



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

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
EN MECATRÓNICA

TEMA:

“ALGORITMO DE VISIÓN ARTIFICIAL PARA APOYAR LA TERAPIA DE
LENGUAJE EN NIÑOS CON PARÁLISIS CEREBRAL.”

AUTOR: REYES INDACOCHEA RENÉ IVÁN

DIRECTOR:

ING. IVÁN IGLESIAS NAVARRO MSC.

IBARRA - ECUADOR

2022



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 IDENTIDAD:	DE	1311208050	
APELLIDOS NOMBRES:	Y	REYES INDACOCHEA RENÉ IVÁN	
DIRECCIÓN:	IBARRA		
EMAIL:	rireyesi@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0980237257

DATOS DE LA OBRA	
TÍTULO:	ALGORITMO DE VISIÓN ARTIFICIAL PARA APOYAR LA TERAPIA DE LENGUAJE EN NIÑOS CON PARÁLISIS CEREBRAL
AUTOR (ES):	REYES INDACOCHEA RENÉ IVÁN
FECHA: DD/MM/AAAA	20/10/2022
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERÍA EN MECATRÓNICA
ASESOR /DIRECTOR:	MSc. Iván Iglesias Navarro

2. CONSTANCIAS

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

Ibarra, a los 20 días del mes de octubre de 2022

EL AUTOR:



René Iván Reyes Indacochea



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

En calidad de director del presente trabajo de titulación denominado: **“ALGORITMO DE VISIÓN ARTIFICIAL PARA APOYAR LA TERAPIA DE LENGUAJE EN NIÑOS CON PARÁLISIS CEREBRAL”**, certifico que fue desarrollado por el Sr. René Iván Reyes Indacochea, con cédula de identidad: 1311208050, bajo mi supervisión.

**IVAN
IGLESIAS
NAVARRO**

Firmado digitalmente por
IVAN IGLESIAS NAVARRO
Motivo: Apruebo este documento
Fecha: 2022-10-19 16:45:05:00

MSc. Iván Iglesias, Ing.

DIRECTOR

DEDICATORIA

Este trabajo de titulación se lo dedico a toda mi familia en especial a mis padres que han sido un gran pilar en el transcurso de esta nueva meta.

A mi madre Jacqueline Indacochea por todo el esfuerzo y consejos que me brindo para seguir adelante en los momentos difíciles y a pesar de la distancia ha sabido brindarme su apoyo.

A mi padre Vicente Reyes por eso sabios consejos que siempre me motivaban a seguir adelante y nunca rendirme.

A toda mi familia por confiar en mí, gracias por su apoyo incondicional.

René Reyes

AGRADECIMIENTO

Agradezco a Dios, por brindarme esta nueva oportunidad de seguir con mis estudios, a mi familia por apoyarme incondicionalmente, por su comprensión y guía que me brindaron a lo largo de mis años de estudio.

Agradezco a todas las personas maravillosas que estuvieron acompañándome en todo este proceso a lo largo de los años, compañeros, docentes de la Universidad Técnica del Norte y a mi tutor Ingeniero Iván Iglesias por haber compartido sus conocimientos a lo largo de mi formación profesional.

René Reyes

RESUMEN

La visión por computadora se ha vuelto muy importante por sus diversas aplicaciones. Este trabajo muestra el desarrollo de un algoritmo de Eye Tracking enfocado en el apoyo en la terapia de lenguaje en niños con parálisis cerebral, el cual está realizado en el lenguaje de programación Python y está implementado en una tarjeta embebida Jetson Nano.

Para el inicio del trabajo, se estudió las diferentes técnicas y dispositivos existentes para el Eye Tracking, además de como implementarlos en niños con parálisis cerebral sin ser muy invasivos. Luego de haber escogido el método a seguir, se procedió con la adquisición y tratamiento de la imagen para eliminar ruidos y otros factores que puedan provocar falsos positivos en la detección del contorno de la pupila.

Finalmente, luego de obtener correctamente el contorno de la pupila, se obtuvieron las coordenadas del centroide para realizar una comparación de posición y poder determinar el movimiento ocular en tres direcciones, izquierda, centro y derecha.

ABSTRACT

Computer vision has become especially important for its various applications. This work shows the development of an Eye Tracking algorithm focused on support in language therapy in children with cerebral palsy, which is made in the Python programming language and is implemented in a Jetson Nano embedded card.

At the beginning of the work, the different existing techniques and devices for Eye Tracking will be studied, as well as how to implement them in children with cerebral palsy without being very invasive. After choosing the method to follow, we proceeded with the acquisition and treatment of the image to eliminate noise and other factors that can cause false positives in the detection of the contour of the pupil.

Finally, after correctly obtaining the contour of the pupil, the coordinates of the centroid were obtained to perform a position comparison and be able to determine the eye movement in three directions, left, center and right.

ÍNDICE DE CONTENIDO

DEDICATORIA	V
AGRADECIMIENTO	VI
RESUMEN	VII
ABSTRACT.....	VIII
ÍNDICE DE TABLAS.....	XII
ÍNDICE DE FIGURAS.....	XIII
CAPÍTULO I	15
INTRODUCCIÓN.....	15
1.1 Revisión de antecedentes	15
1.2 Planteamiento del problema y justificación	16
1.3 Objetivos	18
1.3.1 Objetivo general	18
1.3.2 Objetivo específico	18
1.4 Alcance.....	18
1.5 Justificación.....	19
CAPÍTULO II	20
MARCO TEÓRICO	20
2.1 Parálisis cerebral	20
2.1.1 Etiología de la parálisis cerebral	21
2.1.2 Intervenciones terapéuticas.....	23
2.1.3 Logopedia para niños con parálisis cerebral.....	23
2.1.4 Sistemas alternativos y aumentativos de comunicación (SAAC).....	24
2.1.4.1 Clasificación de los SAAC.....	24
2.2 Sistema Visual Humano (HVS)	26
2.2.1 Atención Visual	26
2.2.2 Atención Visual y movimiento ocular	26

2.3	Eye Tracking	27
2.3.1	Tipos de Eye Trackers	28
2.3.2	Perfil del participante	30
2.3.3	Selección de un Eye Tracker	30
2.4	Sistemas de Visión Artificial.....	31
2.4.1	Etapas de la visión artificial	32
2.4.1.1	Adquisición de la imagen	32
2.4.1.2	Procesamiento de la imagen.....	34
2.4.1.3	Segmentación de la imagen	35
2.4.1.4	Descripción.....	36
2.4.1.5	Reconocimiento	36
2.4.1.6	Interpretación	37
2.4.2	Detección facial.....	37
2.4.3	Campos de aplicación	38
2.4.4	Tarjetas de adquisición y procesamiento de datos	39
2.4.4.1	NVIDIA Jetson nano.....	39
2.4.4.2	Raspberry PI	42
2.4.4.3	Arduino.....	43
2.4.5	Lenguaje de programación Python aplicado en Visión por Computador	44
CAPÍTULO III		46
METODOLOGÍA		46
3.1	Diseño conceptual	46
3.1.1	Requerimiento de usuario	47
3.1.2	Diseño del sistema.....	47
3.1.3	Matriz Morfológica	48
3.1.4	Método de evaluación.....	49
3.1.4.1	Planteamiento de alternativas de solución	49
3.1.4.2	Ponderación de criterios.....	51
3.1.4.2	Ponderación de alternativas	52

3.2	Modelo del Sistema.....	55
3.3	Instalación de Librerías.....	55
3.3.1	Anaconda.....	55
3.3.2	Librería OpenCV.....	56
3.3.3	Librería Numpy.....	57
3.3.4	Librería Matplotlib.....	57
3.4	Algoritmo del sistema.....	57
3.5	Desarrollo del algoritmo del sistema.....	58
3.5.1	Adquisición de imagen.....	59
3.5.2	Detección de rostro.....	60
3.5.3	Procesamiento del video.....	60
3.5.4	Coordenadas del centro de la imagen.....	60
	CAPÍTULO IV.....	62
	PRUEBAS Y ANÁLISIS DE RESULTADOS.....	62
5.1	Especificaciones del sistema a diseñar.....	62
5.1.1	Funcionamiento.....	62
5.2	Implementación del sistema embebido.....	63
5.3	Análisis y valoración del algoritmo.....	64
5.3.1	Prueba de funcionamiento.....	64
5.4	Ejecución del algoritmo.....	68
	CAPÍTULO V.....	71
	CONCLUSIONES Y RECOMENDACIONES.....	71
	Conclusiones.....	71
	Recomendaciones.....	72
	REFERENCIAS.....	73
	ANEXOS.....	80
	Anexo 1. Entrevista al especialista.....	80
	Anexo 2. Scrip del Algoritmo.....	82

ÍNDICE DE TABLAS

Tabla 1: Procesos de la visión por computador	31
Tabla 2: Técnicas de segmentación de imagen.....	35
Tabla 3: Matriz Morfológica.....	48
Tabla 4: Especificaciones de la alternativa 1	49
Tabla 5: Especificaciones de la alternativa 2.....	50
Tabla 6: Especificaciones de la alternativa 3.....	50
Tabla 7: Ponderación de criterios	51
Tabla 8: Ponderación de las alternativas para barato.....	52
Tabla 9: Ponderación de las alternativas para la precisión	52
Tabla 10: Ponderación de las alternativas para el procesamiento	53
Tabla 11: Ponderación de las alternativas para la calidad de imagen.....	53
Tabla 12: Ponderación de las alternativas para el control del entorno	54
Tabla 13: Resultados de las ponderaciones de las alternativas.....	54
Tabla 14: Tabla de Abreviaturas para la matriz de confusión	65
Tabla 15: Matriz de Observación.....	66
Tabla 16: Tabla de Resultados de los Indicadores de Rendimiento	68

ÍNDICE DE FIGURAS

Figura 1- 1. Visión Artificial y sus conceptos.	16
Figura 2- 1. Descripción topográfica en parálisis cerebral y áreas de cerebro involucradas	21
Figura 2- 2. Escala de valoración de grado de dependencia y funcionalidad en la parálisis cerebral.....	22
Figura 2- 3. Eye Tacker Tobii TX 300.	28
Figura 2- 4. Diferentes tipos de configuración de Eye Tracking.	29
Figura 2- 5. Cubo de color RGB.....	33
Figura 2- 6. Modelo HSV.	34
Figura 2- 7. Monitoreo facial que emplea Visión por Computador.....	38
Figura 2- 8. NVIDIA Jetson Nano.....	40
Figura 2- 9. Especificaciones técnicas de NVIDIA Jetson Nano.	41
Figura 2- 10. Raspberry PI.....	42
Figura 2- 11. GPIO Raspberry PI	42
Figura 2- 12. Placa Arduino.....	44
Figura 3- 1. Fases de cumplimiento de objetivos.	46
Figura 3- 2. Diagrama funcional.....	47
Figura 3- 3. Anaconda Navigator.	56

Figura 3- 4. Diagrama de flujo de la solución.	58
Figura 3- 5. Spyder IDE.....	59
Figura 3- 6. Coordenadas del centro de la retina del ojo.	61
Figura 4- 1. Elementos para la implementación del algoritmo.....	64
Figura 4- 2. Matriz de Confusión.....	65
Figura 4- 3. Comprobación del funcionamiento del algoritmo en la dirección izquierda	69
Figura 4- 4. Comprobación del funcionamiento del algoritmo en la dirección del centro	69
Figura 4- 5. Comprobación del funcionamiento del algoritmo en la dirección de la derecha	70

CAPÍTULO I

INTRODUCCIÓN

1.1 Revisión de antecedentes

La Inteligencia Artificial (IA) es el centro del desarrollo de muchos sectores donde se adoptan nuevas tecnologías de la información. Esta se remonta muchos años atrás, pero es claro la importancia que tiene hoy en día en el desarrollo de máquinas inteligentes dotadas de capacidades de aprendizaje, razonamiento y adaptación (Barredo Arrieta et al., 2020).

Las técnicas de Inteligencia artificial como redes neuronales fueron ampliamente usadas desde la década de 1940, sin embargo, las técnicas de visión por computadora aparecieron en la década de 1960 y principios de 1970. Estas técnicas se basan en la imitación del sistema visual humano (Kakani et al., 2020). En el área de visión por computadora, generalmente se trabaja en imágenes con detalles finos, estas son empleadas en aplicaciones de detección de objetos y la clasificación de estos en función de las imágenes (Li et al., 2022).

La visión por computadora tiene aplicaciones en diversos campos, como aprendizaje automático, reconocimiento de patrones, reconstrucción 3D, seguimiento de objetos, etc. (Kakani et al., 2020), como se puede observar en la Figura 1- 1, ilustra la disposición del campo de la visión artificial y sus ramas interrelacionadas.

Figura 1- 1.

Visión Artificial y sus conceptos.



Nota: Este gráfico esta tomado de (Kakani et al., 2020)

1.2 Planteamiento del problema y justificación

La parálisis cerebral es una de las discapacidades físicas más comunes en niños a nivel mundial, en los últimos años se han realizado desarrollos en el diagnóstico, prevención y el tratamiento temprano, ayudando en la incidencia, el pronóstico y la respuesta al tratamiento(Novak et al., 2020). Con base a los resultados obtenidos de los diferentes estudios realizados se afirma que los tratamientos de la parálisis cerebral se han extendido por todo el mundo, lo cual permite a los médicos y familiares la posibilidad de realizar intervenciones más modernas, seguras y efectivas (Novak et al., 2013).

En Latinoamérica no existe un programa de control epidemiológico para la evaluación de pacientes con parálisis cerebral, y los estudios realizados son escasos, aun así, se puede realizar una estimación en los países de la región. En Ecuador, en un estudio retrospectivo realizado en 127 niños diagnosticados con parálisis cerebral infantil (PCI), se determinó que la principal causa de la enfermedad fue la asfixia perinatal (77,2%), seguido de factores postnatales (13,4%), prenatales (6,3%) y genéticas (malformaciones) en un 3,1% (Diaz et al., 2019).

En base a estos datos, es esencial analizar la calidad de vida que llevan los infantes, en el proceso de enseñanza aprendizaje con esta discapacidad. Hoy en día la tecnología ha creado distintas aplicaciones de enseñanza en niños, como aplicaciones móviles que utilizan distintas técnicas como gamificación (Toasa et al., 2019), sistemas educativos inteligentes basado en la Web (Peredo et al., 2011), aplicaciones para mejorar las actitudes de los padres, basado en vídeos (Romantika et al., 2020), con un valor monetario excesivos, lo cual no es un producto accesible para personas con bajos recursos económicos. No obstante, las aplicaciones anteriormente mencionadas, no constan de un nivel gradual de Eye Tracking, de acuerdo con el porcentaje de PCI y la dinámica ocular que tenga el infante, ya que es algo más complejo al nivel patológico y pedagógico, teniendo en cuenta que los factores del diseño, elementos audiovisuales, entre otros, juegan un papel muy importante; ya que ayudaría al desarrollo integral de los infantes.

De esta forma se pretende dar una solución proponiendo el desarrollo de un sistema capaz de contribuir en el apoyo en la terapia de lenguaje a niños con parálisis cerebral.

1.3 Objetivos

1.3.1 *Objetivo general*

Desarrollar un algoritmo de visión artificial para detectar la pupila humana y seleccionar imágenes en niños con parálisis cerebral.

1.3.2 *Objetivo específico*

- Identificar los requerimientos para el desarrollo de la propuesta.
- Diseñar un algoritmo de visión artificial para la asistencia en el proceso de terapia de lenguaje.
- Implementar el algoritmo.
- Validar el algoritmo.

1.4 Alcance

En el presente proyecto se tiene como propósito el desarrollo de un algoritmo de visión artificial utilizando Eye Tracking basándose en tres direcciones, derecha, izquierda y centro, que permita mejorar el aprendizaje y que sirva como herramienta de ayuda en la terapia de lenguaje a niños con parálisis cerebral.

En la elaboración del prototipo se usará tecnología *Open Source* con software y hardware libre. Para el desarrollo técnico de software y hardware se ha dividido en varios módulos los cuales constituyen la arquitectura del sistema, entre los cuales están: módulo de presentación, módulo de comunicación y módulo de procesamiento. Finalmente, para validar los resultados obtenidos se realizará con los terapeutas altamente capacitados en el área de terapia de lenguaje.

1.5 Justificación

La visión por computador es un área de la visión artificial que se encuentra en constante desarrollo. Mediante el presente trabajo de grado se propone un algoritmo de visión artificial utilizando Eye Tracking en un sistema embebido que ayude en el aprendizaje de niños con parálisis cerebral.

Ecuador es un país en vías de desarrollo, donde la implementación de los sistemas de visión artificial contribuye a mejorar la calidad de vida de las personas y su entorno, es así como, la implementación de herramientas tecnológicas en la educación ayuda a encontrar otra forma de enseñanza. En este contexto es necesario el desarrollo de un dispositivo que permita mejorar el proceso de aprendizaje en niños con parálisis cerebral. La falta de tecnología de apoyo en el área de la discapacidad neuromotora da la validación para el desarrollo del dispositivo embebido, el cual permitirá ayudar en la terapia de lenguaje en niños con parálisis cerebral.

El desarrollo de este proyecto tiene la finalidad de ayudar con el proceso de aprendizaje y el desarrollo intelectual de los pacientes, así como a estimular sus relaciones sociales.

CAPÍTULO II

MARCO TEÓRICO

En este capítulo se describe la base teórica del trabajo de titulación, además de aspectos importantes para el diseño del algoritmo de visión artificial, junto con la descripción de la tarjeta de adquisición y el procesamiento de la imagen, aspectos fundamentales para el desarrollo del algoritmo.

2.1 Parálisis cerebral

La parálisis cerebral es un grupo de trastornos en el desarrollo de las funciones motoras y posturales (Upadhyay et al., 2020), como resultado de una lesión no progresiva permanente en el sistema nervioso central (Palisano et al., 1997), que provocan limitaciones de la actividad, que se atribuyen a alteraciones que pasaron en el cerebro fetal o en el desarrollo infantil (Graham et al., 2016).

La parálisis cerebral en ocasiones está relacionada con la encefalopatía hipóxica esquemática (EHI) cuando el cerebro del bebé tiene problemas de flujo sanguíneo adecuado secundario a un evento hipóxico-isquémico en el periodo prenatal, intraparto o posnatal (Kozin & Gardner, 2019). Es la causa más común de discapacidad en la infancia, con una prevalencia de 2.1 casos por cada 1000 nacidos vivos (Novak et al., 2017). Las alteraciones motoras de la parálisis cerebral suelen estar acompañados de alteraciones de la percepción, sensación, comunicación, cognición y conducta, dificultad musculoesquelético y epilepsia (Graham et al., 2016).

Durante mucho tiempo se creía que la principal etiología de la parálisis cerebral era el trabajo de parto prolongado, sin embargo, a pesar de las mejoras en las instalaciones médicas la

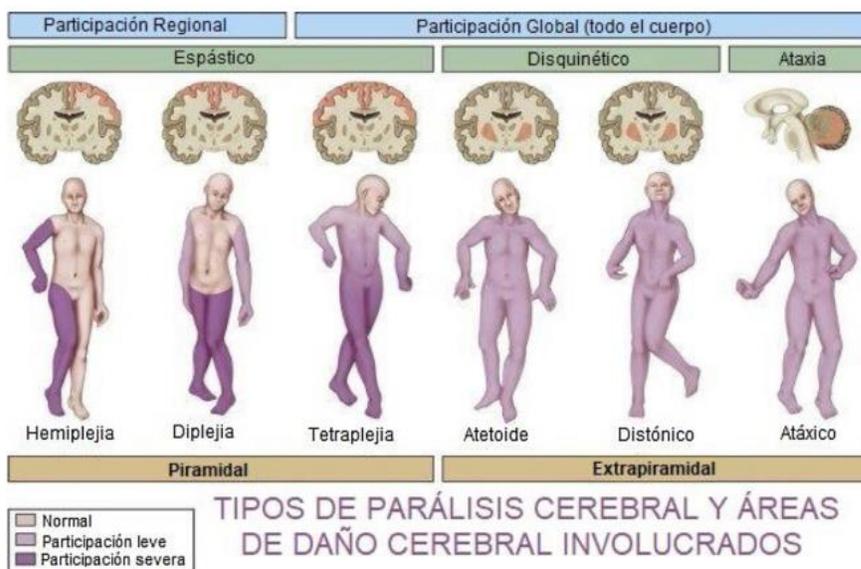
incidencia de parálisis cerebral no disminuyó como se esperaba, diferentes estudios han demostrado que la incidencia de defectos en el desarrollo cerebral se relaciona con la semana gestacional (Vitrikas et al., 2020).

2.1.1 Etiología de la parálisis cerebral

Los factores etiológicos vinculados con la parálisis cerebral varían según la edad y clasificación clínica, estos indican el número de extremidades involucradas y el tipo de espasticidad. El diagnóstico para la parálisis cerebral es una combinación de signos clínicos y neurológicos, el diagnóstico generalmente es realizado entre los 12 y 24 meses de edad. Según la afectación de las extremidades puede ser hemiplejía, diplejía, tetraplejía; Según el trastorno del tono dominante se utilizan términos como espasticidad, discinesia y ataxia (Novak et al., 2017), (Vitrikas et al., 2020).

Figura 2- 1

Descripción topográfica en parálisis cerebral y áreas de cerebro involucradas



Nota: Este gráfico está tomado de (Peláez Cantero et al., 2021).

La parálisis cerebral hemipléjica tiene afectación de un lado (brazo y pierna) y mantiene preservación del otro lado, la diplejía afecta predominantemente a los miembros inferiores afectando la musculatura, la cual se encuentra espástica y prácticamente imposibilitada de flexionar, los niños con tetraplejía tienen afectación de las cuatro extremidades, aunque la gravedad puede variar como se puede observar en la Figura 2- 1

Descripción topográfica en parálisis cerebral (Kozin & Gardner, 2019).

La parálisis cerebral hemipléjica es el tipo más común con una tasa de 70-80%, en ocasiones representa los efectos de un accidente cerebrovascular isquémico perinatal (Vitrikas et al., 2020), la diplejía espástica, suele acompañar de pérdida de sustancia periventricular, la parálisis cerebral atáxica es la forma más rara de parálisis cerebral con una tasa de 5-10%, en ocasiones indica la presencia de una malformación cerebelosa (Graham et al., 2016).

Figura 2- 2

Escala de valoración de grado de dependencia y funcionalidad en la parálisis cerebral.

	GMFCS	MACS	CFCS
I	Marcha sin restricciones, pero limitada para las actividades motoras más demandantes	Utiliza objetos con facilidad y con éxito	Envía y recibe información eficaz y eficientemente con personas conocidas y desconocidas
II	Marcha con restricciones	Utiliza la mayoría de los objetos, pero con una calidad del movimiento reducida o con menor velocidad en la realización	Envía y recibe información de forma eficaz, pero de forma más lenta con conocidos y desconocidos
III	Marcha con asistencia técnica que utiliza con las manos	Utiliza los objetos con dificultad necesitando ayuda en la preparación o realización de actividades	Envía y recibe información de manera eficaz solo con personas conocidas
IV	Limitaciones para la automovilidad, puede utilizar una asistencia técnica controlada por el paciente	Manipulación limitada a una selección de objetos en un entorno adaptado	Envía o recibe información de manera inconsistente con personas conocidas
V	El paciente necesita ser transportado por otra persona en una silla de ruedas	No maneja objetos o tiene limitaciones muy importantes para realizar actividades simples	Casi nunca envía o recibe información

Nota: Este gráfico está tomado de (Peláez Cantero et al., 2021).

La comprensión de las vías que conducen a la parálisis cerebral se ha logrado gracias a estudios epidemiológicos, neuroimagen y post mortem y modelos animales. En el 90% de los casos la parálisis cerebral es producto de procesos destructivos que dañan el tejido cerebral y no de anomalías en el desarrollo del cerebro (Graham et al., 2016).

Para evaluar de forma objetiva el nivel de dependencia y la funcionalidad de los niños con parálisis cerebral se utilizan diferentes medios como se puede observar en la Figura 2- 2

2.1.2 Intervenciones terapéuticas.

Distintas intervenciones terapéuticas se utilizan en el manejo de la parálisis cerebral. Algunas de las técnicas convencionales incluyen la fisioterapia tradicional y terapia ocupacional, el tratamiento del neurodesarrollo, hipoterapia, etc, técnicas recientes utilizan estimulación eléctrica (Upadhyay et al., 2020), las intervenciones terapéuticas más utilizadas son la terapia ocupacional y la fisioterapia tradicional las que han demostrado ser bastante beneficiosas en el tratamiento para la parálisis cerebral.

2.1.3 Logopedia para niños con parálisis cerebral.

El lenguaje en un niño con parálisis cerebral se ve gravemente afectado y este tiende a reducir la longitud de sus frases durante el acto de la comunicación, esto ocasiona reducción del vocabulario y dificultar para adquirir y manejar conceptos básicos (Boquete-Jamardo & Fernández-Méndez, 2015).

El logopeda tiene la función de establecer y potenciar la comunicación, así como en prevenir, rehabilitar y reeducar posibles alteraciones en las funciones neurovegetativas (Boquete-Jamardo & Fernández-Méndez, 2015). En definitiva, lo que se pretende en personas con parálisis

cerebral es facilitar la expresión de sentimientos, permitiéndoles influir y participar en los contextos en los que se desenvuelve (Perlstein et al., 1946).

El enfoque primordial de la intervención logopédica es mejorar el método de comunicación, así como mejorar los aspectos que intervienen en la comunicación. El logopeda a través de los juegos didácticos potencia diferentes áreas del conocimiento, permitiendo desarrollo del lenguaje (Labrada Batchelor et al., 2019).

Los juegos didácticos permiten el desarrollo del lenguaje infantil, desarrollan la sonoridad y expresividad del lenguaje, al mismo tiempo generan sentimientos de afecto, amistad y compañerismo lo cual fomenta otra serie de actividades como la observación y comprensión de estímulos, esto crea en ellos independencia y autonomía (Labrada Batchelor et al., 2019).

2.1.4 Sistemas alternativos y aumentativos de comunicación (SAAC)

También se pueden emplear los Sistemas Aumentativos y Alternativos de Comunicación (SAAC), estos son un conjunto de recursos o estrategias enfocados en facilitar la comprensión y la expresión del lenguaje en personas con dificultad en el habla o la escritura (Montero González, 2003).

2.1.4.1 Clasificación de los SAAC

Los SAAC pueden clasificarse según precisen o no de algún soporte técnico, y estos pueden ser sistemas de comunicación con ayuda y sistemas de comunicación sin ayuda.

Sistemas de comunicación sin ayuda.

Es un sistema que posibilita el intercambiar información usando el cuerpo sin utilizar algún tipo de ayuda o herramienta, un ejemplo son las personas con discapacidad auditiva las

cuales se comunican usando gestos o lenguaje de señas. El aprendizaje de estos sistemas son la Dactilología, la Lengua de Signos, el Bimodal y la Palabra Complementada (Montero González, 2003).

Sistemas de comunicación con ayuda.

Este sistema de comunicación permite el uso de ayudas o herramientas de comunicación. Las herramientas que se usan frecuentemente son papel y cartulina, una carpeta o un libro los cuales contiene palabras, letras o dibujos con los que se pueden elaborar tableros o plantillas de comunicación (Montero González, 2003).

Dentro de este Sistema de Comunicación se puede encontrar el Sistema BLISS, el cual usa símbolos gráfico-visuales que representan significados. Sistema SPC, el cual es un sistema no oral que utiliza símbolos pictográficos sencillos y el Sistema Minspeak, el cual es un sistema diseñado para comunicadores de voz (Montero González, 2003).

Pictogramas

Un pictograma es una representación gráfica de una actividad o una acción (Subcomisión de accesibilidad cognitiva de CERMI, 2019).

Los pictogramas son un sistema de comunicación no oral que permiten lograr una accesibilidad cognitiva y es un sistema alternativo y aumentativo de comunicación que permite brindar terapia en niños con parálisis cerebral (Montero González, 2003).

2.2 Sistema Visual Humano (HVS)

2.2.1 Atención Visual

Ante de abordar el tema del Eye Tracking, es importante determinar la motivación para el registro de los movimientos del ojo humano, los ojos atraen una porción particular del campo de visión para que se pueda observar con gran detalle lo que sea que se encuentre en el campo central de la mirada (Duchowski, 2017).

Para ello es importante comprender los procesos de atención de un observador, y si el movimiento de los ojos revela algo sobre el proceso cognitivo interno conocido como la atención visual (Duchowski, 2017).

2.2.2 Atención Visual y movimiento ocular

La atención visual podría comportarse en un proceso cíclico compuesto por los siguientes pasos:

1. Al percibir un estímulo como una imagen, la escena se observa en paralelo a través de la visión periférica y a baja resolución. En este punto las características relevantes pueden aparecer en el campo de visión, dirigiendo la atención para una inspección más detallada (Duchowski, 2017).
2. La atención se apaga y se desconecta del punto focal y los ojos se reposicionan a la región que atrajo la atención (Duchowski, 2017).
3. La terminar el movimiento de los ojos, el punto focal se dirige a la región de interés y ahora se dedica a percibir características en alta resolución (Duchowski, 2017).

2.3 Eye Tracking

Un Eye Trackers es un dispositivo que registra y graba el moviendo de los ojos, el Eye Tracking comúnmente se refiere a la técnica empleada para registrar y medir el movimiento ocular (Tobii, 2010).

Los Eye Trackers modernos utilizan tecnología basada en video para medir el punto de mirada (POR) (Carter & Luke, 2020), utilizando una o dos cámaras y una fuente de luz infrarroja que ilumina los ojos y permite captura y registrar el reflejo de la luz infrarroja en la córnea o retina del ojo, calculando el punto de mirada comparando la ubicación de la mirada en la cámara con la posición del reflejo de la córnea (Valtakari et al., 2021).

El Eye Tracking es una técnica bien establecida y muy utilizada en la investigación de la cognición humana, fue utilizada por primera vez a principios de XX mediante la utilización de lentes de contacto con un puntero adjunto (Clay et al., 2019).

El desarrollo de la tecnología ha permitido una evolución rápida en los sistemas de Eye Tracking los cuales son más fáciles de operar y son menos intrusivos para los sujetos de prueba, esto ha permitido que el Eye Tracking sea utilizado en diferentes proyectos comerciales y de investigación (Tobii, 2010).

Los datos del Eye Tracking, pueden verse afectados negativamente por una variedad de factores, la precisión del rastreador ocular, las respuestas pupilares y las respuestas reflejas a los cambios en la intensidad de luz son algunos de esos factores que pueden causar problemas en la interpretación de los datos (Tangsgaard Hvelplund, 2014).

2.3.1 Tipos de Eye Trackers

La clasificación de los Eye Trackers se realiza a partir de la precisión donde mira el usuario, y la precisión se mide en los grados de ángulo visual como se puede observar en la **Figura 2- 4**

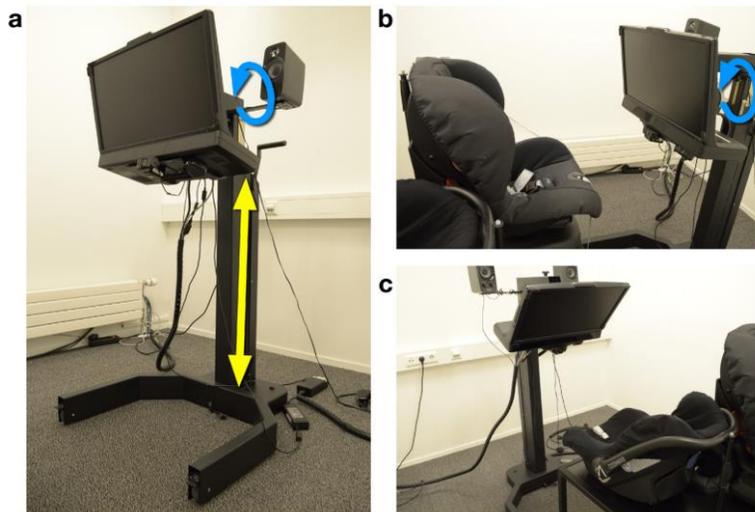
Diferentes tipos de configuración de Eye Tracking.. La imprecisión suele estar registrada entre 0.5 y 1 grados, lo que corresponde aproximadamente a 0.5 a 1 centímetros, para decidir qué tipo de Eye Trackers es adecuado para el objetivo de la investigación es importante considerar el grado de invasión y la velocidad de registro del Eye Trakers medida por la frecuencia de muestras de mirada por segundo (Hz) (Duchowski, 2017).

Los Eye Trackers de escritorio o remotos, son el tipo preferido en la investigación, las cámaras se integran en una caja separada que se coloca frente al monitor o está conectada a él, la serie RED de SMI, X60/120, TX300 de Tobii mostrado en la Figura 2- 3 y EyeLink 1000 de SR Research son un ejemplo de rastreadores menos invasivos como se puede observar en la **Figura 2- 4**

Diferentes tipos de configuración de Eye Tracking. (Tangsgaard Hvelplund, 2014).

Figura 2- 3

Eye Tacker Tobii TX 300.

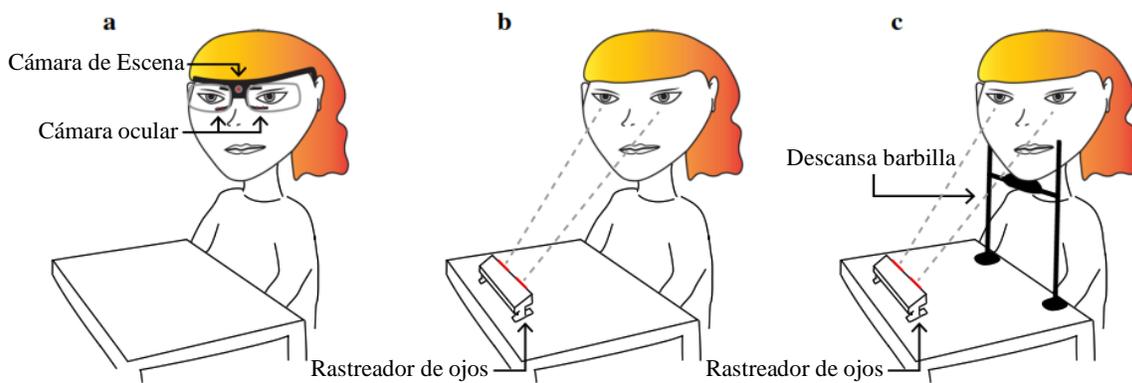


Nota: El gráfico representa la configuración Eye Tracker Tobii TX 300. a. Configuración del Eye Tracker (Tobii TX 300), el rastreador ocular se puede inclinar completamente vertical u horizontal. b. Instalación para bebés o niños pequeños más grandes. c. Configuración para bebés pequeños Este gráfico esta tomado de (Valtakari et al., 2021).

Los sistemas montados en la cabeza son otro tipo de Eye trackers como el iView X HED de SMI y EyeLink II de SR Research, otro tipo de dispositivos son las gafas Eye Tracking como las gafas SMI y las gafas Tobii, estas permiten el libre movimiento de la cabeza y la imprecisión suele ser de 0.5 grados, aunque este tipo de dispositivo son más invasivos que los dispositivos remotos (Tangsgaard Hvelplund, 2014). Sin embargo, los rastreadores montados en la cabeza pueden registrar las actividades que realiza y no solo las que ocurren en el monitor de la computadora.

Figura 2- 4

Diferentes tipos de configuración de Eye Tracking.



Nota: En ese gráfico se puede observar a. Configuración de Eye Tracking portátil. b. Configuración de un Eye Tracking remoto. c. Configuración restringida para la cabeza que

contiene un Eye Tracking remoto y una mentonera. Este gráfico esta tomado de (Valtakari et al., 2021).

2.3.2 Perfil del participante

El Eye Tracking tiene ciertas consideraciones para su correcto funcionamiento, los ojos y el uso de accesorios como anteojos o lentes de contacto pueden influir en la recopilación de los datos del seguimiento ocular, así también el uso de pestañas muy largas y el rímel pesado pueden provocar un impacto negativo en la recolección de datos, ya que estos pueden provocar obstrucciones entre el rastreador ocular y los ojos del participante ocasionando que el registro de los datos no sea bueno, para evitar estos problemas, puede ser necesario instruir al participante para que no use lentes de contacto y anteojos no tengan mucho marco (Tangsgaard Hvelplund, 2014), en todo caso el Eye Tracking debe estar correctamente calibrado, para poder identificar posibles problemas antes de que se comience a registrar los datos.

2.3.3 Selección de un Eye Tracker

Existe una gran variedad de Eye Tracker en el mercado, los fabricantes más destacados son Tobii y SR Research, es importante considerar la aplicación que va a tener el Eye Tracker antes de decidirse por un Eye Tracker, ya que los diferentes sistemas son más adecuados para ciertos usos que para otros (Carter & Luke, 2020).

Las características importantes en la elección del tipo de Eye Tracker comprende la libertad de movimiento para el participante, calidad de los datos del Eye Tracker y facilidad de análisis de los datos, otros aspectos importantes son la frecuencia de muestreo y si la mirada es de forma monocular o binocular (Valtakari et al., 2021).

2.4 Sistemas de Visión Artificial

El área de la Inteligencia Artificial enfocada en imágenes se llama visión por computadora, y es un campo que trata de imitar los ojos humanos.

La visión artificial o visión por computadora utiliza un conjunto de métodos que le permite adquirir, analizar y procesar imágenes convirtiéndolas en información para posterior uso en el computador (Russo et al., 2018).

Tabla 1

Procesos de la visión por computador.

Procesos	Nivel de Visión	Entrada	Salida	Área
1. Captura	Bajo	Imagen	Imagen	Procesamiento de imágenes
2. Preprocesamiento				
3. Segmentación	Medio	Imagen	Grupo de píxeles en bruto (objetos o regiones)	Análisis de Imágenes
4. Descripción		Objetos o regiones	Información cuantitativa de los objetos o regiones	
5. Reconocimiento (clasificación)		Información cuantitativa	Objetos clasificados en categorías	
6. Interpretación	Alto	Objetos clasificados en categorías	Comprensión de la escena	Visión por computador

Nota: Esta tabla esta tomado de (García & Caranqui, 2015).

La constante evolución tecnológica ha permitido el desarrollo de nuevos algoritmo y aplicaciones. La visión por computadora nace del intento de proveer a las máquinas un sistema de visión para poder automatizar un proceso mediante el tratamiento de imágenes (García & Caranqui, 2015).

La visión por computadora es un conjunto de proceso que permite la obtención, caracterización e interpretación de información de imágenes (García & Caranqui, 2015).

*Estos procesos pueden ser subdivididos en seis como se puede observar en la Tabla 1. **Tabla 1***

Procesos de la visión por computador.

2.4.1 Etapas de la visión artificial

El diseño del sistema de visión por computador se basa en las etapas primordiales para el proceso de visión por computador, tomando en consideración el cumplimiento de las necesidades del proyecto.

2.4.1.1 Adquisición de la imagen

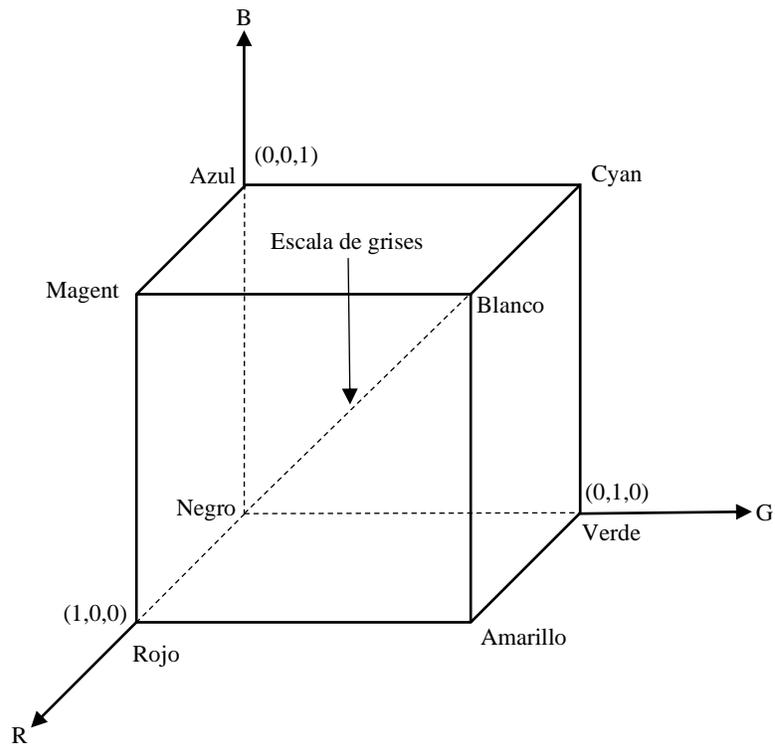
Es el proceso en el cual se obtiene la imagen, a través de un dispositivo como cámara de video, escáner, telescopio, satélite, etc (García & Caranqui, 2015), (Alvear-Puertas et al., 2017).

*Las imágenes son comúnmente digitalizadas en formato RGB, los valores posibles para cada entrada es de 255. Los colores son representados en coordenadas cartesianas y cada color es un vector de origen y la diagonal principal corresponde a la escala de grises como se puede observar en la **Figura 2- 5***

Cubo de color RGB. (Figueredo-Ávila & Ballesteros-Ricaurte, 2016).

Figura 2- 5

Cubo de color RGB.



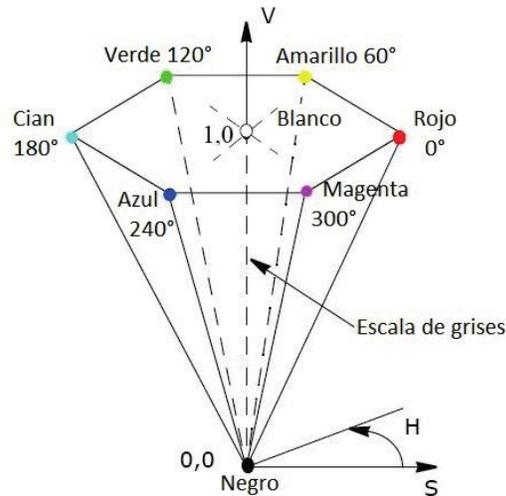
Nota: Este gráfico está tomado de (Figueredo-Ávila & Ballesteros-Ricaurte, 2016).

*Los modelos perceptuales se concentran en cómo se observa el objeto mediante la visión, entre los modelos se encuentra el HSV (Hue, Saturation, Value, o Tono, Saturación, Valor) el cual se obtiene deformando el modelo RGB de tal forma que se convierte en una pirámide hexagonal, donde el eje vertical es la brillantez o valor (V); el eje horizontal, la saturación (S), y el ángulo de la proyección horizontal, el croma (H) como se puede observar en la **Figura 2- 6***

Modelo HSV. (Figueredo-Ávila & Ballesteros-Ricaurte, 2016).

Figura 2- 6

Modelo HSV.



Nota: Este gráfico está tomado de (Figueredo-Ávila & Ballesteros-Ricaurte, 2016).

2.4.1.2 Procesamiento de la imagen

El procesamiento de la imagen utiliza técnicas para la reducción del ruido, mejoramiento del contraste, nitidez de la imagen, realce de detalles o características de la imagen (García & Caranqui, 2015).

En la visión por computador es importante el tratamiento de la imagen con la finalidad de facilitar procesos posteriores (Alvear-Puertas et al., 2017). Dentro del tratamiento de la imagen se encuentran:

- Suavizar-reducir el ruido

Las imágenes usualmente tienen ruido el cual puede afectar la calidad, los cuales pueden ser Gaussiano, Impulsional o multiplicativo, las técnicas más usadas para la

reducción del ruido son Media aritmética, Media ponderada, Media Gaussiana y Media (Figueredo-Ávila & Ballesteros-Ricaurte, 2016).

- Realce del contraste

El contraste en una imagen consiste en la diferencia entre los tonos que la componen y puede ser alto o bajo. Para el tratamiento del contraste se utiliza la modificación del histograma, este proceso se conoce como Ecuilización del histograma (Figueredo-Ávila & Ballesteros-Ricaurte, 2016).

2.4.1.3 Segmentación de la imagen

Es el proceso por el cual se divide la imagen en objetos o regiones que sean importante para el estudio, esta etapa es muy crítica ya que de ella depende el éxito o el fracaso de la aplicación (García & Caranqui, 2015).

Algunos de las técnicas más usadas se pueden observar en la Tabla 2.

Tabla 2

Técnicas de segmentación de imagen.

Técnica	Descripción
Detección de bordes	La técnica de detección de bordes busca diferenciar el objeto de interés de la imagen de fondo, localizando los píxeles del borde.
Marr-Hildreth	Este modelo especifica que la imagen inicialmente se convulsiona con una función gaussiana de dos dimensiones, para posteriormente computar el laplaciano de imagen convulsionada, y así poder hallar los píxeles de los bordes.
Canny	El algoritmo de Canny estima la obtención de los bordes sin omisión alguna, los localiza con la menor distancia posible entre

	los objetos que conforman el borde además evita identificar los múltiples píxeles que existen en el borde cuando solo exista un borde. Este algoritmo proporciona una reducción en el ruido, por lo que la imagen resultante suele verse más borrosa que la original.
Shen-Castan (ISEF)	A diferencia del algoritmo Canny, que aproxima un filtro óptimo por la derivada gaussiana, el algoritmo de Shen-Castan, usa un filtro óptimo directamente, acelerando el proceso de búsqueda.
Laplace	Este modelo está basado en las segundas derivadas o laplaciano de una Gaussiana (LoG), lo que permite estimar con mayor precisión la localización de la orilla. En algunos casos es posible que, por la aplicación de las segundas derivadas, se presente la aparición de dobles orillas.
Sobel	Este modelo no solo permite detectar las orillas o bordes en una imagen, sino que a la vez aplica suavizamiento para que la existencia de ruido no obstruya la detección de los bordes.

Nota: Esta tabla esta tomado de (Figueredo-Ávila & Ballesteros-Ricaurte, 2016).

2.4.1.4 Descripción

Es el proceso en el que se obtiene características relevantes para el estudio, estas características pueden ser externas como el perímetro, la forma, excentricidad; o internas como el centro de gravedad, área, textura, color entre otros (García & Caranqui, 2015).

2.4.1.5 Reconocimiento

Es el proceso mediante el cual se clasifican los objetos detectados dentro de categorías utilizando los descriptores del proceso anterior. Los objetos que presenten descriptores semejantes se agrupan automáticamente dentro de una misma clase o categoría, para esto se

utilizan técnicas como métodos estadísticos avanzados, redes neuronales artificiales, algoritmos entre otros (García & Caranqui, 2015).

2.4.1.6 Interpretación

Este proceso es el que se encarga de dar sentido a las clases o categorías de objetos reconocidos, trata de emular la visión humana y utiliza técnicas cognitivas para la toma de decisiones, esto depende del campo de aplicación (García & Caranqui, 2015).

2.4.2 *Detección facial*

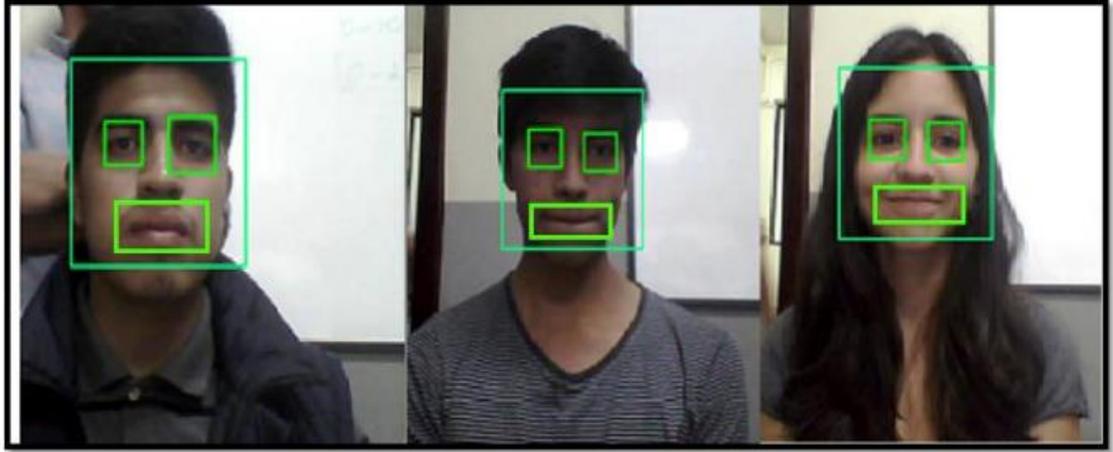
Los sistemas de detección facial en los últimos años son ampliamente utilizados, en la actualidad existen sofisticadas herramientas aplicadas en distintas áreas como el entretenimiento, tarjetas inteligentes, seguridad, etc (Alvear-Puertas et al., 2017).

La implementación de un software de detección facial no ha sido tarea fácil y ha requerido una amplia investigación que ayude a mejorar el proceso (Alvear-Puertas et al., 2017). En la Figura 2- 7, se puede observar una prueba de detección facial. La implementación de los algoritmos necesarios para los sistemas de detección facial se ha subdividido en las siguientes áreas:

- Detección de rostro
- Seguimiento
- Alineamiento
- Extracción de rasgos
- Entrenamiento
- Identificación
- Análisis de expresión facial.

Figura 2- 7

Monitoreo facial que emplea Visión por Computador.



Nota: Este gráfico esta tomado de (Alvear-Puertas et al., 2017).

2.4.3 Campos de aplicación

La visión artificial puede ser aplicada en numerosas situaciones para dar solución a determinados problemas, aunque a veces por la complejidad del caso y las limitaciones propias de la disciplina no resulta ser la solución más adecuada (García & Caranqui, 2015), (Kakani et al., 2020). Aun así, las aplicaciones crecen considerablemente.

Los campos de aplicación de la visión artificial son:

- Agricultura
- Astronomía
- Biología
- Geología
- Medicina
- Modelado y visualización 3D

- Reconocimiento y clasificación
- Robótica
- Seguridad.

2.4.4 Tarjetas de adquisición y procesamiento de datos

Se denomina computadoras de placa única o tarjetas de adquisición a las que se encuentran construidas en una placa de circuito único y contienen microprocesadores, memoria, entradas/salidas y otras características, el uso de los sistemas informáticos de placa única se ha vuelto muy frecuente, ya que puede combinarse con otros campos tecnológicos (Suzen et al., 2020).

El desarrollo tecnológico de las computadoras de placa única y la inserción de CPU y GPU ha potenciado su uso en el campo de la inteligencia artificial. Las computadoras de placa única son la mejor opción para las aplicaciones de aprendizaje profundo donde el objetivo son los sistemas autónomos y móviles (Suzen et al., 2020).

2.4.4.1 NVIDIA Jetson nano

NVIDIA Jetson Nano es una computadora de IA pequeña y poderosa como se puede observar en la

Figura 2- 8, que otorga el rendimiento necesario para ejecutar cargas de trabajo de IA avanzada como clasificación de imágenes, detección de objetos, segmentación y procesamiento de voz, con un tamaño pequeño y de bajo costo (salih & Basman Gh., 2020).

Figura 2- 8

NVIDIA Jetson Nano.



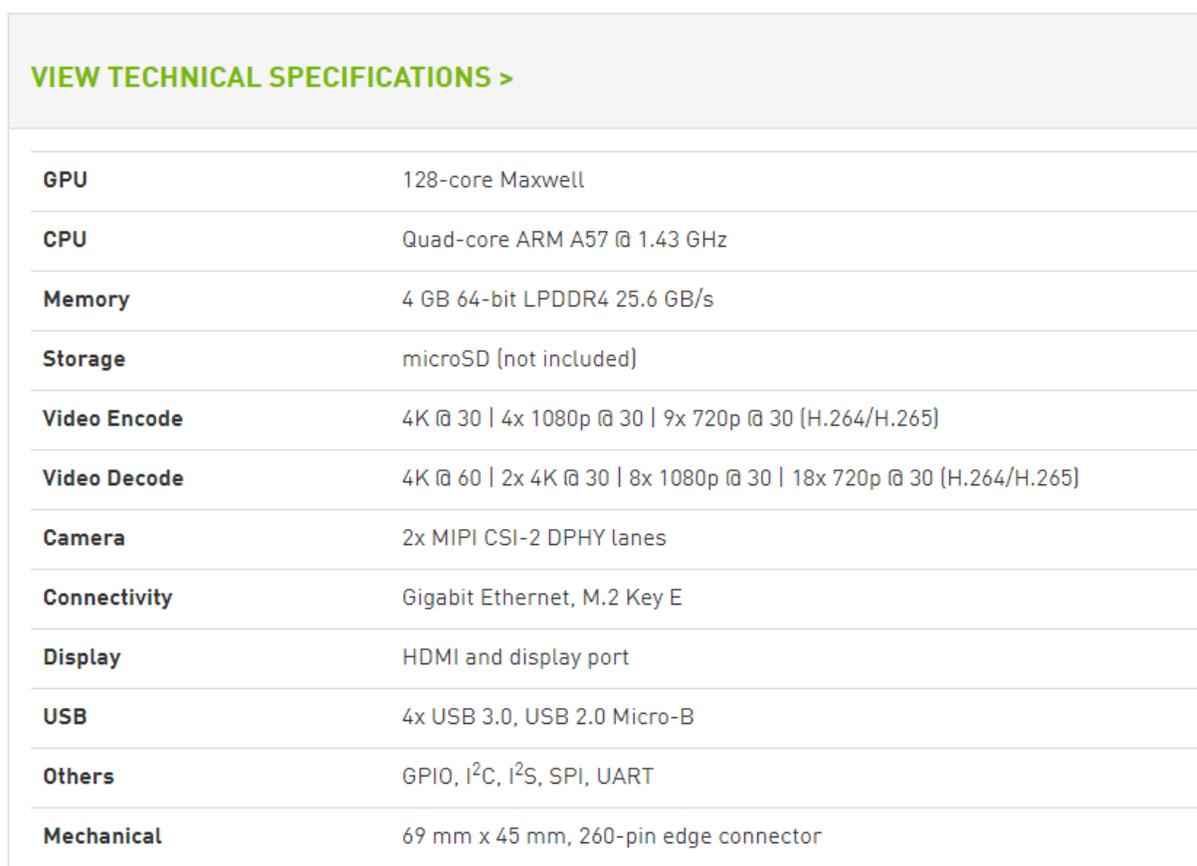
Nota: Este gráfico esta tomado de (NVIDIA, 2019).

NVIDIA Jetson Nano fue diseñado para el desarrollo de inteligencia artificial, aprendizaje automático y tecnología robótica, su arquitectura está basada en un sistema Erista de un solo chip con 4 núcleos CortexA57, una frecuencia de operación de 1,43 GHz y un sistema gráfico con 128

núcleos CUDA en arquitectura Maxwell como se puede observar en las especificaciones técnicas en la Figura 2- 9.

Figura 2- 9

Especificaciones técnicas de NVIDIA Jetson Nano.



VIEW TECHNICAL SPECIFICATIONS >	
GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (not included)
Video Encode	4K @ 30 4x 1080p @ 30 9x 720p @ 30 (H.264/H.265)
Video Decode	4K @ 60 2x 4K @ 30 8x 1080p @ 30 18x 720p @ 30 (H.264/H.265)
Camera	2x MIPI CSI-2 DPHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI and display port
USB	4x USB 3.0, USB 2.0 Micro-B
Others	GPIO, I ² C, I ² S, SPI, UART
Mechanical	69 mm x 45 mm, 260-pin edge connector

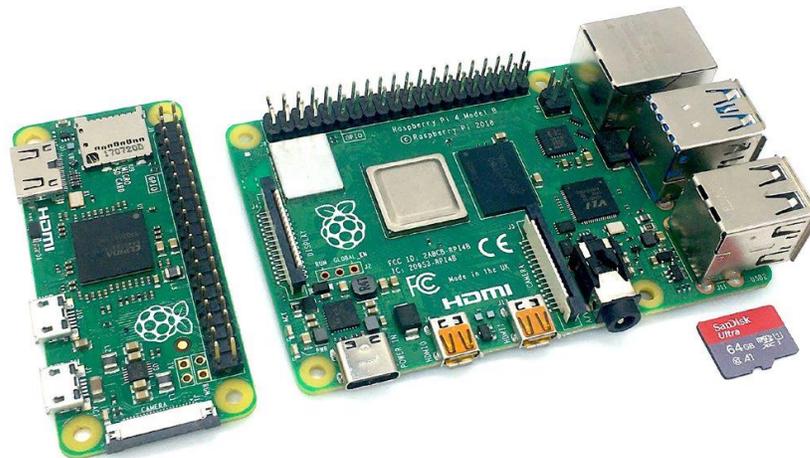
Nota: Este gráfico esta tomado de (NVIDIA, 2019).

2.4.4.2 Raspberry PI

La placa Raspberry PI que se observa en la Figura 2- 10 tiene todas las funcionalidades de una computadora estándar, presenta un sistema en un chip, además contiene una CPU (unidad central de procesamiento) y una GPU (unidad de procesamiento de gráficos) en el chip, memoria integrada y una entrada de alimentación de 5V CC (Jolles, 2021).

Figura 2- 10

Raspberry PI.



Nota: En el gráfico se observa a la izquierda la Raspberry PI zero y a la derecha 4B con una tarjeta microSD. Este gráfico esta tomado de (Jolles, 2021).

Figura 2- 11

GPIO Raspberry PI

Raspberry Pi B+/2 B
Rev 1 (J8)

3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
ID_SD	27	28	ID_SC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

La mayoría de los modelos tienen un puerto para conectar una cámara, conexión Ethernet y conectividad inalámbrica (Wi-Fi y Bluetooth) así como una gran variedad de pines de entrada/salida de uso general (GPIO) como se puede observar en la Figura 2- 11, los que se pueden utilizar para la comunicación de una gran variedad de dispositivos electrónicos (Jolles, 2021).

Las placas de expansión especiales se pueden conectar a los pines GPIO, son denominadas hardware adjunto en la parte superior (HAT), permiten insertar funciones adicionales como identificación de radio frecuencia (RFID), controladores de motor, administración de energía y grabación de audio de alta calidad (Jolles, 2021).

2.4.4.3 Arduino

Arduino es una placa de microcontrolador basada en ATmega328, posee 14 pines de entrada/salida digital, donde 6 de estos pines pueden ser usados como salidas PWM, 6 pueden ser entradas analógicas, posee un oscilador de cristal de 16 MHz, una conexión USB, un conector

de alimentación, un botón de reinicio y un encabezado ICSP como se observa en la Figura 2- 12 (Arduino.cc, 2020).

Figura 2- 12

Placa Arduino



Nota: Este gráfico esta tomado de (Kondaveeti et al., 2021).

Arduino es una placa microcontroladora de código abierto, además también tiene su propio lenguaje de programación basado en lenguaje C, posee un cargador en forma de USB lo que facilita la programación del microcontrolador, mientras que la mayoría de las otras placas de microcontroladores requieren de un circuito de cargador por separado (Susilo et al., 2021).

Arduino es una placa microcontroladora relativamente económica en relación con otras placas microcontroladoras, además es muy versátil el cual puede ejecutarse en sistemas operativos Windows, Macintosh OSX y Linux (Kim et al., 2020).

2.4.5 Lenguaje de programación Python aplicado en Visión por Computador

Python se ha convertido en el lenguaje de programación por excelencia cuando se trata de manejar grandes conjuntos de datos o ajustar modelos de predicción precisos. Python es un

lenguaje de programación muy utilizado en las áreas de investigación mencionadas anteriormente (Blank & Deb, 2020).

Python es un lenguaje de programación de código abierto y de alto nivel, multiplataforma que se centra en la legibilidad del código. Estas características hacen de Python una herramienta adecuada para el desarrollo de proyectos industriales y de investigaciones de alto nivel bastantes complejas (Blank & Deb, 2020).

Teniendo claro lo que es el lenguaje de programación de Python, es importante conocer las librerías más importantes de visión por computadora como lo es OpenCV.

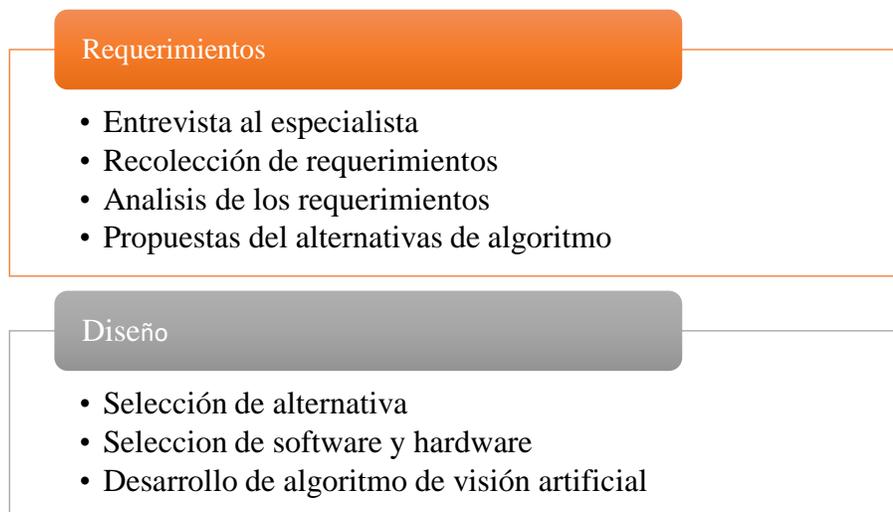
CAPÍTULO III

METODOLOGÍA

En este capítulo se describe la metodología aplicada para el diseño y desarrollo de un algoritmo de visión artificial enfocado en apoyar la terapia de lenguaje en niños con parálisis cerebral. Además, se describen paso a paso los procesos considerados para la selección de software y dispositivos necesarios para lograr el cumplimiento de los objetivos como se observa en la Figura 3- 1.

Figura 3- 1

Fases de cumplimiento de objetivos. ¡Error! No se encuentra el origen de la referencia.



3.1 Diseño conceptual

En el diseño conceptual se definieron los requerimientos del usuario, del sistema de visión artificial y los dispositivos necesarios para el sistema de visión artificial con procesamiento de imagen.

3.1.1 *Requerimiento de usuario*

En la fase de requerimientos del usuario se evalúan las diferentes ideas para el desarrollo del sistema de visión artificial por medio de la entrevista realizada a la especialista de terapia de lenguaje.

El análisis de los criterios de diseño prioritarios del sistema de visión artificial se realiza mediante la utilización de la metodología de diseño conceptual.

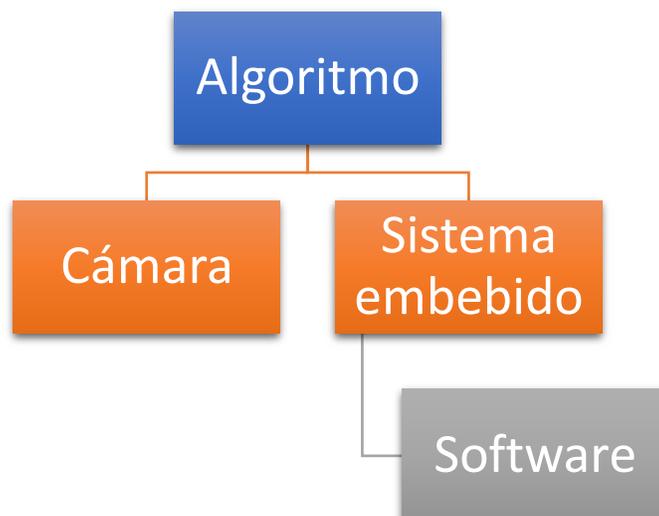
3.1.2 *Diseño del sistema*

Una vez definidas las necesidades del cliente se procede con la fase analítica donde se propone un sistema de visión artificial y los elementos que lo componen de acuerdo con los requerimientos obtenidos por parte de la especialista y que cumpla con las necesidades establecidas.

En la Figura 3- 2, se presenta el diagrama de bloque mediante cuadro conceptual donde se muestra el proceso que se debe llevar a cabo para un sistema de visión artificial con adquisición de imagen.

Figura 3- 2

Diagrama funcional.



3.1.3 Matriz Morfológica

La Matriz Morfológica es una técnica combinatoria de ideas creativas, permite descomponer un problema en sus elementos básicos con el fin de construir una matriz que relacione estos elementos para obtener diferentes ideas.

Tabla 3

Matriz Morfológica.

<p>Tarjeta de adquisición</p>	 <p>Arduino</p>	 <p>Raspberry</p>	 <p>Jetson</p>
<p>Cámara</p>	 <p>RPI Cámara</p>	 <p>Webcam</p>	 <p>Cámara VGA</p>
<p>Software</p>	 <p>Python</p>	 <p>C++</p>	 <p>C</p>

La Tabla 3, presenta una matriz que contiene diferentes dispositivos básicos, proceso y elementos necesarios para un sistema de visión artificial con procesamiento de imagen.

La matriz contiene:

- Cámara. – diferentes tipos de cámaras.
- Sistema embebido. – Procesamiento de la imagen
- Software. – Lenguajes de programación disponibles para el sistema embebido

3.1.4 Método de evaluación

El método de evaluación permite generar las diferentes combinaciones viables, contemplando todas las variaciones de la matriz morfológica, de tal forma que generen soluciones que puedan ser evaluadas para determinar la mejor opción.

3.1.4.1 Planteamiento de alternativas de solución

De acuerdo con las necesidades del usuario y contemplando los requerimientos para el desarrollo del sistema, se proponen las siguientes alternativas de solución que satisfacen estas necesidades.

Tabla 4

Especificaciones de la alternativa 1

Elemento	Criterio	Componente
Software	Medio nivel	Lenguaje C
Sistema embebido	Procesamiento	Arduino
Cámara	Calidad de imagen	Módulo de Arduino

Alternativa 1: El uso de un sistema embebido mediante Arduino, que se conecta con un módulo de cámara VGA y la programación se realiza mediante lenguaje C como se puede observar en la Tabla 4.

Alternativa 2: El uso de un sistema embebido mediante un Raspberry, que se conecta con una RPI cámara y la programación se realiza con lenguaje C++ como se puede observar en la Tabla 5.

Tabla 5

Especificaciones de la alternativa 2

Elemento	Criterio	Componente
Software	Medio nivel	Lenguaje C++
Sistema embebido	Procesamiento	Raspberry
Cámara	Calidad de imagen	Raspberry

Alternativa 3: El uso de un sistema embebido mediante una Jetson, que se conecta con una Webcam y la programación se realiza con lenguaje Python como se observa en la Tabla 6.

Tabla 6

Especificaciones de la alternativa 3

Elemento	Criterio	Componente
Software	Alto nivel	Python
Sistema embebido	Procesamiento	Jetson
Cámara	Calidad de imagen	Webcam

3.1.4.2 Ponderación de criterios

En la ponderación de criterios se realiza un análisis cuantitativo de los criterios más importante por parte del usuario en el desarrollo del sistema.

- Barato
- Precisión
- Procesamiento
- Calidad imagen
- Control de entorno

Como se observa en la Tabla 7, se realizó la ponderación de los criterios de acuerdo con el nivel de importancia de cada criterio respecto a los requerimientos del sistema de visión artificial.

Tabla 7

Ponderación de criterios

	Barato	Precisión	Procesamiento	Imagen	Entorno	$\sum +1$	Ponderación
Barato	-	0.5	0	0	0.5	2	0.13
Precisión	0.5	-	0.5	0.5	1	3.5	0.23
Procesamiento	1	0.5	-	0.5	1	4	0.27
Imagen	1	0.5	0.5	-	1	4	0.27
Entorno	0.5	0	0	0	-	1.5	0.1
Total						15	1

3.1.4.2 Ponderación de alternativas

En la ponderación de alternativas se realiza una comparación de las alternativas de solución con relación a cada criterio.

En la Tabla 8, se realizó la valoración de las alternativas con el criterio barato.

Tabla 8

Ponderación de las alternativas para barato.

Barato	Alternativa 1	Alternativa 2	Alternativa 3	\sum_{+1}	Ponderación
Alternativa 1	-	1	1	3	0.5
Alternativa 2	0	-	0.5	1.5	0.25
Alternativa 3	0	0.5	-	1.5	0.25
Total				6	1

En la Tabla 9, se presenta la valoración de las alternativas con el criterio precisión.

Tabla 9

Ponderación de las alternativas para la precisión.

Precisión	Alternativa 1	Alternativa 2	Alternativa 3	\sum_{+1}	Ponderación
Alternativa 1	-	0	0	1	0.17
Alternativa 2	1	-	0	2	0.33
Alternativa 3	1	1	-	3	0.5
Total				6	1

En la Tabla 10, se realizó una valoración de las alternativas con el criterio procesamiento.

Tabla 10

Ponderación de las alternativas para el procesamiento.

Procesamiento	Alternativa 1	Alternativa 2	Alternativa 3	$\sum +1$	Ponderación
Alternativa 1	-	0	0	1	0.17
Alternativa 2	1	-	0	2	0.33
Alternativa 3	1	1	-	3	0.5
Total				6	1

En la Tabla 11, se muestra la valoración de las alternativas con el criterio calidad de imagen.

Tabla 11

Ponderación de las alternativas para la calidad de imagen.

Imagen	Alternativa 1	Alternativa 2	Alternativa 3	$\sum +1$	Ponderación
Alternativa 1	-	0	0	1	0.17
Alternativa 2	1	-	0	2	0.33
Alternativa 3	1	1	-	3	0.5
Total				6	1

En la Tabla 12, se describe la valoración de las alternativas con el criterio control de entorno.

Tabla 12

Ponderación de las alternativas para el control del entorno.

Entorno	Alternativa 1	Alternativa 2	Alternativa 3	$\sum +1$	Ponderación
Alternativa 1	-	0	0	1	0.17
Alternativa 2	1	-	0.5	2.5	0.41
Alternativa 3	1	0.5	-	2.5	0.42
Total				6	1

Luego de realizar las matrices de los criterios, se procede a comparar las alternativas de solución con cada criterio para determinar cuál es la alternativa optima como se observa en la Tabla 133.

Tabla 13

Resultados de las ponderaciones de las alternativas.

	Barato	Precisión	Procesamiento	Imagen	Entorno	Total	Ponderación
Alternativa 1	0.13*0.5	0.23*0.17	0.27*0.17	0.27*0.17	0.1*0.17	0.23	0,23
Alternativa 2	0.13*0.25	0.23*0.33	0.27*0.33	0.27*0.33	0.1*0.41	0.32	0,32
Alternativa 3	0.13*0.25	0.23*0.5	0.27*0.5	0.27*0.5	0.1*0.42	0.45	0.45

Luego de realizar la evaluación de fiabilidad de criterios ponderados, se observó que la mejor solución es la alternativa 3, debido a que cumple con los requerimientos necesarios del sistema.

3.2 Modelo del Sistema

De acuerdo con el hardware seleccionado, se ha decidido utilizar como software de visión artificial para la detección de la pupila humana el lenguaje de programación Python, debido a que Python es un lenguaje de código abierto, muy fácil de utilizar y cuenta con una extensa biblioteca de herramientas.

Sin embargo, para el desarrollo de los algoritmos en Python, utilizaremos el software Anaconda, el cual es un software gratuito que proporciona un conjunto de herramientas diseñadas para la investigación, Anaconda brinda acceso a entornos que permiten codificar en Python, estos entornos son conocidos como entornos de desarrollo integrado (IDE), son plataformas que facilitan el desarrollo del código (Rolon-Mérette et al., 2020)

3.3 Instalación de Librerías.

Para el desarrollo del algoritmo es importante la instalación de ciertas librerías.

A continuación, se detallan las librerías utilizadas en el desarrollo del algoritmo de visión artificial.

3.3.1 Anaconda

Para la programación y desarrollo del algoritmo es necesario realizarlo en un entorno de desarrollo integrado (IDE), como se había mencionado anteriormente se va a utilizar Anaconda

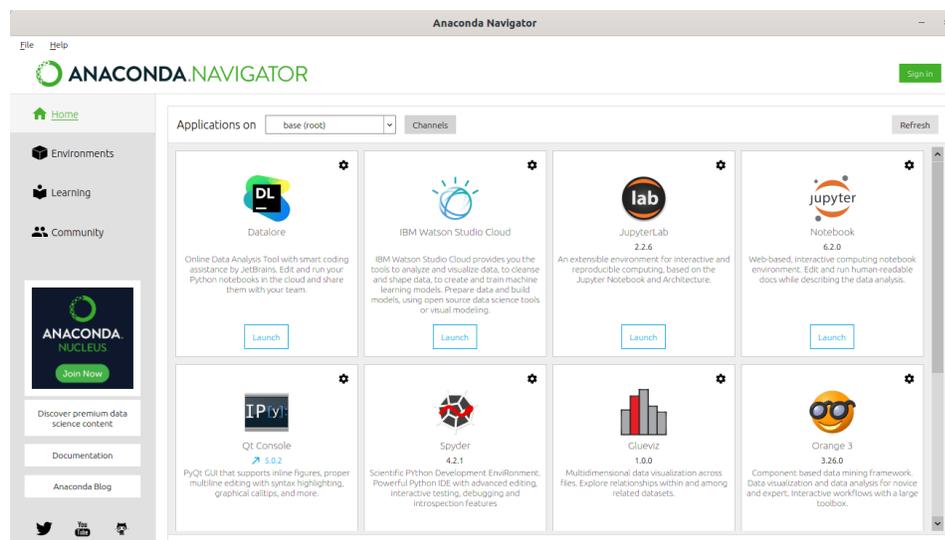
debido a que es un software de codificación gratuito fácil de utilizar que facilita enormemente el desarrollo del código.

Para la instalación del software solo se debe de dirigir a la página oficial y realizar la descarga dependiendo del sistema operativo utilizado (Rolon-Mérette et al., 2020).

Una vez realizada la instalación del software se crea un entorno y se instalan las librerías para el procesamiento de imagen como se observa en la Figura 3- 3.

Figura 3- 3

Anaconda Navigator.



Nota: Este gráfico fue tomado de (Rolon-Mérette et al., 2020)

3.3.2 Librería OpenCV

OpenCV es una librería de código abierto, software de aprendizaje automático y visión artificial, fue diseñada para tener una infraestructura común en las aplicaciones de visión por computadora (About OpenCV, 2022).

Para instalar la librería se debe escribir el siguiente código:

- `pip install opencv-python`

3.3.3 Librería Numpy

Numpy es una librería que tiene como objetivo habilitar la computación numérica en Python, proporciona un objeto de matriz multidimensional, posee operaciones rápidas en matrices, lógica, simulación aleatoria, etc. Numpy siempre será un software de código abierto, de uso gratuito para todos (Numpy, 2022).

Para instalar la librería se debe escribir el siguiente código:

- `pip install numpy`

3.3.4 Librería Matplotlib

Matplotlib es un paquete de gráfico 2D empleado en Python, es utilizado para el desarrollo de aplicaciones, generación de imágenes de alta calidad y entornos interactivos en todas las plataformas. Matplotlib puede utilizarse en scripts de Python, shell de Python e IPython (Hunter, 2007).

Para instalar la librería se debe escribir el siguiente código:

- `pip install matplotlib`

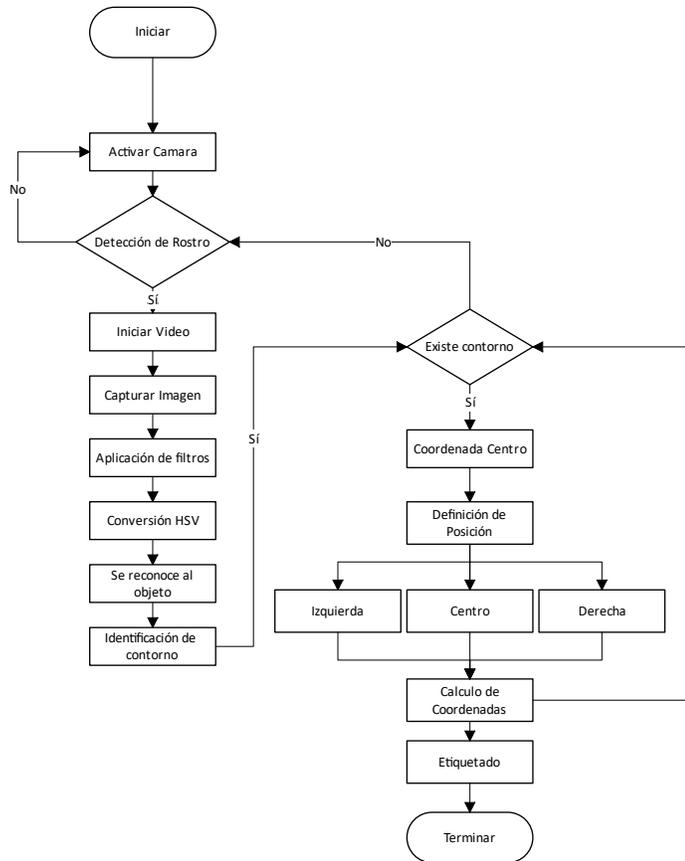
3.4 Algoritmo del sistema

En el desarrollo del algoritmo de visión artificial, se debe tener en cuenta que tiene como objetivo comunicarse con el entorno, por lo tanto, debe tener fácil acceso para el ingreso de datos y la salida de resultado. En la programación del algoritmo se utilizó el lenguaje de programación

Python, debido a que es un lenguaje versátil, de fácil diseño y rápida lectura, en la Figura 3- 4 se puede observar el diseño del algoritmo.

Figura 3- 4

Diagrama de flujo de la solución.

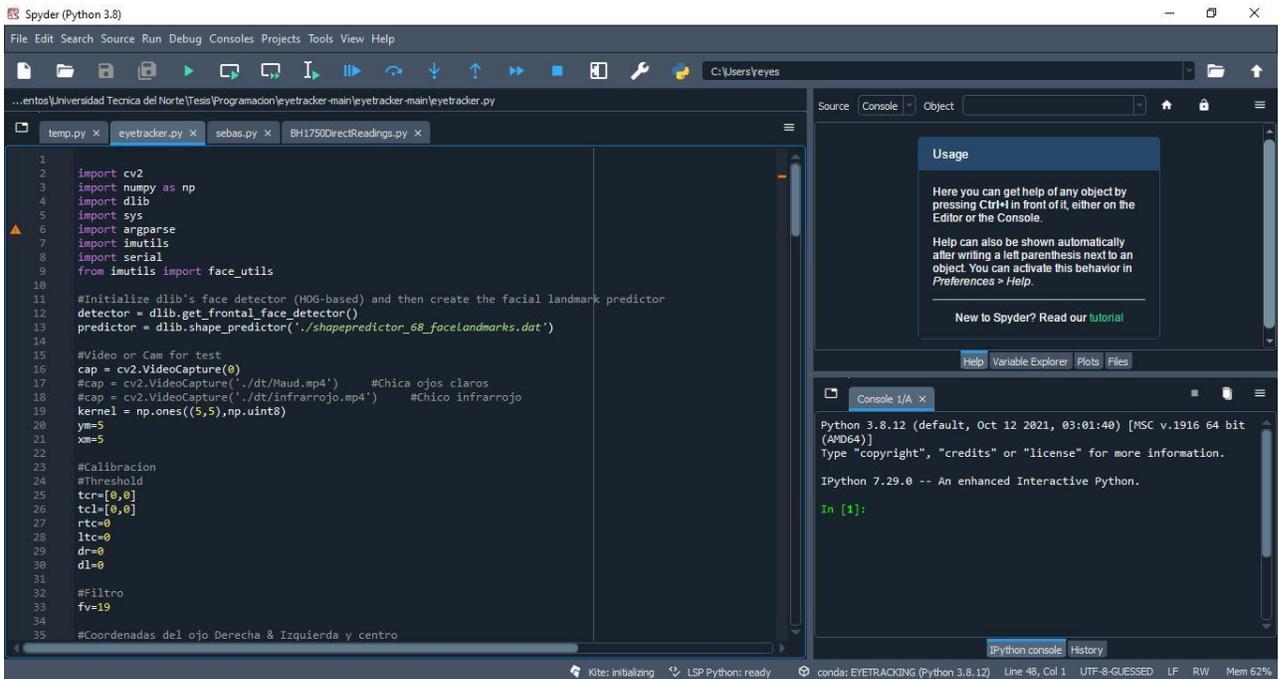


3.5 Desarrollo del algoritmo del sistema

Para el desarrollo del algoritmo fue necesario instalar el editor de código Spyder IDE como se puede observar en la Figura 3- 5, debido a que es compatible con Python y se puede ejecutar en Ubuntu, el sistema operativo de la Jetson Nano.

Figura 3- 5

Spyder IDE.



El desarrollo del algoritmo se realizó con el diagrama de flujo descrito en la Figura 3- 4, se utilizó el lenguaje de programación, el editor de código y librerías de visión artificial ya definidas. Para ello se dividió en los siguientes procesos:

3.5.1 Adquisición de imagen

El sistema empieza con la obtención de los datos de entrada, se utilizó una webcam para adquirir los videos. Ya que el sistema tiene que pre evaluar los datos de entrada para determinar el contorno del rostro y la posición del ojo, para ellos la persona tiene que estar a una distancia de 30 cm de la cámara.

3.5.2 *Detección de rostro*

Para la detección del rostro se utiliza una biblioteca contenedora de código abierto. Se crea un objeto del modelo de detección de rostro en el video. Esto permite delimitar el área de detección y ayuda a minimizar errores en la detección del contorno del ojo.

3.5.3 *Procesamiento del video*

En el procesamiento del video, se utilizaron filtros en el tratamiento de la imagen, ayudando a eliminar ruidos, esto permite realizar la detección de contornos de forma más eficiente.

3.5.4 *Coordenadas del centro de la imagen*

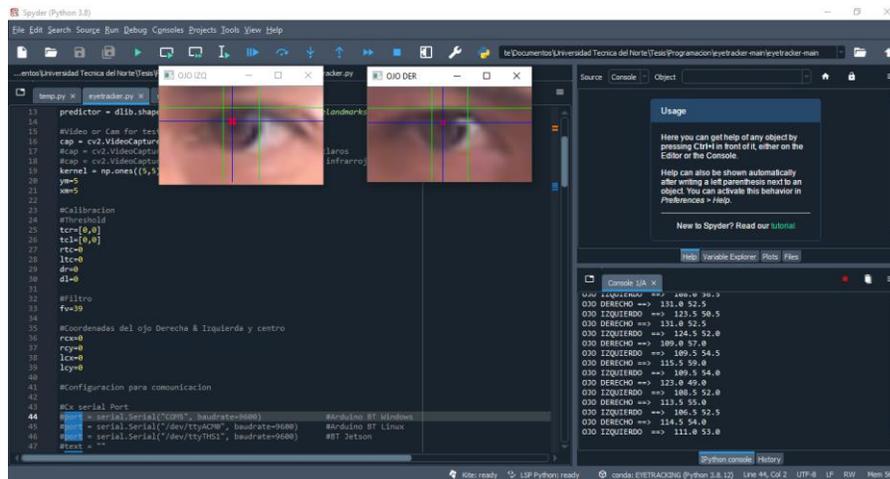
Determinado el contorno del ojo, se define las coordenadas centrales y se realiza una comparación para determinar la posición del ojo utilizando coordenadas determinadas en un mallado como se puede observar en la Figura 3- 6, lo cual permite determinar en qué dirección se está realizando el movimiento ocular.

Realizado el proceso de diseño del algoritmo se procedió a realizar pruebas en su funcionamiento, tomando en cuenta las siguientes condiciones iniciales.

- Ambiente sin control del entorno
- Cámara Logitech 720p
- Distancia de la cámara 30 cm

Figura 3- 6

Coordenadas del centro de la retina del ojo.



El algoritmo cumple con la detección del contorno del ojo, se pudo comprobar que la detección se realiza de forma correcta siempre y cuando el entorno sea controlado. Un sistema de visión artificial para su correcto funcionamiento debe de contar con un buen control del ambiente para obtener un mejor resultado.

CAPÍTULO IV

PRUEBAS Y ANÁLISIS DE RESULTADOS

Se detalla el procedimiento para la implementación y pruebas de funcionamiento del algoritmo propuesto. Así como el análisis de los resultados obtenidos en las pruebas.

5.1 Especificaciones del sistema a diseñar

Para lograr el objetivo de la detección del movimiento de la pupila humana para posteriormente enfocarlo en la terapia de lenguaje en niños con parálisis cerebral, se deben de cumplir ciertos requisitos. Por tal motivo se han determinado las características y condiciones esperadas en este trabajo:

- Determinar el contorno de la pupila
- Las coordenadas de la posición del centroide de la pupila
- El rango de movimiento en las tres direcciones, izquierda, centro y derecha
- El algoritmo debe de poder procesar un alto nivel de fotogramas por segundo (FPS)
- El algoritmo debe de tener buen equilibrio entre velocidad y precisión
- El algoritmo debe de poseer un alto nivel de procesamiento en tiempo real
- La cámara de tener una buena resolución de imagen

5.1.1 *Funcionamiento*

El funcionamiento del algoritmo es a nivel de software, sin embargo, el proceso de funcionamiento tiene nivel de hardware y algoritmo.

Nivel de hardware

En este nivel se encuentra el funcionamiento de la cámara, con la toma de imágenes casi en tiempo real es decir a 144 Hz, sin embargo, por el *stream* y el procesamiento de la imagen baja hasta unos 50 FPS. Esta información es enviada a la tarjeta embebida JETSON Nano para empezar el nivel de algoritmo.

Nivel de algoritmo

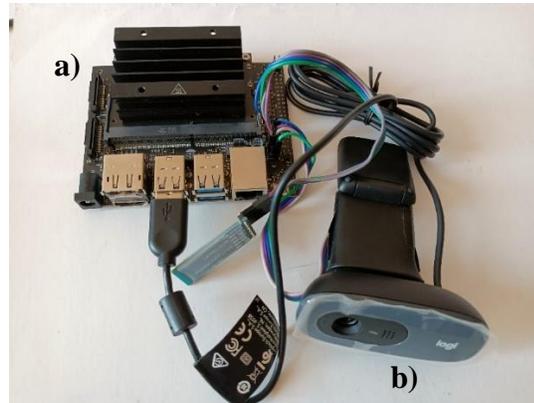
En este nivel inicia el procesamiento de la imagen, dando un tratamiento a la imagen para detectar el contorno de la pupila y determinar las coordenadas del centroide de la pupila, generando un ciclo donde se registra y compara el centroide para determinar el movimiento en las direcciones izquierda, centro y derecha.

5.2 Implementación del sistema embebido

Como se observa en la Figura 4- 1, se implementó el sistema operativo SDK de JetPack que incluye Jetson Linux, Kernel de Linux y entorno de escritorio Ubuntu, en la placa embebida Jetson Nano para la implementación del algoritmo.

Figura 4- 1

Elementos para la implementación del algoritmo.



Nota: En ese gráfico se puede observar a). La tarjeta embebida Jetson NANO. b). La Webcam Logitech C270.

5.3 Análisis y valoración del algoritmo

Una vez concluida la fase de selección y de diseño y desarrollo del algoritmo se realizó el análisis y validación del funcionamiento del algoritmo.

5.3.1 Prueba de funcionamiento

Para el análisis y evaluación de rendimiento del algoritmo de visión por computador, se utilizó la Matriz de Confusión multiclase.

Matriz de Confusión

La Matriz de Confusión es una herramienta fundamental a la hora de evaluar el funcionamiento de un algoritmo de clasificación (koldo, 2018).

En la Tabla 14, se pueden observar los términos y abreviaturas utilizados en la matriz de confusión.

Tabla 14

Abreviaturas para la Matriz de Confusión.

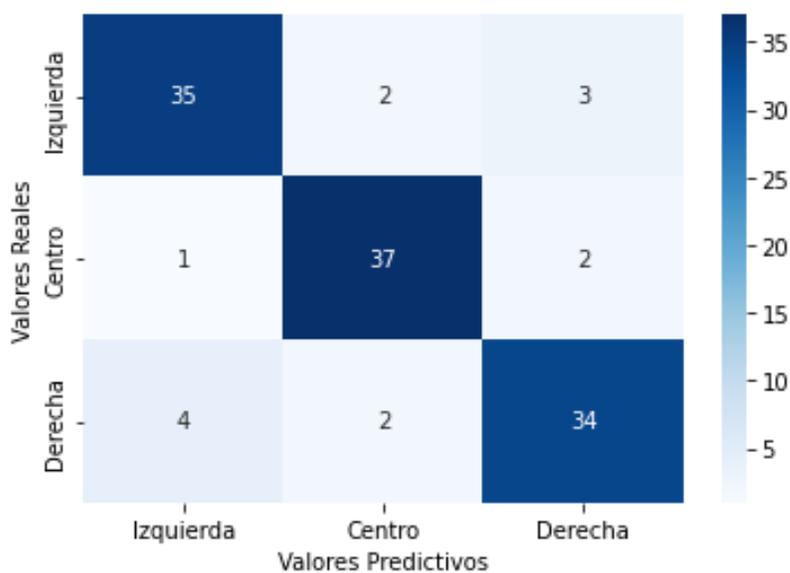
Términos	Abreviaturas	Descripción
Verdaderos positivos	TP	Predichos positivos y que realmente son positivos.
Falsos positivos	FP	Predichos positivos y que realmente son negativos.
Negativos verdaderos	NT	Predichos negativos y que realmente son negativos.
Falsos negativos	FN	Predichos negativos y que realmente son positivos.

Para la realización de la Matriz de Confusión multiclase se realizaron 120 pruebas, donde se realizaron 40 mirando a la izquierda, 40 mirando al centro y 40 mirando a la derecha.

En la Figura 4- 2, se observa el desarrollo de la Matriz de Confusión multiclase, donde las clases definidas son: izquierda, centro y derecha.

Figura 4- 2

Matriz de Confusión



En la Tabla 15 se detallan cuáles fueron los verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos.

Tabla 15

Matriz de Observación

CLASES	Medidas			
	TP	TN	FP	FN
Izquierda	35	75	5	5
Centro	37	76	4	3
Derecha	35	75	5	6

Después de determinar los cuatro estados verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, se determinaron los indicadores de rendimiento como la exactitud, precisión, sensibilidad, especificidad y F1 Score; los que determinan las propiedades básicas del algoritmo. Todos estos valores se miden en porcentaje donde el 100% significa la perfección de la propiedad.

- Precisión

La precisión es lo cerca que esta el resultado obtenido al del valor verdadero, la ecuación (1) es la predicción positiva del modelo realizado a partir de todo el conjunto de los datos, lo que determina en cuanto acierta el modelo.

$$Precisión = \frac{TP}{TP + FP} \quad (1)$$

- Exactitud

La exactitud representa el porcentaje de predicciones correctas frente al total del conjunto de datos. Para calcular la exactitud se utiliza la ecuación (2)

$$Exactitud = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

- Sensibilidad

La sensibilidad es la tasa de verdaderos positivos o TP, es el número de casos positivos bien clasificados frente al total de positivos. Para calcular la sensibilidad se utiliza la ecuación (3)

$$Sensibilidad = \frac{TP}{TP + FN} \quad (3)$$

- Especificidad

La especificidad es la tasa de verdaderos negativos o TN, es el número de casos negativos bien clasificados respecto al total de negativos. Para calcular la sensibilidad se utiliza la ecuación (4)

$$Especificidad = \frac{TN}{TN + FP} \quad (4)$$

- F1 Score

F1 score es la medida de precisión que tiene el modelo, es la relación que existe entre la precisión y sensibilidad. Para calcular la sensibilidad se utiliza la ecuación (5)

$$F1\ Score = \frac{2}{\frac{1}{precisión} + \frac{1}{sensibilidad}} \quad (5)$$

En la Tabla 16, se muestra los resultados a los indicadores de rendimientos del funcionamiento del algoritmo para cada clase evaluada.

Tabla 16

Tabla de Resultados de los Indicadores de Rendimiento

CLASES	Medidas				
	Precisión	Exactitud	Sensibilidad	Especificidad	F1 Score
Izquierda	0,88	0,92	0,88	0,94	0,88
Centro	0,90	0,94	0,93	0,95	0,92
Derecha	0,87	0,91	0,85	0,94	0,86

Los resultados obtenidos de los indicadores de rendimiento en las diferentes clases indican que el rendimiento del algoritmo en las diferentes clases evaluadas es de buena calidad.

5.4 Ejecución del algoritmo

En las Figura 4- 3 a la Figura 4- 5, se puede observar las pruebas de funcionamiento del algoritmo, logrando determinar el movimiento ocular en las direcciones izquierda, centro y derecha, lo cual indica que el algoritmo cumple con su objetivo.

Figura 4- 3

Comprobación del funcionamiento del algoritmo en la dirección izquierda

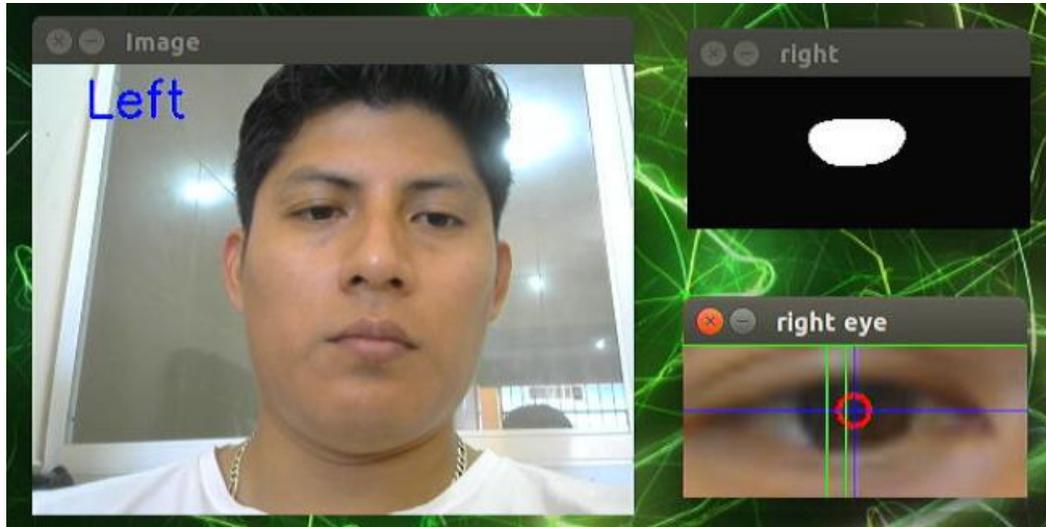


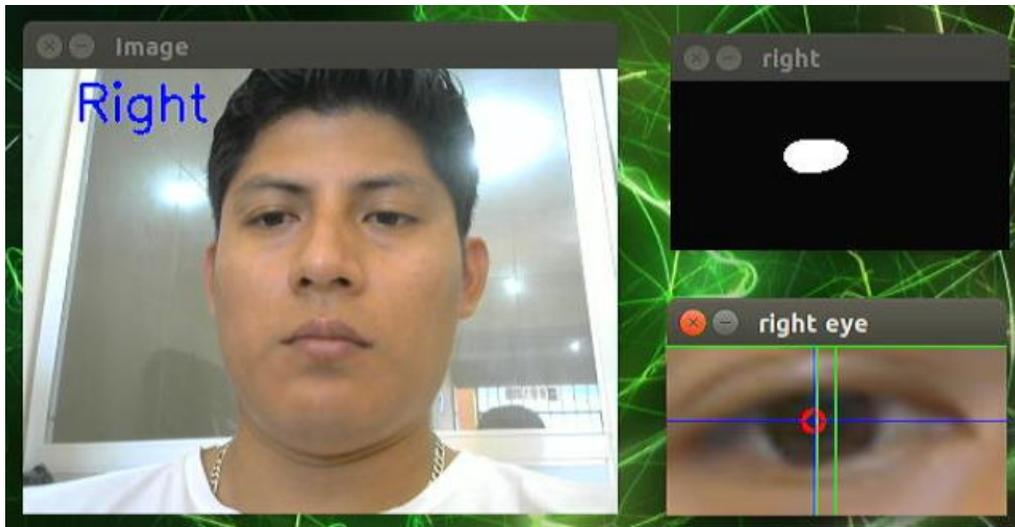
Figura 4- 4

Comprobación del funcionamiento del algoritmo en la dirección del centro



Figura 4- 5

Comprobación del funcionamiento del algoritmo en la dirección de la derecha



CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Este trabajo de visión por computadora presentó un algoritmo de Eye Tracking para en el futuro implementarlo en la terapia de lenguaje en niños con parálisis cerebral. La parálisis cerebral es un grupo de trastornos motoras y posturales, lo que ocasiona movimientos involuntarios por lo cual en el desarrollo del algoritmo se utilizaron diferentes técnicas para lograr detectar los objetos de interés.
- Se diseñó un algoritmo de visión por computadora para la detección del movimiento ocular en niños con parálisis cerebral, tomando en cuenta los requerimientos para la detección de la pupila, obteniendo un algoritmo con un 87% precisión.
- A partir de las pruebas realizadas se determinó que la tarjeta embebida Jetson Nano desarrollada para proyectos en visión por computador, es apta para la implementación del algoritmo, ya que su sistema operativo contiene especificaciones adecuadas que ayuda a tener una buena respuesta del algoritmo.
- La validez del algoritmo se determinó mediante una matriz de confusión multiclase, de la cual se obtuvo un 88% de precisión en la detección del movimiento ocular en la dirección izquierda, un 90% en la dirección del centro y un 88% en la dirección de la derecha, además de 88% de F1 Score en la dirección izquierda, un 92% en la dirección del centro y un 86% en la dirección de la derecha, por lo cual se puede concluir que el algoritmo es confiable y apto para la detección.

Recomendaciones

- En investigaciones posteriores se recomienda realizar la obtención de la imagen en un ambiente controlado para evitar el ruido, ya que esto permitiría reducir el tratamiento de la imagen y aumentar la precisión del algoritmo.
- Mejorar la calidad de la cámara de acuerdo con los FPS requeridos o al tipo ISO que se ocupa de acuerdo con el ambiente de detección.
- Se recomienda realizar una calibración externa del algoritmo de acuerdo con el entorno antes de empezar a la detección.
- Es recomendable utilizar un equipo adecuado a la cantidad de información que se va a procesar, debido a que una mayor cantidad de imágenes necesita un mayor tiempo de procesamiento.

REFERENCIAS

- About OpenCV. (2022). *OpenCV*. <https://opencv.org/about/>
- Alvear-Puertas, V., Rosero-Montalvo, P., Peluffo-Ordóñez, D., & Pijal-Rojas, J. (2017). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *Enfoque UTE*, 8(1). <https://doi.org/10.29019/enfoqueute.v8n1.121>
- Arduino.cc. (2020). Arduino Uno Rev3. *Arduino.Cc*.
- Barredo Arrieta, A., Díaz-Rodríguez, N., del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI. *Information Fusion*, 58, 82–115. <https://doi.org/10.1016/j.inffus.2019.12.012>
- Blank, J., & Deb, K. (2020). Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, 8. <https://doi.org/10.1109/ACCESS.2020.2990567>
- Boquete-Jamardo, A., & Fernández-Méndez, J. C. (2015). Eficacia de un sistema alternativo de comunicación en PC: estudio caso único. *Revista de Estudios e Investigación En Psicología y Educación*, 079–084. <https://doi.org/10.17979/reipe.2015.0.09.642>
- Carter, B. T., & Luke, S. G. (2020). Best practices in eye tracking research. *International Journal of Psychophysiology*, 155, 49–62. <https://doi.org/10.1016/j.ijpsycho.2020.05.010>
- Clay, V., König, P., & König, S. (2019). Eye tracking in virtual reality. *Journal of Eye Movement Research*, 12(1). <https://doi.org/10.16910/jemr.12.1.3>

- Díaz, E., Ignacio, C., Maroto, A., Barrionuevo, C., Moya, E., & Acosta, S. (2019). Prevalencia, factores de riesgo y características clínicas de la parálisis cerebral infantil. *Sociedad Venezolana de Farmacología Clínica y Terapéutica, Venezuela*, 777–796.
- Duchowski, A. T. (2017). Eye tracking methodology: Theory and practice: Third edition. In *Eye Tracking Methodology: Theory and Practice: Third Edition*. <https://doi.org/10.1007/978-3-319-57883-5>
- Figueredo-Ávila, G. A., & Ballesteros-Ricaurte, J. A. (2016). Identificación del estado de madurez de las frutas con redes neuronales artificiales, una revisión. *Ciencia y Agricultura*, 13(1). <https://doi.org/10.19053/01228420.4811>
- García, I., & Caranqui, V. (2015). La visión artificial y los campos de aplicación. *Tierra Infinita*, 1, 94–108. <https://revistasdigitales.upec.edu.ec/index.php/tierrainfinita/article/view/76>
- Graham, H. K., Rosenbaum, P., Paneth, N., Dan, B., Lin, J.-P., Damiano, D. L., Becher, J. G., Gaebler-Spira, D., Colver, A., Reddihough, D. S., Crompton, K. E., & Lieber, R. L. (2016). Cerebral palsy. *Nature Reviews Disease Primers*, 2. <https://doi.org/10.1038/nrdp.2015.82>
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Jolles, J. W. (2021). Broad-scale applications of the Raspberry Pi: A review and guide for biologists. In *Methods in Ecology and Evolution* (Vol. 12, Issue 9). <https://doi.org/10.1111/2041-210X.13652>
- Kakani, V., Nguyen, V. H., Kumar, B. P., Kim, H., & Pasupuleti, V. R. (2020). A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research*, 2, 100033. <https://doi.org/10.1016/j.jafr.2020.100033>

- Kim, S. M., Choi, Y., & Suh, J. (2020). Applications of the open-source hardware Arduino platform in the mining industry: A review. In *Applied Sciences (Switzerland)* (Vol. 10, Issue 14). <https://doi.org/10.3390/app10145018>
- koldo. (2018). *Matriz de confusión*. Koldopina.
- Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. In *Computer Science Review* (Vol. 40). <https://doi.org/10.1016/j.cosrev.2021.100364>
- Kozin, S. H., & Gardner, R. (2019). Cerebral palsy. In *Pediatric Hand Therapy* (pp. 187–199). Elsevier. <https://doi.org/10.1016/B978-0-323-53091-0.00014-2>
- Labrada Batchelor, L. de la C., Batchelor Ramos, M. M., & Rodríguez Aguilar, Y. (2019). Los juegos didácticos su valor para el trabajo del logopeda. *Dilemas Contemporáneos: Educación, Política y Valores*. <https://doi.org/10.46377/dilemas.v28i1.1714>
- Li, B., Ye, L., Jie, L., Wang, Y., & Han, J. (2022). Region-of-interest and channel attention-based joint optimization of image compression and computer vision. *Neurocomputing*, 500, 13–25. <https://doi.org/10.1016/j.neucom.2022.05.047>
- Montero González, P. (2003). *SISTEMAS ALTERNATIVOS Y AUMENTATIVOS DE COMUNICACIÓN (SAAC) Y ACCESIBILIDAD BASES TEÓRICAS DE LOS SAAC*.
- Novak, I., McIntyre, S., Morgan, C., Campbell, L., Dark, L., Morton, N., Stumbles, E., Goldsmith, S., & Wilson, S. (2013). A systematic review of interventions for children with cerebral palsy: state of the evidence Study design. *DEVELOPMENTAL MEDICINE & CHILD NEUROLOGY*. <https://doi.org/10.1111/dmcn.12246>

Novak, I., Morgan, C., Adde, L., Blackman, J., Boyd, R. N., Brunstrom-Hernandez, J., Cioni, G., Damiano, D., Darrah, J., Eliasson, A.-C., de Vries, L. S., Einspieler, C., Fahey, M., Fehlings, D., Ferriero, D. M., Fetters, L., Fiori, S., Forsberg, H., Gordon, A. M., ... Badawi, N. (2017). Early, accurate diagnosis and early intervention in cerebral palsy: Advances in diagnosis and treatment. *JAMA Pediatrics*, *171*(9), 897–907. <https://doi.org/10.1001/jamapediatrics.2017.1689>

Novak, I., Morgan, C., Fahey, M., Finch-edmondson, M., Galea, C., Hines, A., Langdon, K., Namara, M. M., Paton, M. C. B., Popat, H., Shore, B., Khamis, A., Stanton, E., Finemore, O. P., Tricks, A., Velde, A., Dark, L., Morton, N., & Badawi, N. (2020). State of the Evidence Traffic Lights 2019: Systematic Review of Interventions for Preventing and Treating Children with Cerebral Palsy. *Pediatric Neurology*. <https://doi.org/https://doi.org/10.1007/s11910-020-1022-z>

Numpy. (2022). *Some information about the NumPy project and community*. <https://numpy.org/about/>

NVIDIA. (2019). NVIDIA Jetson Nano Developer Kit | NVIDIA Developer. In *Nvidia*.

Palisano, R., Rosenbaum, P., Walter, S., Russell, D., Wood, E., & Galuppi, B. (1997). Development and reliability of a system to classify gross motor function in children with cerebral palsy. *Developmental Medicine and Child Neurology*, *39*(4), 214–223. <https://doi.org/10.1111/j.1469-8749.1997.tb07414.x>

Peláez Cantero, M. J., Moreno Medinilla, E. E., Cerdón Martínez, A., & Gallego Gutiérrez, S. (2021). Comprehensive approach to children with cerebral palsy. *Anales de Pediatría*, *95*(4), 276.e1-276.e11. <https://doi.org/10.1016/j.anpedi.2021.07.011>

- Peredo, R., Canales, A., Menchaca, A., & Peredo, I. (2011). *Intelligent Web-based edation system for adaptive learning*. 38, 14690–14702. <https://doi.org/10.1016/j.eswa.2011.05.013>
- Perlstein, M. A., Shere, M., & Urbana, M. (1946). SPEECH THERAPY FOR CHILDREN WITH CEREBRAL PALSY. *American Journal of Diseases of Children*. <https://doi.org/10.1001/archpedi.1946.02020330021004>
- Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2020). Introduction to Anaconda and Python: Installation and setup. *The Quantitative Methods for Psychology*, 16(5), S3–S11. <https://doi.org/10.20982/tqmp.16.5.s003>
- Romantika, W., Luimilasari, L., Prabandari, Y., & Syahrul, S. (2020). Application of video-based health education in improving mother’s knowledge and attitudes about behavioral problems among preschool children. *Enermería Clínica*, 172–176. <https://doi.org/https://doi.org/10.1016/j.enfcli.2019.07.071>
- Russo, C., Serafino, S., Cicerchia, B., Luengo, P., & Useglio, G. (2018). Visión artificial aplicada en Agricultura de Precisión. *XX Workshop de Investigadores En Ciencias de La Computación*, 1, 992–996. <http://wicc2018.unne.edu.ar/wicc2018librodeactas.pdf>
- salih, T. A., & Basman Gh., M. (2020). Jetson Nano developer kit. *IOP Conference Series: Materials Science and Engineering*, 928(3).
- Subcomisión de accesibilidad cognitiva de CERMI. (2019). *Guía rápida de criterios y recursos sobre pictogramas. Accesibilidad cognitiva*. <https://www.plenainclusion.org/publicaciones/buscador/guia-rapida-de-criterios-y-recursos-sobre-pictogramas-accesibilidad-cognitiva/>

- Susilo, J., Febriani, A., Rahmalisa, U., & Irawan, Y. (2021). Car parking distance controller using ultrasonic sensors based on arduino uno. *Journal of Robotics and Control (JRC)*, 2(5). <https://doi.org/10.18196/jrc.25106>
- Suzen, A. A., Duman, B., & Sen, B. (2020). Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN. *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1–5. <https://doi.org/10.1109/HORA49412.2020.9152915>
- Tangsgaard Hvelplund, K. (2014). Eye tracking and the translation process: reflections on the analysis and interpretation of eye-tracking data. *MonTI Special Issue – Minding Translation*, 201–223. <https://doi.org/10.6035/monti.2014.ne1.6>
- Toasa, R., Burbano, E., Constante, A., Hidalgo, L., & Morales, F. (2019). A Custom and Dynamic Game using Gamification Techniques to Children from 4 to 5 years old. *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, June 1–5.
- Tobii. (2010). Tobii Eye Tracking - An introduction to eye tracking and Tobii Eye Trackers. *Technology*, 12. http://www.tobii.com/Global/Analysis/Training/WhitePapers/Tobii_EyeTracking_Introduction_WhitePaper.pdf?epslanguage=en
- Upadhyay, J., Tiwari, N., & Ansari, M. N. (2020). Cerebral Palsy: Etiology, Pathophysiology and Therapeutic Interventions. *Clinical and Experimental Pharmacology and Physiology*, 47(12), 1891–1901. <https://doi.org/10.1111/1440-1681.13379>

Valtakari, N. v., Hooge, I. T. C., Viktorsson, C., Nyström, P., Falck-Ytter, T., & Hessels, R. S.

(2021). Eye tracking in human interaction: Possibilities and limitations. *Behavior Research Methods*, 53(4), 1592–1608. <https://doi.org/10.3758/s13428-020-01517-x>

Vitrikas, K., Dalton, H., & Breish, D. (2020). Cerebral palsy: An overview. In *American Family*

Physician (Vol. 101, Issue 4, pp. 213–220). American Academy of Family Physicians.

<https://doi.org/10.4274/hamidiyemedj.galenos.2021.72792>

ANEXOS

Anexo 1. Entrevista al especialista

- 1. ¿Cómo se realiza el proceso de la terapia de lenguaje a los niños con parálisis cerebral?**

Al niño se le muestran una serie de imágenes y se le pide que seleccione una de ellas, median gestos o movimientos de la cabeza se interpreta si esta seleccionando la correcta.

- 2. ¿Cuáles son los problemas que se presentan al momento de realizar la terapia de lenguaje?**

El proceso de enseñanza es realizado por medio de interpretación y sería bueno que se optara por una solución tecnológica para mejorar este proceso de terapia.

- 3. ¿Conoce de productos de ayuda en el proceso de la terapia de lenguaje?**

La verdad si tengo conocimiento, anteriormente una empresa extranjera les colaboro con las pruebas de un dispositivo que ayudaba al proceso de terapia, el cual funcionaba por medio de la mirada, el dispositivo funciono muy bien, pero el costo era demasiado elevado para que sea adquirido por parte de los padres del niño.

- 4. ¿Se han realizado prueba con estos productos?**

Si se realizaron prueba con estos dispositivos, y los resultados fueron favorables, ayudo bastante en la terapia de lenguaje.

- 5. ¿Cómo le gustaría que sea el tipo de ayuda brindada por parte del proyecto?**

El proyecto está enfocado en utilizar los medios tecnológicos para ayudar en el proceso de terapia de lenguaje, y este tendría que estar enfocado en la reproducción de dispositivo de ayuda probado anteriormente, pero con un costo mucho más accesible.

6. ¿Qué aspectos considera que son esenciales para la funcionalidad del proyecto?

Que el dispositivo sea portátil, que no sea invasivo y le de comodidad al niño al momento de realizar la terapia.

7. Si se diseñara un sistema de ayuda en la terapia de lenguaje ¿Qué requerimientos debería tener?

Que pueda ser utilizado mediante la mirada del niño, que sea ligero, no invasivo, cómodo al momento de la terapia, que sea sencillo de utilizar y que permita la implementación de juegos de acuerdo con los gustos del niño.

8. ¿Cómo debería de ser el producto final para la realización de las pruebas de validación de funcionamiento?

Para las pruebas sería óptimo que se realice una aplicación en una Tablet donde se pueda realizar la terapia mediante gamificación utilizando la mirada del niño para la selección de la imagen que se le está solicitando.

Anexo 2. Scrip del Algoritmo

```
import cv2
import numpy as np
import dlib
import sys
import argparse
import imutils
import serial
from imutils import face_utils

#Initialize dlib's face detector (HOG-based) and then create the facial landmark predictor
detector = dlib.get_frontal_face_detector ()
#predictor = dlib.shape_predictor("/home/jetson/eyetrack/shapepredictor_68_facelandmarks.dat")
predictor = dlib.shape_predictor('./shapepredictor_68_facelandmarks.dat')

#Cam to test
cap = cv2.VideoCapture(1)
ym=5
xm=5

#Cx serial Port
#port = serial.Serial("COM5", baudrate=9600) #COM Windows
#port = serial.Serial("/dev/ttyACM0", baudrate=9600) #COM Linux
#port = serial.Serial("/dev/ttyTHS1", baudrate=9600) #BT Jetson

#Calibration
#Threshold
tcr=[0,0]
tcl=[0,0]
rtc=0
ltc=0
```

```

dr=0
dl=0
#Filters
fv=11

#Right & Left eye center cords
rcx=0
rcy=0
lcx=0
lcy=0

#Counters for serial Cx
r=0
l=0
u=0
d=0
c=0

#input_dir = './samples/'
#out_dir = './result/'

def extract_roi(img, shape, i, j):
    # extract the ROI of the face region as a separate image
    (x, y, w, h) = cv2.boundingRect(np.array([shape[i:j]]))
    roi = img[y-ym:y+h+ym,x-xm:x+w+xm]
    roi = imutils.resize(roi, width=200, inter=cv2.INTER_CUBIC)
    return roi

while True:

    try:
        # Capture frame-by-frame

```

```

ret, img = cap.read()

# Resize the image for video files.
img = imutils.resize(img, width=350)

# Convert to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Detect face
rects = detector(gray, 1)

# loop over the face detections
for (i, rect) in enumerate(rects):
# determine the facial landmarks for the face region, then
# convert the landmark (x, y) -coordinates to a NumPy array
    shape = predictor(gray, rect)
    shape = face_utils.shape_to_np(shape)

    (ri, rj) = face_utils.FACIAL_LANDMARKS_IDXS['right_eye']
    (li, lj) = face_utils.FACIAL_LANDMARKS_IDXS['left_eye']

#windows sizes
w=extract_roi(img,shape,ri,rj)
rows, cols, _ = w.shape
#print(w.shape)
lbx= int ((cols/9)*(4.1))
rbx= int ((cols/9)*(4.5))
uby= int ((rows/9)*0)
#dby= int ((rows/9)*(11/2))

#Right EYE
re=extract_roi(img,shape,ri,rj)

```

```

gr = cv2.cvtColor(re,cv2.COLOR_BGR2GRAY)
mr = cv2.medianBlur(gr,fv)
br = cv2.GaussianBlur(mr,(fv,fv),0)
#er = cv2.erode(br,kernel,cv2.BORDER_REFLECT)
retr,threshr = cv2.threshold(br,rtc,255,cv2.THRESH_BINARY_INV)
contoursr,
cv2.findContours(threshr,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
#contours,
cv2.RETR_LIST,cv2.CHAIN_APPROX_NONE)
#cv2.drawContours(re, contoursr, -1, (0, 255, 0))
contoursr = sorted(contoursr,key=lambda x:cv2.contourArea(x),reverse=True)

if contoursr:

    if dr==0:
        tcr[dr]=rtc
        dr=dr+1
        tcr[dr]=tcr[0]+5
        dr=dr+1

    for cnt in contoursr:
        rtc=tcr[1]
        (x, y, w, h) = cv2.boundingRect(cnt)
        #cv2.circle(le,(int (colsl/2),int(rowsl/2)),5,(155,0,150),-1) # center
cv2.circle(re,(x+int(w/2),y+int(h/2)),int (h/3),(5,0,250),2) # cv2 circles parameters
(img,center(x,y),ratio,color,thickness)
        #print('right eye',x+int(w/2),y+int(h/2))
        rcx=(x+(w/2))
        rcy=(y+(h/2))
cv2.line(re,(x+int(w/2),0),(x+int(w/2),rows),(250,0,10),1) # cv2 lines parameters
(origin(x,y),end (x,y),color,thickness)
        cv2.line(re,(0,y+int(h/2)),(cols,y+int(h/2)),(250,0,10),1)
        cv2.line(re,(lbx,0),(lbx,rows),(5,250,5),1)

```

```

cv2.line(re,(rbx,0),(rbx,rows),(5,250,5),1)
cv2.line(re,(0,uby),(cols,uby),(5,250,5),1)
break

```

else:

```

rtc=rtc+5

```

#Right

```

if rcx < lbx:

```

```

    if rcy >= uby:

```

```

cv2.putText(img,'Right',(30,30),0,1,(255,0,0),2)
(img,"Txt",cords(x,y),Font,Scale,Color,Thickness)

```

#Text Parameters

```

    r=r+1

```

```

    if (r==2):

```

```

        #port.write(b"R\n")

```

```

        #print('Right')

```

```

        r=0

```

```

        l=0

```

```

        u=0

```

```

        d=0

```

```

        c=0

```

#Center

```

if lbx <= rcx <= rbx and uby <= rcy:

```

```

    cv2.putText(img,'Center',(30,30),2,1,(255,0,0),2)

```

```

    c=c+1

```

```

    if (c==2):

```

```

        #port.write(b"C\n")

```

```

        #print('Center')

```

```

        r=0

```

```

        l=0

```

```

    u=0
    d=0
    c=0

#Left
if rcx > rbx:
    if rcy >= uby:
        cv2.putText(img,'Left',(30,30),0,1,(255,0,0),2)
        l=l+1
        if (l==2):
            #port.write(b"L\n")
            #print('Left')
            r=0
            l=0
            u=0
            d=0
            c=0

#Up
if rcy <= uby:
    cv2.putText(img,'Up',(30,30),2,1,(255,0,0),2)
    u=u+1
    if (u==2):
        #port.write(b"U\n")
        #print('Up')
        r=0
        l=0
        u=0
        d=0
        c=0

print ('Eye Cords ==> ',rcx,rcy)

```

```
cv2.imshow("right eye", re)
cv2.imshow("right", threshr)
cv2.imshow("Image", img)
```

except:

```
print("Error")
rtc=0
dr=0
```

```
if cv2.waitKey(1) & 0xFF == ord('r'):
```

```
    rtc=0
    dr=0
```

```
if cv2.waitKey(1) & 0xFF == 27: # esc key to exit
```

```
    # When everything done, release the capture
    cap.release()
    cv2.destroyAllWindows()
    sys.exit()
```