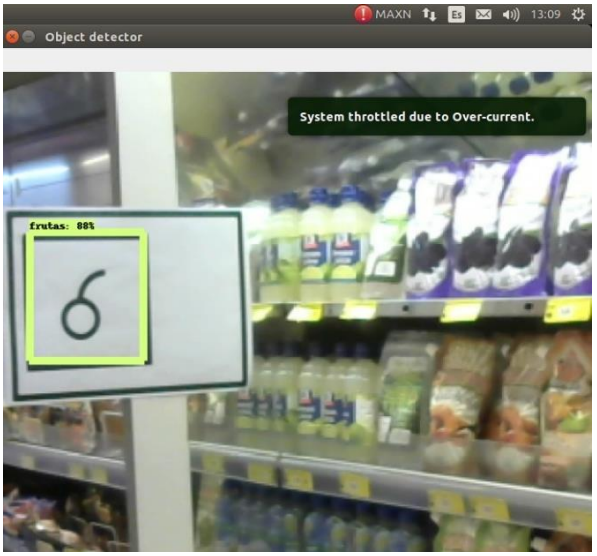
- ***Curva ROC***

Figura 55: Curva de ROC- Objeto: Frutas



- **Funcionamiento – frutas:**

Figura 56. Detección de estante de frutas



3.3. Análisis y resultados

Analizando los resultados del algoritmo queda en evidencia que la detección de cada uno de los objetos, supera las expectativas de ello, ya que ningún objeto dio porcentajes bajos del 50% en precisión, exactitud, especificidad y sensibilidad.

Teniendo un análisis general de la Tabla 20. Tabla general de resultados podemos tener una apreciación por objeto, podemos distinguir que existen porcentajes mayores del 50%, lo cual se considera un algoritmo optimizado.

Tabla 20. Tabla general de resultados

Objeto	Exactitud (%)	Precisión (%)	Sensibilidad (%)	Especificad (%)	F1SCORE (%)
Lácteos	90.12	82.63	84.90	84.90	89.32
Carnes	85.93	86.43	86.29	86.29	86.00
Vegetales	90.52	84.63	95.93	86.25	89.93
Bebidas	89.95	86.63	93.33	86.86	89.86
Frutas	88.32	88.22	88.40	88.25	88.31

Incluso cada una de las curvas analizadas el punto de fluctuantica, ninguno de ellos está por debajo de la pendiente, quiere decir en otras palabras que el algoritmo es optimo ya que tiende a llegar a la perfección.

No obstante, se puede mejorar con más reentrenamiento, e incluso para trabajos futuros se podría implementar funcionalidades IOT para poder no sobresaturar ningún sistema embebido, y se pueda utilizar en aplicaciones móviles.

CAPÍTULO IV

Conclusiones y Recomendaciones

4.1. Conclusiones

- De acuerdo al capítulo 1, se puede debatir que MobileNet-SSD V2 proporciona una velocidad 15 Fps algo similar a la de YOLOv5s, pero le falta precisión. Para fines de tiempo real, la velocidad es un factor determinante, pero la precisión del modelo es óptima es esencial para un funcionamiento fluido. Al pasar a YOLOv3, el modelo proporciona una excelente precisión, pero requiere hardware de computación intensiva. Si tal dispositivo está disponible, entonces este modelo sería suficiente para el requisito de velocidad. Por lo tanto, se puede decir que, dependiendo de los requisitos de varias aplicaciones, se puede elegir cualquiera de los modelos.

- El modelo realizado en SSD MobileNet v2 tiene un IoU (Intersection over Union) de 0.55 o 55%, una precisión de 0.86 o 86%, una recuperación general de 0.57 o 57% y un F1 score de 0.66 o 66%. Y un tiempo de procesamiento de 0.018 ms.

- Se diseño e implemento un sistema de detección de 5 perchas de productos de supermercado como: lácteos, carnes, vegetales, bebidas y frutas. Con un algoritmo presentado en el Anexo 5, desarrollado con TensorFlow y un modelo de Mobilenet SSDv2 con una precisión superior al 50% de cada objeto a detectar.

- La elección del modelo SSD Mobilenet V2 como CNN ahorró recursos informáticos como memoria de procesamiento lo cual da menor tiempo de respuesta, esto gracias al detector multibox de disparo único (SSD) se utiliza para reducir el tamaño y la

complejidad del modelo. en comparación con otros modelos con un precio de memoria más alto.

- Se construyo el prototipo con una tarjeta embebida llamada Jetson NANO la cual incluye una GPU (unidad de procesamiento de gráficos) más potente y de mayor rendimiento) a diferencia de la Raspberry Pi 4 tiene un procesador multimedia VideoCore de bajo consumo.

- Se valido el algoritmo en base a la matriz de confusión lo cual permitió analizar cada una de las etiquetas a detectar, determinando en sí que cada una supera el 50% de precisión y de igual manera su exactitud, no obstante se tiene una pérdida de 0.079 el cuál siendo cercano a cero se lo considera como aceptable.

- La utilidad del sistema de detección de perchas en el supermercado se evidencia ante la notificación de alerta mediante audio instantáneo. Esto permite dar comodidad a los usuarios no videntes reduciendo el impacto del accidente.

4.2. Recomendaciones

- Es importante confirmar la compatibilidad entre las variantes del programa y las bibliotecas de IA (inteligencia artificial) utilizadas. Para evitar problemas que puedan surgir al entrenar o implementar un modelo RCNN.

- Cabe señalar que la última variante del programa se encuentra principalmente en el proceso de prueba, por lo tanto, se indica elegir la variante anterior y más estable del programa para el desarrollo del proyecto. Se solucionó un problema al descargar una variante del programa.

- Este plan utiliza una cámara de computadora local para la detección de perchas de supermecado, en trabajos futuros se recomienda utilizar dispositivos móviles, los cuales ya vienen más optimizados con distintos sensores, que optimizaran la detección.

- Se recomienda en trabajos futuros utilizar un servidor donde aloje este algoritmo de visión artificial, y se pueda conectar con una aplicación móvil, siendo este algoritmo mucho mas portable y económico.

- Aunque se utilizo la tarjeta Jetson NANO obteniendo buenos resultados existen mejores estrategias para optimizar el funcionamiento y el gasto económico, además de ello existen nuevos modelos con una buena calidad de detección, sin necesidad de ocupar tarjetas embebidas repotenciadas.

Bibliografía

AFADACS. (21 de Noviembre de 2020). *afadacs.com*. afadacs.com: https://www.afadacs.com/baston-blanco-que-significan-los-colores/

Altamirano Cabrera, M. (2019). Prototipo de herramienta para la asistencia en el desplazamiento de personas con discapacidad visual. *Mujeres en la tecnología*, 61-75.

Azaña, C. (Diciembre de 2017). Diseño y Construcción de un Prototipo de Guante con Orientación y Localización para Ayuda en la Movilidad de Personas Invidentes. *Diseño y Construcción de un Prototipo de Guante con Orientación y Localización para Ayuda en la Movilidad de Personas Invidentes*. Quito, Pichincha, Ecuador: Escuela Politécnica Nacional.

Brodercick, P., y Blewitt, P. (2006). *The life span: human development for helping professionals* . Upper Saddle River: NJ: Pearson Education.

Chollet, F. (2018). *Deep Learning with Phyton.* Shelter Island, New York: Manning Pubications Co.

Chowdhary, K. (2020). *Fundamentals of Artificial Intelligence.* New Delhi, India: Springer Nature India Private Limited.

CONADIS. (06 de Abril de 2021). *Consejo Nacional para la Igualdad de Discapacidades*. https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/

Condo, I. (Diciembre de 2013). RunaTech: traje tecnológico de no videntes. *RunaTech: traje tecnológico de no videntes*. Quito, Pichinhca, Ecuador: Universidad San Francisco de Quito.

Endelman, G. S., Kok, H. K., Chandra, R. V., Razavi, R. V., Lee, M. J., y Asadi, H. (2018). Machine learning and the future of medicine. *Journal of Internal Medicine*, 603-619.

Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., . . . Socher, R. (2021). Deep learning-enabled medical computer vision. *Nature partner journals*.

Franklin, D. (18 de Septiembre de 2021). *Github*. dusty-nv: https://github.com/dusty-nv/jetson-inference/blob/master/docs/pytorch-cat-dog.md

Girshick, R. (2015). Fast R-CNN. *Microsoft Research, 2*(2), 9. https://doi.org/1504.0808

Gollapudi, S. (2019). *Learn Computer Vision Using OpenCV.* Berkeley, California.

GTI. (08 de Abril de 2020). *tugurium.com*. tugurium.com: http://www.tugurium.com/gti/termino.php?Tr=pixel%20value#:~:text=El%20formato%20de%20p%C3%ADxel%20m%C3%A1s,los%20diferentes%20tonos%20de%20gris.

Guttag, J. (2021). *Introduction to computation and programming using Python: with application to computational modeling and understanding data* (3 ed.). Massachusetts.

Hafsa, A., Vishwesh, R., y Nikhil, S. (2020). *The Computer Vision Workshop.* Packt Publishing Ltd. https://doi.org/978-1-80020-177-4

Halder, S., y Datta, P. (2011). An exploration into sef concept: A comparative analysis between the adolescents who are sighted and blind in Inda. *The British Journal of Visual Impairment*, 31-42.

Health Big Data. (15 de diciembre de 2022). *www.juanbarrios.com*. www.juanbarrios.com: https://www.juanbarrios.com/google-machine-learning/#TensorFlow

Hunt, J. (2019). *Advanced Guide to Python 3 Programming.* Cham, Switzerland: Springer Nature Switzerland AG.

Jiménez, M., Gónzález, D., y Martín, J. (2002). La clasificación internacional del funcionamineto de la discapacidad y de la salud 2001. *Revista Española de Salud Publica*, 271-279.

Jocher, G., Stoken, A., Borovec, J., Changyu, L., y Hogan, A. (s.f.).

Juarez, E. (12 de 02 de 2014). Diseño y desarrollo de sistema de orientación para invidentes. *Diseño y desarrollo de sistema de orientación para invidentes*. México, México, México: Universidad Nacional Autonoma de México.

Kaiming, H., Georgia, G., Piotr, D., y Ross, G. (2016). Mask R-CNN. *Facebook AI Research (FAIR)*, *3*(3), 12. https://doi.org/1703.06870

Lantz, B. (2013). *Machine Learning with R.* Birmingham: Packt Publishing Ltd.

López, B. A. (2021). Detección precoz de covid-19 a partir de imágenes de radiografías de tórax mediante redes neuronales convolucionales. *Detección precoz de covid-19 a partir de imágenes de radiografías de tórax mediante redes neuronales convolucionales.*

Universidad Abierta de Cataluña, Cataluña. Detección precoz de covid-19 a partir de imágenes de radiografías de tórax mediante redes neuronales convolucionales

Lozada, P. A. (17 de Octubre de 2018). *Instituto Nacional para CIegos de Colombia*. inci.gov.co: https://www.inci.gov.co/blog/orientacion-y-movilidad-de-personas-con-discapacidad-visual

Malpartida, E. A. (2003). *Sistema de visión artificial para el reconocimeinto y manipulación de objetos utiizando un brazo robot.* Lima.

McDonald, E. (1985). *Sistema Bliss Enseñanza y Uso.* Toronto: Blissymbolics Communication Institute.

Melillanca, E. (9 de Agosto de 2018). *ericmelillanca*. Evaluación de modelos de clasificación: Matriz de Confusión y Curva ROC: http://ericmelillanca.cl/content/evaluaci-n-modelos-clasificaci-n-matriz-confusi-n-y-curva-roc#:~:text=Evaluaci%C3%B3n%20de%20modelos%20de%20clasificaci%C3%B3n%3A%20Matriz%20de%20Confusi%C3%B3n%20y%20Curva%20ROC,-Publicado%20por%20Eric&text=Una%20Curva%20

Morena, A., Molina, G., y Garay, U. (2019). *Artificial Vision and Language Processing for Robotics.* Packt publishing Ltd. https://doi.org/10.2514/5.9781600868962.0000.0000

Moreno, F. (2016). *Diseño y contrucción de un dispositivo electrónico de ayuda y entretenimiento para personas con discapacidad visual a trav{es de ondas vibratorias e interfaces audibles, para el proyecto HandEyes del banco de ideas del Senescyt.* Sangolquí.

Nighania, K. (30 de Dicimbre de 2018). *towardsdatascience.com*. towardsdatascience.com: https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15

NUMPY. (14 de enero de 2023). *numpy.org*. numpy.org: https://numpy.org/

NVIDIA. (2019). *Jetson Nano Developer Kit User Guide.* NVIDIA.

OCI. (3 de Febrero de 2023). *oracle.com*. oracle.com: https://www.oracle.com/ar/artificial-intelligence/machine-learning/what-is-machine-learning/#:~:text=El%20aprendizaje%20autom%C3%A1tico%20(ML)%20es,que%20imitan%20la%20inteligencia%20humana.

ONCE. (9 de Diciembre de 2018). *perrosguia.once.es*. perrosguia.once.es: https://perrosguia.once.es/es/que-hacemos/nuestros-perros

ORIENTATECH. (28 de febrero de 2023). *orientatech.es*. orientatech.es: https://www.orientatech.es/sunu-band

Ortega, C., y Sanchez, D. (Junio de 2016). Diseño e implementación de un sistema de navegación para asistencia de personas no videntes. *Diseño e implementación de un sistema de navegación para asistencia de personas no videntes*. Quito, Pichincha, Ecuador: Escuela Politécnica Nacional.

Pallai, D. (2021). *Python 3 and Data Analytics Pocker Primer.* Dulles, Virginia: Mercury Learning and Information LLC.

Pandas.org. (12 de Diciembre de 2022). *pandas.pydata.org*. pandas.pydata.org: https://pandas.pydata.org/

Pérez, D., y Jiménez., A. (16 de enero de 2023). *tifloeduca.eu*. tifloeduca.eu: https://www.tifloeduca.eu/strap-dispositivo-sustituto-del-baston-blanco/

Perez, J. A., Deligianni, F., Ravi, D., y Yang, G.-Z. (2018). *arXiv*. https://arxiv.org/ftp/arxiv/papers/1803/1803.10813.pdf

PyImageSearch/imutils. (3 de enero de 2023). *github.com*. github.com: https://github.com/PyImageSearch/imutils

Qiang, Y. (2018). Optimizatio of Swap System on DAPHIC: Data Acces Pattern Hint Instrumenting Compiler. *Optimizatio of Swap System on DAPHIC: Data Acces Pattern Hint Instrumenting Compiler.* Universidad Nacional de Seoul, Seoul. https://s-space.snu.ac.kr/bitstream/10371/144020/1/000000152343.pdf

Rabiner, L., y Juang, B.-H. (2006). *Encyclopedia of Language & Linguistics* (2 ed.).

Reyes, F. (2016). *Diseño y construccion de un dispositivo electronico de ayuda y entrenamiento para personas con discapacidad visual a traves de ondas vibratorias e interfaces audibles, para el proyecto handeyes del banco de ideas del senescyt.* Repositorio ESPE: http://repositorio.espe.edu.ec/handle/21000/11626

Rouhiainen, L. (2018). *Inteligencia Artificial 101 cosas que debes saber hoy sobre nuestro futuro.* Barcelona: Alienta Editorial.

Saitoh, K. (2021). *Deep Learning from the Basics.* Birmingham: Packt Publishing.

Silva, E. (18 de febrero de 2019). *escninosciegos.org*. escninosciegos.org: https://escninosciegos.org/index.php/servicios/talleres/orientacion-y-movilidad/

Solano, M. (10 de Mayo de 2021). (D. Borja, Entrevistador)

Stanford Vision and Learning Lab. (2021). *Convolutional Neural Networks for Visual Recognition Course*. https://cs231n.github.io/convolutional-networks/

Stevens, E., y Antiga, L. (2019). *Deep Learning with Pytorch.* Shelter Island: Manning Publications.

Varadharajan, R., Shashikant, R., y Bhensdadiya, K. (2021). Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Time. *International Research Journal of Engineering and Technology (IRJET), 8*(7), 5. https://doi.org/2395-0072

Vargas, A. (Febrero de 2015). Sistema embebido de movilización y posicionamiento para personas no videntes mediante harware libre. *Sistema embebido de movilización y posicionamiento para personas no videntes mediante harware libre*. Ambato, Tungurahua, Ecuador: Universidad Técnica de Ambato.

Vasquez Salazar, R. D., y Cardona Mesa, A. A. (2015). Dispositivos de asistencia para la movilidad en personas con discapacidad visual: una revisión bibliográfica. *Revista Politécnica*, 107-116.

Vejarano, R. (2019). Eye QR-aplicación identificadora de objetos para personas con discapacidad visual. *Revista de Iniciación Científica* , 77-82.

Villán, A. F. (2019). *Mastering OpenCV 4 with Python.* Birmingham: Packt Publishing Ltd.

Young, J. (2016 de Abril de 2016). *beyondinfinity.com*. beyondinfinity.com: https://beyondinfinity.com.au/the-life-of-a-scientist/

Yu-Chen, C., Chi-Yi, T., Mind-Da, R., Guan-Yu, S., y Tsu-Tian, L. (2020). Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems. *2020 International Conference on System Science and Engineering (ICSSE)*, *2*(8), 5. https://doi.org/10.1109/ICSSE50014.2020.9219319

Anexos

Anexo1: Guion de entrevista

Guion de entrevista

Tiempo estimado de la entrevista:

90min

Criterios para comprobar antes de la entrevista:

El entrevistado es Docente y directiva de la asociación de no videntes de Imbabura (ANVI, quien en su trabajo desempeña mucho el enseñarse a desplazarse y orientarse.

En este caso usaremos su experiencia para sacar los diferentes requerimientos de parte del especialista, para el desarrollo de este trabajo de investigación.

Antes de la entrevista, se le pedirá a la entrevistada que confirme oralmente su consentimiento en la grabación de la entrevista.

Para la entrevista usaremos la técnica del embudo, empezando con preguntas generales y siguiéndolas con preguntas cada vez más concretas.

Descripción del proyecto:

EL proyecto consiste en el desarrollo de un Sistema electrónico con inteligencia artificial, capas de identificar los percheros de los super mercados, para ayudar a la orientación y desplazamiento de personas no videntes.

Entrevista

- ¿Cuáles son las técnicas para la orientación y movilidad de los no videntes?
- ¿El desarrollo de estas habilidades depende de la edad?
- ¿Cómo enseña a las personas no videntes a movilizarse y orientarse?
- ¿Cuáles son los principales problemas en la movilidad de las personas no videntes?
- ¿Cuáles son las herramientas que utilizan para movilizarse?
- ¿Existen equipos tecnológicos que ayude a la movilidad y orientación de las personas no videntes?
- ¿Cuáles son los principales problemas que se identificó en el uso de los diversos equipos tecnológicos?
- ¿Cómo se desplazan en un supermercado?
- ¿Cuáles son los sentidos que ayuda más en la orientación de la persona no vidente?
- ¿Existe algún medio de orientación en los centros comerciales para las personas no videntes?
- ¿Cuáles son las recomendaciones que puede darnos para la construcción del prototipo?

Anexo 2: entrevista

Ibarra,16 de noviembre del 2022.

La entrevista se realizó en la ciudad de Ibarra, en la provincia de Imbabura, en la Asociación de No videntes de Imbabura (ANVI) ubicada en la calle Juan Hernández 1-56 y Raúl Montalvo, esta entrevista documenta un guion junto con un documento contextual de toda la entrevista ubicada en el Anexo 2.

Actualmente ANVI cuenta con más de 140 miembros anexados de todos los alrededores de la provincia de Imbabura y de todos los rangos etarios a los cuales se les brinda atención, capacitación y formación para su inclusión en la sociedad.

ANVI está representada por la Srta presidenta Noemi Trejo la cual dio acogida a este proyecto aportando con la experiencia de los diferentes miembros, en esta entrevista referida en el ANEXO 2, dio a conocer la realidad en la que viven las personas no videntes.

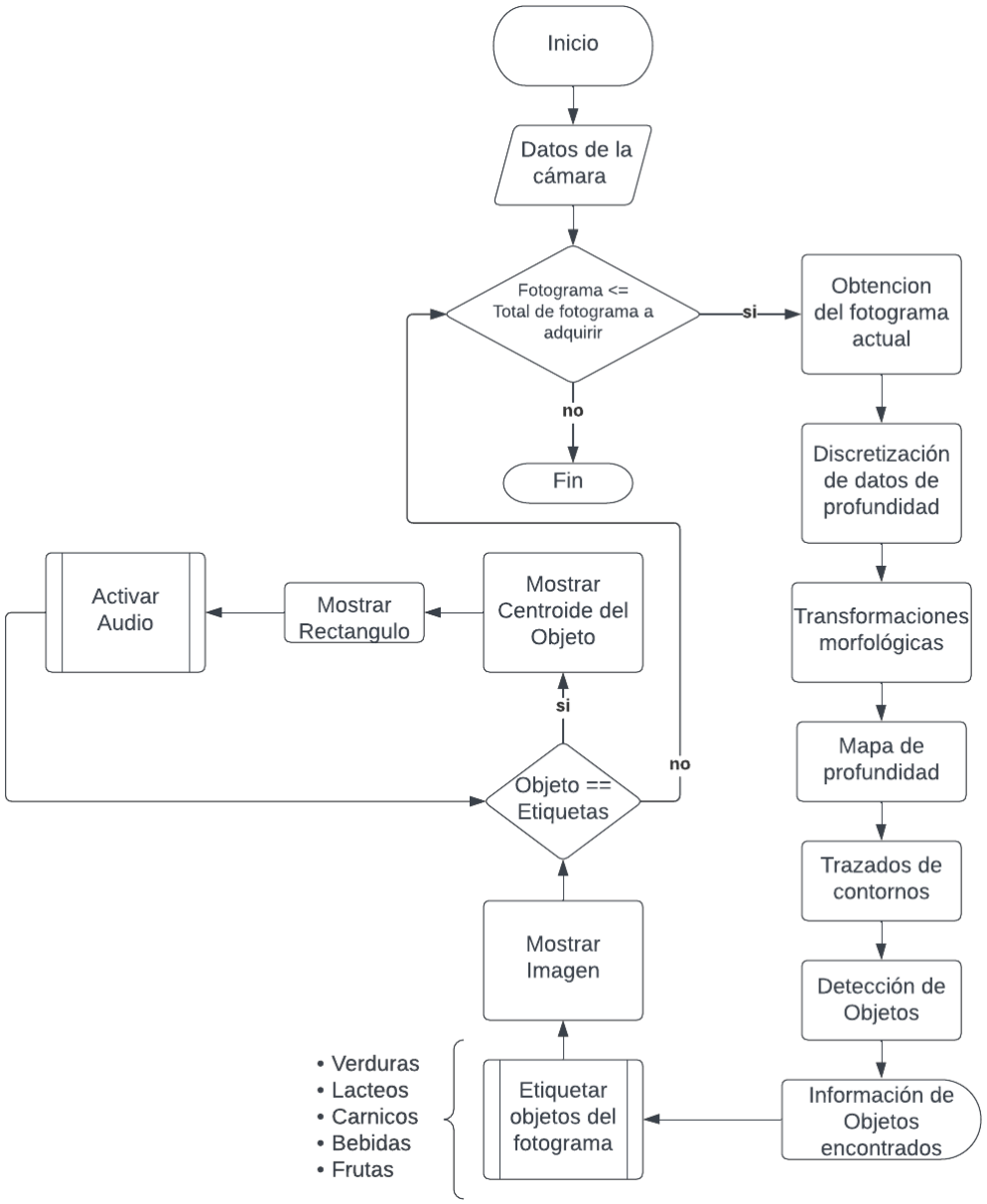
Específicamente con respecto a la habitualidad de la movilidad y orientación , dijo que las personas no videntes determinan la orientación y movilidad, dependiendo de la edad en la que se adquiriere la discapacidad, ya que si es de nacimiento, desde niños se va adquiriendo el desarrollo prematuro de los demás sentidos, obteniendo en si el manejo del espacio en el que se encuentra, al contrario cuando las personas de edad avanzada adquieren esta discapacidad, es un poco más difícil adaptarse, ya que los sentidos están acostumbrados a manejarse con la visión, y eso que independientemente depende del porcentaje de la discapacidad visual.

Determino también que tienen diversas herramientas para su movilidad, como el bastón, el cual ocupa tres colores diferentes de acuerdo a la gravedad de su discapacidad, y que la

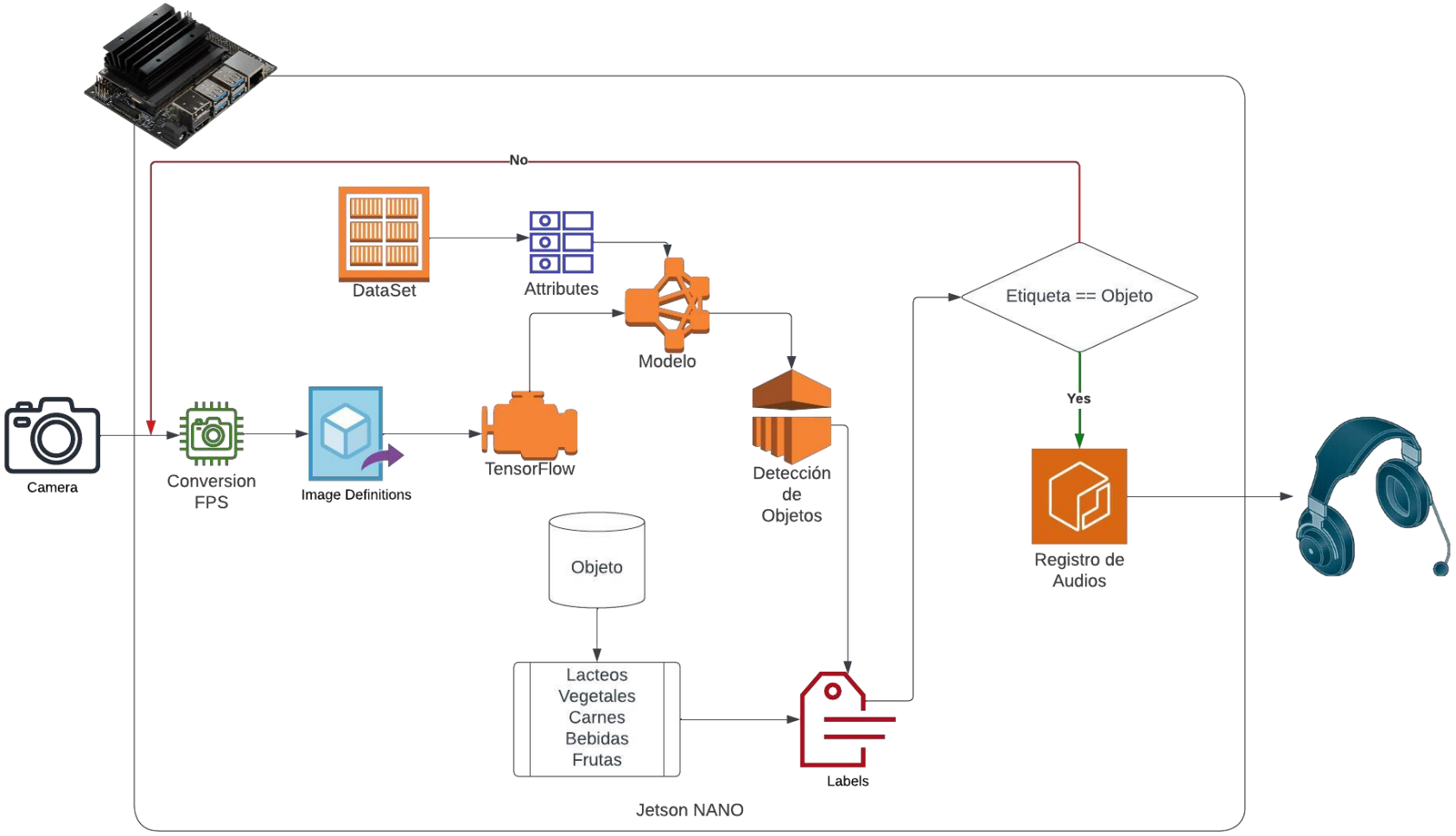
orientación se da de acuerdo a la experiencia que vayan adquiriendo, de acuerdo a las veces que transcurran en los diversos sitios, en los cuales agudizan los sentidos, teniendo en contexto sonidos o olores principales de cada lugar, para determinar en donde están.

El problema habitual al que se enfrentan es prácticamente en los diversos supermercados, en los cuales el problema no es la movilidad ni la orientación, si no mas bien en la identificación de algunos objetos, que ellos desean comprar, ya que ellos buscan su autonomía y un lugar dentro de la sociedad; luego argumento que si existen diversos equipos de los cuales han experimentado, pero por lo general se experimenta mucho estrés en los diversos avisos audibles, ya que tornan en ser repetitivos y redundantes, en los cuales a veces dan la respuesta equivocada del objeto, ya que los productos contienen diversas texturas y contornos similares.

Anexo 3: Flujograma



Anexo 4: Diagrama de bloques


No
DataSet
Attributes
Modelo
Etiqueta == Objeto
Yes
Camera
Conversion
FPS
Image Definitions
TensorFlow
Detección
de
Objetos
Registro de
Audios
Objeto
Lacteos
Vegetales
Carnes
Bebidas
Frutas
Labels
Jetson NANO

Anexo 5: Algoritmo

```
# Import packages
import os
import cv2
import numpy as np
import tensorflow as tf
import imutils
import sys

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")

# Import utilites
from utils import label_map_util
from utils import visualization_utils as vis_util

# Name of the directory containing the object detection module we're using
MODEL_NAME = 'inference_graph'

# Grab path to current working directory
CWD_PATH = os.getcwd()

# Path to frozen detection graph .pb file, which contains the model that is used
# for object detection.
PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,'frozen_inference_graph.pb')
```

```
# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH,'training','labelmap.pbtxt')

# Number of classes the object detector can identify
NUM_CLASSES = 6
i=0
## Load the label map.
# Label maps map indices to category names, so that when our convolution
# network predicts '5', we know that this corresponds to 'king'.
# Here we use internal utility functions, but anything that returns a
# dictionary mapping integers to appropriate string labels would be fine
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,  max_num_classes=NUM_CLASSES,
use_display_name=True)
category_index = label_map_util.create_category_index(categories)

# Load the Tensorflow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT,  'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def,  name='')

    sess = tf.Session(graph=detection_graph)
```

```
# Define input and output tensors (i.e. data) for the object detection classifier

# Input tensor is the image
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

# Output tensors are the detection boxes, scores, and classes
# Each box represents a part of the image where a particular object was detected
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

# Each score represents level of confidence for each of the objects.
# The score is shown on the result image, together with the class label.
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

# Number of objects detected
num_detections = detection_graph.get_tensor_by_name('num_detections:0')

# Initialize webcam feed
video = cv2.VideoCapture(0)


while(True):

    # Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]
    # i.e. a single-column array, where each item in the column has the pixel RGB value
    ret, frame = video.read()
    frame=imutils.resize(frame, width=580)
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

```
    frame_expanded = np.expand_dims(frame_rgb, axis=0)

    # Perform the actual detection by running the model with the image as input
    (boxes, scores, classes, num) = sess.run(
        [detection_boxes, detection_scores, detection_classes, num_detections],
        feed_dict={image_tensor: frame_expanded})

    # Draw the results of the detection (aka 'visulaize the results')
    vis_util.visualize_boxes_and_labels_on_image_array(
        frame, np.squeeze(boxes),
        np.squeeze(classes).astype(np.int32),
        np.squeeze(scores),
        category_index,
        use_normalized_coordinates=True,
        line_thickness=7,
        min_score_thresh=0.5)
    #print(np.squeeze(scores))
    # All the results have been drawn on the frame, so it's time to display it.
    cv2.imshow('Object detector', frame)
```