

UNIVERSIDAD SANTO TOMÁS

DESARROLLO DE UN SOFTWARE BASADO EN VISIÓN ARTIFICIAL PARA ATENDER NECESIDADES DE PERSONAS CON DISCAPACIDAD VISUAL EN UN ESCENARIO DE ENSEÑANZA-APRENDIZAJE

Realizado por

Hernán Felipe Salamanca Montoya
Juan David Avendaño Moreno

Anteproyecto enviado en cumplimiento del requisito parcial
para optar por el grado de Ingeniería Electrónica



Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

Septiembre de 2022

DESARROLLO DE UN SOFTWARE BASADO EN VISIÓN ARTIFICIAL PARA ATENDER NECESIDADES DE PERSONAS CON DISCAPACIDAD VISUAL EN UN ESCENARIO DE ENSEÑANZA-APRENDIZAJE

Realizado por

**Hernán Felipe Salamanca Montoya
Juan David Avendaño Moreno**

Anteproyecto enviado en cumplimiento del requisito parcial
para optar por el grado de Ingeniería Electrónica

Dirigido por

Sindy Paola Amaya

Codirigido por

Edgar Camilo Camacho Poveda

Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

Septiembre de 2022

Índice general

1. Glosario	1
2. Resumen	2
3. Introducción	3
4. Problema	5
5. Antecedentes	7
6. Justificación	10
7. Objetivos	12
7.1. Objetivo General	12
7.2. Objetivos específicos	12
8. Marco Teórico	13
8.1. Procesamiento de Imágenes	13
8.1.1. Fotografías como Datos	13
8.1.2. Dilatación y Erosión	15
8.2. Redes Neuronales	16
8.2.1. Neurona Artificial	16
8.2.2. Red Neuronal en el Procesamiento de Imágenes	18
8.2.2.1. Filtros	18
8.2.2.2. Pooling	19
8.2.2.3. Vectorización y Red de Salida	19
8.3. Acompañamiento y asesorías externas	21
8.3.1. Instituto Nacional Para Ciegos	21
9. Desarrollo Metodológico	23
9.1. Investigación y caracterización de los elementos visuales	24
9.2. Recopilación de algoritmos	25
9.3. Pruebas de algoritmos	26
9.3.1. Pruebas de algoritmo Tesseract (Google)	26
9.3.2. Pruebas de algoritmo gtts (Google)	27

9.3.3.	Pruebas de algoritmo gtts (Google)	28
9.3.4.	Pruebas de algoritmo FER Library	28
9.3.5.	Pruebas de los algoritmos <i>Priya, facial-emotion-recognition</i> y <i>Atulapra-Emotion-detection</i>	30
9.3.6.	Pruebas de algoritmo Mathpix	31
9.3.7.	Recopilación de pruebas algoritmos	33
9.4.	Integración de algoritmos	33
9.4.1.	Revisar Compatibilidad	34
9.4.2.	Separación de Elementos	34
9.4.2.1.	Detección de Tableros Blancos	35
9.4.2.2.	Detección de Personas	36
9.4.3.	Integración	37
9.4.4.	Parámetros de cámara	37
9.4.5.	Generación de LaTeX y MathML	40
9.4.6.	Generación de PDF y HTML	41
9.4.7.	Integración Interfaz Gráfica	42
9.4.7.1.	Parte Visual	42
9.4.7.2.	Parte Funcional	42
9.4.7.3.	Parte Técnica	43
9.5.	Ejecutable	43
10.	Resultados	44
10.1.	Caracterización de Elementos Visuales.	44
10.2.	Un Algoritmos para Detectar Elementos Visuales.	44
10.2.1.	Reentrenamiento de la Red de Detección de Tableros Blancos.	45
10.2.2.	Pruebas del <i>Software</i>	52
10.2.2.1.	Primer Prueba	52
10.2.2.2.	Segunda Prueba	55
10.2.2.3.	Tercer Prueba	58
10.3.	Algoritmo del Aplicativo	61
10.4.	Presentación al Instituto Nacional Para Ciegos	66
10.4.1.	Accesibilidad	66
10.4.2.	Educación	66
10.4.3.	Tecnología	66
11.	Conclusiones y Trabajo Futuro	67
11.1.	Conclusiones	67
11.2.	Trabajo Futuro	68
12.	Referencias Bibliográficas	69
13.	Anexos	73
13.1.	Anexo 1.	73
13.2.	Anexo 2.	73
13.3.	Anexo 3.	76

13.4. Anexo 4.	78
13.5. Anexo 5.	80
13.6. Anexo 6.	82
13.7. Anexo 7.	82
13.8. Anexo 8.	82

Capítulo 1

Glosario

INCI: (Instituto Nacional Para Ciegos) Entidad Colombiana adscrita al Ministerio de Educación, de carácter técnico, que brinda apoyo a las personas con baja visión o ciegas en el territorio colombiano.

LaTeX: Es un lenguaje utilizado para describir y crear documentos, ecuaciones matemáticas y gráficos.

MathML: (*Mathematical Markup Language*) lenguaje basado en etiquetas, cuyo fin es facilitar la interpretación de estructuras matemáticas para intercambio de información.

PDF: (*Portable Document Format*) Es un formato de almacenamiento para documentos digitales que como característica principal es independientes de plataformas de software o hardware.

HTML: (*HyperText Markup Language*). lenguaje basado en etiquetas, para describir la estructura de un documento. Comúnmente es utilizado en páginas web.

OCR: (Reconocimiento óptico de caracteres) proceso que se realiza para reconocer textos o expresiones, que contengan caracteres para su transformación en cadenas útiles para manejo de los mismos de forma digital.

TTS: (*Text to Speech*). desarrollo para reproducir voz humana de manera artificial.

gts libreria de python: (*Speech Services by Google*) biblioteca de Python que interactúa con el servicios de Google para transformar texto a voz.

GUI: (*Graphical user interface*) Conjunto de elementos gráficos para comunicar información de manera gráfica a un usuario.

Capítulo 2

Resumen

En este documento se describe el proceso de desarrollo de un software construido a partir de diversos algoritmos de visión artificial que cumplen diferentes funciones, como lo son: la extracción de textos a partir de imágenes, la estimación de los sentimientos a partir de un rostro humano, y la descripción de los gestos corporales de una persona, entre otros. Esto con la finalidad de obtener un software capaz de brindar apoyo a una persona que tenga alguna discapacidad visual, en un aula de clase presencial.

Capítulo 3

Introducción

La vista es uno de los sentidos más importantes, con base a este sentido los seres humanos plantean diferentes actividades en el día a día, puesto que la mayor parte de la información de su entorno es percibida por dicho sentido. Sin embargo y a pesar de su gran importancia para los seres humanos, no todos pueden contar con este sentido ya sea por una pérdida parcial o total del mismo, dichas pérdidas pueden ser generadas por diferentes factores como lo pueden ser la genética, lo que provoca que su sentido de la vista se vea afectado; aún así, la vista puede verse afectada no solamente por aleatoriedades genéticas sino también por accidentes en la vida cotidiana [18]. Es por ello que al no poseer un sentido de la vista en condiciones óptimas, las personas tienen grandes dificultades en muchos contextos, entre ellos, el académico. El Instituto Nacional para Ciegos (INCI) ha identificado diferentes elementos que son cruciales para estudiantes que asisten a clases presenciales y que además posean alguna dificultad visual. Es entonces que, desde la perspectiva de la Ingeniería Electrónica se hace la siguiente pregunta: ¿Cómo desarrollar un software basado en visión artificial que permita contribuir al proceso de enseñanza de personas en condición de discapacidad visual?

Para tratar de aminorar las consecuencias de éstos problemas, y para dar una respuesta a la pregunta problema, éste proyecto plantea el uso de varios algoritmos de inteligencia artificial, enfocados a la visión artificial, con la finalidad de poder obtener y transmitir información a las personas que no puedan acceder tan fácilmente a ésta, como lo son las personas con discapacidades visuales parciales o totales, aumentando así los contenidos recibidos dentro de un área de clases presencial. De esta manera también respondiendo a la pregunta problema, determinando el proceso por el cual se utiliza la visión artificial para contribuir al proceso de enseñanza en personas en condición de discapacidad visual.

Teniendo en cuenta lo anterior se plasmará el desarrollo de este proyecto por medio de una serie de capítulos, teniendo en cuenta el inicio del proyecto desde el planteamiento del problema donde se habla sobre pregunta problema a desarrollar, con dicho planteamiento se desarrolla un análisis de diferentes dispositivos ya existentes en el mercado que estén ligados de alguna forma al proyecto, con esto se presenta una justificación y un objetivo general del proyecto, del cual se derivan diversos objetivos específicos, los cuales darán la ruta para el desarrollo del proyecto. Con lo anterior se podrá hacer un marco teórico donde se discutirá sobre los principales referentes, bases teóricas e hipótesis que soportan el proyecto. Teniendo en cuenta todo lo anterior se plantea el desarrollo y ejecución donde se plasmará una estructura por capítulos que detallaran el paso a paso para alcanzar los objetivos general y específicos planteados anteriormente. Luego se desarrollarán los resultados así como su impacto social, con esto se plantea una serie de conclusiones. Para finalizar, se plasmará una serie trabajos futuros que corresponden a propuestas y sugerencias de mejoramiento a futuro del proyecto actual, también se presentarán las diferentes citas bibliográficas y anexos utilizados en el proyecto.

Capítulo 4

Problema

De acuerdo a los reportes del Censo Nacional de Población y Vivienda 2018, en Colombia hay 1.948.332 personas (alrededor del 4 % de la población en Colombia) que expresaron tener dificultades en la realización de actividades básicas diarias relacionadas con ver de cerca, de lejos o alrededor, dichos problemas pueden deberse a varios factores, entre ellos accidentes, problemas genéticos, edad avanzada y enfermedades [19]. Dichas dificultades de la visión pueden perjudicar la calidad de vida de estas personas en múltiples aspectos como, en los procesos de aprendizaje. Los procesos por los cuales una persona adquiere conocimientos, en la sociedad de hoy en día, depende en su mayoría de la visión puesto que los conocimientos se adquieren en gran parte de manera visual y auditiva [20]. Esto es un problema ya que dicha situación afecta la inclusión de estos individuos a la sociedad puesto que tendrán menos oportunidades en la vida laboral.

Por otra parte, las diferentes herramientas tecnológicas no están enfocadas actualmente en brindar un apoyo en un escenario de enseñanza-aprendizaje, en su lugar buscan atender otras necesidades como la asistencia en la movilidad de los usuarios y la detección de diferentes objetos en la cotidianidad. Éste software existente está ligado al hardware específico de la herramienta, por lo tanto, se incrementan los costos de su adquisición además el posible mantenimiento que requiere dicho hardware puede resultar en una tarea tediosa para los usuarios debido a su déficit visual. Por ejemplo, dentro de las diferentes herramientas encontradas existe un dispositivo que ayuda a las personas ciegas para detectar obstáculos y hacer su trabajo fácilmente, por medio de hardware y software de Arduino implementando sensores infrarrojos, de ultrasonido y un módulo GPS para dar notificaciones de ubicación cuando la persona ciega está en amenaza [21]; esta herramienta les brinda ayuda en la movilidad, pero no en un escenario de enseñanza aprendizaje. Otra herramienta encontrada permite al usuario moverse por su teléfono móvil

con gestos y sonido llamada Foogues; la cual tampoco aporta directamente al proceso de enseñanza y aprendizaje. A pesar de que hay muchas más herramientas, pocas aportan al proceso de aprendizaje [22][23].

Gracias a los avances en el campo de la inteligencia artificial, más concretamente en la visión artificial, se pueden tomar datos de la naturaleza y procesarlos para generar información útil. con todo esto se plantea la siguiente pregunta:

¿Cómo desarrollar un software basado en visión artificial que permita contribuir al proceso de enseñanza de personas en condición de discapacidad visual?

Capítulo 5

Antecedentes

Herramientas han sido desarrolladas con el fin de mejorar la calidad de vida de las personas con alguna discapacidad visual, pero no todas pertenecen o se relacionan con la temática que se presenta en este texto, por tanto, ésta se ha establecido una ponderación para cada parte de ésta, realizando un enfoque en la visión artificial aplicada a las necesidades de las personas con discapacidad visual. Se ha seleccionado una ventana de observación entre los años 1999 a 2020 dado que en éstos años se presentan la mayor cantidad de avances en éste campo. En 1999 se implementó un instrumento de lectura de textos básicos para la enseñanza a personas con discapacidad visual. El reconocimiento de caracteres utilizado en el instrumento se basa en una red neuronal de preclasificación y retropropagación, los caracteres se leen, a través del escáner de texto de libros de enseñanza elemental, y el programa localiza las palabras reconocidas en una base de palabras grabadas para emitirlas a través de la voz con ayuda de los sistemas multimedia, luego un instrumento de lectura para transformación de texto a voz y reconocimiento de caracteres [24]. En el 2009 se realizó una investigación que busca verificar la aplicación de la tecnología de asistencia en la educación de estudiantes ciegos o con problemas de visión desde la perspectiva de sus maestros; de acuerdo con los profesores las tecnologías de asistencia tienen aplicaciones diferentes para cada estudiante, y aunque resultan ser bastante útiles, requieren una mayor inversión para poder brindar acceso a dichas tecnologías a cada estudiante, lo que limita su uso [25]. Para 2010 se desarrolló una aplicación para dispositivos móviles llamada Foogues, la cual por medio del sonido y del movimiento del usuario, le permite a la persona moverse por el teléfono buscando la información que pueda necesitar y así eliminar la pantalla táctil [26]. Para el 2014 se han desarrollado diferentes redes neuronales para ser implementadas junto con cámaras que transforman el texto que se observe ya sea en una pantalla o en un libro, en sonidos, sistemas como este existen en casi todos los dispositivos y se usan en

gran amplitud [27]. Se hicieron también estudios acerca de cómo la tecnología asistida facilita los procesos de enseñanza-aprendizaje y administrativos para la gente con discapacidad visual. En uno de ellos se obtuvo la información a partir de entrevistas que tomaron alrededor de 25 a 30 minutos, en dicho estudio los investigadores tenían como objetivo conocer las opiniones de los participantes sobre qué tipo de tecnología se utiliza en la Cyprus Turkish Blind Association (Turquía), qué beneficios y/o problemas han tenido los profesores, los administradores y los que enfrentan los estudiantes en el uso de la tecnología y cómo la tecnología facilita los procesos administrativos (toma de decisiones, cambio, motivación, liderazgo y comunicación). El estudio encontró que las personas se sentían más motivadas con dichas tecnologías y podían comunicarse más fácilmente con gente de otras organizaciones, les permitió comunicarse mejor con los profesores; en resumidas cuentas, el estudio encontró que las tecnologías asistivas son un excelente apoyo para las personas con problemas de visión [28]. En el 2018 ya existían variedad de aplicaciones para diferentes dispositivos que buscan facilitar la interacción de las personas con problemas de visión y las nuevas tecnologías. Una de estas aplicaciones es Voice Maps 96: un sistema para la navegación punto a punto y movilidad independiente para usuarios con discapacidad visual en áreas urbanas que opera en un teléfono inteligente con pantalla táctil; Voice Maps aprovecha el mecanismo de texto a voz de Android para generar mensajes de voz, vibración para la accesibilidad de la pantalla y gestos reconocimiento para la entrada de texto. Una característica interesante del sistema es que, además de encontrar la ruta óptima, monitorea continuamente la dirección y la posición de un usuario. Existen además apps que complementan herramientas ya utilizadas por personas invidentes, como lo es SmartVision50, es una ayuda a la navegación que mejora y complementa electrónicamente el bastón para guiar a los usuarios a un destino evitando obstáculos en el camino. Ayuda en la navegación local por seguimiento de caminos y detección de obstáculos y cubre el área frente al usuario y más allá del alcance del bastón de tal manera que el sistema puede alertar a los usuarios sobre los obstáculos por delante de ellos antes de que su bastón alcance a tocarlos. En el mismo año se desarrolló un software de traducción de lenguaje de señas a voz y viceversa para que las personas con dificultades visuales puedan comunicarse con las personas con discapacidades auditivas, el software es llamado VirtualSign [29]. Finalmente dentro de la colección de documentos hallados se puede encontrar gran variedad de conjuntos de dispositivos electrónicos que intentan emular la visión humana con el fin de intentar comunicar a la persona invidente o con poca visión, la gran mayoría de dispositivos ayudan a las personas ciegas para detectar obstáculos y hacer su trabajo fácilmente, por medio de hardware y software, uno de ellos está implementado por medio de Arduino con sensores infrarrojos, de ultrasonido y un módulo GPS para dar notificaciones de ubicación cuando la persona ciega se encuentre en amenaza con lo que se encuentra a su alrededor, como lo podría ser un robo con el cual se activara por medio

de la lectura de la voz del usuario enviando su ubicación, como también peligros con vehículos que parezcan poner en riesgo la vida de la persona, esta alerta llegará a un ente externo definido por la persona como lo podría ser un familiar, amigo, entre otros, también se puede encontrar una detección de objetos implementando un algoritmo de estimación que se ejecuta en hardware de procesamiento limitado y de bajo costo. El algoritmo se basa en dos pasos principales: en la fase de detección, emplea una Red Neuronal Convolucional (CNN) con el fin de reconocer y hacer seguimiento de objetos de interés; mientras que en la fase de estimación de la pose el algoritmo utiliza correspondencias estéreo para reconstruir en 3D las coordenadas espaciales de múltiples características ORB al rededor del objeto reconocido. El algoritmo también estima su posición 3D por medio de herramientas de ROS, esto con el fin de ayudar en el desplazamiento del usuario [30]. En este año se propuso el desarrollo de un dispositivo portátil con inteligencia artificial basado en Edge Computing para asistir el paso por cruces peatonales para la gente con discapacidad visual. El dispositivo se compone de un par de gafas inteligentes, un bastón y un cinturón con una caja. Cuando el peatón con discapacidad visual vaya a cruzar el paso peatonal, el peatón con discapacidad visual recibirá el mensaje actual de la señal del semáforo, luego un servicio de voz de orientación para personas con discapacidad visual se proporcionará a través de Bluetooth. Los resultados experimentales mostraron que la tasa de reconocimiento se puede lograr hasta en un 90 % [31].

Capítulo 6

Justificación

Con el paso del tiempo y con el desarrollo de las herramientas y el progreso de la civilización, poco a poco se ha ido mitigando el efecto que causan problemas evolutivos, como las mutaciones no provechosas. La humanidad, gracias a la tecnología, ha venido reduciendo las muertes y dando esperanza tanto a los individuos a los que la genética les ha jugado una mala pasada, como a los que han tenido infortunios en forma de accidentes que resultan en pérdidas de algunas de las capacidades de sus cuerpos, como lo son problemas del sentido de la vista[19]. La discapacidad visual ha sido un inconveniente con el paso de los años en diferentes aspectos, y aunque en los últimos siglos se ha evitado casi totalmente el efecto de la selección natural, la calidad de vida de las personas con dichas discapacidades se ha visto afectada de diversas formas, dependiendo si han perdido la visibilidad total o parcialmente. Los problemas por los que deben pasar, si no han perdido en su totalidad su capacidad visual son principalmente: complicación a la hora de identificar de forma visual personas, objetos, espacios, etc; dificultad para leer textos; dificultad para detectar obstáculos y para poder orientarse en nuevos espacios. Por otro lado, si han perdido en su totalidad su capacidad visual tendrán una diferente percepción de la orientación y una diferente movilidad, que se han mitigado con el uso de un bastón o de un perro guía; para poder leer deben hacerlo únicamente con textos en Braille; no pueden recibir información que se encuentre solamente en un medio gráfico, como avisos, escrituras en carteles o tableros. Para tratar de solucionar dichos problemas, se han desarrollado diferentes dispositivos y aplicaciones que tratan de mitigar el efecto que tiene la discapacidad visual en la calidad de vida de las personas que la padecen, en Colombia, ya existen algunos dispositivos gratuitos, pero que no están enfocados directamente a brindar apoyo en la enseñanza y/o aprendizaje presencial de los individuos, sino que, están más enfocados a la cotidianidad[35]. Si existiera una solución de bajo costo y fácil acceso, más personas con discapacidad visual o

instituciones educativas podrían acceder a ella y ésta ayudaría a al proceso de enseñanza y aprendizaje de las personas con problemas de visión [36]. Además, por conversaciones provenientes de la intención de colaboración de la Facultad de Ingeniería Electrónica y el Instituto Nacional Para Ciegos (INCI), se ha planteado la idea general del proyecto; el INCI considera que el desarrollo de éste podría determinar una solución que puede ayudar a mejorar la calidad de vida de las personas con problemas de visión y/o ceguera.

Capítulo 7

Objetivos

7.1. Objetivo General

Desarrollar un software basado en algoritmos de visión artificial para identificar elementos visuales que hagan parte de un escenario de enseñanza presencial, para facilitar el aprendizaje en personas con discapacidad visual.

7.2. Objetivos específicos

- Caracterizar los elementos visuales a detectar en el escenario de enseñanza presencial para población con discapacidad visual, de acuerdo a las recomendaciones del Instituto Nacional para Ciegos (INCI).
- Implementar un algoritmo existente de visión artificial que permita identificar los elementos caracterizados en el escenario de enseñanza presencial.
- Integrar el algoritmo de visión artificial y un algoritmo de texto a voz, para lograr comunicar información al usuario.
- Validar el funcionamiento del sistema de acuerdo a los criterios aportados por el INCI.

Capítulo 8

Marco Teórico

En este capítulo se desarrollarán una serie de conceptos necesarios para el entendimiento de las temáticas que se abordan en este proyecto. Conceptos como lo son: el procesamiento de imágenes, la visión artificial, la transformación de texto a voz, como también características de algunos algoritmos.

8.1. Procesamiento de Imágenes

El procesamiento de imágenes consiste en tomar imágenes con un dispositivo como lo puede ser una cámara y a través de diversos algoritmos acceder a la información que está en ella. Para ello es necesario entender la imagen como una serie de datos que representan los colores, figuras y posiciones que se pueden ver en la misma. Algunos de los algoritmos y técnicas que se utilizan para trabajar estos datos se presentan en las siguientes subsecciones.

8.1.1. Fotografías como Datos

Una fotografía es entendida por una máquina como un conjunto de números, ordenados en 3 matrices, donde cada número se halla en una de las celdas de cada matriz. En total hay 3 matrices, una para cada color: verde, rojo y azul. Son estos 3 colores porque se imita el funcionamiento del ojo humano: éste tiene células denominadas conos, que son las encargadas de percibir los colores. Hay 3 tipos, uno para cada uno de los colores mencionados anteriormente [1].

Para poder obtener más fácilmente información de la imagen, se eliminan 2 de las 3 matrices de colores, quedando solo con una, y con una imagen en escala de grises como resultado. Seguidamente la escala de grises es transformada a una imagen en blanco y negro, es decir, los valores de la matriz solo podrán tener dos valores el máximo y el mínimo, lo cuales son 255 y 0 si se trabaja con formato de 8 bits, o 1 y 0 si se trabaja con formato de 1 bit. El valor máximo es 255 puesto que cada valor de la matriz es la unidad de información computacional y es llamado byte; un byte puede tener como máximo un valor de 255.

Un ejemplo de cómo se ven los datos en una imagen a escala de grises se puede ver en la figura 1.

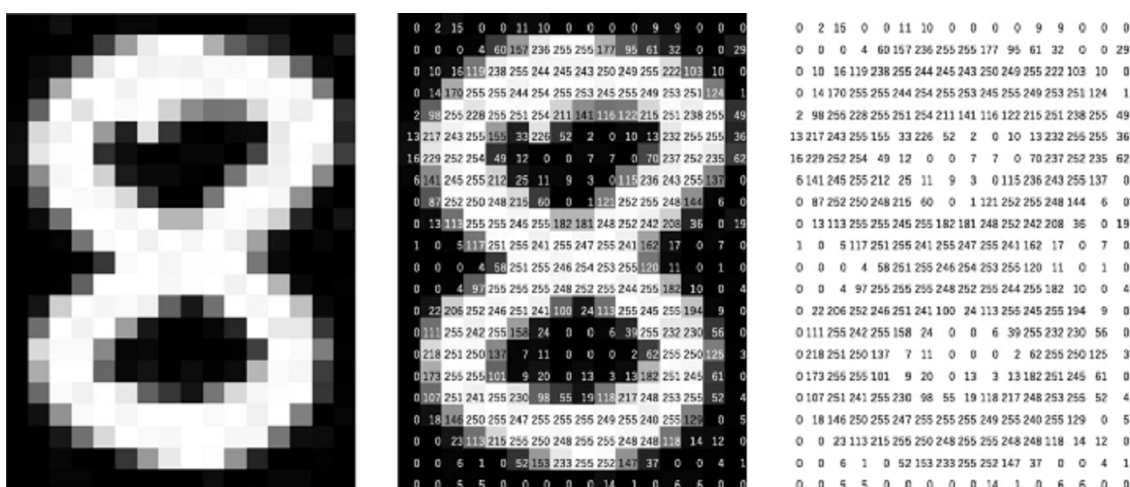


FIGURA 1: Demostración de datos en una imagen, tomado de [15]. En el primer recuadro se puede observar una imagen en escala de grises de un número 8; en el siguiente recuadro se puede observar la misma imagen, pero para cada píxel se muestra también su valor numérico; el el tercer recuadro se muestran únicamente los valores numéricos de los píxeles en la misma ubicación que se encuentran en la imagen original.

Para repartir todos los diferentes valores de la matriz de escala de grises en solo dos valores se utiliza un umbral, o sea, un valor de referencia. Si el valor que se encuentra en la matriz de grises (dato) es mayor que el valor umbral, dicho dato será elevado a 255; por otro lado, si el dato es menor que el valor umbral, dicho dato será reducido a 0 [2].

Una vez que se tiene la imagen en blanco y negro es más fácil obtener información de ella, ésta información se obtiene empleando otras técnicas. En la figura 2 se puede observar cómo a partir de una imagen en blanco y negro se obtienen únicamente los contornos de las figuras.

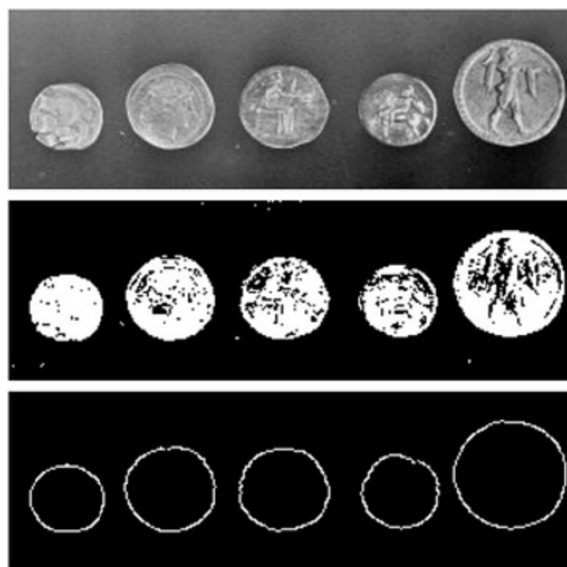


FIGURA 2: Proceso de adecuación de imagen (adaptada de [11]). En el primer recuadro se observa una imagen de unas monedas en escala de grises; en el segundo recuadro se observa la misma imagen pero en blanco y negro; en el tercer recuadro se observan los contornos de las monedas obtenidos a partir del proceso de adecuación de imagen.

8.1.2. Dilatación y Erosión

Después de los procedimientos vistos anteriormente, es muy común que en la imagen a blanco y negro quede información no necesaria o inservible y es requerido eliminarla, esto se hace mediante dos procesos llamados dilatación y erosión. El primer método, Dilatación, consiste en aumentar los bordes de los objetos en blanco que se encuentren en la imagen, provocando que todas las secciones que se vean blancas aumenten su tamaño, esto con el fin de que información importante se vuelva más visible y fácil de detectar. Seguidamente se utiliza el método Erosión, el cual disminuye los bordes de los objetos blancos, haciéndolos más pequeños, esto para eliminar cualquier ruido o información no deseada (manchas blancas), eliminando datos que pueden entorpecer el proceso de adquisición de información [3].

Luego de esto se procede a emplear redes neuronales para extraer la información de la imagen.

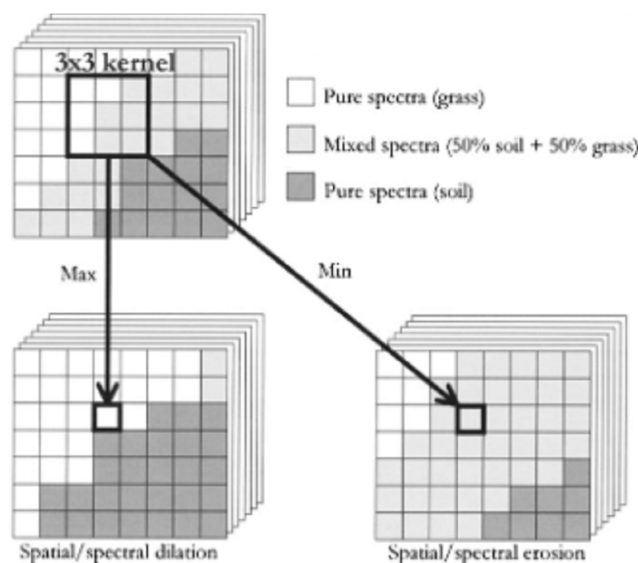


FIGURA 3: Ejemplos de Dilatación y Erosión, tomado de [12], la cuadrícula que se encuentra en la parte superior es una representación de una imagen la cuál será sometida a procesos de Erosión y Dilatación utilizando un *kernel* de 3x3 píxeles; la cuadrícula de la parte inferior izquierda muestra el resultado del proceso de Dilatación sobre el primer recuadro, la cuadrícula de la parte inferior derecha muestra el resultado de el proceso de Erosión sobre el primer recuadro.

8.2. Redes Neuronales

Las redes neuronales son, como su nombre lo dice, un conjunto de neuronas modeladas matemáticamente a partir del funcionamiento de una neurona humana. Cada neurona individual tiene su función matemática.

8.2.1. Neurona Artificial

Una neurona artificial es un procesador de información en la manera de que recibe varias entradas simultáneas, como lo puede ser un vector y produce una salida o respuesta única.

Cada una de las entradas está multiplicada por un valor llamado peso, este valor es diferente para cada entrada y es una de las partes más importantes de la neurona artificial; el peso le da más o menos valor a cada una de las entradas, y conforme la neurona va procesando más información, los pesos en sus entradas van variando con el fin de obtener la salida deseada. Este proceso de modificación de los pesos es el que se conoce como aprendizaje [4].

Luego de que las entradas sean multiplicadas por el peso, éstas son sumadas para obtener un único valor, luego este valor es restado con otro propio de la neurona llamado Umbral, el cual aumenta o disminuye el valor de la neurona, siendo útil dentro de una red de neuronas. Se puede encontrar una representación gráfica de éste proceso en la figura 5

Finalmente esta información pasa por una Función de Activación, la cual es una función matemática que retorna un valor dependiendo del valor de entrada. Hay varios tipos de funciones de activación. Una de las más usadas es llamada lineal, y entrega el mismo valor que recibe, pero si es mayor a 1 o menor a -1 entregará 1 y -1 respectivamente, limitando los valores entre éstas unidades [5], en la figura 4 se pueden observar tres diferentes gráficas para 3 diferentes funciones de activación.

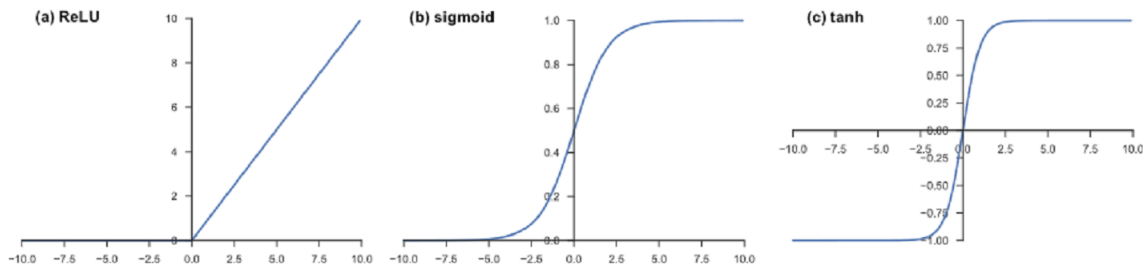


FIGURA 4: Funciones de activación más utilizadas, tomado de [15]. Se pueden ver las gráficas de respuesta correspondientes a 3 tipos Funciones de Activación, de izquierda a derecha, ReLU, Sigmoide y Tangente hiperbólica

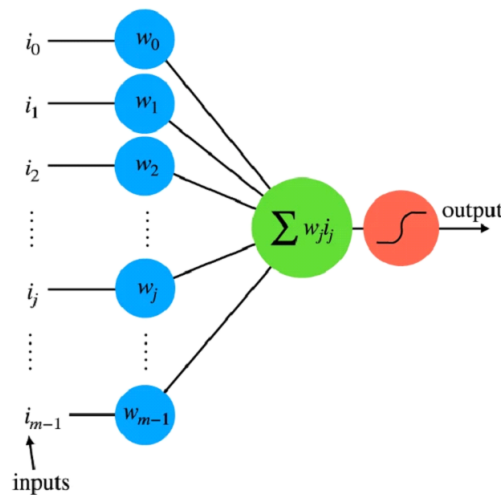


FIGURA 5: Representación de una neurona artificial (adaptado de [13]). Se observa una representación gráfica de los procesos matemáticos que se llevan a cabo en una neurona artificial, en éste caso se pueden ver las entradas de la neurona en la parte izquierda, seguidas de una sumatoria en el medio y finalmente una Función de Activación en la parte derecha.

8.2.2. Red Neuronal en el Procesamiento de Imágenes

Cuando 2 o más neuronas artificiales se conectan entre ellas se forma una red neuronal, básicamente es la unión de las entradas de una neurona, con las salidas de otras. Ésta unión permite que más neuronas procesen la información, generando modificaciones en los pesos de las entradas de cada una, por lo tanto aprendizaje. Además cada neurona modifica su umbral, lo que permite otorgar más o menos importancia a la información que sea procesada por una neurona en específico [6].

Para que los pesos de las neuronas varíen, es necesario entrenar la red. Básicamente lo que se hace es pasarle información que procesar para obtener un resultado, para que la red sepa qué resultado debe obtener, ésta información ya debe contener el resultado correcto. Lo que hace la red es tratar de modificar sus pesos para obtener los mismos resultados que los datos que se le han pasado, aprendiendo “cómo hacerlo” [7].

8.2.2.1. Filtros

Las redes neuronales convolucionales tienen varias capas, la primera de ellas es llamada capa convolucional. Lo que ocurre con la imagen en esta capa es que se le aplican diversos filtros, que pueden ser de diferentes tamaños. Los filtros son pequeñas matrices, llamadas *kernels*, con las cuales se realizan operaciones matemáticas por toda la matriz de la imagen (la imagen es una matriz de datos, como se mencionó anteriormente), barriendo todos sus datos. Como resultado de las operaciones matemáticas del *kernel* con la matriz de la imagen se obtiene otra matriz, que es la imagen con modificaciones o filtros, estas modificaciones dependen de los valores que tenga el *kernel* y de la operación matemática que se haya utilizado para operarlo con la matriz de la imagen [8].

Como resultado de estos filtros se pueden tener resultados como que las líneas verticales que existan en la imagen se muestren más fuertemente, mientras que las líneas horizontales sean más tenues y viceversa. También puede ocurrir que esto pase con líneas diagonales, que se vuelvan más fuertes los bordes o incluso que se resalten las curvas [8]. En la figura 7 se pueden observar diferentes imágenes procesadas con diferentes filtros.

Todas esas variaciones presentes luego del filtro resaltan características de la imagen original y serán muy útiles para la red neuronal que espera más adelante.

En la figura 6 se puede observar el proceso de filtrado píxel a píxel, utilizando un *kernel* de 3x3 píxeles.

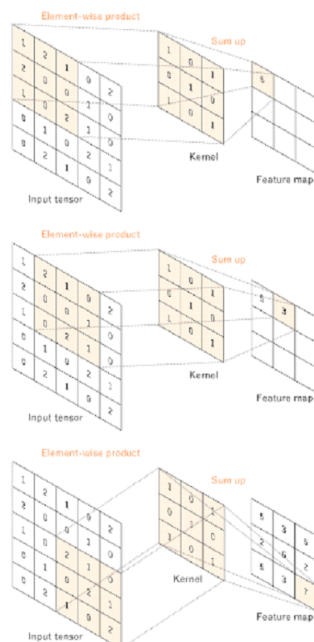


FIGURA 6: Funcionamiento de filtros (adaptado de [15]). Se observa una representación a nivel de píxeles del funcionamiento de filtros utilizando un *kernel* de 3x3 píxeles; se observa como se van obteniendo los píxeles de la imagen filtrada de manera progresiva desde la parte superior hasta la inferior.

8.2.2.2. Pooling

Luego del proceso de filtrado es necesario reducir la cantidad de datos que se tienen, por lo que el siguiente procedimiento se encarga de eso precisamente y es llamado *Pooling*. Éste procedimiento toma cada una de las matrices generadas y agrupa sus datos (posiciones de la matriz) en pequeñas matrices, dependiendo del tipo de *Pooling* estas matrices pueden ser de diversos tamaños. Luego, se toma el valor más grande que se encuentre en cada una de las matrices pequeñas y forma otra matriz pero solo con estos valores más grandes, resultando en una imagen más pequeña que la original, pero conservando la información importante, con menos datos. Este procedimiento se hace varias veces hasta conseguir la cantidad de datos deseada [9][10].

8.2.2.3. Vectorización y Red de Salida

Para que la red pueda procesar la imagen es necesario tomar la matriz y convertirla en un vector, reorganizando y pasándola a la red neuronal. Éste proceso es llamado vectorización. Éste vector entra a una red neuronal la cual procesa estos datos obteniendo una salida que se compara con el resultado que se debería obtener, y realizando los ajustes necesarios en sus pesos.

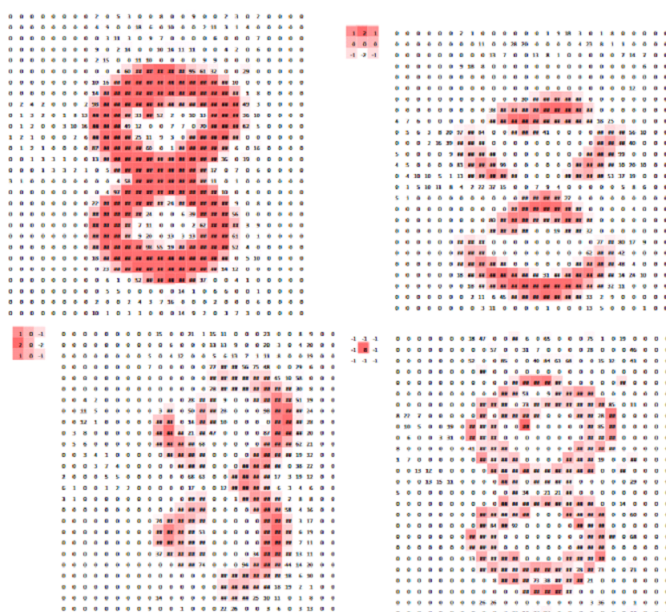


FIGURA 7: Ejemplos de características obtenidas con varios filtros (adaptado de [15]). Se observan 4 imágenes con valores numéricos para cada píxel. La imagen en la parte superior izquierda muestra la imagen original, las demás imágenes muestran el la misma imagen luego de ser sometida a un proceso de filtro empleando diferentes *kernel*, el *kernel* en cuestión se observa a la izquierda de cada imagen.

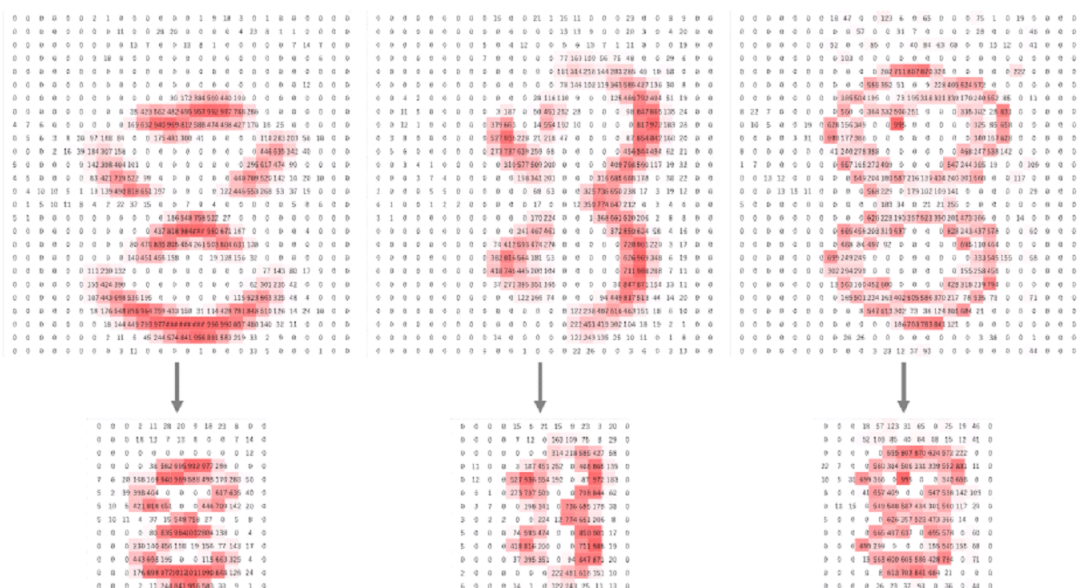


FIGURA 8: Funcionamiento de filtros (adaptado de [15]). Se pueden observar 3 diferentes imágenes con valores numéricos para cada píxel a las que se les ha aplicado Pooling, el resultado de éste proceso para cada una se puede observar debajo de la misma.

La red neuronal del final entrega como resultado la información que logró obtener a partir de las características encontradas en la imagen. En el caso de una imagen de la hoja de un libro, entregará el texto que hay en ella [10].

8.3. Acompañamiento y asesorías externas

En el inicio del proyecto surgen una serie necesidades por el desconocimiento de algunas temáticas, como lo son necesidades de las personas con falta parcial o total del sentido de la vista en escenarios de enseñanza presencial, por lo que se determina tomar una serie de acompañamientos y asesorías de instituciones o personal externo de la Facultad de Ingeniería Electrónica de la universidad Santo Tomás entre ellas podemos encontrar:

8.3.1. Instituto Nacional Para Ciegos

El Instituto Nacional para Ciegos (INCI) es una entidad de carácter técnico asesor adscrita al ministerio de Educación, creada mediante el Decreto 1955 del 15 de Julio de 1955, esta institución trabaja para la comunidad de discapacitados por ceguera o pérdida parcial de la visión en Colombia para garantizar y ayudar en temas de inclusión social, educativa, económica, política y cultural.

Como guía para el desarrollo del proyecto se hace una serie de reuniones con el INCI con el propósito de afianzar el proyecto e informar a dicha institución del proyecto para su aprobación y futuras colaboraciones en el mismo, en dicha reunión se plantearon temas como lo son:

- Objetivo esperado por el INCI hacia este tipo de proyectos
- Problemáticas a tratar en un aula de enseñanza presencial
- Herramientas tecnológicas existentes y conocidas por el INCI
- Sugerencias para el proyecto

Teniendo en cuenta los temas que se plantearon en dicha reunión se encontró que el objetivo esperado por el INCI respecto al proyecto es la búsqueda de estrategias para garantizar la entrega de información de los estudiantes en un aula de enseñanza presencial de manera equitativa para cada uno de los estudiantes sin importar las circunstancias. Como justificación a lo anterior

se plantea un problema que es resumido a la problemática de la falta de acceso a la información de los estudiantes con algún tipo de discapacidad visual, por medios como el profesor que da explicación verbal (explicación por medio de gestos) y algunas expresiones gráficas que no son entendidas por dichos estudiantes sin algún tipo de ayuda, para esto ya existen algunas herramientas tecnológicas como lo son:

- Lecturas de libros por reconocimiento óptico de caracteres y conversión del mismo en voz sintética.
- Masificación de imágenes o textos en dispositivos.
- Lector de pantalla.
- Impresoras braille.
- Sistema OCR.

Para finalizar se le plantea el proyecto al INCI obteniendo su aval para el desarrollo y su acompañamiento, con esto se comunica un tipo de sugerencia para el funcionamiento del software el cual consiste en tratar de interpretar la información entregada a los estudiantes por medio gestos y explicarlo o avisar al profesor que está haciendo cosas que los estudiantes no pueden interpretar como también en el caso que el programa no pueda detectar las diferentes formas gráficas.

Ésta reunión se desarrolló con personal autorizado del INCI, entre los participantes podemos encontrar a este personal:

- Miryam Herrera (Formación optómetra, Grupo de investigación)
- Patricia Montoya (Sub dirección técnica, Grupo de investigación)
- Gustavo Pulido (Sub dirección del INCI)
- Santiago Adolfo Rodriguez Alvarez (Profesional Especializado)

Capítulo 9

Desarrollo Metodológico

Para la realización del presente trabajo es necesario seguir un procedimiento que busca completar, uno a uno, los objetivos específicos planteados en este documento. Para abordarlos de manera sencilla se dispondrán en diferentes subsecciones, presentadas a continuación:

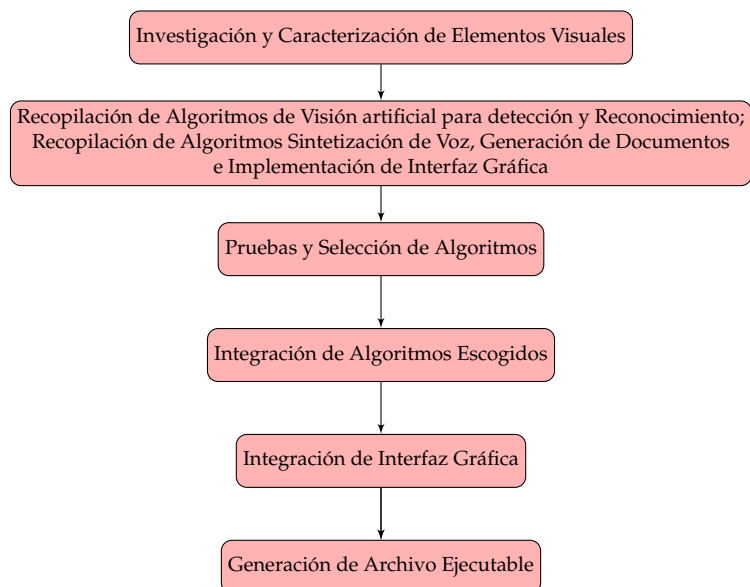


FIGURA 9: Esquema del Desarrollo Metodológico. Este esquema resume a rasgos generales la metodología empleada. Fuente: autores

9.1. Investigación y caracterización de los elementos visuales

Este proyecto está enfocado en la población ciega y con problemas visuales que se encuentra en estratos bajos, por ello las herramientas brindadas no están diseñadas para representar un gran coste computacional y monetario, lo que conlleva una sencillez en sí mismas.

Para el desarrollo de este proyecto es necesario caracterizar elementos visuales en el escenario de enseñanza presencial, para población con discapacidad visual. Esto se lleva a cabo en primer lugar por reuniones con personal autorizado del INCI, con el fin de recolectar información sobre estos elementos. Después, a partir de estos datos recolectados, se determinan las diferentes características de dichos elementos (el agendamiento de éstas reuniones se puede ver en los anexos 4 y 5).

Para determinar estas características se parte de todos los elementos que albergan información o contenido útil para las personas que se encuentran en el aula de clase presencial. Mucha de la información de un aula se encuentra en carteleras, tableros, pantallas, paredes y, por supuesto, profesores. De estos objetos, los más comunes y utilizados suelen ser las carteleras y tableros, además de los maestros.

Estos elementos, la mayoría de las veces, contienen información en forma de texto, lo cual los hace de difícil acceso para las personas con problemas de visión parcial o total y por ello son seleccionados para su caracterización y análisis más adelante.

Los elementos visuales caracterizados se presentan en el cuadro 1:

Elementos de un Aula de Clase Presencial		
Elementos	Características	
Tablero	Escritos	Información de tipo texto como frases, párrafos, entre otros.
	Gráficas	Datos o información presentados en forma gráfica
	Dibujos	Expresiones gráficas para la ayuda del aprendizaje
	Ecuaciones	Expresiones matemáticas (números y símbolos.)
Carteleras	Información en forma de texto	Información de tipo texto como frases, párrafos, entre otros.
Profesor	Gesticulación Facial	Comunicación por medio de gestos (rostro) y sus diferentes emociones
	Movimientos del Cuerpo	Comunicación por medio de gestos (brazos y manos)

CUADRO 1: Elementos identificados del aula de clases

Se determinan para el uso de los elementos tablero y profesor presentados en la tabla, con sus diferentes características exceptuando a las gráficas y los dibujos.

9.2. Recopilación de algoritmos

Teniendo en cuenta la caracterización de los elementos visuales, se empieza un proceso de investigación respecto a los diferentes algoritmos de visión artificial existentes, esto lleva a tener una recopilación de algoritmos para la detección e identificación de elementos por medio de una cámara, con esto se llevan a cabo pruebas a los algoritmos encontrados, con el fin de determinar cuál sería el algoritmo o algoritmos útiles para esta aplicación.

Los algoritmos encontrados se muestran en el cuadro 2:

Algoritmos		
Nombre	Descripción (Características)	Características Necesarias
Tesseract (Google)	-Abstracción de caracteres a partir de imágenes -Online y Offline[37]	Cámara, Python 3.6 o superior, conexión a internet
gtts (Google)	-TTS de Google (Python) -Online[38]	Cámara, Python 3.6 o superior, conexión a internet
pyttsx3	-Librería TTS python -Supports multiple TTS engines, including Sapi5, nsss, and espeak. -Offline [39]	Dispositivos de audio, Python 3.6 o superior
FER Library	-Detección de emociones de caras humanas -Requiere OpenCV y TensorFlow [40]	Cámara, Python 3.6 o superior
Priya face_and_emotion_detection	-No está muy trabajado -Poca documentación [41]	Cámara, Python 3.6 o superior
Facial-emotion-recognition 0.3.4	-Necesita Pytorch, Torchvision -No está muy trabajado -Poca documentación [42]	Cámara, Python 3.6 o superior
Atulapra/Emotion-detection	-Real-time Facial Emotion Detection -No está muy trabajado -Poca documentación [43]	Cámara, Python 3.6 o superior
Mathpix	-Imágenes de matemáticas a texto [44]	Cámara, Python 3.6 o superior

CUADRO 2: Recopilación de algoritmos

9.3. Pruebas de algoritmos

9.3.1. Pruebas de algoritmo Tesseract (Google)

Teniendo en cuenta la información de la tabla 2 se escoge una serie de algoritmos para probar y observar su funcionamiento, mirando sus características. Ésto con la finalidad de escoger entre lo algoritmos que más se adaptan a las necesidades del proyecto, teniendo en cuenta que las pruebas se realizaron en Ubuntu 18.04 LTS y Ubuntu 20.04 LTS

Para Las pruebas realizadas al algoritmo Tesseract se emplean distintas imágenes que contengan textos y se observan los datos entregados por Tesseract. Para determinar la efectividad se realizaron 20 pruebas con diferentes fotos, en las cuales su característica principal es contener texto escritos, en donde el algoritmo tuvo un 90 % de acierto.

El pseudocódigo correspondiente al código utilizado para la prueba es el siguiente:

```
1      Importar librerias
2
3      ruta_imagen = entrada_texto()
4      imagen = abrir_imagen(ruta_imagen)
5      texto = pytesseract_imagen_a_texto(imagen)
6      imprimir(imagen)
```

Entre las pruebas efectuadas se realiza una la cual contiene a la figura 10 como entrada, donde el resultado de la prueba se muestra en la figura 11.

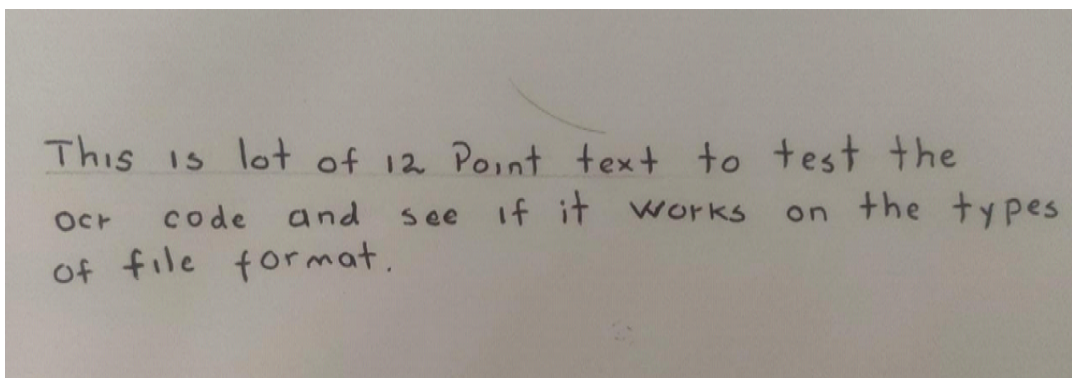
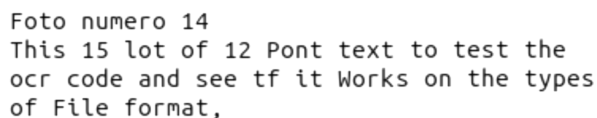


FIGURA 10: Imagen de entrada para un prueba de Tesseract Fuente: Autores.



```
Foto numero 14
This 15 lot of 12 Pont text to test the
ocr code and see tf it Works on the types
of File format,
```

FIGURA 11: Resultado de la prueba para la imagen de la Figura 9 Fuente: Autores.

Teniendo en cuenta el resultado de las pruebas se realiza un pequeño resumen de algunas características que presenta dicho algoritmo, se encontró que el algoritmo funciona con fotos de texto de diferente color, con algunas restricciones, el texto se debe encontrar de forma horizontal sin inclinaciones, los caracteres especiales o información que no son texto las detecta como ruido y las traduce en información basura y no importa qué color sea el fondo de la imagen desde que se diferencia notoriamente del color de la letra.

9.3.2. Pruebas de algoritmo gtts (Google)

Para las pruebas realizadas al algoritmo gTTS se emplean diferentes textos como información de entrada y se observan que los datos de salida del algoritmo es un archivo mp3 que se guarda en el dispositivo, con la información del texto entrada en forma de audio.

En ésta prueba se ingresa el siguiente texto "La USTA es una universidad con buenos maestros pero nos hacen trabajar mucho y no nos dejan dormir" y se obtiene como resultado un archivo .mp3 (anexo 1).

El pseudocódigo correspondiente al código utilizado para la prueba es el siguiente:

```
1  importar librerias
2
3  texto = entrada_texto()
4  audio = gTTS(texto, lenguaje='español')
5  audio.guardar('audio.mp3')
```

El archivo de audio resultante se presenta en el Anexo 1.

Teniendo en cuenta el resultado de las pruebas se realiza un pequeño resumen de algunas características que presenta dicho algoritmo, se encuentra que algoritmo gTTS toma cualquier texto y lo guarda en un archivo mp3, pero no cuenta con la funcionalidad de reproducir directamente el audio, por lo cual es necesario guardar el archivo mp3 y reproducirlo mediante un módulo de una sola función sin dependencias para reproducir sonidos llamado Playsound.

9.3.3. Pruebas de algoritmo gtts (Google)

Para PyTTS se emplea la misma metodología que con gTTS en donde se emplean diferentes textos como información de entrada y se observan que los datos de salida del algoritmo en un archivo mp3, teniendo en cuenta el resultado de las pruebas se realiza un pequeño resumen de algunas características que presenta dicho algoritmo, el algoritmo toma cualquier texto y lo guarda en un archivo mp3 y tiene la capacidad de reproducir el audio automáticamente.

El Pseudocódigo correspondiente para probar PyTTS es el siguiente:

```
1     importar librerias
2
3     texto = entrada_texto()
4     motor = pytts.iniciar()
5     motor.poner_propiedad('voz', 'español-latino')
6
7     motor.guardar_a_archivo(texto)
8     motor.correr_esperar()
```

9.3.4. Pruebas de algoritmo FER Library

Para las pruebas realizadas al algoritmo FER se emplean diferentes imágenes como información de entrada y como información de salida se obtendría la clasificación de la emoción que presenta la persona en ese momento, dichas información corresponde a una imagen de una persona, en donde la persona podía encontrarse de cuerpo completo o solamente de su parte superior de su cuerpo, pero manteniendo siempre la cara fácilmente reconocible.

El Pseudocódigo correspondiente para probar la librería FER es el siguiente:

```
1     importar librerias
2
3     ubicacion_imagen = entrada_texto()
4     imagen = leer_imagen(ubicacion_imagen)
5     detector = FER()
6     resultados = detector.detectar_emociones(imagen)
7     imprimir(resultados)
```

A continuación, se presentara el resultado de una prueba realizada con la figura 12 en donde se busca clasificar la emoción de la persona en cuestión obteniendo como resultado los datos

obtenidos en la figura 13 en donde la emoción con mayor probabilidad es *Happy* (Contento) con un 0.8



FIGURA 12: Imagen empleada para probar la librería FER Fuente: Autores.

Foto numero 13

```
2021-05-04 08:24:42.824409: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not
creating XLA devices, tf_xla_enable_xla_devices not set
```

RESULTADOS:

```
[{'box': (683, 369, 539, 539), 'emotions': {'angry': 0.0, 'disgust': 0.0, 'fear':
: 0.01, 'happy': 0.8, 'sad': 0.06, 'surprise': 0.01, 'neutral': 0.1}}]
```

Emoción más probable: happy : 0.8

FIGURA 13: Resultados de la prueba con FER utilizando la figura 12 como entrada
Fuente: Autores.

A continuación, se presentara el resultado de una prueba realizada con la (figura 14) en donde se busca clasificar la emoción de la persona diferente a la prueba realizada en la figura 12), teniendo en cuenta que se cambia el sujeto en cuestión y la emoción presentada en el rostro del sujeto, obteniendo como resultado los datos obtenidos en la figura (figura 15) en donde la emoción con mayor probabilidad es *Sad* (triste) con un 0.82



FIGURA 14: Imagen empleada para probar la librería FER Fuente: Autores.

Foto numero 16

```
2021-05-04 08:24:53.810241: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not  
creating XLA devices, tf_xla_enable_xla_devices not set
```

RESULTADOS:

```
[{'box': (674, 141, 595, 595), 'emotions': {'angry': 0.03, 'disgust': 0.0, 'fear': 0.08, 'happy': 0.02, 'sad': 0.82, 'surprise': 0.01, 'neutral': 0.06}}]  
Emoción más probable: sad : 0.82
```

FIGURA 15: Resultados de la prueba con FER utilizando la figura 15 como entrada
Fuente: Autores.

9.3.5. Pruebas de los algoritmos *Priya*, *facial-emotion-recognition* y *Atulapra-Emotion-detection*

Para los algoritmos *Priya*, *facial-emotion-recognition* y *Atulapra-Emotion-detection* se emplean las mismas pruebas que para la librería FER, pero sus resultados no son nada fiables debido a que sus resultados podían variar respecto a la misma imagen, provocando que la probabilidad de las emociones fluctuara considerablemente en las diferentes pruebas en una misma imagen.

9.3.6. Pruebas de algoritmo Mathpix

Para las pruebas realizadas al algoritmo Mathpix se emplean diferentes imágenes como información de entrada, donde se pueden evidenciar expresiones matemáticas y como información de salida se obtendría un json en el cual se puede encontrar campos con la información en forma de texto y código latex.

El Pseudocódigo correspondiente para probar la librería Mathpix es el siguiente:

```
1  importar librerías
2
3  ubicacion_archivo = entrada_texo()
4  uri_imagen = "data:image/jpeg;base64," + convertir_base64(abrir(ubicacion_archivo))
5  peticion = peticiones_http.pedir("https://api.mathpix.com/v3/text",
6                                   data=json.(uri_imagen))
7  imprimir(peticion.texto)
```

En las (figura 16) y (figura 18) se presentan imágenes con ejemplos de expresiones matemáticas y la respuesta dada por el algoritmo son plasmadas en las imágenes (figura 17) y (19), en donde se puede encontrar que el campo "latex_styled" nos da la expresión matemática en un formato LaTeX.

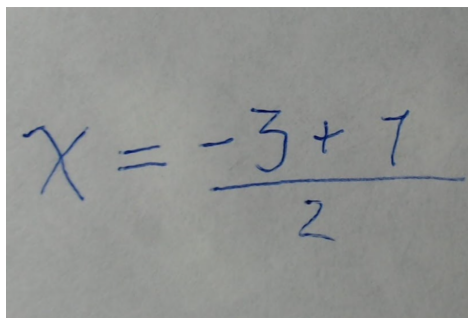

$$x = \frac{-3 + 7}{2}$$

FIGURA 16: Foto como entrada para una prueba Mathpix Fuente: Autores.

```
Foto numero 6
{
  "auto_rotate_confidence": 0.027831692615791326,
  "auto_rotate_degrees": 0,
  "confidence": 0.915820067981258,
  "confidence_rate": 0.915820067981258,
  "is_handwritten": true,
  "is_printed": false,
  "latex_styled": "x=\\frac{-3+1}{2}",
  "request_id": "3e59e0108c375d83ee2fb9fcd605fee6",
  "text": "\\( x=\\frac{-3+1}{2} \\)"
}
```

FIGURA 17: Resultado de la prueba de Mathpix utilizando la figura 16 como entrada Fuente: Autores.

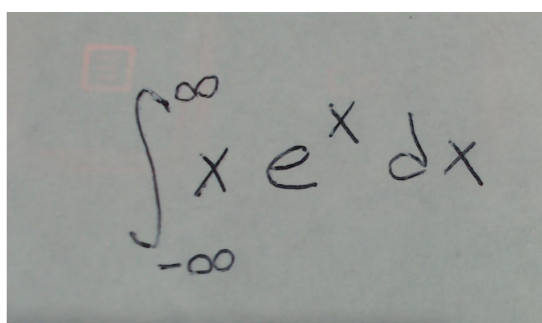


FIGURA 18: Foto como entrada para una prueba Mathpix Fuente: Autores.

```
Foto numero 2
{
  "auto_rotate_confidence": 0.035545827793825424,
  "auto_rotate_degrees": 0,
  "confidence": 1,
  "confidence_rate": 1,
  "is_handwritten": true,
  "is_printed": false,
  "latex_styled": "\\int_{-\\infty}^{\\infty} x e^{x} d x",
  "request_id": "28535a88ba73a158d85d978af2b37694",
  "text": "\\( \\int_{-\\infty}^{\\infty} x e^{x} d x \\)"
}
```

FIGURA 19: Resultado de la prueba de Mathpix utilizando la figura 18 como entrada Fuente: Autores.

El algoritmo Mathpix es capaz de reconocer todas las ecuaciones y las entrega en formato LaTeX. También puede entregarlas en otros formatos: markdown de Mathpix, HTML y solo la información HTML, sin etiquetas, según la configuración.

9.3.7. Recopilación de pruebas algoritmos

Se presentan los resultados de las diferentes pruebas de los algoritmos encontrados en el cuadro 3, en donde se plasman algunas características de las diferentes pruebas, como también algunas restricciones encontradas respecto a las pruebas realizadas y un cálculo de precisión respecto a las pruebas realizadas:

Pruebas			
Algoritmo	Resultado Pruebas	Restricciones	Precisión
Tesseract (Google)	El algoritmo funciona con fotos de texto de diferente color que desde que el texto se encuentra derecho, las cosas que no son texto las detecta como ruido y las traduce a basura, no importa qué color sea el fondo desde que se diferencia notoriamente del color de la letra.	Podría funcionar pero molesta con el tipo de letra, tienen que ser letra clara, se tienen que tener en cuenta la cuadrícula de algunos elementos. y se decidió que sólo se utilizará para textos.	Se realizaron 20 pruebas con textos diferentes y un 90 % de efectividad
gtts (Google)	El algoritmo toma cualquier texto lo guarda en un archivo mp3	Encontrar la forma de que se reproduzca sin guardar ese archivo	Se realizaron 15 pruebas con textos diferentes y un 100 % de efectividad
pyttsx3	El algoritmo toma cualquier texto lo guarda en un archivo mp3 y también reproduce el texto automáticamente	Buscar que sea simultáneo el guardado y la reproducción y la voz es menos entendible que el gtts	Se realizaron 10 pruebas con textos diferentes y un 100 % de efectividad, pero la voz no es entendible
FER Library	Fer funciona bien detecta tanto pasando solo la cara como pasando una foto de cuerpo completo y es preciso en las pruebas que se efectuaron	Si la emoción no está dentro de las predeterminadas eroga un vector nulo	Se realizaron 30 pruebas con textos diferentes y un 90 % de efectividad, pero la voz no es entendible
Priya face_and_emotion_detection	El programa no detecta muy bien las emociones tiene alto nivel de error en la predicción	Hacer mas pruebas mayor cantidad de fotos	Se realizaron 20 pruebas con textos diferentes y un 10 % de efectividad, pero la voz no es entendible
facial-emotion-recognition 0.3.4	El programa no detecta muy bien las emociones tiene alto nivel de error en la predicción	Hacer más pruebas mayor cantidad de fotos	Se realizaron 20 pruebas con textos diferentes y un 15 % de efectividad, pero la voz no es entendible
Atulapra/Emotion-detection	El programa no detecta muy bien las emociones tiene alto nivel de error en la predicción	Hacer más pruebas mayor cantidad de fotos	Se realizaron 20 pruebas con textos diferentes y un 35 % de efectividad, pero la voz no es entendible
Mathpix	Detecta todas las ecuaciones y nos las entrega en formato latex	Codificar el formato latex a algo entendible para utilizar el gtts	Se realizaron 25 pruebas con textos diferentes y un 95 % de efectividad, pero la voz no es entendible

CUADRO 3: Recopilación de pruebas algoritmos

9.4. Integración de algoritmos

Con base a las pruebas realizadas se procede a realizar la integración de algoritmos con mayor porcentaje de efectividad en un solo software, de tal forma que todos sean utilizables sin presentar conflictos y las utilidades de cada uno se sumen para lograr el objetivo de obtener

datos significativos en el contexto del aula de clase presencial. Además de ello al grupo de algoritmos se incorporan algoritmos de TTS con la finalidad de poder comunicarle al usuario la información correspondiente al profesor de una forma entendible, adquirida de los otros algoritmos mediante sonido, lo que terminará de dar una funcionalidad completa al software. Ésta integración se correrá sobre un ordenador que tendrá varios periféricos como lo pueden ser cámara, micrófono, entre otros. Es menester aclarar que en este proyecto no se evalúa la optimización de dichos algoritmos para ejecutar en plataformas de bajo costo o bajo consumo.

9.4.1. Revisar Compatibilidad

Teniendo en cuenta los algoritmos escogidos se desarrolla un primer archivo en Python, con la finalidad de probar la compatibilidad entre las diferentes dependencias y librerías que cada algoritmo requiere. Este nuevo algoritmo emplea un menú por consola, el cual permite escoger la funcionalidad (de los otros algoritmos) que se desee probar como lo podemos ver a continuación en la figura 20.

```
Bienvenido a el menú de ayuda para personas con discapacidad visual a continuación
encontrará los algoritmos:
1 Tesseractt
2 GTTs
3 Fer
4 Matpix
5 Conversor latex PDF
```

FIGURA 20: Menú en consola del primer algoritmo de integración Fuente: Autores.

Mediante el menú se acceden y prueban una a una las funcionalidades, y a su vez, la compatibilidad entre las mismas y sus librerías. Se encuentra que todas las pruebas resultan satisfactorias. Las pruebas se realizan de la misma forma y con las mismas entradas que las pruebas anteriores con el fin de evitar variaciones debido a diferentes contextos.

9.4.2. Separación de Elementos

Es necesario separar los elementos captados por la cámara, puesto que en un solo fotograma pueden encontrarse varios elementos. Pasar estos elementos a todos los algoritmos no es eficiente, puesto que cada algoritmo funciona para cosas diferentes, además se procesaría la misma información varias veces. Por ello es necesario separar la imagen en varias imágenes más

pequeñas, que contengan un elemento en específico según la necesidad y según la información que contengan serán procesadas por un algoritmo en específico.

Para lograr esto es necesario implementar una red de detección que separe los elementos existentes en un escenario de enseñanza aprendizaje, los cuales ya fueron escogidos anteriormente en la sección 8.1.

9.4.2.1. Detección de Tableros Blancos

Para detectar tableros se requiere entrenar una red a partir de datos de tableros blancos, para esto se utiliza la librería de imágenes de Google (Open Images Dataset), la cuál es una gran base de datos de imágenes diseñadas para ser usadas en entrenamiento de redes neuronales y, en este caso, se buscan imágenes de tableros blancos, que son los que convencionalmente se encuentran en los salones de clase. Se seleccionan 2143 imágenes de para realizar el entrenamiento; el 87 % (1885) de las imágenes son utilizadas para entrenar la red, el 9 % (193) son utilizadas para validar y el 3 % (65) para probar los resultados.

Con el fin de ahorrar tiempo y recursos se reentrenará una red neuronal ya existente y de código abierto conocida como You Only Look Once (YOLO). Esta red neuronal tiene distintas versiones, se escoge la más pequeña en el caso de este proyecto puesto que lo que se desea detectar (tableros blancos) no requiere una red muy potente. Para entrenar ésta red se utilizan las imágenes obtenidas de la librería de imágenes de Google.

Ésta red entrenada es capaz de detectar tableros blancos en una imagen y encerrarlos en una caja, las coordenadas de ésta caja sirven para extraer el tablero de la imagen original en donde sea detectado. La red entrega 4 pares de números enteros, estos pares son las coordenadas de los vértices del cuadrado en donde el tablero blanco es encerrado figura 21.

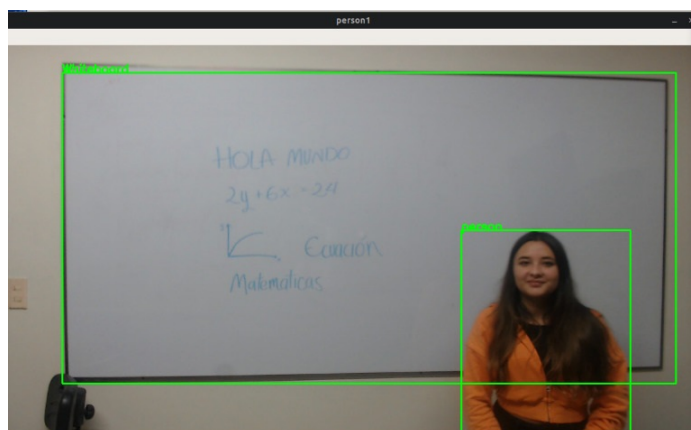


FIGURA 21: Prueba detección de Tableros Blancos Fuente: Autores.

Los resultados de éste entrenamiento se pueden ver con más detalle en la sección 9.2, Resultados del entrenamiento de la Red de Detección de Tableros Blancos.

9.4.2.2. Detección de Personas

Para la detección de Personas se hace uso nuevamente de la red neuronal de YOLO, ésta red está preentrenada para detectar gran cantidad de objetos y es bastante precisa, por lo cual brinda resultados coherentes para la función que se necesita, ahorra trabajo y tiempo en gran medida. De esta red solo se necesita detectar personas, por ello se filtran los resultados entregados por la misma, eliminando todo aquello que no sea una persona. De la misma forma que ocurre con la detección de tableros blancos, para cada cuadrado en donde sea detectada una persona, la red nos entrega coordenadas de sus 4 vértices figura 22.

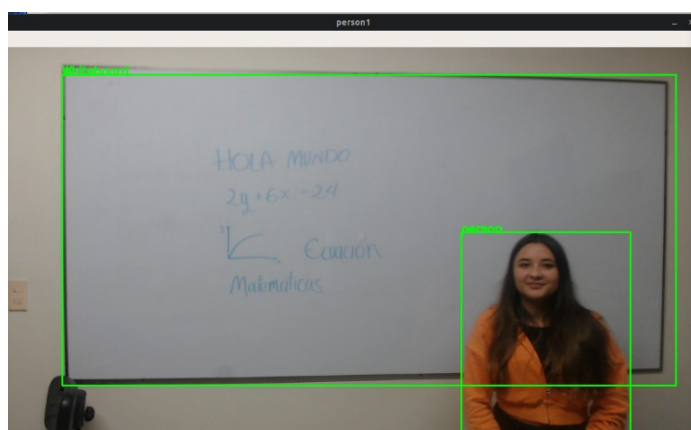


FIGURA 22: Prueba detección de Personas Fuente: Autores.

9.4.3. Integración

Teniendo en cuenta los pasos anteriores se desarrolla un programa en python el en cual se integra todos los algoritmos escogidos y la cámara, con un funcionamiento simultáneo y constante, teniendo en cuenta las diferentes funciones de cada algoritmo se toman algunas restricciones de toma de datos con el fin de generar la información en los momentos indicados para evitar distracciones en el estudiante a la hora de las explicaciones verbales del profesor.

Éste algoritmo realiza operaciones y análisis tomando un *frame* de la cámara cada 3 segundos. Éste tiempo cambia más adelante para que el usuario pueda indicarle al programa cuándo es el mejor momento para empezar con la adquisición de información. A cada uno de estos *frames* se le realizan las operaciones correspondientes a cada algoritmo, tomando la información pertinente para cada uno y enviándola al usuario mediante audio para que pueda acceder a ésta en algún otro momento según lo requiera.

9.4.4. Parámetros de cámara

Teniendo en cuenta las pruebas realizadas en los diferentes algoritmos (MathPix, Tesseract y FER), se encontró una necesidad de modificar los diferentes parámetros básicos de la cámara dado que dichos parámetros enfocan los objetos o personas más cercanas a la cámara y por ende haciendo que los objetos más lejanos no se vean claramente, esto conlleva a fallos en los diferentes algoritmos de detección de caracteres y reconocimiento facial; estos fallos incluyen:

- Detección errónea de los diferentes caracteres o emociones.
- No detección de ningún carácter o emoción.

Teniendo en cuenta esta necesidad se modifican los siguientes parámetros de la cámara, con el fin de eliminar la distracción del fondo y mejorar la calidad de la imagen capturada por medio de la cámara, este proceso es realizado utilizando la librería de OpenCV la cual nos da una gama de parámetros que podemos modificar: (poner referencia a la página de los parámetros documentación)

- Control del Valor de Zoom realizado por la cámara (CAP_PROP_ZOOM).
- Control de activación o desactivación de asistencia al enfoque realizado por la cámara (CAP_PROP_AUTOFOCUS).

- Control de exposición realizado por la cámara. (CAP_PROP_AUTO_EXPOSURE).
- Contraste de la imagen capturada por la cámara.(CAP_PROP_CONTRAST).
- Saturación de la imagen capturada por la cámara.(CAP_PROP_SATURATION).
- Exposición de la imagen capturada por la cámara.(CAP_PROP_EXPOSURE).
- nitidez de la imagen capturada por la cámara. (CAP_PROP_SHARPNESS).
- Brillo de la imagen capturada por la cámara. (CAP_PROP_BRIGHTNESS).
- Ganancia de la imagen capturada por la cámara. (CAP_PROP_GAIN).

Para finalizar se determina el tamaño específico de la ventana con el fin de controlar el tamaño del cuadro de transmisión del vídeo, el tamaño determinado es 854x480; Este proceso se realiza modificando los siguientes parámetros.

- Ancho de los cuadros en la transmisión de video (CAP_PROP_FRAME_WIDTH).
- Altura de los fotogramas en la secuencia de vídeo (CAP_PROP_FRAME_HEIGHT).

A continuación se evidencia el código utilizado para modificar los parámetros de la cámara mencionados anteriormente:

```
1  import cv2
2  vid = cv2.VideoCapture(1)
3  vid.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc('M', 'J', 'P', 'G'))
4  vid.set(cv2.CAP_PROP_FRAME_WIDTH, 854)
5  vid.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
6  vid.set(cv2.CAP_PROP_ZOOM, 0)
7  vid.set(cv2.CAP_PROP_AUTOFOCUS, 0.0)
8  vid.set(cv2.CAP_PROP_FOCUS, 0)
9  vid.set(cv2.CAP_PROP_AUTO_EXPOSURE, 1)
10 vid.set(cv2.CAP_PROP_CONTRAST, valor de contraste)
11 vid.set(cv2.CAP_PROP_SATURATION, valor de saturacion)
12 vid.set(cv2.CAP_PROP_EXPOSURE, valor de Exposición)
13 vid.set(cv2.CAP_PROP_SHARPNESS, valor de nitidez)
14 vid.set(cv2.CAP_PROP_BRIGHTNESS, valor de brillo)
15 vid.set(cv2.CAP_PROP_GAIN, valor de ganancia)
```

El tablero y la persona requieren diferentes conjuntos de parámetros, puesto que poseen diferentes colores y están en posiciones diferentes. Los parámetros del tablero modifican el enfoque

entre otros parámetros, para que los objetos más lejanos no se distorsionen, como el tablero; además los parámetros son modificados teniendo como objetivo los tableros blancos. Los parámetros para la cara, en cambio, enfocan objetos un poco más cercanos para obtener mejores imágenes del profesor.

Se muestra a continuación una imagen a la cual ya se le aplica el cambio de parámetros en la cámara con el fin de hacer más legible los diferentes contenidos del tablero figura 23.



FIGURA 23: Cambio Parámetros Captura Tablero Fuente: Autores.

Se muestra a continuación una imagen a la cual ya se le aplica el cambio de parámetros en la cámara con el fin de hacer mas visible la cara de la persona figura 24.

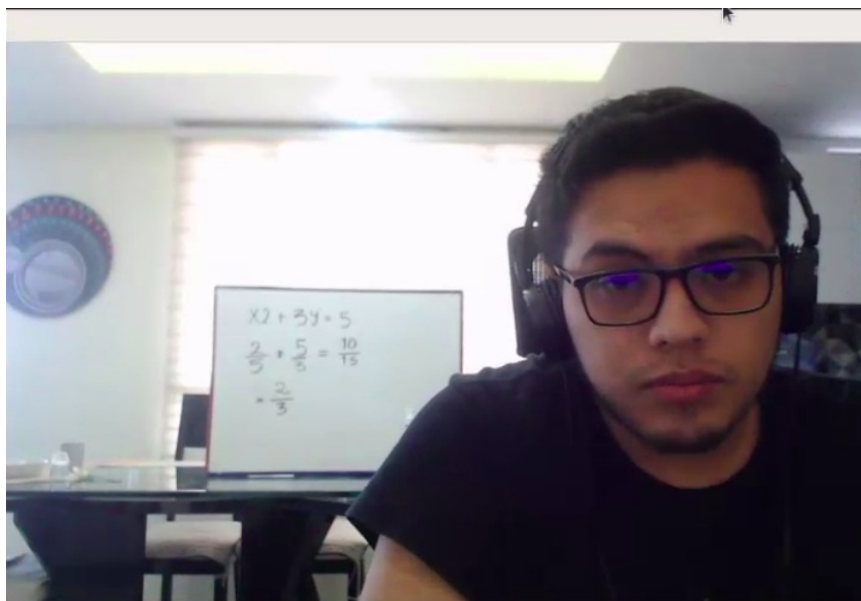


FIGURA 24: Cambio Parámetros Captura Tablero Fuente: Autores.

9.4.5. Generación de LaTeX y MathML

Del tablero blanco detectado por la red neuronal se extrae la información (texto y ecuaciones matemáticas), para esto se emplea otra red neuronal de reconocimiento óptico de caracteres (OCR por sus siglas en inglés) pre-entrenada y de pago. Ésta red ha sido entrenada con gran cantidad de datos lo que le otorga gran valor para el proyecto pues, nuevamente, es una red que ahorra trabajo y tiempo. Se hace uso de esta red enviando y recibiendo información vía HTML, para ello se emplea una librería de Python conocida como *requests*; ésta librería se encarga de enviar y recibir peticiones HTML, que son mensajes escritos y leídos para y por un servidor, con el fin de comunicarse con él.

Como resultado se obtienen, en formato JSON, los textos y expresiones matemáticas encontradas en la imagen, éste JSON contiene los resultados en diferentes formatos, entre ellos MathML y LaTeX.

Luego de recibir los resultados, el formato JSON facilita tomar los resultados en los formatos relevantes, que en este caso son los mismos MathML y LaTeX, luego, se guardan en una variable string para ser utilizados posteriormente.

9.4.6. Generación de PDF y HTML

Luego de que los datos son procesados por los distintos algoritmos, éstos son almacenados en un mismo documento PDF, el cual incluye la fecha, hora y materia en su nombre; en su interior se encuentra el contenido de LaTeX ya compilado figura 25.

$$\int_{-\infty}^4 x dx = 0$$

FIGURA 25: Ecuación almacenada PDF Fuente: Autores.

De la misma forma un documento HTML es generado con el contenido detectado en formato MathML, a continuación se presenta un ejemplo de un archivo HTML con una de las ecuaciones figura 26 en dicho formato.

```

1 <div><span class="math-inline "><mathml style=""><math xmlns="http://www.w3.org/1998/Math/MathML"
2   <msubsup>
3     <mo data-mjx-texclass="OP">&#x222B;</mo>
4     <mrow>
5       <mo>&#x2212;</mo>
6       <mi mathvariant="normal">&#x221E;</mi>
7     </mrow>
8     <mrow>
9       <mn>4</mn>
10    </mrow>
11  </msubsup>
12  <mi>x</mi>
13  <mi>d</mi>
14  <mi>x</mi>
15  <mo>=</mo>
16  <mn>0</mn>
17 </math></mathml></span></div>

```

$$\int_{-\infty}^4 x dx = 0$$

FIGURA 26: Ecuación almacenada archivo HTML-MathML Fuente: Autores.

9.4.7. Integración Interfaz Gráfica

Finalmente se diseña una interfaz gráfica con el fin de facilitar la interacción de los usuarios con el proyecto. Esta interfaz gráfica cuenta con dos pestañas, tienen el propósito evitar abrumar a los usuarios y mantener organizados los componentes de la interfaz.

9.4.7.1. Parte Visual

La primer pestaña contiene, en el medio, el vídeo capturado por la cámara en tiempo real, esto con el fin de facilitar al usuario la configuración y disposición de la cámara; a la izquierda de ésta imagen se encuentra el nombre completo del proyecto, el nombre de los desarrolladores y de la directora, además se encuentra un botón para escoger la ubicación en donde se desea guardar los archivos PDF y HTML, un texto que muestra la ubicación actual y otro texto que muestra las teclas a presionar con su respectiva función (se hablará de ello más adelante); sobre vídeo en tiempo real se encuentran dos imágenes, la primera contiene el recorte del tablero obtenido de la imagen capturada por la cámara (si lo hay), del mismo modo, la segunda contiene el recorte de la persona detectada (si la hay); En la parte superior se puede encontrar el nombre de la aplicación junto a los logos de la Universidad Santo Tomás y la Facultad de Ingeniería Electrónica, a la izquierda de éstos elementos se encuentra la fecha actual y a la derecha de éstos elementos se encuentra un desplegable que permite seleccionar la materia para la que se desea tomar datos.

La segunda pestaña contiene únicamente una imagen en donde se puede apreciar el contenido detectado del tablero, incluyendo ecuaciones matemáticas.

9.4.7.2. Parte Funcional

La interfaz reacciona en tiempo real al teclado, teclas en específico realizan acciones en específico. La tecla 'F' tomará una foto por medio de la cámara y la pasará por los distintos algoritmos

para procesar el tablero; La tecla 'E' hará lo mismo que la 'F' pero procesando las emociones de la persona detectada; la tecla 'J' terminará todos los subprocesos y cerrará el programa; finalmente, la tecla 'R' leerá la ruta de guardado para los archivos de salida (PDF y HTML) en voz alta.

El botón de cambio de ruta abrirá una nueva ventana al ser pulsado, ésta ventana permitirá al usuario escoger la ruta en donde desea que los archivos de salida sean guardados.

9.4.7.3. Parte Técnica

La ventana principal de la interfaz gráfica está programada utilizando una librería llamada Qt5, en su versión para Python (Pyqt5). El proceso principal se encarga de crear ésta ventana principal, cargar las librerías en memoria y de inicializar la cámara, además, éste crea un hilo, el cual se encarga de mostrar el vídeo capturado por la cámara en tiempo real. Cuando la tecla de lectura de ruta en voz alta ('R') es pulsada, un nuevo hilo es creado con el propósito de ejecutar el algoritmo Text To Speech para que la ruta sea leída en voz alta, luego, el hilo es terminado.

9.5. Ejecutable

Hasta éste momento, el software se compone de un conjunto de scripts, cada uno con un propósito diferente, además, las diferentes librerías integradas tienen sus propios scripts, que también hacen parte del software. Debido a ésta composición, el ejecutar la aplicación requiere el uso de la consola del sistema operativo y de comandos para que el script correcto sea ejecutado de la forma correcta.

Este funcionamiento supone una incomodidad para los usuarios con buen conocimiento del sistema operativo y sus herramientas, y un gran obstáculo para usuarios regulares que tienen conocimientos básicos; éste problema se soluciona fácilmente integrando todos los diferentes scripts y librerías (con sus respectivos scripts) en un solo archivo ejecutable convencional, de esta forma los usuarios con conocimientos básicos podrán ejecutar la herramienta sin complicaciones. Además, si se integran todas las librerías en el ejecutable, el usuario no tendrá que instalarlas aparte, una a una (como se hizo durante el desarrollo de la aplicación).

Para generar el ejecutable y hacer ése proceso sencillo se hace uso de otra librería (que no es incluida en el ejecutable), llamada Pyinstaller. Ésta librería se encargará de agrupar todo y generar el ejecutable, todo mediante comandos sencillos, lo que facilitará el proceso.

Capítulo 10

Resultados

10.1. Caracterización de Elementos Visuales.

Los Elementos Visuales más importantes en un aula de clases presencial para los estudiantes son el profesor y el tablero. El profesor es la persona que transmite la información a los estudiantes y por tanto, a los usuarios, al ser la principal fuente de información es considerado el elemento más importante, aunque la mayoría de información es suministrada de manera auditiva, su importancia hace que la información que transmita de manera visual sea muy importante también; por otra parte, el tablero es una parte esencial en un aula de clases presencial, puesto que allí es donde el profesor expondrá gran parte de la información que transmite a sus estudiantes, ésta información es totalmente visual.

10.2. Un Algoritmos para Detectar Elementos Visuales.

Para poder obtener la información que se encuentra en el tablero y las emociones del profesor, es necesario poder detectarlos primero, es decir, encontrar sus ubicaciones (coordenadas) dentro de una imagen. Para ello se opta por utilizar una Red Neuronal de Detección llamada *You Only Look Once* (YOLO). Esta red cuenta con la capacidad de detectar gran cantidad de diversos objetos en una imagen, pero los tableros blancos no están entre estos objetos, es por ello que se reentrena para brindarle la capacidad de detectarlos.

10.2.1. Reentrenamiento de la Red de Detección de Tableros Blancos.

Para reentrenar YOLO se utilizan imágenes de distintos contextos, pero que posean por lo menos un tablero blanco. Estas imágenes son utilizadas para entrenar, probar y validar la detección que presenta YOLO respecto a Tableros blancos. Como YOLO ya es una Red Definida, no es necesario jugar con la estructura de la red en sí, sino que se modificarán únicamente los parámetros de entrenamiento.

Luego de este entrenamiento, se obtiene una red que es capaz de detectar tableros blancos y extraer sus contenidos, para ésta red, los hiper-parámetros finales son los siguientes:

- lr0: 0.01
- lrf: 0.1
- momentum: 0.937
- weight_decay: 0.0005
- warmup_epochs: 3.0
- warmup_momentum: 0.8
- warmup_bias_lr: 0.1
- box: 0.05
- cls: 0.5
- cls_pw: 1.0
- obj: 1.0
- obj_pw: 1.0
- iou_t: 0.2
- anchor_t: 4.0
- fl_gamma: 0.0
- hsv_h: 0.015
- hsv_s: 0.7

- hsv_v: 0.4
- degrees: 0.0
- translate: 0.1
- scale: 0.5
- shear: 0.0
- perspective: 0.0
- flipud: 0.0
- fliplr: 0.5
- mosaic: 1.0
- mixup: 0.0
- copy_paste: 0.0

En la figura 27 se puede observar el comportamiento de la red por cada época, todo esto mientras se está entrenando:

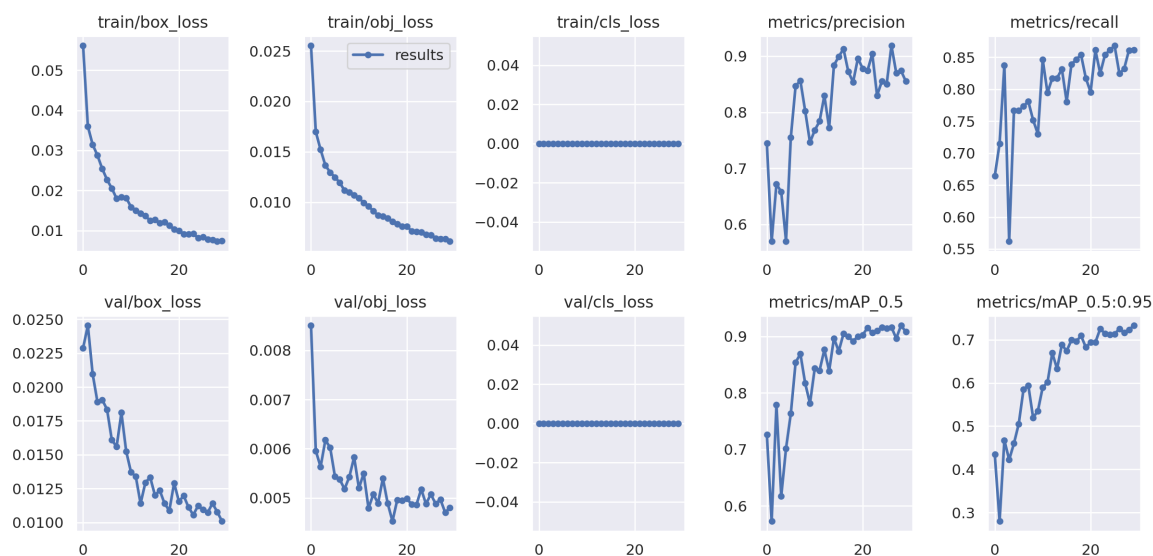


FIGURA 27: Evolución de resultados de la red en entrenamiento por época. Fuente: Autores

Con los datos de evaluación, se asegura que la confianza de la red va en aumento durante su entrenamiento y es registrada durante el proceso, con ésta información se compone la siguiente gráfica figura 28:

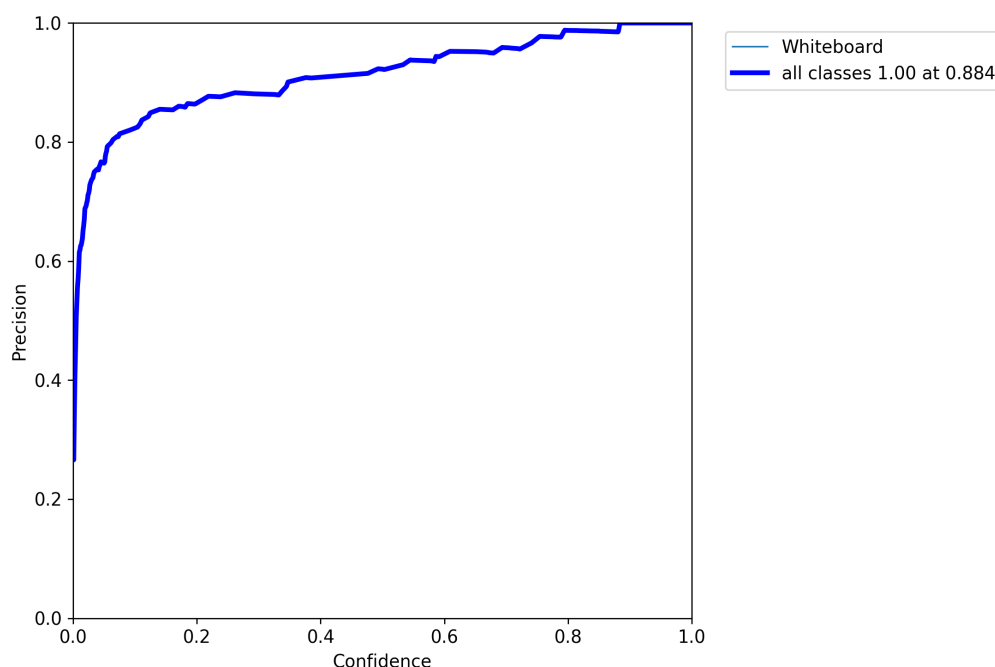


FIGURA 28: Evolución de la confianza de la red por época Fuente: Autores.

A manera de reporte, se genera la matriz de confusión para el entrenamiento de la red de detección. Con ésta matriz podemos determinar que tan buenos resultados ha dado el entrenamiento, y para nuestra red, ha dado resultados deseables.

La red resultante es capaz de detectar tableros en diversos escenarios como se presentan algunos resultados en la imagen figura 30. Después del entrenamiento, la red ha sido probada con diferentes imágenes que contienen tableros blancos en diferentes contextos. La red marca las ubicaciones en donde cree que existe un tablero blanco con un rectángulo rojo, en las imágenes de prueba la red ha marcado y encontrado los tableros de forma acertada figura 30.

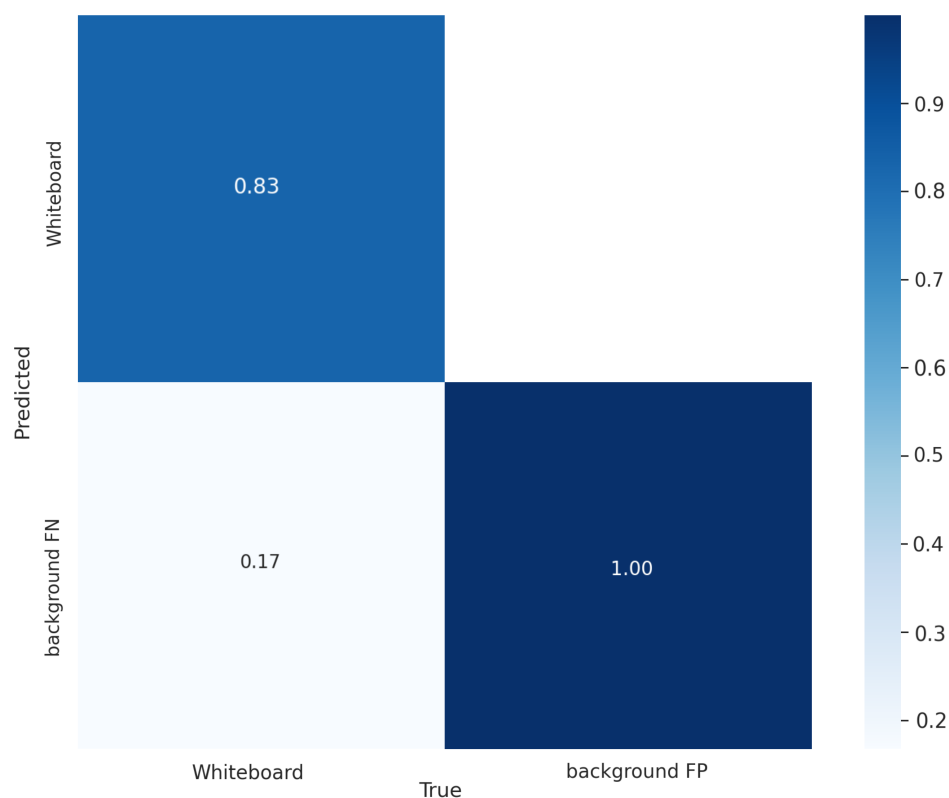


FIGURA 29: Matriz de confusión de la red de detección de tableros blancos Fuente: Autores.

En la matriz de confusión resultante se observa que los falsos positivos y falsos negativos (cuadrante superior derecho y cuadrante inferior izquierdo) tienen un porcentaje muy bajo, por tanto, demuestran que la red tiene una alta efectividad en detección de tableros blancos. En otras palabras, el 83 % de los tableros fueron detectados como tableros y no otra cosa; al únicamente existir una clase (tablero blanco), todas las otras cosas que no son esa clase (denominadas background en la matriz de confusión) nunca son detectadas como tableros blancos, por ello se tiene un 0 % de falsos positivos figura 29.



FIGURA 30: Resultados de algunas detecciones de prueba. Fuente: Autores.

De acuerdo con la metodología y los contenidos que se presentan en la interfaz gráfica, se ha optado por diseñar dos pestañas, pues el contenido queda mejor distribuido y es más cómodo y fácil de comprender figura 30 y figura 31.

En la figura 29 se muestra la pestaña principal de la interfaz gráfica la cual consta de varios elementos

- Botón de cambio de ubicación para guardado de archivos.

- *Labels* con instrucciones básicas
 - F para Captura
 - E Detección de emoción
 - j Cerrar programa
 - R Lectura de ruta de guardado
- Desplegable para escoger la materia
- Fecha de ejecución
- *Labels* de información básica desarrolladores *Software*
- Recuadros para las diferentes imágenes como lo son
 - Captura tablero
 - Captura persona
 - Captura prueba (cámara tiempo real)

Con los diferentes elementos de estas pestañas de la interfaz, se busca dar diferentes funcionalidades o dar información al usuario sobre el *software*, como lo es escoger la materia por medio del desplegable, con el fin de poder guardar los archivos con un nombre de materia específico y una fecha que será dada por el sistema a la hora de guardar los archivos, como también cambiar la ruta en donde se guardan dichos archivos, teniendo en cuenta que si no es cambiada los archivos se guardarán en la misma carpeta del ejecutable,

En la figura 31 se muestra la pestaña secundaria de la interfaz gráfica la cual consta de un solo elemento, el cual es un recuadro para mostrar la información capturada del tablero, inicialmente este recuadro estará en blanco hasta que se ejecute la instrucción de la letra F y sea detectada información en el tablero, si no se encuentra información en el tablero el recuadro seguirá en blanco

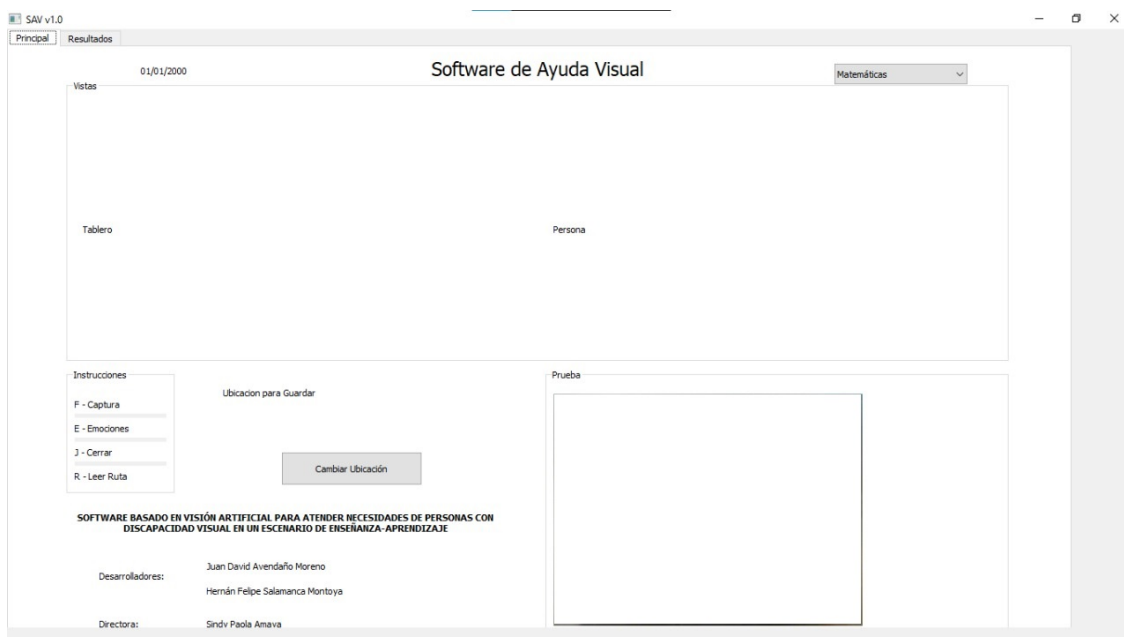


FIGURA 31: Diseño de la interfaz gráfica en la primer pestaña Fuente: Autores.

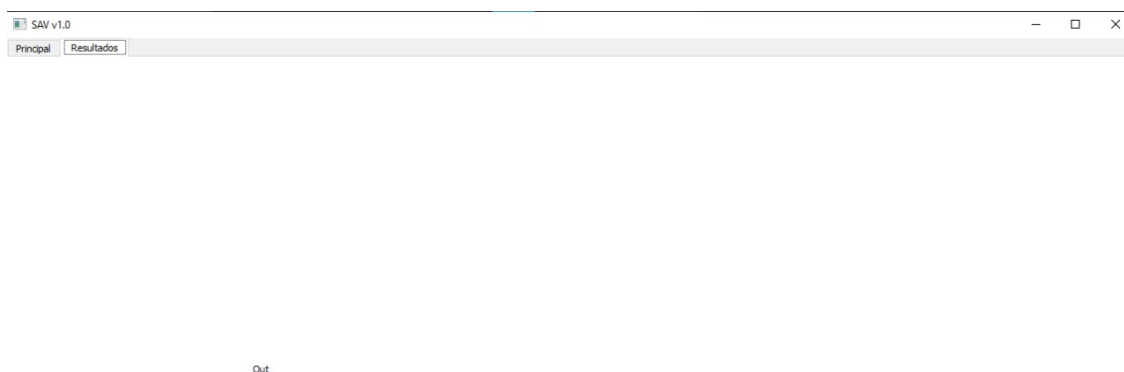


FIGURA 32: Diseño de la interfaz gráfica en la segunda pestaña Fuente: Autores.

Además, se realizan pruebas con un tablero blanco y escritura manual, con el objetivo de probar si el programa funciona correctamente en entornos que nunca ha visto. En éstas pruebas se observa que es capaz de detectar correctamente los textos y las ecuaciones matemáticas escritas.

10.2.2. Pruebas del *Software*

A continuación se presenta una serie de pruebas realizadas en el software:

10.2.2.1. Primer Prueba

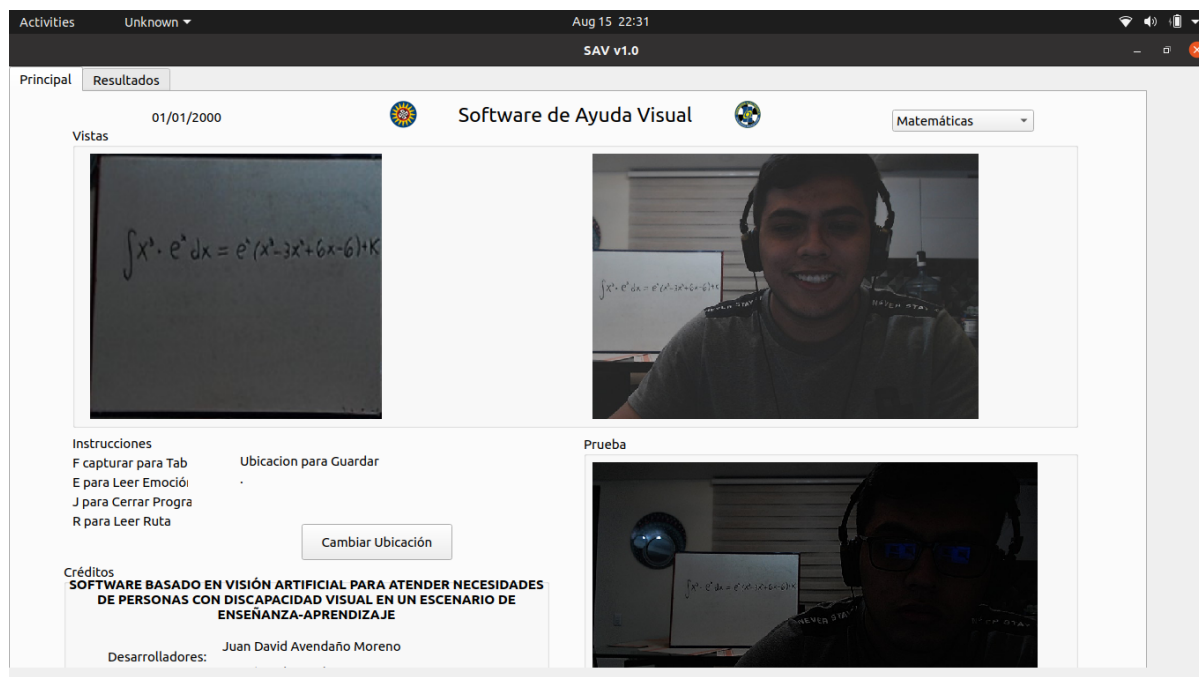


FIGURA 33: Interfaz gráfica correspondiente a la primer prueba del *software*.
Fuente: Autores.

Para la primer prueba, en la figura 33 se plantea una ecuación con una integral y una expresión de felicidad en el rostro de individuo que actúa como profesor y como resultado se obtiene que, en la segunda pestaña, aparece la ecuación correspondiente (figura 34) a la toma de información que se observa en la figura 33.

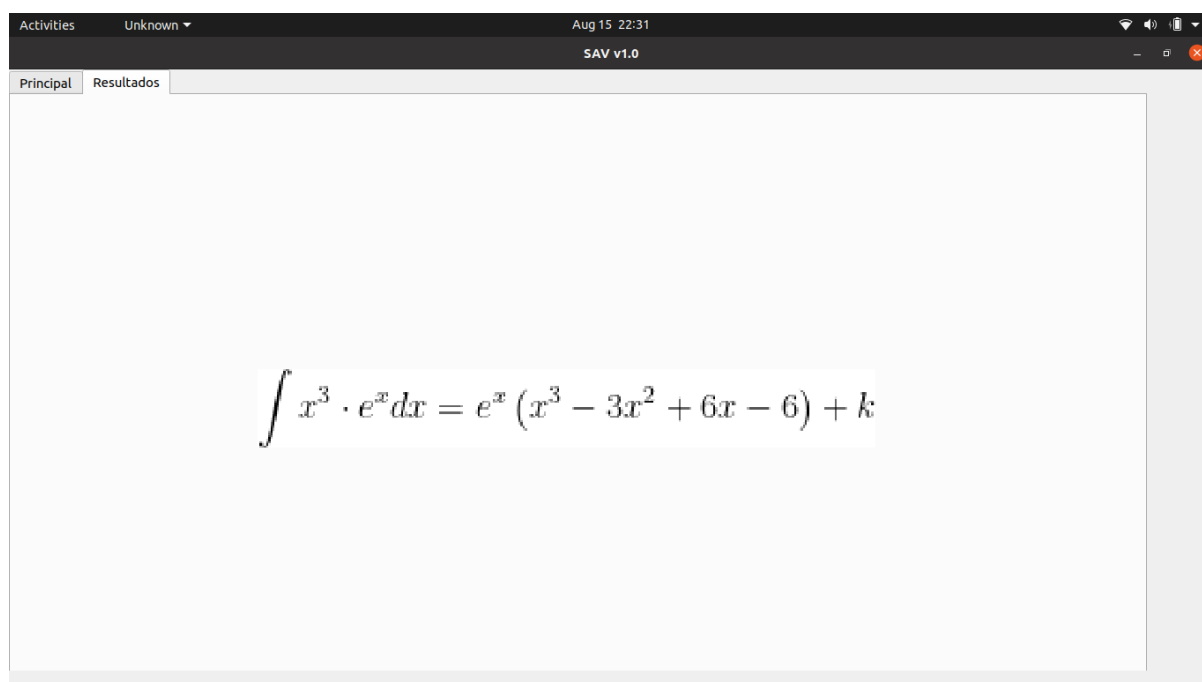


FIGURA 34: Resultado mostrado en la segunda pestaña del *software* para la primer prueba. Fuente: Autores.

De la misma forma se generan dos archivos, el primero es un documento PDF, que se puede observar en la figura 35 y el segundo un es documento HTML como se ve en la figura 36; éstos documentos contienen la información detectada en el tablero para la primer prueba (figura 33).

$$\int x^3 \cdot e^x dx = e^x (x^3 - 3x^2 + 6x - 6) + k$$

FIGURA 35: Contenido del documento PDF para la primer prueba. Fuente: Autores.

$$\int x^3 \cdot e^x dx = e^x (x^3 - 3x^2 + 6x - 6) + k$$

FIGURA 36: Contenido del documento HTML para la primer prueba. Fuente: Autores.

A continuación se presenta el contenido del archivo HTML de la figura 36:

```

1 <div><span class="math-inline "><mathml style=""><math xmlns="http://www.w3.org/1998
2 /Math/MathML">
3   <mfrac>
4     <mn>2</mn>
5     <mn>5</mn>
6   </mfrac>
7   <mo>⋅</mo>
8   <mfrac>
9     <mn>4</mn>
10    <mn>7</mn>
11  </mfrac>
12  <mo>=</mo>
13  <mfrac>
14    <mrow>
15      <mn>2</mn>
16      <mo>⋅</mo>
17      <mn>4</mn>
18    </mrow>
19    <mrow>
20      <mn>5</mn>
21      <mo>⋅</mo>
22      <mn>7</mn>
23    </mrow>
24  </mfrac>
25  <mo>=</mo>
26  <mfrac>
27    <mn>8</mn>
28    <mn>35</mn>
29  </mfrac>
30 </math></mathml></span></div>

```

Para la primer prueba (figura 33) el *software* genera un audio que corresponde a la emoción mas probable del individuo (anexo 6).

10.2.2.2. Segunda Prueba

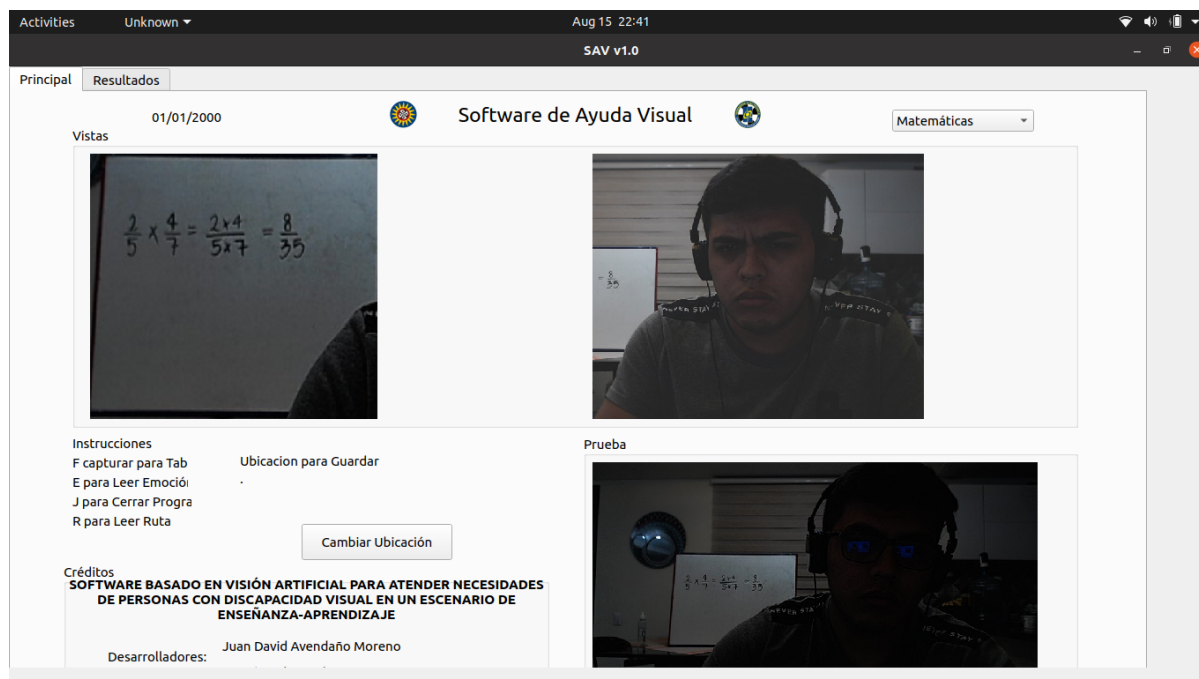


FIGURA 37: Interfaz gráfica correspondiente a la segunda prueba del *software*.
Fuente: Autores.

Para la segunda prueba, en la figura 37 se plantea una ecuación con fraccionarios y una expresión de enojo en el rostro del individuo que actúa como profesor, como resultado se obtiene que en la segunda pestaña aparece la ecuación de la figura 38, correspondiente a la toma de información de la figura 37.

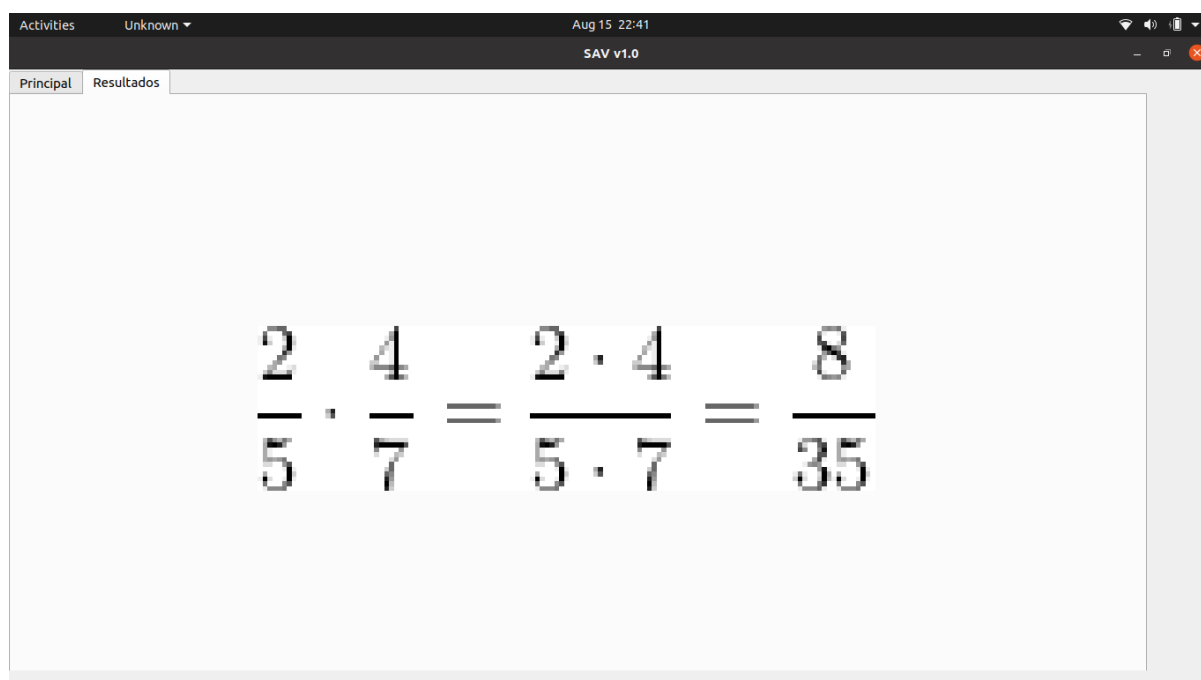


FIGURA 38: Resultado mostrado en la segunda pestaña del *software* para la segunda prueba. Fuente: Autores.

De la misma forma se generan dos archivos, el primero un documento PDF, que se puede observar en la figura 39 y el segundo es un documento HTML como se ve en la figura 40; éstos documentos contienen la información detectada en el tablero para la segunda prueba (figura 37).

$$\frac{2}{5} \cdot \frac{4}{7} = \frac{2 \cdot 4}{5 \cdot 7} = \frac{8}{35}$$

FIGURA 39: Contenido del documento PDF para la segunda prueba. Fuente: Autores.

$$\frac{2}{5} \times \frac{4}{7} = \frac{2 \times 4}{5 \times 7} = \frac{8}{35}$$

FIGURA 40: Contenido del documento HTML para la segunda prueba. Fuente: Autores.

A continuación se presenta el contenido del archivo HTML de la figura 40:

```

1  <div><span class="math-inline "><mathml style=""><math xmlns="http://www.w3.org/1998
2  /Math/MathML">
3    <mfrac>
4      <mn>2</mn>
5      <mn>5</mn>
6    </mfrac>
7    <mo>⋅</mo>
8    <mfrac>
9      <mn>4</mn>
10     <mn>7</mn>
11   </mfrac>
12   <mo>=</mo>
13   <mfrac>
14     <mrow>
15       <mn>2</mn>
16       <mo>⋅</mo>
17       <mn>4</mn>
18     </mrow>
19     <mrow>
20       <mn>5</mn>
21       <mo>⋅</mo>
22       <mn>7</mn>
23     </mrow>
24   </mfrac>
25   <mo>=</mo>
26   <mfrac>
27     <mn>8</mn>
28     <mn>35</mn>
29   </mfrac>
30 </math></mathml></span></div>

```

Para la segunda prueba (figura 33) se genera un audio que corresponde a la emoción mas probable del individuo (anexo 7).

10.2.2.3. Tercer Prueba

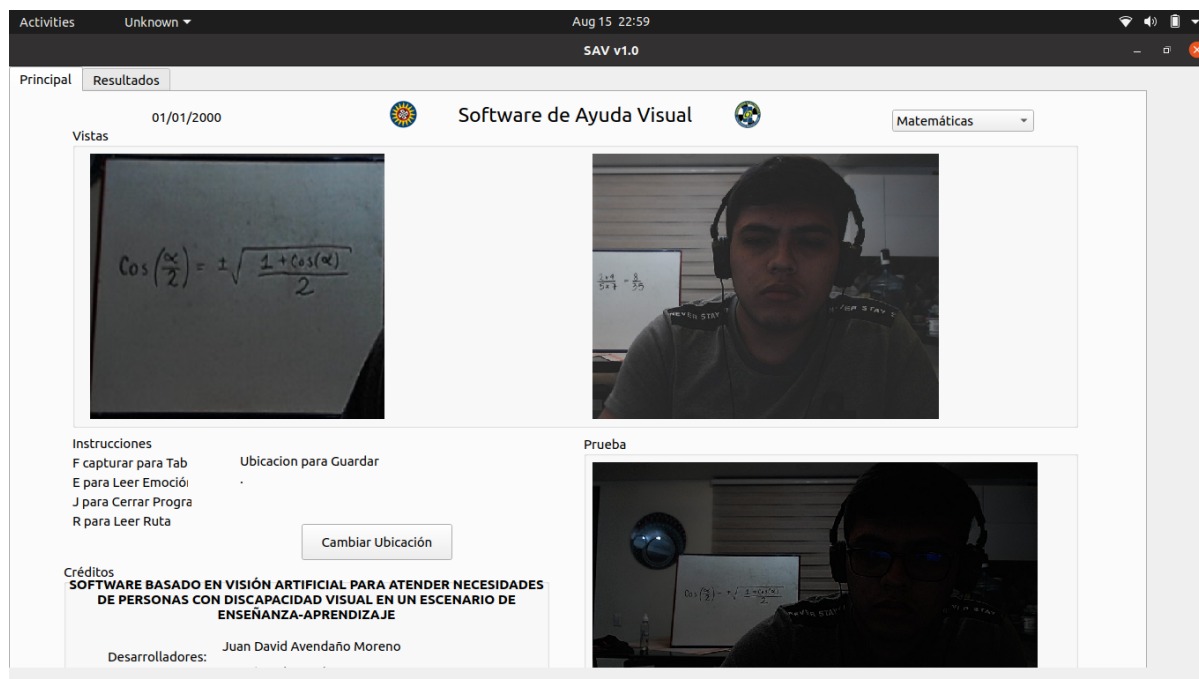


FIGURA 41: Interfaz gráfica correspondiente a la tercer prueba del *software*. Fuente: Autores.

En la tercer prueba (figura 41) se plantea una ecuación matemática y una expresión de tranquilidad en el rostro de individuo que actúa como profesor, como resultado se obtiene que en la segunda pestaña aparece la ecuación de la figura 42, correspondiente a la toma de información de la figura 41.

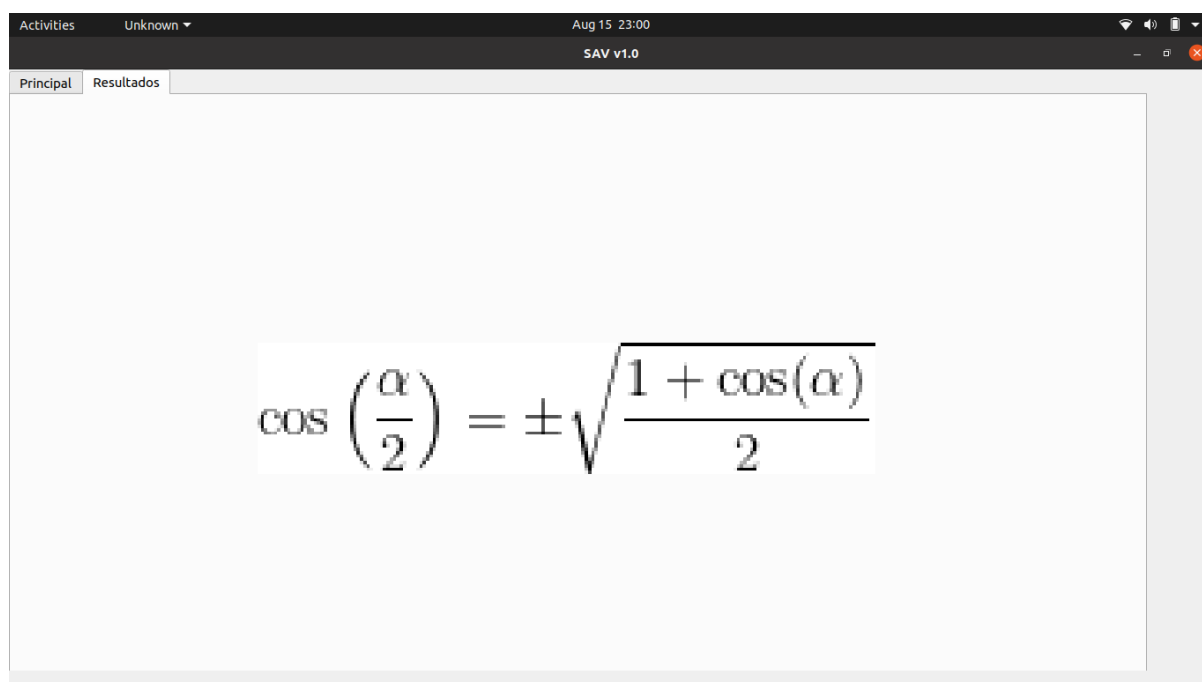


FIGURA 42: Resultado mostrado en la segunda pestaña del *software* para la tercer prueba. Fuente: Autores.

De la misma forma se generan dos archivos, el primero un documento PDF, que se puede observar en la figura 43 y el segundo es un documento HTML como se ve en la figura 44; estos documentos contienen la información detectada en el tablero para la tercer prueba (figura 41).

$$\cos\left(\frac{\alpha}{2}\right) = \pm \sqrt{\frac{1 + \cos(\alpha)}{2}}$$

FIGURA 43: Contenido del documento PDF para la tercer prueba. Fuente: Autores.

$$\cos\left(\frac{\alpha}{2}\right) = \pm \sqrt{\frac{1 + \cos(\alpha)}{2}}$$

FIGURA 44: Contenido del documento HTML para la tercer prueba. Fuente: Autores.

A continuación se presenta el contenido del archivo HTML de la figura 44:

```

1  <div><span class="math-inline "><mathml style=""><math xmlns="http://www.w3.org/1998
2  /Math/MathML">
3    <mi>cos</mi>
4    <mo data-mjx-texclass="NONE">±</mo>
5    <mrow data-mjx-texclass="INNER">
6      <mo data-mjx-texclass="OPEN">(</mo>
7      <mfrac>
8        <mi>1</mi>
9        <mn>2</mn>
10     </mfrac>
11     <mo data-mjx-texclass="CLOSE">)</mo>
12   </mrow>
13   <mo>=</mo>
14   <mo>±</mo>
15   <msqrt>
16     <mfrac>
17       <mrow>
18         <mn>1</mn>
19         <mo>+</mo>
20         <mi>cos</mi>
21         <mo data-mjx-texclass="NONE">±</mo>
22         <mo stretchy="false">(</mo>
23         <mi>±</mi>
24         <mo stretchy="false">)</mo>
25       </mrow>
26       <mn>2</mn>
27     </mfrac>
28   </msqrt>
29 </math></mathml></span></div>

```

En la tercer prueba (figura 41) se genera un audio que corresponde a la emoción mas probable del individuo (anexo 8).

10.3. Algoritmo del Aplicativo

El algoritmo desarrollado para éste proyecto se encarga de tomar imágenes a través de la cámara, procesarlas por las diferentes Redes Neuronales y generar documentos con los resultados de los reconocimientos que éstas redes entregan; además, permite que el usuario interactúe con la misma mediante teclas, presentando una interfaz gráfica compatible con lectores de pantalla, para que los usuarios puedan utilizarla con facilidad; también se muestra en todo momento lo que la cámara está observando para comodidad de los asistentes que pueda tener el usuario y para que se sepa qué está viendo la cámara. Todo esto se realiza de simultáneamente. Para adquirir imágenes de la cámara en tiempo real se lanza un hilo, el cual tiene únicamente éste propósito.

Este algoritmo al ser bastante complejo es representado en diferentes diagramas de flujo, los cuales son las figuras 45, 46, 47 y 48.

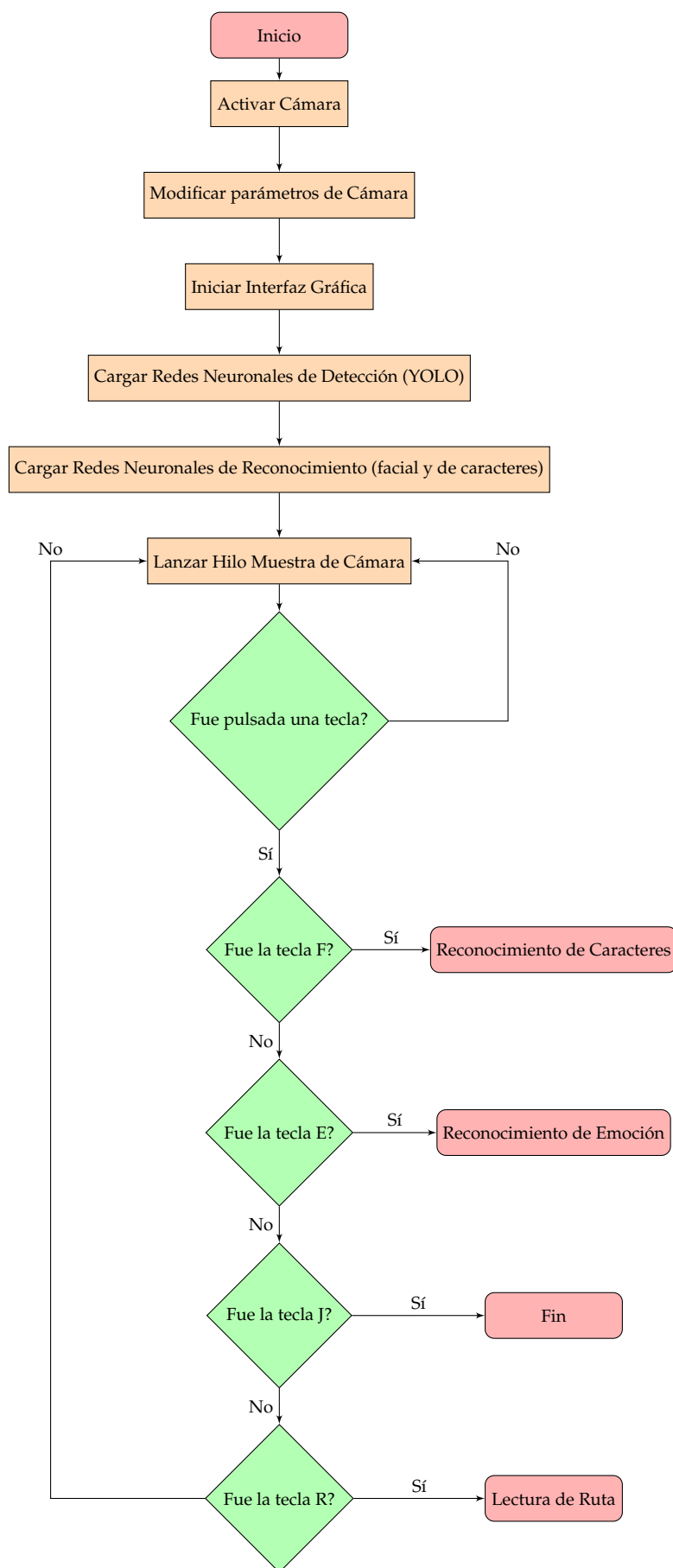


FIGURA 45: Diagrama de flujo del algoritmo principal. Fuente: Autores

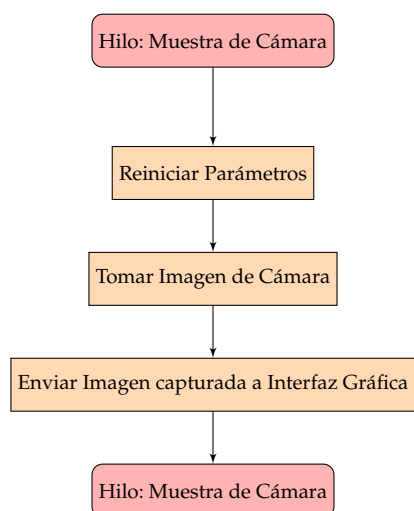


FIGURA 46: Diagrama de flujo del Hilo "Muestra de Cámara". Fuente: Autores

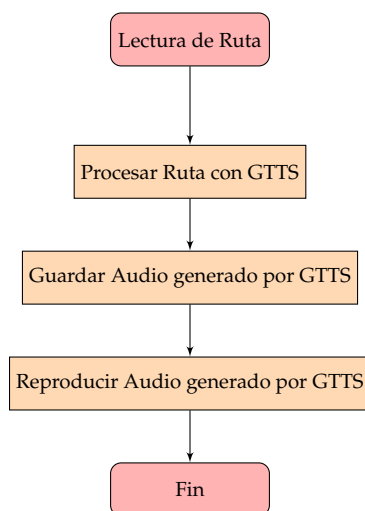


FIGURA 47: Diagrama de flujo del Algoritmo "Leer Ruta". Fuente: Autores

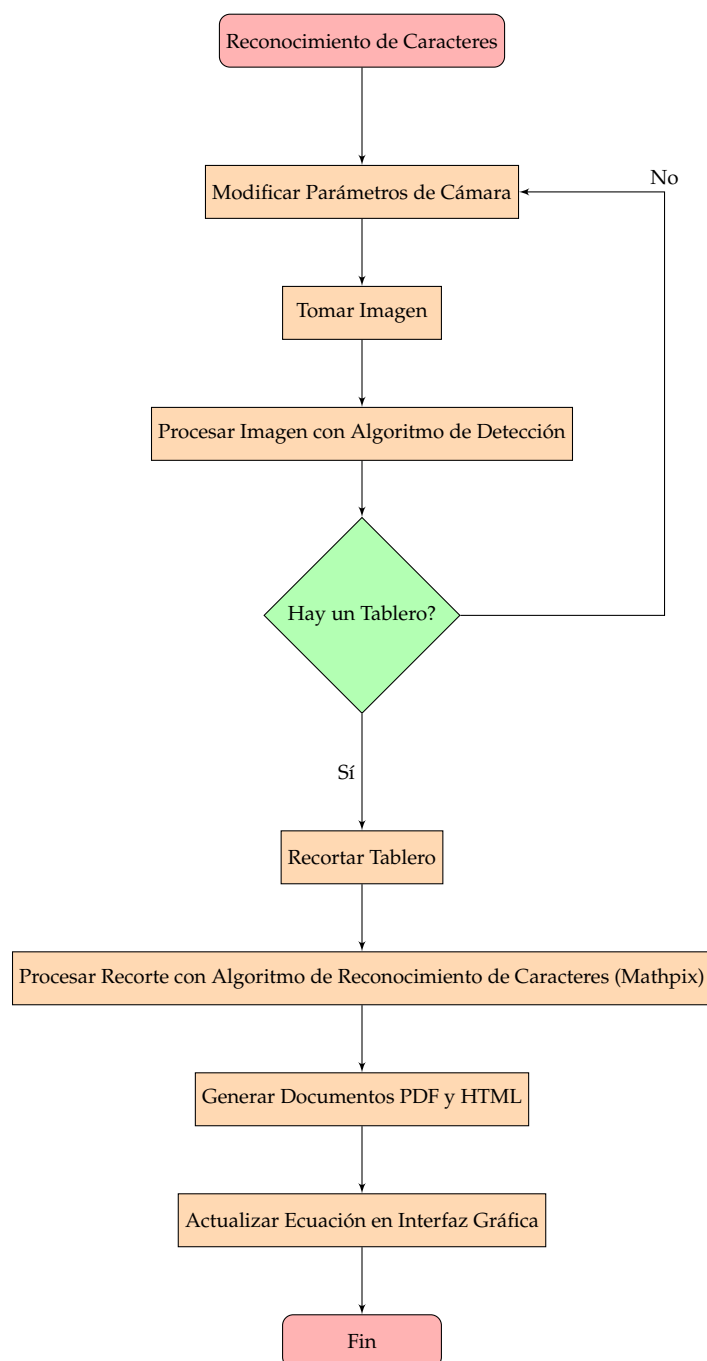


FIGURA 48: Diagrama de flujo del Algoritmo “Reconocimiento de Caracteres”.
Fuente: Autores

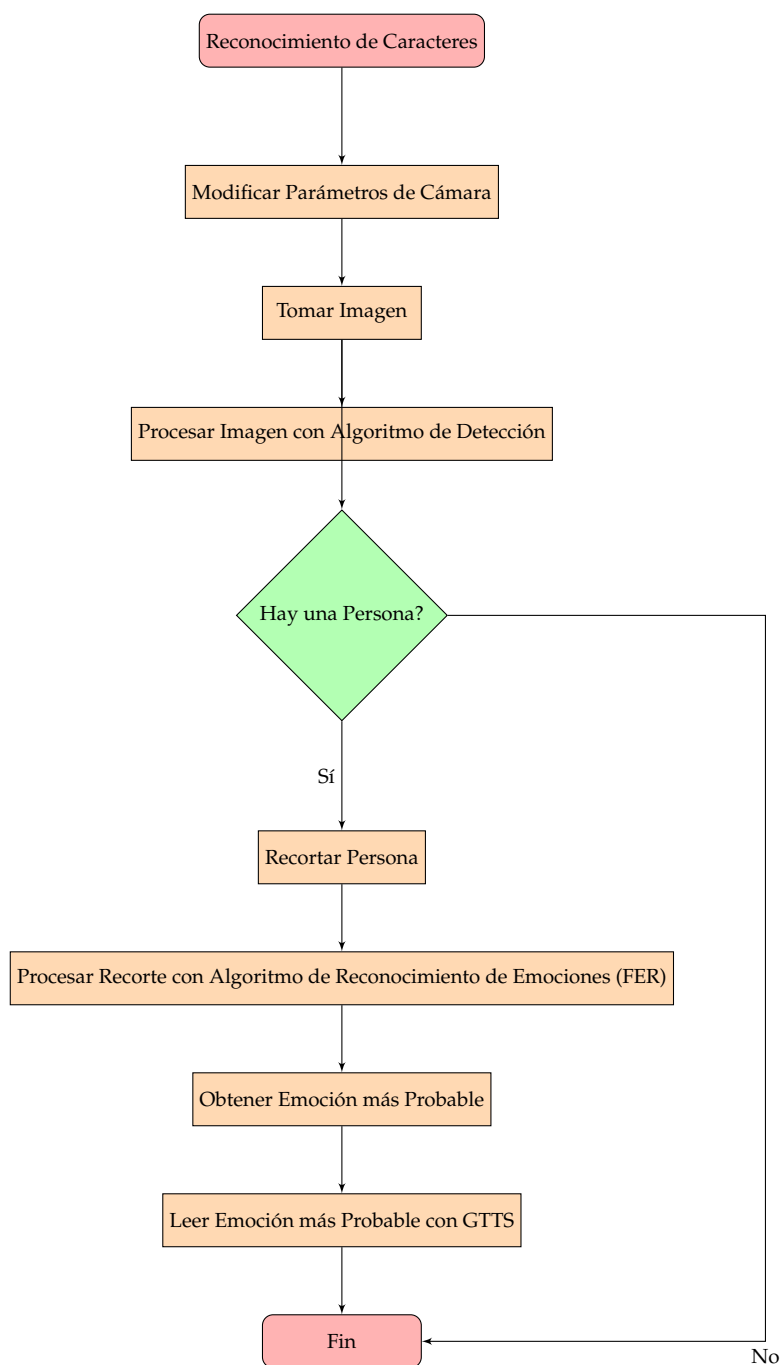


FIGURA 49: Diagrama de flujo del Algoritmo “Reconocimiento de Emocion”.
Fuente: Autores

Una vez se encuentran todos los algoritmos integrados y funcionando simultáneamente como un único proceso, se presenta al Instituto Nacional Para Ciegos, con la finalidad de obtener una retroalimentación y validación respecto al mismo.

10.4. Presentación al Instituto Nacional Para Ciegos

Se presentó el aplicativo resultante del desarrollo del proyecto al Instituto Nacional Para Ciegos con el objetivo de recibir una retroalimentación. Se acordó una reunión y allí se expuso el funcionamiento y características del aplicativo; los asistentes y el acta de la reunión se presentan en el Anexo 1. Esta retroalimentación es dada respecto a criterios que coinciden con líneas de investigación del INCI:

10.4.1. Accesibilidad

El INCI encontró que el aplicativo posee características que permiten que sea utilizado con facilidad por cualquier persona que posea alguna condición de ceguera o discapacidades visuales. Esto es debido a que la aplicación es compatible con software de lectura de pantalla, tal como lo es JAWS; además su utilización es explicada de manera textual sobre la ventana principal, de manera que cualquier software de lectura de pantalla pueda hacer saber al usuario el uso del aplicativo, y su uso es totalmente realizado por comandos del teclado.

10.4.2. Educación

El INCI resaltó que el aplicativo desarrollado contribuye a la inclusión de estudiantes con alguna condición de ceguera o discapacidades visuales pues les permite acceder fácilmente a información, que de no contar con el aplicativo, sería difícil obtener; por lo tanto, permitiéndoles ser partícipes de una clase presencial convencional.

10.4.3. Tecnología

El desarrollo de un nuevo aplicativo que permite que estudiantes con alguna discapacidad visual o ceguera puedan acceder a contenidos que antes eran de difícil acceso es apreciado por el INCI, puesto que implica que el uso de tecnologías de Visión e Inteligencia Artificial puede hacer un cambio significativo y positivo en el día a día de muchas personas con éstos problemas. Esto efectivamente implica un excelente uso de la tecnología para el bien común

Capítulo 11

Conclusiones y Trabajo Futuro

11.1. Conclusiones

Se caracterizaron diferentes elementos visuales en un escenario de enseñanza presencial de acuerdo a su utilidad a la hora de brindar información a un estudiante. Entre muchos elementos visuales, con ayuda de recomendaciones aportadas por el Instituto Nacional para Ciegos se escogieron unos pocos de acuerdo con su relevancia en contenidos para el estudiante, estos elementos son el tablero y el profesor.

Para el tablero es esencial extraer la información escrita y dibujada. Actualmente existen algoritmos de reconocimiento óptico de caracteres en textos; pero no hay muchos enfocados en reconocimiento de ecuaciones matemáticas y por ésta razón se decide enfocar el proyecto a éste ámbito.

El otro elemento, el profesor, al ser el que dirige la clase, provee la información de forma sonora, pero, su rostro puede transmitir también información que puede ser valiosa para el estudiante, por ello se ha optado por tomar la información que un estudiante con problemas de visión no puede obtener, y en este proyecto esa información es la obtenida a través de las emociones del profesor, las cuales serán comunicadas al usuario estudiante por medio de audio.

Se buscan algoritmos existentes que sean útiles a la hora de capturar la información que es proporcionada por los elementos visuales identificados. Concretamente, se necesitan algoritmos para obtener información que esté escrita en el tablero en forma de texto y ecuaciones matemáticas; a su vez se necesita otro algoritmo para identificar las emociones del profesor a partir de imágenes de su rostro.

Para lograr capturar esta información se integran los dos algoritmos mencionados en uno solo, que pueda ejecutarlos de manera simultánea. Esto se logra haciendo uso de Python y librerías como Open Computer Vision (Open CV) y NumPy; estas librerías permiten obtener información de la cámara, procesar ésta información y hacerla llegar a los algoritmos previamente mencionados.

Adicionalmente, se agregan otros dos algoritmos a los ya integrados, el primero de ellos para detectar la ubicación de los elementos visuales en la imagen y el segundo para transformar texto en voz, y entregar así la información al usuario. El primero de ellos, el detector, toma la imagen capturada por la cámara y busca en ella las coordenadas en donde los elementos caracterizados que se encuentren, es decir, las ubicaciones de el o los tableros blancos y de el profesor (si los hay); el segundo algoritmo, el sintetizador de voz, le hace saber al usuario los resultados entregados por las redes de reconocimiento de caracteres y reconocimiento de emociones, éstos resultados los entregará en forma de audio, de manera clara.

Para validar el funcionamiento del aplicativo, éste es presentado al Instituto Nacional para Ciegos. Los miembros del Instituto dan su retroalimentación respecto al proyecto, resaltando los aspectos a mejorar, a adicionar y los puntos fuertes.

11.2. Trabajo Futuro

La aplicación resultante posee un código que funciona a partir de hilos, lo que abre la posibilidad de añadir nuevas funcionalidades. El siguiente paso es crear una versión para dispositivos móviles con sistema operativo Android y iOS, esto para facilitar la utilización por parte del usuario y brindar la comodidad de la portabilidad.

Una característica útil que puede ser agregada al aplicativo es una red de detección de gestos. Esta red puede ser implementada en otro hilo de programa. La finalidad de esta red es reconocer los diferentes gestos que pueda hacer el profesor durante la clase presencial, de ésta manera se le puede hacer saber cuándo está transmitiendo información de forma que un estudiante con problemas visuales o ceguera no puede recibir, que en este caso es información gestual.

Esta red de detección puede ser entrenada hasta que sea capaz de no solamente detectar cuándo ocurren gestos, sino que puede reconocer los gestos en específico y comunicárselos al usuario.

Capítulo 12

Referencias Bibliográficas

- [1] PETROU, Maria; BOSGOGIANNI, Panagiota; BOSDOGIANNI, Panagiota. Image processing. John Wiley Sons Limited, 1999.
- [2] RUSS, John C., et al. The image processing handbook. Computers in Physics, 1994, vol. 8, no 2, p. 177-178.
- [3] COMER, Mary L.; DELP III, Edward J. Morphological operations for color image processing. Journal of electronic imaging, 1999, vol. 8, no 3, p. 279-289.
- [4] HARMON, Leon D. Artificial neuron. Science, 1959, vol. 129, no 3354, p. 962-963.
- [5] GERSHENSON, Carlos. Artificial neural networks for beginners. arXiv preprint cs/0308031, 2003.
- [6] WANG, Sun-Chong. Artificial neural network. En Interdisciplinary computing in java programming. Springer, Boston, MA, 2003. p. 81-100.
- [7] GALLANT, Stephen I.; GALLANT, Stephen I. Neural network learning and expert systems. MIT press, 1993.
- [8] HE, Kaiming; SUN, Jian; TANG, Xiaoou. Guided image filtering. En European conference on computer vision. Springer, Berlin, Heidelberg, 2010. p. 1-14.
- [9] WISCHIK, Damon; HANDLEY, Mark; BRAUN, Marcelo Bagnulo. The resource pooling principle. ACM SIGCOMM Computer Communication Review, 2008, vol. 38, no 5, p. 47-52.

-
- [10] ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. En 2017 International Conference on Engineering and Technology (ICET). Ieee, 2017. p. 1-6.
- [11] PETROU, Maria; BOSGOGIANNI, Panagiota; BOSDOGIANNI, Panagiota. Image processing. John Wiley Sons Limited, 1999.
- [12] PLAZA, Antonio, et al. Spatial/spectral endmember extraction by multidimensional morphological operations. IEEE transactions on geoscience and remote sensing, 2002, vol. 40, no 9, p. 2025-2041.
- [13] TACCHINO, Francesco, et al. An artificial neuron implemented on an actual quantum processor. npj Quantum Information, 2019, vol. 5, no 1, p. 1-8.
- [15] YAMASHITA, Rikiya, et al. Convolutional neural networks: an overview and application in radiology. Insights into imaging, 2018, vol. 9, no 4, p. 611-629.
- [16] El Instituto Nacional para Ciegos (INCI) Accessed on: Mar. 11, 2021. [Online]. Available: <https://www.inci.gov.co/elinstituto>
- [18] JAMES, William, et al. On a certain blindness in human beings. Pragmatism and other writings, 1899, p. 269.
- [19] Gobierno de Colombia. «Resultados Censo Nacional de Población y Vivienda 2018». En: Resultados Censo Nacional de Población y Vivienda 2018 1 (2018).
- [20] Alexis Hidalgo Maldonado y Revista Educativa Internacional. «DIFICULTADES DE APRENDIZAJE EN EL ALUMNADO CON DÉFICIT VISUAL y CIEGO». En: 2016 International Conference Communications and Informatics, ICACCI 2016 1 (2016).
- [21] S. Divya y col. «Smart Assistance Navigational System for Visually Impaired Individuals». En: IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing, INCOS 2019 (2019). DOI: 10.1109/INCOS45849.2019.8951333.
- [22] Lilit Hakobyan y col. Mobile assistive technologies for the visually impaired. Nov. de 2013. URL: <http://trap.ncirl.ie/3170/>.
- [23] Başak Süller Zor y Arzu Vuruşkan. «ASSISTIVE TECHNOLOGIES FOR INDIVIDUALS WITH VISUAL IMPAIRMENT: CONTRIBUTION OF DESIGN IN THE FIELD OF CLOTHING». En: Mühendislik Bilimleri ve Tasarım Dergisi 7.4 (2019), págs. 913-925. ISSN: 1308- 6693. DOI: 10.21923/jesd.482180.

- [24] L. Leija, P. Hernandez y S. Santiago. «Reader instrument of basic texts to the teaching of blind people». En: Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings 1 (1999), pág. 588. ISSN: 05891019. DOI: 10.1109/iembs.1999. 802666.
- [25] Cássia Cristiane De Freitas Alves y col. «Assistive technology applied to education of students with visual impairment». En: Revista Panamericana de Salud Publica/Pan American Journal of Public Health 26.2 (2009), págs. 148-152. ISSN: 16805348. DOI: 10.1590/s1020-49892009000800007.
- [26] Christina Dicke, Katrin Wolf y Yaroslav Tal. «Foogues: Eyes-Free Interaction for Smartphones Christina». En: (2010), págs. 1-3.
- [27] Narcisa Isaila. «The Assistive Software, Useful and Necessary Tool for Blind Student's Abilities Development». En: Procedia - Social and Behavioral Sciences 116 (2014), págs. 2189-2192. ISSN: 18770428. DOI: 10.1016/j.sbspro.2014.01.541. URL: <http://dx.doi.org/10.1016/j.sbspro.2014.01.541>.
- [28] Fatoş Silman, Hüseyin Yaratan y Tolgay Karanfiller. «Use of assistive technology for teaching-learning and administrative processes for the visually impaired people». En: Eurasia Journal of Mathematics, Science and Technology Education 13.8 (2017), págs. 4805-4813. ISSN: 13058223. DOI: 10.12973/eurasia.2017.00945a.
- [29] Joao Ulisses y col. «ACE assisted communication for education: Architecture to support blind deaf communication». En: IEEE Global Engineering Education Conference, EDUCON 2018-April (2018), págs. 1015-1023. ISSN: 21659567. DOI: 10.1109/EDUCON.2018. 8363341.
- [30] Jacket Demby's y col. «Object Detection and Pose Estimation Using CNN in Embedded Hardware for Assistive Technology». En: 2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019 (2019), págs. 3165-3171. DOI: 10.1109/SSCI44817.2019.9002767.
- [31] Wan Jung Chang y col. «An AI edge computing based Wearable assistive device for visually impaired people zebra-crossing walking». En: Digest of Technical Papers - IEEE International Conference on Consumer Electronics 2020-Janua (2020), págs. 4-5. ISSN: 0747668X. DOI: 10.1109/ICCE46568.2020.9043132
- [35] MinTIC. ConVerTic. es. URL: <http://convertic.gov.co/641/w3-propertyvalue-15341.html>.
- [36] Marion A. Hershey y Michael A. Johnson. Assistive Technology for Visually Impaired and Blind People. Ed. por David Keating y col. Springer. ISBN: 9781846288661.
- [37] Tesseract OCR. en. URL: <https://github.com/tesseract-ocr/tesseract>.
- [38] Google Text to Speech. en. URL: <https://github.com/pndurette/gTTS>.

[39] PyTTSx3. en. URL: <https://github.com/nateshmbhat/pyttsx3>.

[40] Facial Expression Recognition. en. URL: <https://github.com/justinshenk/fer>.

[41] Priya-Dwivedi. en. URL: https://github.com/priya-dwivedi/face_and_emotion_detection.

[42] Facial Emotion Recognition. en. URL: <https://pypi.org/project/facial-emotion-recognition/>.

[43] Atulapra Emotion Detection. en. URL: <https://github.com/atulapra/Emotion-detection>.

[44] Mathpix. en. URL: <https://github.com/Mathpix>.

[45] Boletines Poblacionales. Personas con Discapacidad -PCD. en. URL: <https://www.minsalud.gov.co/sites/poblacionales-personas-discapacidadI-2020.pdf>

Capítulo 13

Anexos

13.1. Anexo 1.

Se presenta el audio generado por el Sintetizador a Voz, en un archivo en la nube.

<https://drive.google.com/file/d/1qxn12doGhRmB0tw7ezR1GNclT9BHfF-/view?usp=sharing>

13.2. Anexo 2.

Se presenta el acta de la reunión de muestra del aplicativo hacia el Instituto Nacional para Ciegos. Se mencionan las actividades llevadas a cabo y la lista de asistentes junto con sus firmas.

**PRESENTACIÓN DE PROYECTO DE GRADO TITULADO: DESARROLLO DE UN
SOFTWARE BASADO EN VISIÓN ARTIFICIAL PARA ATENDER NECESIDADES DE
PERSONAS CON DISCAPACIDAD VISUAL EN UN ESCENARIO DE
ENSEÑANZA-APRENDIZAJE**

ACTA

BOGOTÁ, 13/06/2022

HORA: 14:00

LUGAR: SEDE PRINCIPAL DEL INCI

ASISTENTES:

Edwin Beltrán
Santiago Rodríguez
Alejandro Pabón
María del Rosario Yepes
Patricia Montoya F.

INVITADOS:

Juan David Avendaño Moreno - Universidad Santo Tomás
Hernán Felipe Salamanca Montoya- Universidad Santo Tomás
Armando Mateus Rojas - Universidad Santo Tomás

AUSENTES: No

ORDEN DEL DÍA

1 - Socialización sobre los resultados del proyecto de grado titulado "DESARROLLO DE UN SOFTWARE BASADO EN VISIÓN ARTIFICIAL PARA ATENDER NECESIDADES DE PERSONAS CON DISCAPACIDAD VISUAL EN UN ESCENARIO DE ENSEÑANZA-APRENDIZAJE" por parte de los estudiantes Juan David Avendaño Moreno y Hernán Felipe Salamanca Montoya bajo la dirección de la Profesora Sindy Paola Amaya.

2 - Retroalimentación del punto anterior por parte de los integrantes del Instituto Nacional para Ciegos

DESARROLLO

1 - Los estudiantes Juan David Avendaño Moreno y Hernán Felipe Salamanca Montoya expusieron el proyecto de grado titulado "DESARROLLO DE UN SOFTWARE BASADO EN VISIÓN ARTIFICIAL PARA ATENDER NECESIDADES DE PERSONAS CON DISCAPACIDAD VISUAL EN UN ESCENARIO DE ENSEÑANZA-APRENDIZAJE" en nombre de la Facultad de Ingeniería Electrónica de la Universidad Santo Tomás

Código: SG-110-FM-038 - Versión 7 - Vigencia 29/01/2019

Carrera 13 No. 34 - 91 / www.inci.gov.co
aciudadano@inci.gov.co / PBX:(57 1) 384 66 66
Bogotá D.C. Colombia



**La educación
es de todos**

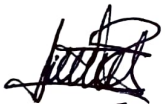
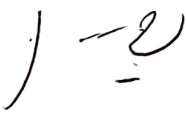
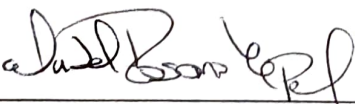
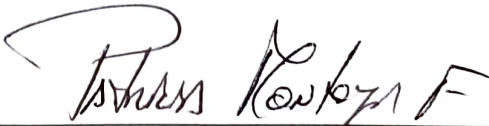
Mineducación

2- Los integrantes del Instituto Nacional para Ciegos brindaron retroalimentación respecto al proyecto de grado, mencionando que cumple con los objetivos propuestos para ésta fase del proyecto de la Universidad Santo Tomás, y por tanto, cumpliendo los objetivos del proyecto de grado anteriormente mencionado. Mencionaron también proyectos futuros respecto al mismo proyecto de grado.

COMPROMISO	RESPONSABLE	FECHA
No Aplica	No Aplica	No Aplica

CONVOCATORIA.

Firmas:

 Juan David Avendaño Moreno Estudiante	 Hernán Felipe Salamanca Montoya Estudiante
 Armando Mateus Rojas Docente encargado	 Edwin Beltrán Coordinador accesibilidad
 Santiago Rodríguez Profesional Especializado	 Alejandro Pabón Técnico
 Maria del Rosario Yepes Coordinadora Gestión Interinstitucional	 Patricia Montoya F. Profesional Especializado

Anexos: Asistencia del día de la reunión

Transcriptor: No aplica

Código: SG-110-FM-038 - Versión 7 - Vigencia 29/01/2019

Carrera 13 No. 34 - 91 / www.inci.gov.co
 aciudadano@inci.gov.co / PBX:(57 1) 384 66 66
 Bogotá D.C. Colombia



La educación
es de todos

Mineducación

13.3. Anexo 3.

Se presenta una carta firmada y entregada por el Instituto Nacional para Ciegos que da constancia de que el proyecto realizado cumple con los criterios para cada línea que se maneja en la institución.



20221200017971

22-07-2022

Bogotá D.C.

Señores
Facultad de Ingeniería Electrónica
Universidad Santo Tomás
Bogotá

Asunto: Criterios Instituto Nacional para Ciegos – INCI-

Cordial saludo.

De acuerdo a la reunión sostenida el pasado 13 de junio de 2022 en las instalaciones del Instituto Nacional para Ciegos - INCI, en la cual los estudiantes Hernán Felipe Salamanca Montoya y Juan David Avendaño Moreno, socializaron los desarrollos y resultados de su proyecto de grado "Desarrollo de un software basado en visión artificial para atender necesidades de personas con discapacidad visual en un escenario de enseñanza-aprendizaje" y dirigido por la docente Sindy Paola Amaya, podemos dar constancia que el proyecto cumple con los siguientes criterios asociados a cada una de las líneas que se manejan en la institución:

Asequibilidad: el software se constituye en una herramienta de bajo costo que permite el acceso a la información gráfica para personas con discapacidad visual

Educación: la herramienta potencia los procesos de enseñanza-aprendizaje en el aula, específicamente para las matemáticas

Tecnología: integra elementos tecnológicos de uso común como computadores y cámaras web mediante la utilización de inteligencia computacional y visión artificial para la mejora de la calidad de vida de las personas.

En constancia y a solicitud de los estudiantes interesados se firma la presente carta para los procesos académicos correspondientes

Cordialmente,

PEDRO ANDRADE LOZADA
Subdirector General

Proyectó: Edwin E. Beltrán
Revisó: Pedro Andrade Lozada

Código: SG-110-FM-039 – Versión: 14 – Vigencia: 29/01/2019

Carrera 13 No. 34 - 91 / www.inci.gov.co
aciudadano@inci.gov.co / PBX: (57 1) 384 66 66
Bogotá D.C., Colombia



La educación
es de todos

Mineducación

13.4. Anexo 4.

Se presenta un correo electrónico con la información del agendamiento para una de las reuniones con el Instituto Nacional para Ciegos.



Invitation: Reunión INCI - Santo Tomás Ingeniería Electrónica @ Mon Sep 21, 2020 2pm - 3pm (COT) (hernansalamanca@usantotomas.edu.co)

1 mensaje

Sindy Paola Amaya <sindyamaya@usantotomas.edu.co>

16 de septiembre de 2020, 17:56

Responder a: Sindy Paola Amaya <sindyamaya@usantotomas.edu.co>

Para: hernansalamanca@usantotomas.edu.co, edgarcamacho@usantotomas.edu.co, pmontoya@inci.gov.co, juan.avendano@usantotomas.edu.co, srodriguez@inci.gov.co, myepes@inci.gov.co

You have been invited to the following event.

Reunión INCI - Santo Tomás Ingeniería Electrónica

When Mon Sep 21, 2020 2pm – 3pm Colombia Standard Time

[more details »](#)

Joining info Join with Google Meet
meet.google.com/yxs-uydj-sbi

Join by phone
(US) [+1 443-671-8031](tel:+14436718031) (PIN: 408178164)

Calendar hernansalamanca@usantotomas.edu.co

Who

- sindyamaya@usantotomas.edu.co - organizer
- edgarcamacho@usantotomas.edu.co
- pmontoya@inci.gov.co
- juan.avendano@usantotomas.edu.co
- hernansalamanca@usantotomas.edu.co
- srodriguez@inci.gov.co
- myepes@inci.gov.co

Going (hernansalamanca@usantotomas.edu.co)? **Yes** - **Maybe** - **No** [more options »](#)

Invitation from [Google Calendar](#)

You are receiving this email at the account hernansalamanca@usantotomas.edu.co because you are subscribed for invitations on calendar hernansalamanca@usantotomas.edu.co.

To stop receiving these emails, please log in to <https://www.google.com/calendar/> and change your notification settings for this calendar.

Forwarding this invitation could allow any recipient to send a response to the organizer and be added to the guest list, or invite others regardless of their own invitation status, or to modify your RSVP. [Learn More](#).

 **invite.ics**
3K

13.5. Anexo 5.

Se presenta un correo electrónico con la información del agendamiento para una de las reuniones con el Instituto Nacional para Ciegos.

Universidad Santo Tomás: Reunión avances Proyecto de Grado Ingeniería Electrónica

1 mensaje

Edwin Beltrán Chamorro <ebeltran@inci.gov.co>

6 de mayo de 2021, 16:20

Para: Sindy Paola Amaya <sindyamaya@usantotomas.edu.co>

Cc: "hernansalamanca@usantotomas.edu.co" <hernansalamanca@usantotomas.edu.co>, "juan.avendano@usantotomas.edu.co"

<juan.avendano@usantotomas.edu.co>, Patricia Montoya Falla <pmontoya@inci.gov.co>, Santiago Adolfo Rodríguez <srodriguez@inci.gov.co>, ronys lopez <ronyslopezinci@gmail.com>

Buenas tardes compañeros,

Enviamos el enlace de acceso a la reunión para el miércoles 12 de 4:00 p.m. a 5:00 p.m.

Reunión de Microsoft Teams

Únase desde su equipo o aplicación móvil

Haga clic aquí para unirse a la reunión

Infórmese | Opciones de reunión

De: Sindy Paola Amaya <sindyamaya@usantotomas.edu.co>

Enviado el: jueves, 6 de mayo de 2021 4:17 p. m.

Para: Edwin Beltrán Chamorro <ebeltran@inci.gov.co>

CC: hernansalamanca@usantotomas.edu.co; juan.avendano@usantotomas.edu.co; Patricia Montoya Falla <pmontoya@inci.gov.co>; Santiago Adolfo Rodríguez <srodriguez@inci.gov.co>

Asunto: Re: Universidad Santo Tomás: Reunión avances Proyecto de Grado Ingeniería Electrónica

Buenas tardes Edwin,

Nos permitimos confirmar reunión. ¿Les parece que la programemos nosotros?, en ese caso sería por medio de Meet.

Quedo atenta.

Muchas gracias.

El jue, 6 may 2021 a las 16:11, Edwin Beltrán Chamorro (<ebeltran@inci.gov.co>) escribió:

Cordial saludo Sindy,

Muchas gracias, de las fechas propuestas, proponemos la del miércoles 12 de mayo a las 4:00 p.m.

Quedamos atentos.

Edwin Enrique Beltrán Chamorro.

Profesional Universitario (E)

Subdirección Técnica

[Tel:3846666](tel:3846666) extensión: 406

Carrera 13 No. 34 – 91

www.inci.gov.co

13.6. Anexo 6.

Se presenta el audio generado por el Sintetizador de Texto a Voz para la primer prueba del *Software*, en un archivo en la nube.

https://drive.google.com/file/d/1gB6OJDcPk3wPovAOwmQ0VSlXFYm_FkxC/view?usp=sharing

13.7. Anexo 7.

Se presenta el audio generado por el Sintetizador de Texto a Voz para la segunda prueba del *Software*, en un archivo en la nube.

<https://drive.google.com/file/d/14cDv1l7h57FNco1JnBxLCpq420hhI8h3/view?usp=sharing>

13.8. Anexo 8.

Se presenta el audio generado por el Sintetizador de Texto a Voz para la tercer prueba del *Software*, en un archivo en la nube.

<https://drive.google.com/file/d/1dZaBbadcPwG4Cz5HAV2qojDRSbdLCXQ0/view?usp=sharing>