

UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA
PROGRAMA DE TITULACIÓN POR TESIS
ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA



SIMULACIÓN DE MÓDULO RECONOCEDOR DE
SÍMBOLOS MEDIANTE TÉCNICAS DE
PROCESAMIENTO DE IMÁGENES PARA APRENDIZAJE
DEL CÓDIGO COLOR ADD EN PERSONAS
DALTÓNICAS

TESIS
PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO
PRESENTADA POR

Bach. Gallegos Jara, Hilary Dharmia
Bach. Gonzales del Valle Romero, Angello

Asesor: Dr. Ing. Huamaní Navarrete, Pedro

LIMA-PERÚ

2019

DEDICATORIA

Este trabajo está dedicado a nuestro Señor Dios por ser siempre guía y fuerza en nuestro camino para lograr nuestras metas.

A mis queridos padres y familia por su amor, sacrificio y apoyo incondicional con el fin de alcanzar nuestros sueños y objetivos.

Bach. Gallegos Jara, Hilary

Esta investigación la dejo dedicada a mis padres, los cuales fueron el motor de avance para llegar hasta este punto, por su incondicional apoyo y ferviente esfuerzo.

Bach. Gonzales del valle Romero, Angello

AGRADECIMIENTO

A Dios por sus bendiciones, a mis padres por ser ejemplo de emprendimiento y lucha constante frente a adversidades, a nuestro tutor de tesis por su orientación no solo en el desarrollo de la tesis sino también a lo largo de nuestra carrera profesional.

Bach. Gallegos Jara, Hilary y Bach. Gonzales del valle Romero, Angello

INDICE

RESUMEN	x
ABSTRACT.....	xi
INTRODUCCIÓN	1
CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA.....	3
1.1. Problemática.....	3
1.1.1. Problema General.....	3
1.1.2. Problemas Específicos	3
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos.....	4
1.3. Importancia y justificación.....	4
1.3.1. Importancia	4
1.3.2. Justificación	4
1.4. Alcances y limitaciones	5
1.4.1. Alcances	5
1.4.2. Limitaciones.....	5
CAPÍTULO II: MARCO TEÓRICO	6
2.1. Antecedentes del estudio de investigación.....	6
2.1.1. Antecedentes Nacionales.....	6
2.1.2. Antecedentes Internacionales	6
2.2. Bases teóricas relacionadas con el tema.....	7
2.2.1. Técnicas de Procesamiento Digital de Imágenes	7
2.2.1.1. Etapas de Procesamiento digital de Imágenes	7
2.2.1.2. Momentos Invariantes a la rotación, traslación y escalado: Momentos de Hu9	
2.2.2. Redes Neuronales	11
2.2.3. Reconocedor de Símbolos	12
2.2.4. Conexión inalámbrica.....	14
2.3. Variables de Estudio	15
2.4. Tipo y método de investigación	16
2.4.1. Tipo de investigación	16

2.4.2. Método de investigación.....	16
2.4.3. Diseño de investigación.....	17
2.5. Instrumento de recolección de datos	17
2.6. Técnicas de recolección de datos	17
2.7. Definición de términos	17
CAPÍTULO III: DISEÑO DE LA INVESTIGACIÓN.....	18
3.1. Diseño planteado del módulo reconocedor de símbolos del código colorADD .	18
3.2. Conectividad de dispositivo IP hacia interfaz usuaria	18
3.2.1. Conexión con software MATLAB.....	19
3.2.2. Captura de muestra a procesar	20
3.3. Adecuación y filtrado de muestra capturada.....	21
3.3.1. Adecuación	21
3.3.2. Filtrado: Filtro de Mediana.....	23
3.4. Procesamiento de muestras	24
3.3.3. Generación de base de datos ideal.....	24
3.4.1.1. Aplicación de los momentos de Hu sobre imágenes ideales	24
3.4.1.2. Generación de Red Neuronal Multicapa Perceptrón	27
3.4.1.3. Entrenamiento de RNMP	28
3.4.1. Aplicación de los momentos de Hu sobre la muestra capturada post filtrado.	30
3.4.2. Normalización de los momentos de Hu	30
3.5. Generación de GUIDE y conexionado con el algoritmo.	30
CAPÍTULO IV: PRUEBAS Y RESULTADOS	35
4.1. Análisis del resultado color amarillo	35
4.2. Análisis del resultado color azul	36
4.3. Análisis del resultado color amarillo claro	37
4.4. Resultados generales	38
CONCLUSIONES	41
RECOMENDACIONES.....	42
REFERENCIAS BIBLIOGRÁFICAS	43
ANEXOS	A

INDICE DE FIGURAS

Figura N° 1: Etapas de procesamiento digital de imágenes.....	8
Figura N° 2: Ejemplificación de vecindad de píxeles de una matriz 3 x 3	9
Figura N° 3: Proceso de imagen en el dominio de la frecuencia.	9
Figura N° 4: Arquitectura del Perceptrón Multicapa	12
Figura N° 5: Cinco símbolos básicos del código colorADD	13
Figura N° 6: Combinación de símbolos del código colorADD	13
Figura N° 7: Líneas de Señalización del Metro de Porto, Portugal	14
Figura N° 8: Software Droidcam	15
Figura N° 9: Diagrama de bloques del módulo reconocedor de símbolos del código color ADD	18
Figura N° 10: Icono en menú móvil del aplicativo DroidCam	19
Figura N° 11: Pruebas de Cámara IP (Vista de PC en navegador vs Vista de Celular)	19
Figura N° 12: Algoritmo de conexión de Cámara IP (aplicativo DroidCam) con software Matlab.	20
Figura N° 13: Identificación de la variable “cam”	20
Figura N° 14: Identificación de la variable “ap”	21
Figura N° 15: Imagen tomada como muestra (Color Amarillo Claro) mostrado en MATLAB	21
Figura N° 16: Histograma de la muestra capturada	22
Figura N° 17: Algoritmo para convertir imagen de color a blanco y negro	22
Figura N° 18: Aplicación del algoritmo convertidor de blanco y negro sobre muestra capturada.....	23
Figura N° 19: Algoritmo de filtrado.....	23
Figura N° 20: Imagen capturada sin filtro e imagen capturada con filtro.....	24
Figura N° 21: Tabla de momentos de Hu diferenciados de los colores con una rotación a 0°	24
Figura N° 22: Tabla de momentos de Hu diferenciados de los colores con una rotación a -15°	25
Figura N° 23: Tabla de momentos de Hu diferenciados de los colores con una rotación a 15°	25
Figura N° 24: Tabla de momentos de Hu diferenciados de los colores con una rotación a -5°	25
Figura N° 25: Tabla de momentos de Hu diferenciados de los colores con una rotación a +5°	25
Figura N° 26: Tabla de momentos de Hu diferenciados de los colores con una rotación a -10°	26
Figura N° 27: Tabla de momentos de Hu diferenciados de los colores con una rotación a 10°	26

Figura N° 28: Tabla de momentos de Hu diferenciados de los colores con una rotación a -20°	26
Figura N° 29: Tabla de momentos de Hu diferenciados de los colores con una rotación a 20°	27
Figura N° 30: RNMP mostrada en Software MATLAB.....	28
Figura N° 31: Interfaz de entrenamiento de RNMP con 4 capas	29
Figura N° 32: Interfaz de entrenamiento de RNMP con 2 capas	29
Figura N° 33: Interfaz de creación de GUIDE.....	30
Figura N° 34: Área de diseño de GUIDE.....	31
Figura N° 35: Área de diseño de GUIDE mostrando el uso de herramienta axis.....	31
Figura N° 36: Área de diseño de GUIDE mostrando el uso de herramienta static text	32
Figura N° 37: Área de diseño de GUIDE mostrando el uso de herramienta pushbutton	32
Figura N° 38: Diseño de GUIDE	33
Figura N° 39: Código para imágenes de la GUIDE.....	33
Figura N° 40: Código para resultados de la GUIDE.....	34
Figura N° 41: GUIDE	34
Figura N° 42: Prueba con color amarillo sin rotación.....	35
Figura N° 43: Prueba con color amarillo con rotación	35
Figura N° 44: Prueba con color amarillo con rotación	36
Figura N° 45: Prueba con color azul sin rotación	36
Figura N° 46: Prueba con color azul con rotación	37
Figura N° 47: Prueba con color amarillo claro con rotación	37
Figura N° 48: Prueba con color amarillo claro con rotación	38

ÍNDICE DE TABLAS

Tabla N° 1: Tabla de valores asociados y rangos permisibles de los colores detectados.....	28
Tabla N° 2: Tabla de Detección de imágenes Ideales	38
Tabla N° 3: Tabla de Detección de imágenes Reales	39
Tabla N° 4: Tabla de Funcionamiento de Guide.....	39

ÍNDICE DE ECUACIONES

Ecuación N° 1: Momentos centrales.....	10
Ecuación N° 2: Normalización de Momentos	10
Ecuación N° 3: 7 Momentos Invariantes	11

RESUMEN

La presente investigación se desarrolló en Lima, Perú durante los meses de Mayo a Setiembre del año 2019, la cual tuvo como propósito alcanzar el mayor porcentaje de reconocimiento de símbolos del código colorADD utilizando el algoritmo de los momentos de Hu, como técnica de procesamiento de imagen, para lograr dicho cometido. El presente estudio se justificó por cuanto posee valor teórico, utilidad práctica y por relevancia social, debido a que generará mayor impacto en personas daltónicas. El referencial teórico se fundamenta en el concepto Rice (1999) en su libro *Optical Character Recognition*, quien sustenta que la aplicación de distintos métodos de procesamiento de imágenes permitieron el reconocimiento de caracteres ópticos de acuerdo al tipo de condiciones a las cuales se sometieron.

Se utilizó como parte de la investigación una muestra de 10 símbolos del código colorADD ideales. Se empleó una red neuronal multicapa perceptrón de 4 capas entrenada con los valores dados por los momentos de Hu aplicada a la muestra para obtener resultados con imágenes reales impresas donde intervinieron factores de ruido.

Los resultados evidencian que la simulación efectuada tuvo un 70% de éxito de reconocimiento de símbolos del código colorADD, y a su vez mantuvo su porcentaje de éxito al haberse introducido en una *GUIDE*. Concluyéndose que el módulo reconocedor de símbolos del código colorADD requiere de aplicar diferentes técnicas de procesamiento de imágenes para mejorar el porcentaje de reconocimiento.

Palabras claves: Código colorADD, GUIDE, Momentos de Hu, Red Neuronal
Multicapa Perceptrón

ABSTRACT

The present investigation was carried out in Lima, Peru during the months of May to September of the year 2019, hence the purpose of the investigation was to achieve the highest percentage of recognition of symbols of the colorADD code using the algorithm of the moments of Hu, as an image processing technique, to achieve this task.

This study was justified because it has theoretical value, practical utility and social relevance, because it will generate greater impact on colorblind people. The theoretical reference is based on the Rice concept (1999) in his book Optical Character Recognition, who argues that the application of different image processing methods allowed the recognition of optical characters according to the type of conditions to which they were subjected.

A sample of 10 ideal color ADD code symbols was used as part of the investigation. A 4-layer multilayer perceptron neuronal network trained with the values given by the moments of Hu applied to the sample was used to obtain results with real printed images where noise factors intervened.

The results show that the simulation carried out had a 70% success in recognizing symbols of the colorADD code, and in turn maintained its success rate after having entered a GUIDE. Concluding that the symbol recognition module of the colorADD code requires applying different image processing techniques to improve the recognition rate.

Keywords: colorADD code, GUIDE, Hu Moments, Perceptron Multilayer Neural Network.

INTRODUCCIÓN

Para el desarrollo de la presente investigación **SIMULACIÓN DE MÓDULO RECONOCEDOR DE SÍMBOLOS MEDIANTE TÉCNICAS DE PROCESAMIENTO DE IMÁGENES PARA APRENDIZAJE DEL CÓDIGO COLOR ADD EN PERSONAS DALTÓNICAS** se consideró los siguientes antecedentes: Sobrado E. en sus tesis Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot, Lema L. en su tesis Sistema de reconocimiento de imagen para detección de placas de vehículo obtenidas con cámara web, así mismo Casas A. en su tesis Reconocimiento de imágenes con Redes convolucionales en C, finalmente la investigación de Palacios R. en su tesis Reconocimiento de imágenes mediante Raspberry PI.

Para la investigación se ha formulado el problema general de la siguiente manera: ¿Cómo será la simulación de un módulo reconocedor de símbolos mediante técnicas de procesamiento digital de imágenes para el aprendizaje del código colorADD?, interrogante que a través de la investigación se ha dado respuesta.

El motivo principal de la investigación considera conocer el Procesamiento de Imágenes y cómo esta impacta en el Reconocimiento de Símbolos. Existe la necesidad de inducir y promover este código de colores como se viene desarrollando en Portugal para el aprendizaje y uso no solo de las personas con esta afección, sino también de las de su entorno, de esta forma la sociedad se compromete a ser más empática e inclusiva. En ese sentido, los resultados obtenidos de la investigación servirán de base para incentivar en mejorar la simulación e implementar nuevos programas basados en otros requerimientos. El objetivo principal es Simular un módulo identificador de símbolos mediante técnicas de procesamiento de imágenes para el aprendizaje del código colorADD en personas con daltonismo.

Así mismo para un estudio sistematizado del problema, la investigación se ha estructurado de la siguiente manera:

Capítulo I: Planteamiento y delimitación del problema, donde hemos determinado el origen del problema y sus delimitaciones, capítulo II: Marco Teórico, donde se expuso los antecedentes a nuestra investigación, capítulo III: Diseño de la Investigación, en donde hemos mostrado el diseño planteado para satisfacer las necesidades de la investigación, y capítulo IV: Pruebas y resultados, donde vimos el funcionamiento de lo simulado.

Para finalizar la investigación se redactan las conclusiones, recomendaciones y se describe la referencia bibliográfica utilizada como también se consideran los anexos con información que valida la investigación realizada.

CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA

1.1. Problemática

El daltonismo es considerado una discapacidad visual cromática, esa condición acarrea un fenómeno social generando situaciones de exclusión y violencia, así como problemas de autoestima y autonomía. Según Neiva (2010), el 8% de la población masculina y el 0.5% de la población femenina padece en cierto grado daltonismo. Los mensajes y la comunicación en su mayoría se basan en colores, esta desventaja afecta a las personas que padecen esta anomalía debido a que la sociedad no cuenta con políticas ni recursos para que puedan desenvolverse sin problemas como lo hace una persona sana, no tomando en cuenta la dificultad que supone insertarse en un sistema de desigualdad de oportunidades.

Los infantes son más proclives a esta discriminación en la etapa escolar, debido a que los materiales cromáticos son básicos y fundamentales y al encontrarse con estos obstáculos el aprendizaje se ralentiza. La falta de difusión y utilización de códigos empleados por personas con este problema es mínima, por lo que la enseñanza de este lenguaje no se realiza limitando su crecimiento personal y laboral. Así mismo, se desaprovecha la utilidad de medios tecnológicos para afrontar esta situación en la sociedad.

1.1.1. Problema General

¿Cómo será la simulación de un módulo reconocedor de símbolos mediante técnicas de procesamiento digital de imágenes para el aprendizaje del código colorADD?

1.1.2. Problemas Específicos

- a) ¿Cómo se establecerá la conexión inalámbrica entre PC y cámara IP para simular el módulo reconocedor de símbolos?
- b) ¿Cómo se diseñará una GUI en el software Matlab para simular el funcionamiento del módulo reconocedor de símbolos del código colorADD?
- c) ¿Cómo se implementará algorítmicamente las técnicas de procesamiento digital de imágenes, en el software Matlab, para el reconocimiento de 9 símbolos del código colorADD?

1.2. Objetivos

1.2.1. Objetivo General

Simular un módulo reconocedor de símbolos mediante técnicas de procesamiento de imágenes para el aprendizaje del código colorADD en personas con daltonismo.

1.2.2. Objetivos Específicos

- a) Establecer una conexión inalámbrica entre PC y cámara IP para simular el módulo reconocedor de símbolos.
- b) Diseñar una GUI en el software Matlab para simular el funcionamiento del módulo reconocedor de símbolos del código colorADD.
- c) Implementar algorítmicamente técnicas de procesamiento digital de imágenes, en el software Matlab, para el reconocimiento de 9 símbolos del código colorADD.

1.3. Importancia y justificación

1.3.1. Importancia

El desarrollo de esta investigación es importante para encaminar, en un futuro, un aprendizaje igualitario y la inclusión de personas que padecen del daltonismo empleando el código colorADD.

1.3.2. Justificación

En la actualidad, el daltonismo se concibe como un problema social en donde poco esfuerzo se ha dedicado para eliminar las barreras de comunicación cromáticas.

Al lograr este aprendizaje, se podrá constituir una sociedad más segura para las personas que no reconocen todos los colores visualmente, además de desarrollar iniciativas en cuanto al sistema de inserción laboral y educacional. A su vez, se incentivará el uso de este lenguaje que ya se viene trabajando en otros países con gran éxito y por ende la empleabilidad de este módulo de aprendizaje que será de gran utilidad para las personas diagnosticados con esta enfermedad.

1.4. Alcances y limitaciones

1.4.1. Alcances

Se desarrollará una interfaz GUIDE capaz de realizar el reconocimiento de símbolos del código colorADD a partir de imágenes preexistentes.

Dicho GUI será lo suficientemente práctica para que cualquier persona pueda utilizarla y a su vez pueda reconocer los símbolos del código colorADD de manera textual y auditiva.

1.4.2. Limitaciones

Debido a la gran cantidad de símbolos diferentes que existen en el código colorADD, solo se utiliza como muestra 10 símbolos escogidos. A su vez, el porcentaje de reconocimiento de símbolos fluctuará mucho de acuerdo al nivel de iluminación que presente el ambiente donde se realizará la captura virtual.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes del estudio de investigación

2.1.1. Antecedentes Nacionales

Sobrado, E. (2003) en su tesis diseñó con el fin de distinguir objetos como tuercas, tornillos, etc., sin importar la posición u orientación de estos, un brazo robot. Para lograr esta selección de objetos, dividió el proceso en 6 etapas: Adquisición de la imagen, pre procesamiento, segmentación, extracción de características, clasificación y utilización del brazo. La técnica empleada fue la de aprendizaje de la red neuronal y clasificación automática para sistema de visión. Ni bien el objeto es reconocido y localizado, el brazo robótico lo alzaría para situarlo en la posición predefinida por el operador. Para lograr discriminar los objetos de acuerdo a sus características en tiempo real, utilizar 4 de 7 momentos de Hu fue suficientes; además, el uso de las redes neuronales como procedimiento de clasificación tuvo un buen rédito para los objetos seleccionados a tratar.

2.1.2. Antecedentes Internacionales

Lema, L. (2014) en su tesis propuso un sistema de reconocimiento de imagen en el cual implementó y analizó algoritmos con el fin de comparar las imágenes con patrones desarrollados con ciertas particularidades. Con el fin de desarrollar el sistema de reconocimiento automático, utilizó los algoritmos de programación basados en técnicas de morfología matemática y sistemas de reconocimiento óptico elaborados en software Matlab 2008b. Así mismo poseía una base de datos con la información necesaria para la comparación y reconocimiento de los caracteres de las placas. Para la efectividad de la identificación, se consideró definir la distancia y tiempo máximo a la que debe estar y permanecer la cámara con referencia a la placa del vehículo. Además, el algoritmo creado logró el reconocimiento con facilidad de los caracteres sin necesidad de utilizar herramientas de toolbox adicionales.

Casas, A. (2017) en su tesis realizó el entrenamiento de una red neuronal artificial, específicamente de una red convolucional, con la finalidad de reconocer imágenes. Esta red supervisada fue entrenada a partir de un vasto grupo de símbolos ofrecidos por el dataset MNIST siendo capaz de clasificar los símbolos con una. El reconocimiento de imágenes alcanza un 97% de

asertividad utilizando redes neuronales. Por otro lado, el tiempo que demandó el entrenamiento de estas redes neuronales e identificar los filtros y neuronas ocultas para lograr la fiabilidad en el resultado fue alto.

Palacios, R. (2018) en su tesis detectó los objetos por forma o color a través del robot que utilizaba el sistema operativo robótico en dicha universidad. Se realizó la investigación del sistema operativo y se trabajó con el lenguaje de programación Python de forma básica con el propósito de desarrollar a futuro acciones con mayor nivel de dificultad y realizables. El reconocimiento de las imágenes se realizó con la Raspberry cámara o webcam, la programación del Raspberry PI y la creación de base de datos. El resultado en las pruebas de acuerdo a una resolución de 224px de la imagen indicó que el re-entrenamiento en un tiempo de 42 minutos ofreció una precisión del 82% y para un tiempo de 32 minutos la precisión disminuye a 71.1%

Sahare P. y Dhok S. (2018) indicaron que “el área de análisis de imágenes de documentos es un área que trata con imágenes escaneadas, debido que se utiliza la técnica de OCR en medio offline para el reconocimiento de caracteres, que en este caso pertenecen a documentos indios con antigüedad.” (p. 2)

Faiz, A. y Muhamad, A. et Al (2018) presentan en su paper “un reconocedor de caracteres offline para el lenguaje arábigo, la cual consiste en aplicar redes neuronales, el modelo oculto de Markov y lógica difusa para reconocer caracteres coránicos.” (p. 1)

2.2. Bases teóricas relacionadas con el tema

2.2.1. Técnicas de Procesamiento Digital de Imágenes

Dominguez, A (1996) expone que “el procesamiento digital de imágenes es conocido como el grupo de técnicas y procesos para obtener información desde una imagen con la ayuda de una computadora.” (p. 72)

2.2.1.1. Etapas de Procesamiento digital de Imágenes

Mejía, J. (2004) refiere que, para realizar un procesamiento digital de imágenes es necesario plantear etapas que expliquen el proceso. Como primera etapa describe la adquisición de imágenes, la cual involucra un pre procesamiento. Una vez obtenida la adquisición, como segunda etapa, es necesario optimizar la imagen adquirida, ya que en muchas ocasiones no se encuentra apta para el

procesamiento; esto involucra mejorar las características particulares, las cuales serán las que buscaremos procesar. Como tercera etapa, se encuentra el procesamiento de color, para la cual, la imagen a procesar ya debe encontrarse en condiciones óptimas para aplicarles los algoritmos. Es aquí donde, para simpleza del proceso, se convierte la imagen a una imagen monocromática. Finalmente se aplicarán los algoritmos necesarios a fin de obtener una respuesta al procesamiento de acuerdo a lo procesado. En la Figura N° 1, se podrá observar a manera gráfica las etapas del procesamiento de imágenes. (p. 12)

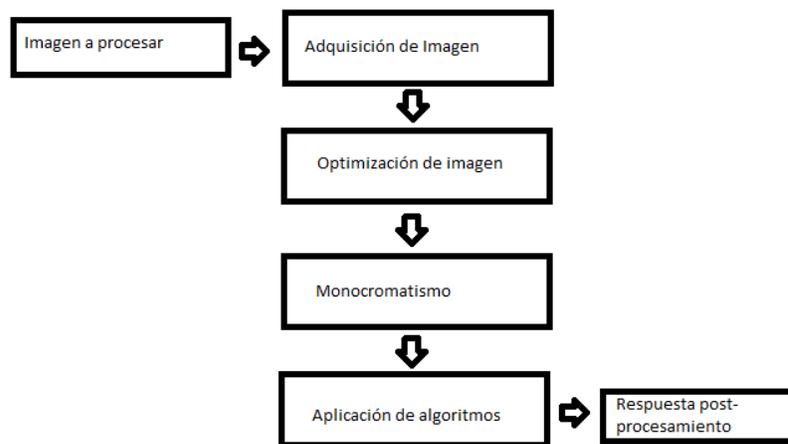


Figura N° 1: Etapas de procesamiento digital de imágenes

Fuente: J. Mejía (2004). Apuntes de Procesamiento Digital de Imágenes.

Mejía, J. (2004) acota que “las técnicas de procesamiento digital de imagen se diferencian una de la otra por el método de adquisición, siendo la principal diferenciación el tratamiento espacial y el tratamiento en frecuencia de las imágenes.” (p. 83)

Con referencia al tratamiento espacial de las imágenes, Aguirre, N. (2002) expone que “para poder hacer un tratamiento espacial, es requerido relacionar los pixeles vecinos como se ejemplifica en la Figura N°2, los cuales generan una región que se genera si y solo si tienen en común una de sus fronteras o esquinas para una adquisición sencilla por asociación.” (p. 52)

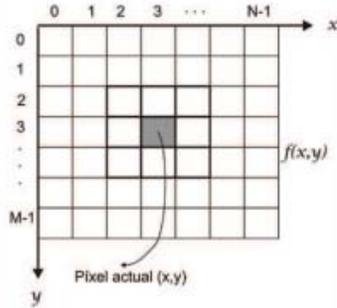


Figura N° 2: Ejemplificación de vecindad de píxeles de una matriz 3 x 3

Fuente: Aguirre, N (2002). Procesamiento de Imágenes, (52)

En cuanto al tratamiento en frecuencia de las imágenes, Aguirre, N. (2002) plantea que para poder hacer un tratamiento en frecuencia es requerido convertir la imagen a una representación en el dominio de la frecuencia. Los filtros en el dominio de la frecuencia se usan, principalmente, para eliminar altas o bajas frecuencias de la imagen, lo que se traduce en suavizar la imagen, o bien, realzar o detectar bordes. (p. 55). En la Figura N°3, se podrá observar a manera gráfica lo que expone el autor.

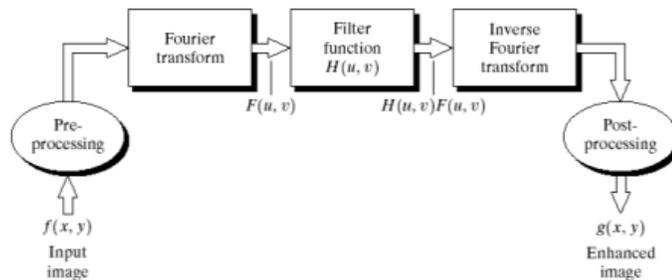


Figura N° 3: Proceso de imagen en el dominio de la frecuencia.

Fuente: Aguirre, N (2002). Procesamiento de Imágenes, (55)

2.2.1.2. Momentos Invariantes a la rotación, traslación y escalado: Momentos de Hu

Las imágenes presentan transformaciones simples de coordenadas espaciales como la traslación, rotación y escalado de acuerdo con Flussner, J. y Suk, T. et Al (2009) “La invariancia en traslación se define como el cambio del centroide de la imagen de tal forma que coincida con el sistema de coordenadas” (p. 13). Los momentos centrales de una imagen se representan como sigue a continuación:

$$\mu_{pq} = \iint_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x, y) dx dy \dots\dots\dots (1)$$

Donde:

$$x_c = \frac{m_{10}}{m_{00}}, \quad y_c = \frac{m_{01}}{m_{00}} \quad , \quad m_{10}: \text{Momento de orden 0 y 1}$$

La transformación del escalado se puede mantener invariante normalizando de forma adecuada cada momento. El momento a usar para normalizar podría ser cualquiera de los calculados, verificando que al aplicarlos sea diferente de cero en todas las imágenes. Los momentos de bajo orden se consideran lo más adecuados debido a que son más estables y más fáciles de calcular por lo que se utiliza comúnmente el momento μ_{00} .

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^w} \dots\dots\dots (2)$$

Donde:

$$w = \frac{p+q}{2} + 1 \quad , \quad w: \text{Potencia de } \mu_{00}$$

Dentro de los métodos de reconocimiento para extraer características de imágenes se encuentra los Momentos de Hu. El momento de rotación invariante nació de la aplicación de la teoría de los invariantes algebraicos resultando en los 7 momentos invariantes en rotación, traslación y escalado:

$$\begin{aligned}
\phi_1 &= m_{20} + m_{02} \\
\phi_2 &= (m_{20} - m_{02})^2 + 4m_{11}^2 \\
\phi_3 &= (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2 \\
\phi_4 &= (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2 \\
\phi_5 &= (m_{30} - 3m_{12})(m_{30} + m_{12})((m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2) \\
&\quad + (3m_{21} - m_{03})(m_{21} + m_{03})(3(m_{30} + m_{12})^2 \\
&\quad - (m_{21} + m_{03})^2) \\
\phi_6 &= (m_{20} - m_{02})((m_{30} + m_{12})^2 - (m_{21} + m_{03})^2) \\
&\quad + 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03}) \\
\phi_7 &= (3m_{21} - m_{03})(m_{30} + m_{12})((m_{30} + m_{12})^2 - 3(m_{21} + \\
&\quad m_{03})^2) - (m_{30} - 3m_{12})(m_{21} + m_{03})(3(m_{30} + m_{12})^2 - \\
&\quad (m_{21} + m_{03})^2) \dots \dots \dots (3)
\end{aligned}$$

Este método tiene como fin definir las particularidades en global e invariantes de las imágenes. Estas se representan con valores estadísticos para establecerse constantes, posterior a alguna modificación como la rotación, traslación o escalado de la imagen. Los momentos trabajan particularmente con las regiones de píxeles suministrando una representación global de las imágenes.

2.2.2. Redes Neuronales

Rivas, W. y Mazón, B. (2017) manifiestan que las redes neuronales también llamados sistemas conexionistas pertenecen a las técnicas de la Inteligencia Artificial. Se encargan de procesar la información a partir de entradas asignadas las cuales se multiplican por pesos determinados al azar, construyendo nodos enlazados por conexiones o enlaces sinápticos. Además, puntualiza que este método es empleado principalmente para la clasificación, identificación y diagnóstico donde los datos tengan mayor relevancia que el conocimiento. (p. 12)

Uno de los modelos de Redes Neuronales que permite resolver los problemas de clasificación es el perceptrón multicapa. Flórez, R. y Fernández, J. (2008) describen al “perceptrón multicapa como un modelo neuronal con propagación hacia adelante, evitando así las conexiones hacia atrás, de tal forma que ninguna de las salidas

neuronales se comporte como una entrada para las neuronas de capas previas o de la misma.” (p. 60)

A diferencia del perceptrón simple, en la arquitectura del perceptrón multicapa se incluyen una o varias capas ocultas como se presenta a continuación en la Figura N°4:

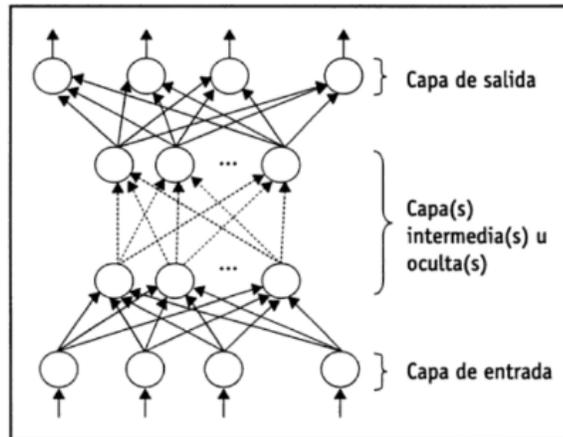


Figura N° 4: Arquitectura del Perceptrón Multicapa

Fuente: Flórez y Fernández (2008). Las Redes Neuronales Artificiales, (61)

2.2.3. Reconocedor de Símbolos

El reconocimiento de símbolos puede ser descrito en diferentes contextos. Según los autores Gonzalez, R. y Woods, R. (2002) definen el reconocimiento como “la asignación de una etiqueta a un elemento en base a sus descripciones las cuales determinarán la clasificación que tendrá dentro de un global.” (p 693)

Buchler, J. (1955) plantea que “los símbolos se establecen a partir de una norma dictada por su interpretante, considera que es una representación como por ejemplo las oraciones, libros y otros signos habituales.” (p 112)

La identificación de símbolos utiliza descriptores y rasgos para definir a qué tipo de objeto hace referencia, luego de definir estas características físicas se debe emplear técnicas para el reconocimiento. De acuerdo a De la Fuente, M. y Calonge, T. (1995) “las técnicas de reconocimiento se dividen en 2 tipos: Método de decisión teórica y métodos estructurales” (p 99). En el primer método utiliza las características para representarlos en vectores a fin de encontrar la mayor cantidad de similitudes y agruparlos para clasificar los vectores en grupos. El segundo método se basa en representar los rasgos de forma simbólica. En general estos métodos implican crear el lenguaje de alto nivel en el que se logre identificar el nivel de coincidencia de los símbolos.

2.2.3.1. Código colorADD

En el caso particular de los símbolos del colorADD, estos son descritos a partir de cinco símbolos mostrados en la Figura N°5, los cuales representan los colores básicos azul, amarillo y rojo, y los colores blanco y negro.



Figura N° 5: Cinco símbolos básicos del código colorADD

Fuente: Neiva, M (2010) colorADD-Color Identification System [Imagen].

Recuperado de <http://www.coloradd.net/code.asp>

Estos símbolos básicos se combinan entre ellos o con secundarios para formar una amplia variedad de colores, como los que se observan en la Figura N°6.



Figura N° 6: Combinación de símbolos del código colorADD

Fuente: Neiva, M (2010) colorADD-Color Identification System [Imagen].

Recuperado de <http://www.coloradd.net/code.asp>

Es importante mencionar que este código colorADD se viene utilizando en el país en donde se originó la idea, Portugal. La noticia “Tres símbolos para que 350 millones de daltónicos puedan identificar colores” da a conocer que se viene empleando en áreas

estratégicas como medicamentos, líneas de señalización en hospitales, líneas de metros ilustrado en la Figura N°7 y banderas de seguridad en las playas; proyectándose la idea a países como Brasil, Alemania, Japón o Estados Unidos. Las empresas que han adoptado este código de colores a partir de la licencia otorgada por el creador Miguel Neiva y lo implementan en los productos que comercializa son: la fábrica de lápices Viarco y la marca de ropa Zippy. De acuerdo al artículo “El Código “braile” para daltónicos se inventa en el siglo XXI”, la empresa Viarco goza con un mercado internacional el cual se expande en más de 25 países y la marca de ropa Zippy se instaló en España logrando un gran impacto con la publicidad del código colorADD.



Figura N° 7: Líneas de Señalización del Metro de Porto, Portugal

Fuente: Lázaro, N (2018) Metro de Porto. A vida em movimento [Imagen].

Recuperado de <https://brands.elconfidencial.com/sociedad/2018-06-27/daltonicos-simbolos-colores-abecedario-bra-1584172/>

2.2.4. Conexión inalámbrica

Dentro del mundo de las conexiones inalámbricas existen muchas clases, las cuales se clasifican por rango de cobertura que poseen.

a) Redes WAN

De acuerdo con Andreu, F. y Pellejero, L. (2006) “las redes WAN, o redes de área amplia, se identifican por tener rangos de coberturas altas que traspasan los kilómetros de radio.” (p. 108)

b) Redes WLAN

A su vez, Andreu, F. y Pellejero, L. (2006) indican que “las redes WLAN son una mixtura de las redes WAN con las redes LAN, la cual en términos simples sería una red LAN con mayor cobertura.” (p. 108)

c) Redes PAN

Del mismo modo, Andreu, F. y Pellejero, L. (2006), “las redes PAN, o redes de área personal, se caracterizan por tener rangos de cobertura cortos, que no superan 1 metro de distancia.” (p. 108)

La herramienta que permite realizar la conexión inalámbrica es la DroidCam. En la Figura N°8 se visualiza la interfaz de la aplicación, la cual muestra la dirección IP lo que permite convertir el dispositivo con operador Android en una cámara web IP de tal forma que se conecta con cualquier red de internet y permite visualizar las imágenes capturadas por el dispositivo Android en la plataforma seleccionada.

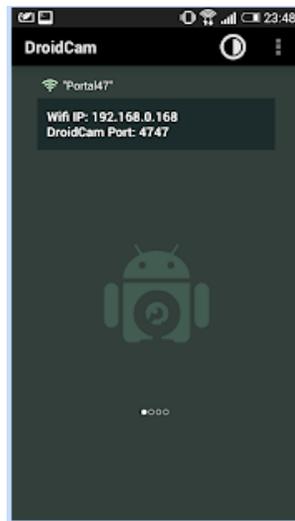


Figura N° 8: Software Droidcam

Fuente: Elaboración Propia

2.3. Variables de Estudio

En cuanto a las variables de estudio empleadas en función al tipo de investigación correlacional desarrollada, se tiene:

a) Variable Independiente: Procesamiento Digital de Imágenes

- Indicadores:
- Técnicas de Procesamiento
 - Información de la imagen

b) Variable Dependiente: Reconocimiento de Símbolos

- Indicadores:
- Porcentaje de reconocimiento
 - Tiempo de conexión entre PC y cámara IP

2.4. Tipo y método de investigación

2.4.1. Tipo de investigación

La siguiente investigación fue de tipo experimental, debido a que se simularon las técnicas de procesamiento digital de imágenes para el módulo reconocedor de símbolos, teniéndose como resultados el grado de identificación y asociación de estos con respecto a una base prediseñada.

2.4.2. Método de investigación

El desarrollo de la siguiente investigación constó de la siguiente metodología:

- a. Búsqueda de información y asesoramiento
- b. Trabajo en laboratorio, a fin de poder realizar las pruebas de funcionamiento correspondientes.
- c. Aplicación de información obtenida para reconocimiento de caracteres en el software Matlab.
- d. Desarrollo de los procesos de programación y pruebas de los algoritmos generados para identificación de símbolos.
- e. Desarrollo y pruebas de la interfaz a desarrollarse.
- f. El estudio es de tipo longitudinal y se realizó entre los meses de Mayo 2019 a Noviembre 2019.
- g. Optimización de los algoritmos post análisis de las conclusiones y observaciones.

2.4.3. Diseño de investigación

Para la presente investigación, la población es igual a la muestra debido a que se utilizarán a personas sin deficiencias visuales para las pruebas de reconocimiento de símbolos.

2.5. Instrumento de recolección de datos

Se utilizó el dispositivo Android con la cámara IP para la toma real y la plataforma de internet para la obtención de las imágenes ideales.

2.6. Técnicas de recolección de datos

La recolección de datos se obtuvo mediante las siguientes actividades:

- a. Revisión de datos y levantamiento de información online
- b. Procesamiento de imágenes y comparación con muestras prediseñadas.
- c. Pruebas simuladas de obtención de data y comparación.

2.7. Definición de términos

Código colorADD: Sistema identificador de colores para daltónicos creado por el portugués Miguel Neiva, Máster en Diseño. De acuerdo a Neiva (2010) “este lenguaje está basado en los tres colores primarios representados en símbolos gráficos”. Estos símbolos al combinarse generan una amplia paleta de colores logrando ser asociados e identificados por las personas con daltonismo. Este vocabulario está siendo empleado actualmente en hospitales, transportes públicos, escuelas y universidades, así como en la industria textil y calzado.

Daltonismo: Montenegro (2004) lo define como la incapacidad de distinguir ciertos colores. Esta afección ocurre cuando los conos, encargados de detectar los colores, están ausentes o no trabajan de forma adecuada. Esta enfermedad es hereditaria y se manifiesta en el cromosoma X, por lo que si el hombre es afectado por este gen desarrollará la enfermedad, en cambio la mujer los dos cromosomas deberán ser afectados para manifestar este padecimiento, si solo tiene uno será considerada portadora. (p. 170)

GUI: Siglas que significan Graphic User Interface, que traducido significa interfaz de usuario gráfico.

CAPÍTULO III: DISEÑO DE LA INVESTIGACIÓN

En este capítulo se muestra paso a paso, cómo es que se logró definir los algoritmos para la ejecución del módulo reconocedor de símbolos del código colorADD

3.1. Diseño planteado del módulo reconocedor de símbolos del código colorADD

Para el diseño planteado, ha sido necesario identificar cada punto esencial del correcto funcionamiento del módulo, siendo estas mostradas a continuación en la Figura N° 9:

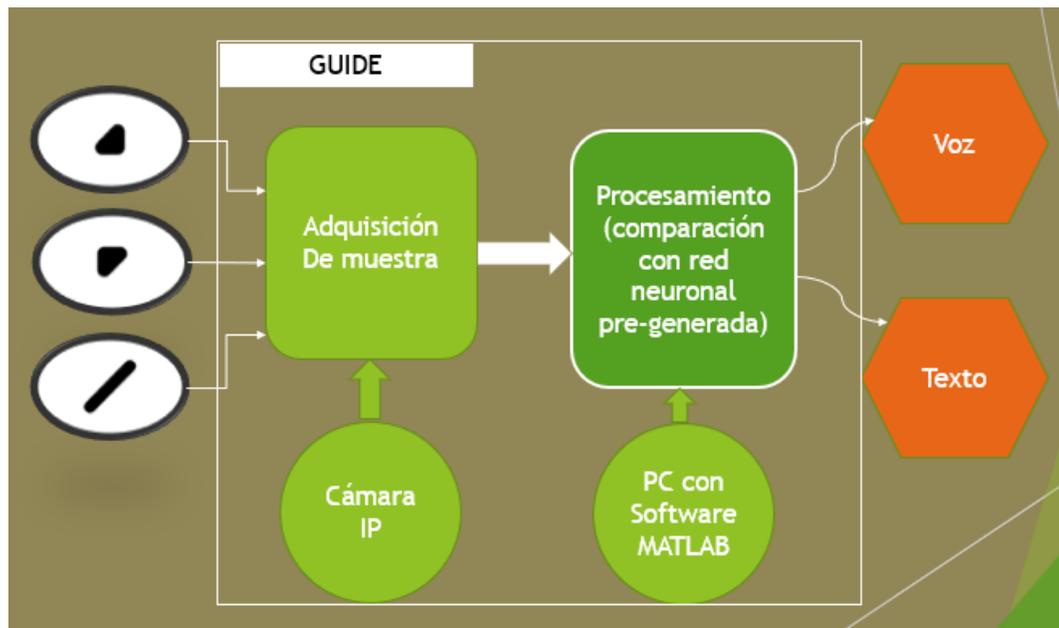


Figura N° 9: Diagrama de bloques del módulo reconocedor de símbolos del código color ADD

Fuente: Elaboración Propia

Puntualmente el módulo reconocedor constó de 4 etapas, la captura de la muestra, la adaptación y filtrado de la muestra obtenida y por último su procesamiento con el fin de obtener la respuesta del módulo que fue la voz y el texto. En puntos subsecuentes se indica el proceso paso a paso de cómo es que se logró que el módulo funcione de manera correcta.

3.2. Conectividad de dispositivo IP hacia interfaz usuaria

Para iniciar con la primera etapa fue necesario definir 2 puntos, primero, ¿cómo se obtendría las muestras?, y segundo, ¿cómo las ingresaremos a un medio procesable virtual? La respuesta a la primera pregunta fue sencilla: se optó por elegir nuestro dispositivo móvil, la herramienta más accesible para lograr la conectividad, un

Smartphone con ayuda de una aplicación (Droidcam), la cual ayuda a simular una cámara IP, debido a que utilizar una real tendría mucho costo. A continuación, en la Figura N°10, se muestra el ícono del aplicativo DroidCam en la interfaz del Smartphone; y en la Figura N°11, se muestra una prueba del enlazado del aplicativo DroidCam en una PC.



Figura N° 10: Icono en menú móvil del aplicativo DroidCam

Fuente: Elaboración Propia

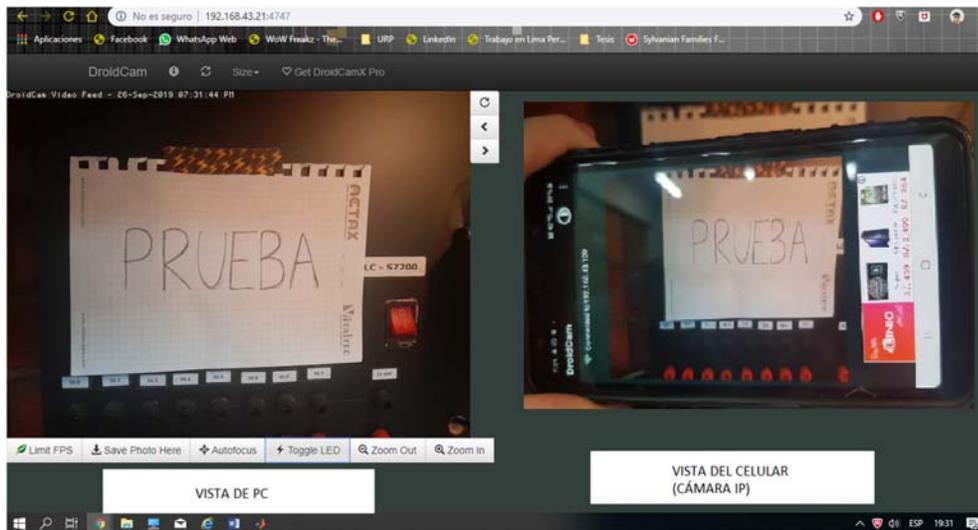


Figura N° 11: Pruebas de Cámara IP (Vista de PC en navegador vs Vista de Celular)

Fuente: Elaboración Propia

La respuesta a la segunda pregunta fue más sencilla, en este caso usamos el software Matlab puesto que se tenía en mente hacer uso de algoritmos matemáticos para poder reconocer los símbolos como tal.

3.2.1. Conexión con software MATLAB

Se conectó el aplicativo Droidcam del Smartphone al software Matlab gracias a los módulos adicionales que presenta el software como Add-ons y además los códigos

“cam=ipcam(‘dirección ip : puerto/video’)”, de tal forma que permita agregar a la variable “cam” la cámara IP simulada por el Smartphone.

Para establecer la conexión fue necesario que tanto el equipo que poseía el software Matlab como el Smartphone, que tiene el aplicativo DroidCam, permanezcan dentro de una misma red WI-FI , con el objetivo de lograr la asignación de la dirección IPv4. Se consideró otorgar un tiempo de procesamiento de conexión para no perder el enlace. A continuación en las Figuras N°12 y N°13 se observará el algoritmo utilizado para conectar el aplicativo DroidCam y la definición de la variable “cam” vista desde MATLAB.

```
3 %% Conectar con camara
4 - close all;
5 - clc;clear all;
6 - pause(2)
7 - cam=ipcam('http://192.168.43.21:4747/video');
8 - pause(2)
```

Figura N° 12: Algoritmo de conexión de Cámara IP (aplicativo DroidCam) con software Matlab.

Fuente: Elaboración Propia



```
Command Window
>> cam
cam =
    ipcam with properties:
        URL: 'http://192.168.43.21:4747/video'
        Username: ''
        Password: ''
        Timeout: 10
```

Figura N° 13: Identificación de la variable “cam”

Fuente: Elaboración Propia

3.2.2. Captura de muestra a procesar

Una vez realizada la conexión a Matlab de la cámara IP virtual, la captura de las muestras se ejecutó utilizando la línea de código “ap=snapshot(cam);” la cual asigna una fotografía en resolución 640x420 a la variable “ap”. Cabe mencionar que para este proyecto de tesis, se tuvieron ciertas condiciones lumínicas a fin de obtener una

imagen lo menos ruidosa posible. A continuación en las Figuras N°14 y N°15 se mostrará la definición de la variable ap post captura, y su presentación en MATLAB.

```
Command Window
>> whos ap
Name      Size      Bytes  Class  Attributes
ap       480x640x3  921600  uint8
```

Figura N° 14: Identificación de la variable “ap”

Fuente: Elaboración Propia

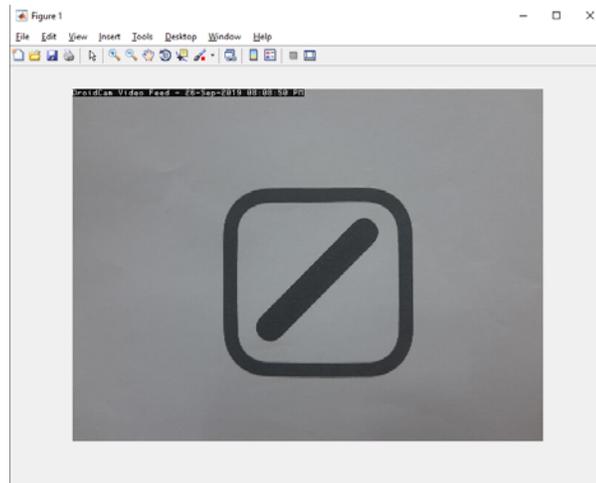


Figura N° 15: Imagen tomada como muestra (Color Amarillo Claro) mostrado en MATLAB

Fuente: Elaboración Propia

3.3. Adecuación y filtrado de muestra capturada

Una vez obtenida la imagen a partir del MATLAB, fue indispensable la adaptación y el filtrado, esto por simpleza al procesado

3.3.1. Adecuación

Debido a que la imagen entregada es una matriz de 3 capas (R-G-B) fue necesaria su adecuación a una imagen de 1 capa (B-N), para ello se hizo uso del histograma (con la línea de código “imhist(ap)”, así se pudo determinar el pixel valle de la imagen, para este caso asumimos el valor 180, la cual permitió convertir una imagen a color en una imagen binaria. El proceso que siguió al histograma, consistió en asignar una muestra de muchas (y restarla de 255 pixeles), a fin de obtener la imagen compuesta por ceros y unos de manera invertida. En las Figuras N°16 y N°17 mostradas a continuación se observa el histograma de la captura y el algoritmo

para convertir la captura en blanco y negro. Obteniendo como resultando la Figura N° 18.

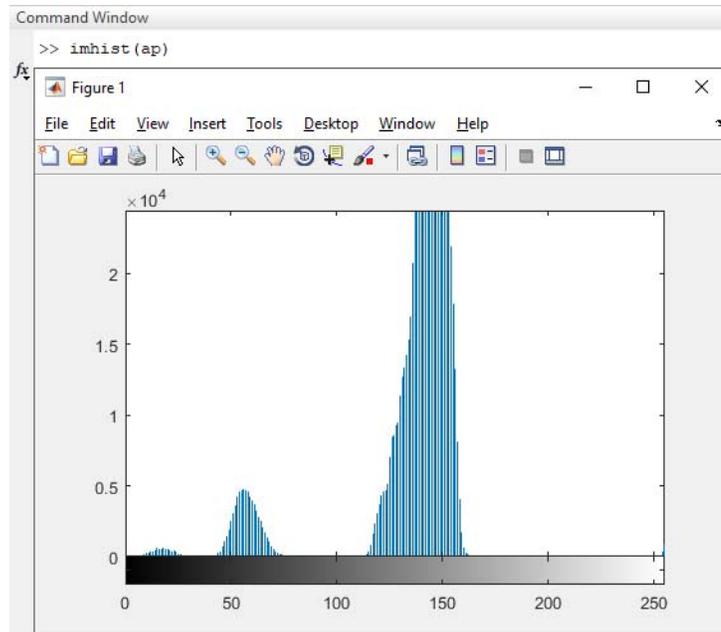


Figura N° 16: Histograma de la muestra capturada

Fuente: Elaboración Propia

```
11 %% Convertir de blanco a negro
12 - b= 255-double(ap(:,:,1));
13 - b = uint8(b);
14 - b = double(b)>=180;
```

Figura N° 17: Algoritmo para convertir imagen de color a blanco y negro

Fuente: Elaboración Propia

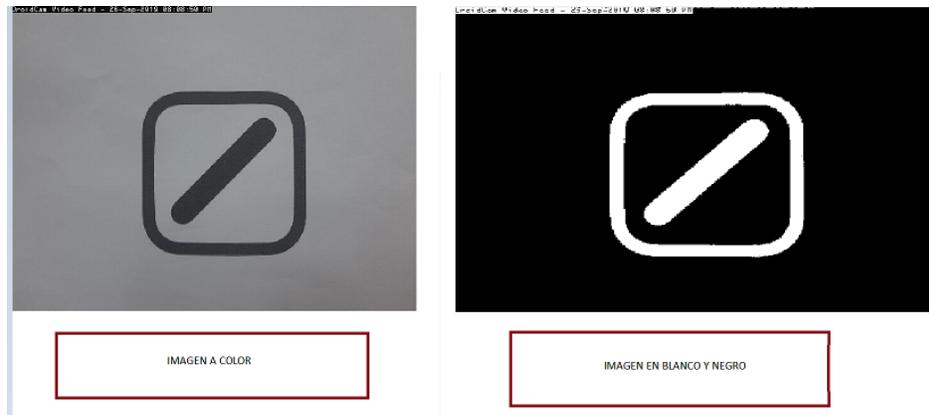


Figura N° 18: Aplicación del algoritmo convertidor de blanco y negro sobre muestra capturada

Fuente: Elaboración Propia

3.3.2. Filtrado: Filtro de Mediana

Al haber obtenido del punto anterior la imagen en blanco y negro invertida, se obtuvo una imagen ruidosa, la cual tuvo que ser filtrada. Cabe recordar que nuestra imagen muestra fue obtenida gracias a ciertos factores de iluminación indicados previamente, por lo que se presentó ruido al momento de la captura. Para la eliminación del ruido como tal, fue imprescindible hacer uso del filtro Mediana, el cual se encarga de obtener interinamente el valor de la mediana de los píxeles en la imagen. En las Figuras N°19 y N°20 mostradas a continuación, se observa el algoritmo de filtrado y cómo se manifiesta poniendo en contraste una figura sin filtro y la otra con filtro.

```

16 %% Filtrado
17 - b(1:20 , : ) = zeros(20,640);
18 - b = medfilt2(b);

```

Figura N° 19: Algoritmo de filtrado

Fuente: Elaboración Propia



Figura N° 20: Imagen capturada sin filtro e imagen capturada con filtro

Fuente: Elaboración Propia

3.4. Procesamiento de muestras

3.3.3. Generación de base de datos ideal

Con el fin de realizar el procesamiento de las muestras obtenidas en los puntos anteriores, se generó una base de datos ideal utilizando imágenes ideales (libres de ruido) a fin de que las muestras puedan ser procesadas.

3.4.1.1. Aplicación de los momentos de Hu sobre imágenes ideales

Los momentos de HU son 7 valores que identifican a una imagen, estos no varían si la imagen rota o cambia de tamaño. Para la generación de la base de datos, se procedió a la toma de imágenes ideales de los símbolos que escogimos, los cuales representan el marrón, amarillo, amarillo claro, violeta, rojo, azul, verde, naranja, celeste y rosado, para ser rotadas desde 20° hasta -20° cada 5°, obteniéndose los siguientes valores representados a partir de la Figura N°21 hasta la Figura N°29.

a 0°								
n°	Color	10*M1	100*M2	1000*M3	10000*M4	100000*M5	1000000*M6	10000000*M7
1	MARRON	2.099996645	0.10880404	0.000345861	0.009023372	-1.32456476527957E-08	0.005192888	-1.13425E-05
2	AMARILLO	4.757471651	19.93220992	0.104525008	0.411175244	7.30E-06	7.917343855	-0.001845872
3	AMARILLO CLARO	5.357268983	0.251824041	0.000513678	0.148512244	3.51E-06	-0.348121352	-0.001993736
4	VIOLETA	2.413703937	2.087457296	0.003107344	0.007102011	-2.68503E-08	-0.070019825	-8.5261E-06
5	ROJO	1.787741534	0.349445579	1.043996542	0.357972346	-0.005190123	-2.020903247	0.029339879
6	AZUL	1.783183805	0.348573234	1.063024698	0.131409379	-0.002759152	-0.597345875	-0.013897817
7	VERDE	2.418948245	0.692047391	7.941689517	1.996658643	-0.250812143	-16.55230722	0.257641936
8	NARANJA	2.41821748	0.690886852	7.749511917	2.608090045	-0.266335216	-20.11281732	1.818053533
9	CELESTE	5.587652156	0.00207029	0.011516788	3.7870338	-0.004277283	-1.708243698	-0.405060613
10	ROSADO	5.571111282	0.001976904	0.006394388	5.459461015	0.00857199	-2.412997656	0.002087769

Figura N° 21: Tabla de momentos de Hu diferenciados de los colores con una rotación a 0°

Fuente: Elaboración Propia

a -15° (derecha)								
n°	Color	10*M1	100*M2	1000*M3	10000*M4	100000*M5	1000000*M6	10000000*M7
1	MARRON	2.10168328	0.109984259	0.000479175	0.012595333	2.20433E-08	0.0235551	-2.22046E-05
2	AMARILLO	4.766214428	20.01820207	0.098705995	0.773979513	0.000245142	29.72740937	-0.000596082
3	AMARILLO CLARO	5.350538154	0.250047081	0.001506696	0.331393524	5.75132E-05	1.095024888	0.016411773
4	VIOLETA	2.412977333	2.093474326	0.005373797	0.013215948	-1.03197E-07	-0.171230621	8.87617E-06
5	ROJO	1.78465939	0.34904725	0.938283054	0.200798412	0.001453244	-1.074500497	-0.002873274
6	AZUL	1.784787528	0.349271232	1.091585404	0.165733128	-0.000696299	-0.330443332	-0.013808839
7	VERDE	2.418502546	0.690570342	8.103540349	2.298866084	0.035579422	-19.06525206	-2.286973411
8	NARANJA	2.420101475	0.692233073	7.744995854	2.720915455	0.083007929	-20.39839283	-0.556733737
9	CELESTE	5.58102922	0.001939663	0.012531391	3.125844189	-0.000725108	-1.350546288	-0.806832941
10	ROSADO	5.578264272	0.002109767	0.003386165	3.869453537	0.005664337	-1.676174151	-0.096486323

Figura N° 22: Tabla de momentos de Hu diferenciados de los colores con una rotación a -15°

Fuente: Elaboración Propia

a +15° (izquierda)								
n°	Color	10*M1	100*M2	1000*M3	10000*M4	100000*M5	1000000*M6	10000000*M7
1	MARRON	2.101479128	0.110443092	0.00090552	9.47793E-05	1.45272E-11	-0.00027185	-6.68502E-09
2	AMARILLO	4.766614952	20.02238695	0.215841921	0.863415238	9.78E-04	27.56374782	0.18469846
3	AMARILLO CLARO	5.352689477	0.25025551	0.006911764	0.451935923	8.61E-05	1.270766911	-0.016575801
4	VIOLETA	2.412760753	2.090572959	0.0014028	0.001701021	2.88E-08	0.01709726	6.05E-07
5	ROJO	1.784860352	0.346110779	1.096976853	0.351893391	5.99E-04	-2.063373832	0.010110381
6	AZUL	1.784419604	0.347251456	1.005569033	0.166170265	-2.71E-04	-0.288777135	-0.009695918
7	VERDE	2.418832147	0.688678878	7.579498503	1.570640049	-0.007837584	-13.03419414	0.378135136
8	NARANJA	2.419299784	0.689859714	7.831861711	2.256063235	0.011033792	-17.91064672	0.619912314
9	CELESTE	5.58241819	0.001923648	0.007958371	3.118055934	-0.002187426	-1.338670122	-0.203103299
10	ROSADO	5.581903044	0.001932438	0.00917302	5.287032036	0.001424748	-2.083445663	-1.964961382

Figura N° 23: Tabla de momentos de Hu diferenciados de los colores con una rotación a 15°

Fuente: Elaboración Propia

a -5°								
n°	Color	10*M1	100*M2	1000*M3	10000*M4	100000*M5	1000000*M6	10000000*M7
1	MARRON	2.101804289	0.110172649	8.21E-04	0.004041692	3.67E-08	0.012467137	-1.10E-06
2	AMARILLO	4.769358636	20.05385197	0.037559367	0.056409814	8.93E-06	2.122873249	0.001025739
3	AMARILLO CLARO	5.353885315	0.248207675	0.001095984	0.408303086	6.79E-05	-0.653936267	-0.01629833
4	VIOLETA	2.41174623	2.089448034	7.71E-04	4.18E-04	-1.48E-10	-5.58E-04	4.43E-08
5	ROJO	1.785862012	0.351889179	0.970818768	0.283313703	-0.001583544	-1.242164271	0.036292435
6	AZUL	1.785424696	0.34935425	1.054688178	0.079088325	-0.00156088	-0.232850935	-0.00826824
7	VERDE	2.4196667	0.692763122	8.061063566	2.20255124	-0.228522667	-18.33185015	-0.942424602
8	NARANJA	2.419585418	0.691952445	7.68174996	2.390886895	-0.235211181	-19.82367264	-0.570979772
9	CELESTE	5.58188304	0.001677016	0.013114105	2.82405953	-0.003679314	-1.15609246	-0.121164247
10	ROSADO	5.580530517	0.001761603	0.005155413	4.975519528	0.004908454	-1.859888638	-0.191430843

Figura N° 24: Tabla de momentos de Hu diferenciados de los colores con una rotación a -5°

Fuente: Elaboración Propia

a 5°								
n°	Color	10*M1	100*M2	1000*M3	10000*M4	100000*M5	1000000*M6	10000000*M7
1	MARRON	2.100937093	0.109958874	8.68E-04	0.001227582	5.83E-09	0.003036796	2.46E-07
2	AMARILLO	4.772933373	20.08157093	0.199039	1.042824869	3.73E-04	37.42515732	0.159203572
3	AMARILLO CLARO	5.35290035	0.24954284	0.002236542	0.390696523	7.13E-05	0.504624016	-0.011396425
4	VIOLETA	2.414382186	2.095381834	6.13E-04	3.35E-04	-2.49E-10	-0.001004194	1.27E-08
5	ROJO	1.786779612	0.352028097	1.029554764	0.42994666	-1.63E-03	-0.918128571	0.043613266
6	AZUL	1.784736221	0.349079182	1.086897922	0.21270763	-0.003532933	-0.720144978	-0.028294884
7	VERDE	2.419829729	0.691484969	7.970702081	2.19489882	-0.269567633	-17.69900096	-0.3183854
8	NARANJA	2.419580521	0.693811224	7.807766769	2.644796552	-0.226505943	-19.85041607	2.193925797
9	CELESTE	5.582948716	0.001910407	0.010701502	4.122642371	-0.005856106	-1.793003645	0.02564105
10	ROSADO	5.578845342	0.00201436	0.008874878	6.852316574	0.005567634	-2.747282841	-1.415986405

Figura N° 25: Tabla de momentos de Hu diferenciados de los colores con una rotación a +5°

Fuente: Elaboración Propia

a -10°								
n°	Color	10*M1	100*M2	1000*M3	10000*M4	100000*M5	1000000*M6	10000000*M7
1	MARRON	2.100715523	0.109688745	8.99E-04	0.003119216	3.17E-08	0.01029699	5.06E-07
2	AMARILLO	4.76760867	20.03876422	0.030445828	0.041296793	4.18E-06	1.143634283	4.44E-04
3	AMARILLO CLARO	5.353111499	0.251468782	8.47E-04	0.601591446	1.65E-04	2.262750621	0.049998187
4	VIOLETA	2.412935322	2.091359415	0.00100172	9.79E-04	1.44E-09	0.002966839	1.26E-07
5	ROJO	1.78471451	0.347742168	0.975834015	0.298124407	-9.03E-04	-1.613045324	0.006704928
6	AZUL	1.78528773	0.350938618	1.083683146	0.162942838	-0.001746225	-0.164838266	-0.014189115
7	VERDE	2.419600667	0.693561923	8.002454349	2.083887293	-0.12530197	-16.76419826	-2.574449674
8	NARANJA	2.419264343	0.691676307	7.74789757	2.660500456	-0.115390039	-21.73971545	-1.125271284
9	CELESTE	5.580010693	0.00210277	0.014120857	4.613331956	-0.005198758	-2.11548494	-0.867271261
10	ROSADO	5.580848764	0.00189137	0.00301907	6.869899383	0.010450767	-2.745732719	-0.003431748

Figura N° 26: Tabla de momentos de Hu diferenciados de los colores con una rotación a -10°

Fuente: Elaboración Propia

a 10°								
n°	Color	10*M1	100*M2	1000*M3	10000*M4	100000*M5	1000000*M6	10000000*M7
1	MARRON	2.101676726	0.109972316	7.33E-04	5.52E-04	1.06E-09	0.001652646	5.47E-09
2	AMARILLO	4.766109572	20.01415064	0.211404306	0.903566233	4.12E-04	28.03958913	0.140141368
3	AMARILLO CLARO	5.353944956	0.249584037	0.00406732	0.38752153	8.15E-05	1.056514868	-0.008401116
4	VIOLETA	2.412984006	2.092623515	4.82E-04	2.46E-04	-8.72E-11	-0.003137216	8.19E-09
5	ROJO	1.785126988	0.348795793	1.046974465	0.314594901	-2.23E-03	-1.747095719	0.001126448
6	AZUL	1.784895763	0.349432411	1.077440021	0.21388044	-0.002465877	-1.099037764	-0.005001515
7	VERDE	2.418351804	0.689205423	7.820550004	1.974948325	-0.133834339	-16.39430265	0.723094736
8	NARANJA	2.420182051	0.690533834	8.012215487	2.955639063	-0.150123881	-23.67417579	1.801740131
9	CELESTE	5.582521447	0.001938536	0.010307451	4.013325777	-0.005517087	-1.761783329	0.025581468
10	ROSADO	5.583098733	0.00177556	0.008138012	6.881298558	0.005674442	-2.84605904	-1.254214859

Figura N° 27: Tabla de momentos de Hu diferenciados de los colores con una rotación a 10°

Fuente: Elaboración Propia

a -20°								
n°	Color	10*M1	100*M2	1000*M3	10000*M4	100000*M5	1000000*M6	10000000*M7
1	MARRON	2.101645102	0.110023245	1.18E-03	0.00154241	5.16E-09	0.004093375	-7.09E-09
2	AMARILLO	4.768232299	20.04599716	0.017667309	0.017205985	5.33E-07	-0.222765782	-1.89E-05
3	AMARILLO CLARO	5.353274655	0.250204748	0.002569042	0.615917455	2.65E-04	1.129191444	0.047340116
4	VIOLETA	2.414183265	2.094566895	0.00144131	0.001280089	2.46E-10	0.014632142	-7.62E-08
5	ROJO	1.784549722	0.347992056	0.956697486	0.243907131	0.003179108	-1.149466397	-0.001713634
6	AZUL	1.785303049	0.350060509	1.075653637	0.155144823	3.51E-05	-0.164393841	-7.99E-04
7	VERDE	2.419255012	0.692250712	8.029500062	2.135167465	0.119275129	-17.72499862	-2.643873389
8	NARANJA	2.420101236	0.693395832	7.756166814	2.676476368	0.175409485	-20.84757885	-1.906609458
9	CELESTE	5.581658803	0.001965225	0.013671792	3.812096141	7.22E-04	-1.655521801	-1.333752734
10	ROSADO	5.580704301	0.002011173	0.002538737	5.366736583	0.009733563	-2.224292429	-0.368783391

Figura N° 28: Tabla de momentos de Hu diferenciados de los colores con una rotación a -20°

Fuente: Elaboración Propia

a 20°								
n°	Color	10°*M1	100°*M2	1000°*M3	10000°*M4	100000°*M5	1000000°*M6	10000000°*M7
1	MARRON	2.101122295	0.110178181	1.12E-03	0.001700949	-5.20E-09	-0.003381034	3.51E-07
2	AMARILLO	4.768232299	20.04599716	0.017667309	0.017205985	5.33E-07	-0.222765782	-1.89E-05
3	AMARILLO CLARO	5.352846616	0.248140191	0.009990751	0.432088176	1.53E-04	1.992701661	-0.001200528
4	VIOLETA	2.411426487	2.088139253	4.22E-04	6.02E-04	5.83E-10	5.25E-04	-9.29E-08
5	ROJO	1.785433474	0.349609116	1.09055265	0.280341604	2.24E-03	-1.373427842	3.72E-05
6	AZUL	1.784777194	0.34797279	1.001506396	0.178893264	9.48E-04	-0.743054804	-3.84E-04
7	VERDE	2.419453285	0.691199427	7.72158336	2.061633069	0.094592438	-16.92616725	0.052336898
8	NARANJA	2.41919449	0.690119992	8.209940773	3.113194594	0.151092398	-25.78678169	-0.302063522
9	CELESTE	5.582893143	0.001767487	0.008007896	2.645990439	-0.001824512	-1.112403915	-0.109619392
10	ROSADO	5.580534577	0.001882567	0.011189982	5.579072729	0.002054434	-2.345595596	-2.181047098

Figura N° 29: Tabla de momentos de Hu diferenciados de los colores con una rotación a 20°

Fuente: Elaboración Propia

3.4.1.2. Generación de Red Neuronal Multicapa Perceptrón

Adquiridos los datos que requerimos para procesar las muestras, el siguiente paso fue elaborar un motor de procesamiento, en este caso una red neuronal multicapa perceptrón (en adelante RNMP). Para la generación de la RNMP fue conveniente agrupar en matrices los momentos de Hu de acuerdo a color y grado de rotación, obteniéndose 10 matrices (por ser 10 colores) de 7 momentos de Hu por 10 colores. Una vez obtenidas las 10 matrices de entrada, procedimos a crear una variable con el nombre del color a trabajar, la cual contendría los valores de los momentos de Hu por cada grado de rotación, resultando finalmente una variable general la cual contiene cada variable de color. Es necesario precisar que fue necesario normalizar los valores que se encontraran en las matrices, debido a que los valores que se manejaron como salidas en la RNMP constaron de 0 hasta 1.

Por último, fue necesario determinar las salidas que poseerán las matrices, por lo que a cada color se le asignó un valor numérico entre 0.1 hasta 0.95. No se utilizaron los valores puntuales 0 y 1 porque se encuentran en la zona indeterminada de acción de la RNMP. A continuación en la tabla N°1, se podrá observar el rango de valores asociado a cada color a reconocer.

Tabla N° 1: Tabla de valores asociados y rangos permisibles de los colores detectados

Color Asociado	Valor del asociado	Rango de aceptación
Marrón	0.1	0.05 – 0.145
Amarillo	0.2	0.146 – 0.245
Amarillo Claro	0.3	0.246 – 0.345
Violeta	0.4	0.346 – 0.45
Rojo	0.5	0.446 – 0.545
Azul	0.6	0.546 – 0.645
Verde	0.7	0.646 – 0.745
Naranja	0.8	0.746 – 0.845
Celeste	0.9	0.846 – 0.94
Rosado	0.95	0.945 – 1.00

Fuente: Elaboración Propia

Luego se generó la red neuronal gracias a la función “newff”, en la cual podemos determinar la cantidad de capas que utilizaremos. Inicialmente se planteó utilizar 2 capas donde se mostró una eficiencia de $8.34E-0.5$. No obstante en son de mejoría se utilizó 4 capas donde se mostró una eficiencia $5.32E-0.2$. En la Figura N° 30 se aprecia la RNMP, una red de 7 entradas, y 3 salidas (del mismo valor).

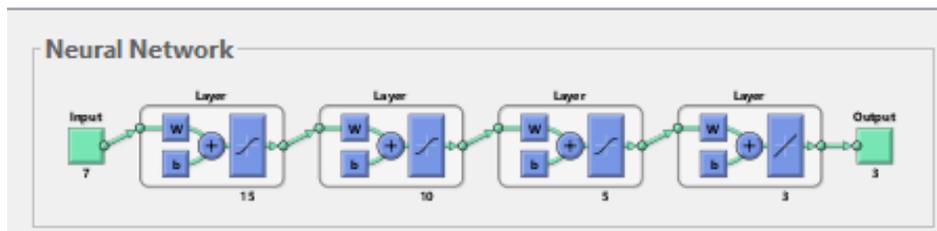


Figura N° 30: RNMP mostrada en Software MATLAB

Fuente: Elaboración Propia

3.4.1.3. Entrenamiento de RNMP

Una vez generada la RNMP, se continuó con su entrenamiento que se puede visualizar en las Figuras N°31 y N°32 con 4 y 2 capas respectivamente. Para ello se utilizó la función “train”, la cual permite asociar las entradas con las salidas previamente definidas. Seguidamente, se determinó la cantidad de repercusiones o “epochs” que manejaría la red neuronal para su entrenamiento. Para el caso puntual, se determinó un número alto, en este caso 100000. Cabe precisar que este valor fue determinado gracias a diferentes pruebas empíricas de respuesta de la RNMP.

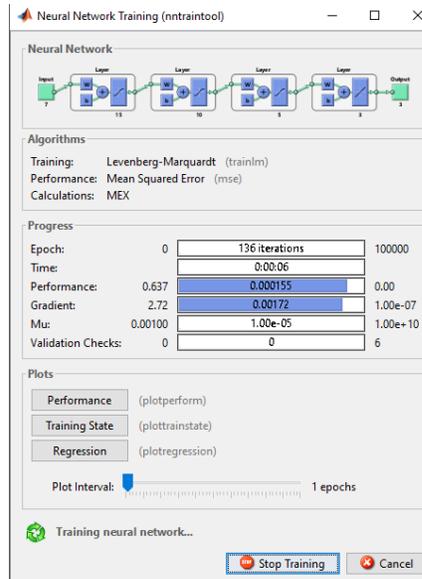


Figura N° 31: Interfaz de entrenamiento de RNMP con 4 capas

Fuente: Elaboración Propia

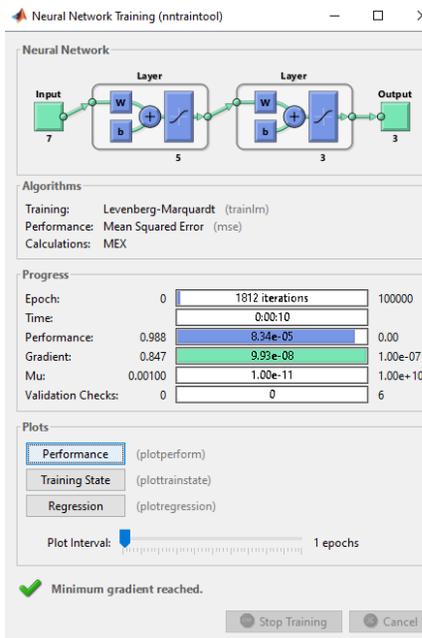


Figura N° 32: Interfaz de entrenamiento de RNMP con 2 capas

Fuente: Elaboración Propia

Como resultado final al entrenamiento, obtuvimos una RNMP o base de datos lista para responder ante el ingreso de valores reales obtenidos de muestras reales.

3.4.1. Aplicación de los momentos de Hu sobre la muestra capturada post filtrado.

Considerando las muestras reales obtenidas de los puntos anteriores, fue necesario el haber obtenido los momentos de HU de la imagen capturada a fin de ingresarla a la red neuronal y posteriormente procesarla, ejecutando el mismo proceso que se dio para las imágenes ideales.

3.4.2. Normalización de los momentos de Hu

Cabe precisar que la normalización de los momentos de HU fue imprescindible debido a que a nivel de entrenamiento de RNMP se normalizó los momentos de HU ideales.

3.5. Generación de GUIDE y conexión con el algoritmo.

Se ejecutó la instrucción >>guide en la ventana de comandos, se escogió la opción Blank GUI(Default) el cual nos presenta la interfaz gráfica de usuario en blanco como en la Figura N° 33 para crear y modificar el diseño de nuestro programa.

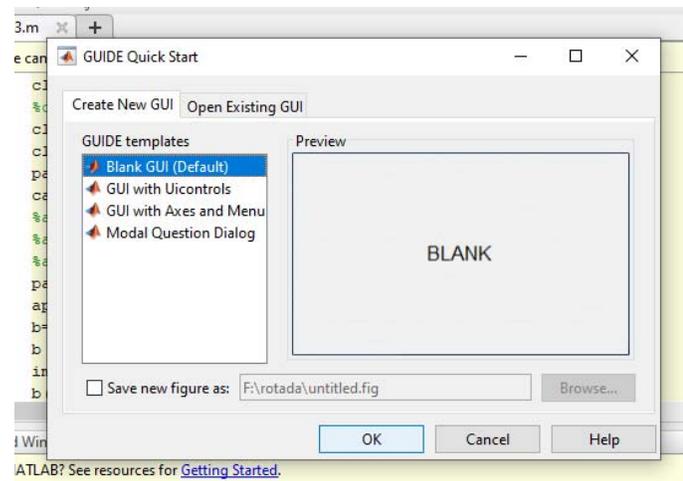


Figura N° 33: Interfaz de creación de GUIDE

Fuente: Elaboración Propia

Se obtuvo el área de diseño representado en la Figura N° 34 para realizar las modificaciones de acuerdo al diseño planteado a través de las herramientas ubicadas en el lado izquierdo de la ventana

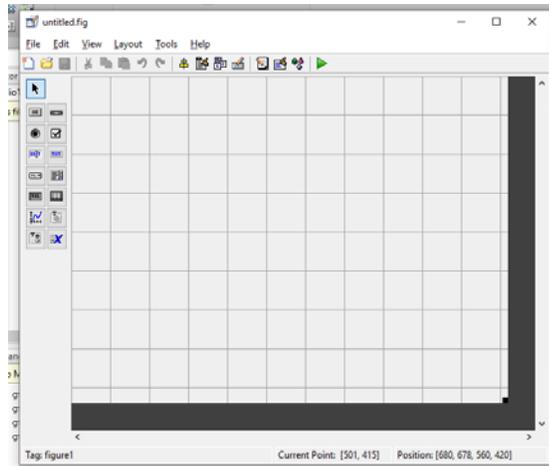


Figura N° 34: Área de diseño de GUIDE

Fuente: Elaboración Propia

Dentro de los componentes utilizados para realizar la GUIDE estuvo la herramienta “axis”. Como se observa en la Figura N°35, esta se empleó para colocar la imagen de fondo de la interfaz, el logo de la universidad en el extremo superior derecho y para ubicar las imágenes que fueron capturadas por el dispositivo Smartphone. La opción Property Inspector nos permitió modificar la dirección asociada editando el campo “Tag” para usarla en la programación.

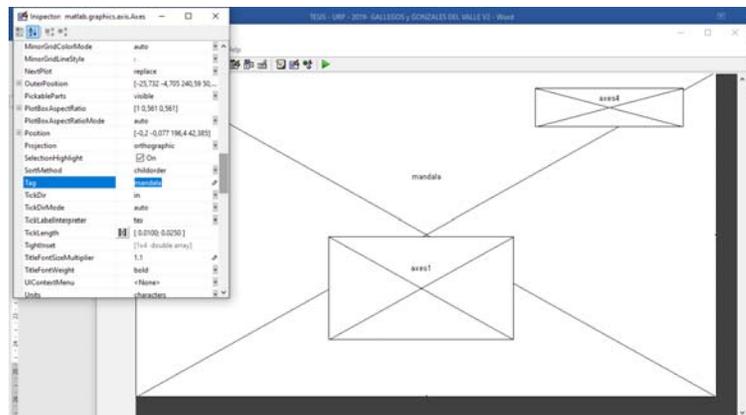


Figura N° 35: Área de diseño de GUIDE mostrando el uso de herramienta axis

Fuente: Elaboración Propia

Se hizo uso de la herramienta “Static Text” para colocar el título de la interfaz gráfica identificado con el siguiente texto “MÓDULO RECONOCEDOR colorADD”; el lugar donde aparecerá el color asociado a la imagen fotografiada identificado con la palabra “COLOR”; y nuestros nombres identificado con el texto “Bach. Hilary Gallegos Bach Angello Gonzales del Valle”. Estas modificaciones

se lograron con la opción Property Inspector, en el que se editó el campo “String” con la información detallada al inicio resultando como la Figura N°36

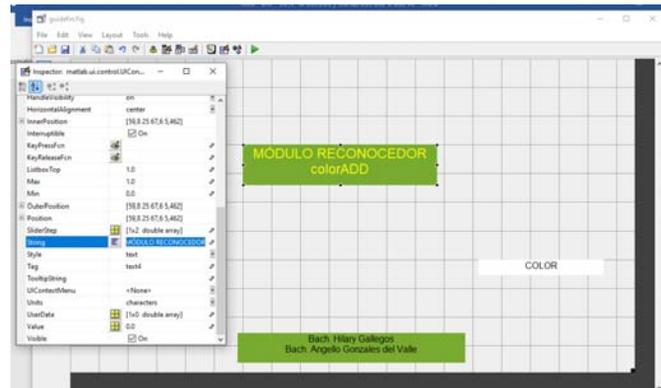


Figura N° 36: Área de diseño de GUIDE mostrando el uso de herramienta static text

Fuente: Elaboración Propia

Adicionalmente, se empleó la herramienta “Pushbutton” como se visualiza en la Figura N°37. Este botón ejecutará el programa cuando se presione sobre este, dando inicio al reconocimiento de los símbolos colorADD obtenidas por el dispositivo Smartphone y procesamiento del símbolo para reconocimiento del color. De igual forma, modificando el campo “String”, ingresando a la opción Property Inspector, se identificó el botón con el nombre de “INICIO”.

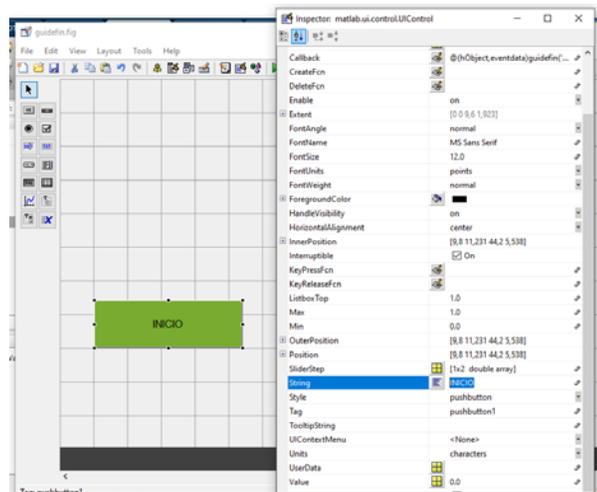


Figura N° 37: Área de diseño de GUIDE mostrando el uso de herramienta pushbutton

Fuente: Elaboración Propia

Finalmente, el diseño de la GUIDE se obtuvo como se muestra a continuación en la Figura N° 38.

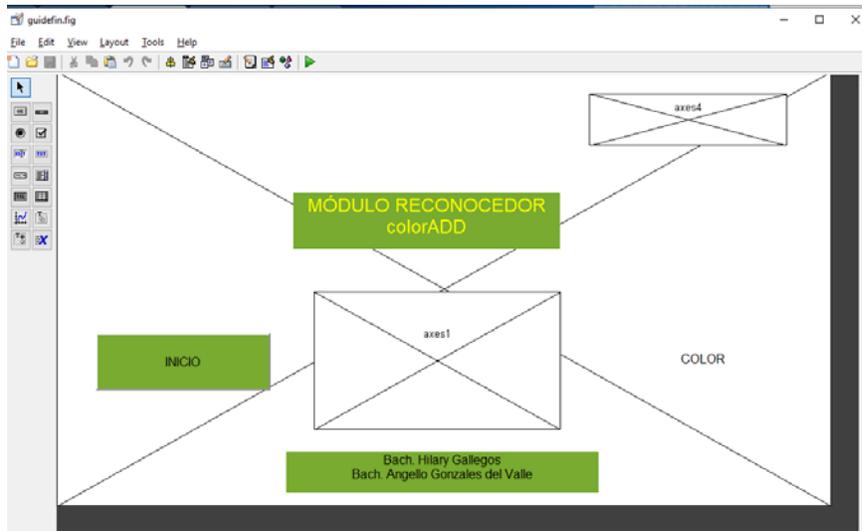


Figura N° 38: Diseño de GUIDE

Fuente: Elaboración Propia

Se guardó la GUIDE con el nombre de “guidefin” en la carpeta de nuestro programa, creando el m-file asociado a nuestro diseño con los archivos: guidefin.fig y guidefin.m.

En el archivo guidefin.m se editó el programa que se carga por default y ubica las funciones por secciones de acuerdo a nuestro diseño.

Se trabajó el código de la Figura N° 39 para insertar la imagen de fondo y el logo de la universidad en la sección “guidefin_Opening”

```

61 | %Colocar imagen de fondo
62 | mandala = imread('mandala.jpg'); %Leer imagen
63 | axes(handles.mandala); %Carga la imagen en mandala
64 | axis off;
65 | imshow(mandala); %Presenta la imagen
66 | handles.output = hObject;
67 | guidata(hObject, handles);
68 | %Colocar logo URP
69 | axes(handles.axes4)
70 | urp=imread('urp.jpg');
71 | axis off;
72 | imshow(urp);

```

Figura N° 39: Código para imágenes de la GUIDE

Fuente: Elaboración Propia

En la sección “pushbutton1_Callback”, para el inicio de la ejecución del módulo de reconocimiento, se colocó la programación que se detalla en el capítulo “3.2.1

Conexión con el software Matlab”. De igual forma, para la presentación de los resultados de reconocimiento de imágenes tanto en forma textual como auditiva se trabajó en la misma sección adicionando la programación de la Figura N° 40, se toma como ejemplo el color rojo:

```
228 %Resultado en forma textual
229 set(handles.text5,'String','ROJO')
230 %Resultado en forma auditiva
231 audiorojo='colorrojo.mp4';
232 in_play=audiorojo;
233 [y,Fs]=audioread(in_play);
234 handles.player=audioplayer(y,Fs);
235 play(handles.player)
236 guidata(hObject, handles);
```

Figura N° 40: Código para resultados de la GUIDE

Fuente: Elaboración Propia

Al ejecutar la GUIDE resultó como se observa en la Figura N° 41:



Figura N° 41: GUIDE

Fuente: Elaboración Propia

CAPÍTULO IV: PRUEBAS Y RESULTADOS

En este capítulo se muestra las pruebas y resultados del módulo reconocedor de símbolos para los colores: amarillo, azul y amarillo claro con y sin rotación simulada.

4.1. Análisis del resultado color amarillo

A continuación en la Figura N°42 se muestra la prueba del GUIDE con los algoritmos, teniendo la imagen real capturada colocada sin rotación.



Figura N° 42: Prueba con color amarillo sin rotación

Fuente: Elaboración Propia

En seguida en las Figuras N° 43 y N° 44 se muestra la prueba del GUIDE con los algoritmos y con rotación simulada.

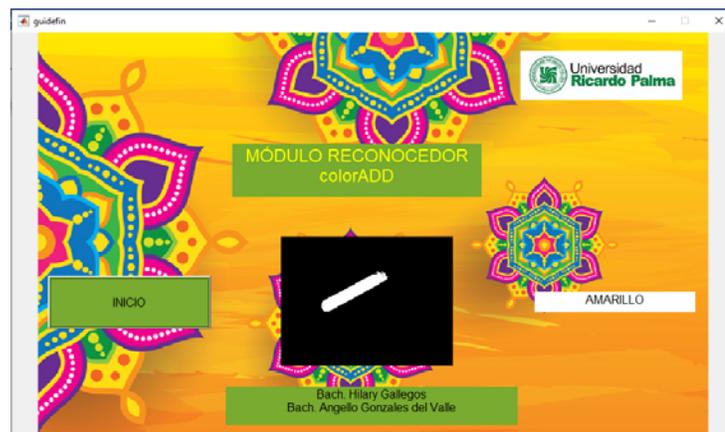


Figura N° 43: Prueba con color amarillo con rotación

Fuente: Elaboración Propia



Figura N° 44: Prueba con color amarillo con rotación

Fuente: Elaboración Propia

Se puede apreciar en las Figuras N° 43 y N° 44 que para el color amarillo el módulo reconocedor funciona adecuadamente sea rotado o no el símbolo.

4.2. Análisis del resultado color azul

a) Sin rotación:



Figura N° 45: Prueba con color azul sin rotación

Fuente: Elaboración Propia

b) Con rotación:



Figura N° 46: Prueba con color azul con rotación

Fuente: Elaboración Propia

Evaluando las Figuras N° 45 y N° 46, el módulo reconocedor funciona adecuadamente cuando la imagen no es rotada; mas cuando la imagen se rota el módulo no reconoce el símbolo como azul sino como rojo

4.3. Análisis del resultado color amarillo claro

a) Sin rotación:

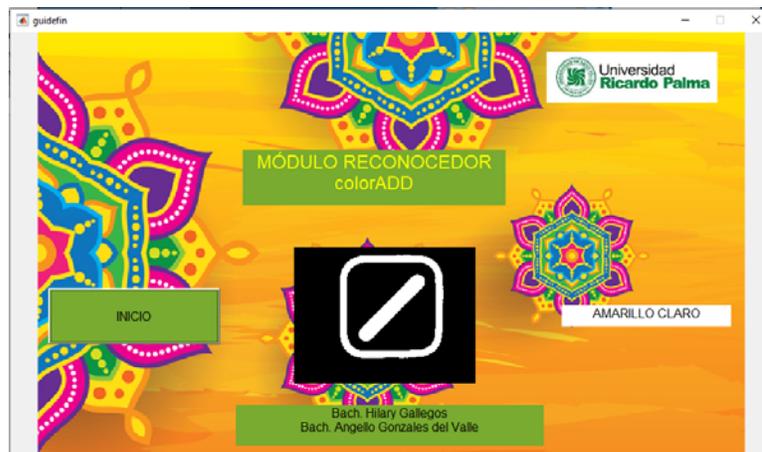


Figura N° 47: Prueba con color amarillo claro con rotación

Fuente: Elaboración Propia

b) Con rotación:



Figura N° 48: Prueba con color amarillo claro con rotación

Fuente: Elaboración Propia

Las Figuras N° 47 y N° 48 nos muestran que para el color amarillo claro el módulo reconocedor funciona adecuadamente sea rotado o no el símbolo.

4.4. Resultados generales

Tabla N° 2: Tabla de Detección de imágenes Ideales

Color	¿Reconocido a 0°?	¿Reconocido a 5°?	¿Reconocido a -5°?
Marrón	Reconocido	Reconocido	Reconocido
Amarillo	Reconocido	Reconocido	Reconocido
Amarillo Claro	Reconocido	Reconocido	Reconocido
Violeta	Reconocido	Reconocido	Reconocido
Rojo	Reconocido	Reconocido	Reconocido
Azul	Reconocido	Reconocido	Reconocido
Naranja	Reconocido	Reconocido	Reconocido
Verde	Reconocido	Reconocido	Reconocido
Celeste	Reconocido	Reconocido	Reconocido
Rosa	Reconocido	Reconocido	Reconocido

Fuente: Elaboración Propia

Porcentaje de éxito: 100%

En la tabla N°2 se realizaron las pruebas virtuales de los símbolos reconocidos, obteniéndose Porcentaje de éxito: 100%. Cabe rescatar que al ser las mismas imágenes con las cuales se entrenó, los resultados obtenidos fueron perfectos.

Tabla N° 3: Tabla de Detección de imágenes Reales

Color	¿Reconocido a 0°?	¿Reconocido a 5°?	¿Reconocido a -5°?
Marrón	Reconocido	Reconocido	Reconocido
Amarillo	Reconocido	Reconocido	Reconocido
Amarillo Claro	Reconocido	No Reconocido	Reconocido
Violeta	Reconocido	Reconocido	Reconocido
Rojo	Reconocido	Reconocido	Reconocido
Azul	No Reconocido	Reconocido	No Reconocido
Naranja	No Reconocido	Reconocido	No Reconocido
Verde	Reconocido	No Reconocido	Reconocido
Celeste	No Reconocido	Reconocido	Reconocido
Rosa	Reconocido	No Reconocido	No Reconocido

Porcentaje de éxito: 70%

En la tabla N°3 se realizaron las pruebas reales impresas de los símbolos reconocidos, obteniéndose un Porcentaje de éxito: 70%. Se indica a su vez, que estos resultados dependieron de ciertas condiciones lumínicas y de enfoque de cámara.

Tabla N° 4: Tabla de Funcionamiento de Guide

Sub Prueba	Rspta.
Tiempo de Conexión de cámara IP mínima sin Falla	2 <u>seg.</u>
Tiempo de Captura y enfoque de muestra mínima sin Falla	3 <u>seg.</u>
Tiempo de procesamiento de imagen	3 <u>seg.</u>
¿Tiempo de respuesta entre captura y resultado cortos?	SI
¿Texto de respuesta de GUIDE se mostró exitosamente?	SI
¿Audio de respuesta de GUIDE se reprodujo exitosamente?	SI

En la tabla N°4 se observan los resultados de pruebas en específico para medir la efectividad del simulador de reconocimiento de símbolos.

CONCLUSIONES

Las conclusiones correspondientes al desarrollo de esta investigación, son:

- 1) Se logró el reconocimiento de un 70% de los 10 símbolos escogidos para el proyecto de tesis, generando un módulo mediante la GUI, la cual se comportó de forma muy intuitiva como interfaz para identificar los símbolos.
- 2) Se logró establecer una conexión inalámbrica entre PC y cámara IP para simular el módulo reconocedor, no obstante, se añade que pueden haber fallas de conexión si es que no se denota un tiempo prudente de reconocimiento de dispositivo al momento de configurar el enlace, tal como se indica en el punto 3.2.1.
- 3) La realización de la GUIDE fue exitosa, no obstante, su funcionamiento fue simple en el sentido de que se utilizó un solo botón interactivo para ejecutar todo el algoritmo de reconocimiento.
- 4) El haber implementado algorítmicamente la técnica de procesamiento de imágenes (Momentos de Hu) denotó significativamente el porcentaje de acierto del reconocimiento de 10 símbolos del código colorADD.

RECOMENDACIONES

- 1) Se recomienda que el tiempo de enlace entre cámara IP y Pc sea mayor a 2seg.
- 2) Se recomienda que al GUIDE diseñado se profundice a nivel de funcionamiento disgregado.
- 3) Para lograr un mayor porcentaje de reconocimiento, se recomienda utilizar otras técnicas de procesamiento a fin de reducir los fallos de algoritmo

REFERENCIAS BIBLIOGRÁFICAS

- Aguirre, N. (2002) Procesamiento de imágenes. (52-55).
- Andreu F. y Pellejero I. (2006) *Redes WLAN: Fundamentos y aplicaciones de seguridad*. [Libro en línea] [Consultado 22, agosto, 2019]. Disponible en: https://books.google.com.pe/books?id=k3JuVG2D9IMC&pg=PA108&dq=redes+lan&hl=es-419&sa=X&ved=0ahUKEwjH_tSuh_3IAhXHmVkkHb8MAaMQ6AEIKDAA#v=onepage&q&f=false
- Al-Shatnawi, M (2012). Un nuevo método en esteganografía de imagen con mejora de calidad de imagen. *Appl. Math, SCI*, vol 6, (3907-3915).
- Buchler, J. (2012) *Philosophical Writings of Peirce*. [Libro en línea] [Consultado 18, julio, 2019]. Disponible en: <http://theory.theasitheas.org/wp-content/uploads/2013/02/Peirce-C-S-Logic-Semiotic.pdf>
- Casas Martínez, A (2017). Reconocimiento de Imágenes con Redes Convolucionales (Tesis de Pregrado) Universidad de Sevilla. España.
- De la Fuente, M. y Calonge, T. (1999) Aplicaciones de las redes de neuronas en supervisión, diagnóstico y control de procesos. [Libro en línea] [Consultado 11, agosto, 2019]. Disponible en: <https://books.google.com.pe/books?id=jUHGRXd5xU8C&pg=PA99&lpg=PA99&dq=de+la+fuentes+y+calonge+tecnicas+de+reconocimiento&source=bl&ots=aCWWsQMFvY&sig=ACfU3U0mvcKSbBiBTCkzq6G0t8VWoPBW3g&hl=es&sa=X&ved=2ahUKEwiI05iGqPnkAhXtzVkkHcsuBJQQ6AEwAHoECAkQAQ#v=onepage&q=de%20la%20fuente%20y%20calonge%20tecnicas%20de%20reconocimiento&f=false>
- Domínguez Torres, A. (1996). Procesamiento digital de imágenes. *Perfiles Educativos*, (72).
- Faiz, A; Muhamad A et Al (2018) Reconocimiento óptico de caracteres para la coincidencia de similitud de imágenes coránicas.
- Flusser, J; Suk, T et Al (2009) *Moments and Moment Invariants in Pattern Recognition*. [Libro en línea] [Consultado 15, julio, 2019]. Disponible en: <https://books.google.com.pe/books?id=IAFDvarvAAC&printsec=frontcover&dq=Moments+and+Moment+Invariants+in+Pattern+Recognition&hl=es&sa=X&ved=0ahUKEwisqpyL6PjkAhUh0FkKHZ74CicQ6AEIKDAA#v=onepage&q=Moments%20and%20Moment%20Invariants%20in%20Pattern%20Recognition&f=false>
- Flórez R. y Fernández J. (2008) *Las Redes Neuronales Artificiales*. [Libro en línea] [Consultado 22, agosto, 2019]. Disponible en: https://books.google.com.pe/books?id=X0uLwi1Ap4QC&pg=PA60&dq=perceptron+multicapa&hl=es&sa=X&ved=0ahUKEwix9JKrn_nkAhWlslk

[KHXYAFsQ6AEIKDAA#v=onepage&q=perceptron%20multicapa&f=false](https://www.researchgate.net/publication/327703478_Capitulo_1_Generali-dades_de_las_redes_neuronales_artificiales/link/5b9fe3c0299bf13e6038a1d8/download)

- Gomez, D; Guerrero, A (2016). Estudio y Análisis de técnicas para procesamiento digital de imágenes. Universidad Tecnológica de Pereira. Colombia.
- González, R. y Woods, R. (2002) *Digital Image Processing*. [Libro en línea] [Consultado 10, junio, 2019]. Disponible en: https://www.academia.edu/8240316/Digital_Image_Processing_Gonzalez_and_Woods_3rd_Ed._Pearson
- Langella, R; Testa, A; Caicedo, J et al (2018). Sobre el uso de descriptores de Fourier para la evaluación de matrices de acoplamiento de frecuencia de dispositivos electrónicos de potencia, 2018 18ava Conferencia internacional en Armónicas y calidad de la potencia, Ljubljana, 2018, pp. 1-6.
- Lázaro, N. (27 de junio del 2018). Tres símbolos para que 350 millones de daltónicos puedan identificar colores. El Confidencial. Recuperado de https://brands.elconfidencial.com/sociedad/2018-06-27/daltonicos-simbolos-colores-abecedario-bra_1584172/
- Lema, L. (2014). Sistema de reconocimiento de imagen para detección de placas donde vehículo obtenidas con cámara web (Tesis de Pregrado) Universidad Tecnológica Israel. Ecuador .
- López, M. (27 de julio del 2015). El código “braille” para daltónicos se inventa en el siglo XXI. El País. Recuperado de https://elpais.com/elpais/2015/07/24/planeta_futuro/1437744306_549219.html
- Mejía, Jose (2004). Apuntes de procesamiento digital de imágenes. Facultad de Ingeniería de la UASLP. Bolivia. (21-23).
- Montenegro, R. (2004). *Biología Evolutiva*. [Libro en línea] [Consultado: 21, junio, 2019]. Disponible en: <https://books.google.com.pe/books?id=WA56vndSXNIC&printsec=frontcover&hl=es#v=onepage&q&f=false>
- Neiva, M. (2010). *colorADD-Color Identification System*. Recuperado en enero 21, 2019. Disponible en <http://www.coloradd.net/publications.asp>
- Palacios, R. (2018). Reconocimiento de imágenes mediante Raspberry PI(Tesis de Maestría) Universidad Abierta de Cataluña. España.
- Rivas, W. y Mazón B. (2018) *Redes Neuronales artificiales aplicadas al reconocimiento de patrones*. [Libro en línea] [Consultado: 03, agosto, 2019]. Disponible en: https://www.researchgate.net/publication/327703478_Capitulo_1_Generali-dades_de_las_redes_neuronales_artificiales/link/5b9fe3c0299bf13e6038a1d8/download

- Sahare, P y Dhok, S. (2018). Segmentación de caracteres multi lingüísticos y esquemas de reconocimiento para imágenes de documentos de la India, (1).
- Sobrado. E. (2003). Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. (Tesis de Posgrado) Pontificia Universidad Católica del Perú.

ANEXOS

Programación de la GUIDE en Matlab

```
function varargout = guidefin(varargin)
% GUIDEFIN MATLAB code for guidefin.fig
%     GUIDEFIN, by itself, creates a new GUIDEFIN or raises the
existing
%     singleton*.
%
%     H = GUIDEFIN returns the handle to a new GUIDEFIN or the handle
to
%     the existing singleton*.
%
%     GUIDEFIN('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in GUIDEFIN.M with the given input
arguments.
%
%     GUIDEFIN('Property','Value',...) creates a new GUIDEFIN or
raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before guidefin_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to guidefin_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help guidefin

% Last Modified by GUIDE v2.5 28-Sep-2019 04:19:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @guidefin_OpeningFcn, ...
                  'gui_OutputFcn',  @guidefin_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before guidefin is made visible.
function guidefin_OpeningFcn(hObject, eventdata, handles, varargin)
```

```

% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to guidefin (see VARARGIN)

% Choose default command line output for guidefin
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

%Colocar imagen de fondo
mandala = imread('mandala.jpg'); %Leer imagen
axes(handles.mandala); %Carga la imagen en mandala
axis off;
imshow(mandala); %Presenta la imagen
handles.output = hObject;
guidata(hObject, handles);
%Colocar logo URP
axes(handles.axes4)
urp=imread('urp.jpg');
axis off;
imshow(urp);

axes(handles.axes1),image(ones(4000,6000))

% UIWAIT makes guidefin wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = guidefin_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

pause(2)
cam=ipcam('http://172.20.60.230:4747/video');
pause(2)
ap=snapshot(cam);
b= 255-double(ap(:,:,1)); b = uint8(b);% imshow(b)
b = double(b)>=180; %b = uint8(255*b);
imshow(b)
b(1:20 , : ) = zeros(20,640);
b = medfilt2(b);
imshow(b);
axes(handles.axes1), imshow( b )

```

```

load('tesisv4.mat')
%Calculando los momentos de Hu

%Primer momento de Hu
m00 = sum(sum(b));
%Segundo momento de Hu
[f,c] = size(b);
m10 = 0; m20 = 0; m22= 0; m33 = 0 ;
m01 = 0; m02 = 0; m11 = 0; m30 = 0 ; m03 = 0;
for i=1:f
    for j=1:c
        m10 = i*b(i,j) + m10;
        m20 = i*i*b(i,j) + m20;
        m01 = j*b(i,j) + m01;
        m02 = j*j*b(i,j) + m02;
        m11 = i*j*b(i,j) + m11;
        m22 = i*i*j*j*b(i,j) + m22;
        m33 = i*i*i*j*j*j*b(i,j) + m33;
        m30 = i*i*i*b(i,j) + m30;
        m03 = j*j*j*b(i,j) + m03;
    end
end
% m10 , m01 , m20 , m02 , m11, m22 , m33 ,m30, m03

xp = round( m10/m00 );
yp = round( m01/m00 );
%colormap(gray(256)), image(b*255)

%Momentos centrales
clear u
p = 1; q = 1;
cc = 0;
for p = 0:3
    for q = 0:3
        for x=1:f
            for y=1:c
                cc = ((x-xp)^(p))*((y-yp)^(q))*b(x,y) + cc ;
            end
        end
        u(p+1,q+1) = cc;
    end
    cc = 0;
end
u;
%
% %Momentos centrales normalizados
u00 = u(1,1);
for p=0:3
    for q=0:3
        L = 0.5*(p+q)+1;
        npq( p+1 , q+1 ) = u(p+1,q+1) / ( u00^L );
    end
end
npq;

FI1 = npq(3,1) + npq(1,3);
FI2 = (npq(3,1)-npq(1,3))^2 + 4*npq(2,2)^2;
FI3 = (npq(4,1)-3*npq(2,3))^2 + (3*npq(3,2) - npq(1,4) )^2;
FI4 = (npq(4,1)+npq(2,3))^2 + (npq(3,2) + npq(1,4) )^2;
FI5 = (npq(4,1)-npq(2,3))*(npq(4,1)+npq(2,3))*((npq(4,1)+npq(2,3)))^2
- 3*(npq(3,2)-npq(1,4))^2 + (3*npq(3,2)-

```

```

npq(1,4))*(npq(3,2)+npq(1,4))*((3*npq(4,1)+(npq(2,3)))^2 - (npq(3,2)-
npq(1,4))^2);
FI6 = (npq(3,1)-npq(1,3))*((npq(4,1)+npq(2,3))^2 -
(npq(3,2)+npq(1,4))^2)+4*npq(2,2)*(npq(4,1)+npq(2,3))*(npq(3,2)+npq(1,
4));
FI7 = (3*npq(3,2)-
npq(4,1))*(npq(4,1)+npq(2,3))*((npq(4,1)+npq(2,3))^2-
3*(npq(3,2)+npq(1,4))^2) + (3*npq(3,2)-
npq(4,1))*(npq(3,2)+npq(1,4))*(3*(npq(4,1)+npq(2,3))^2-
(npq(3,2)+npq(1,4))^2);

%MO = [ FI1 FI2 FI3 FI4 FI5 FI6 FI7 ]
%if FI1>0.1783 && FI1<0.2100
    %disp('ESTE COLOR ES ROJO')
%else
    %disp('ESTE COLOR ES AZUL')
%end

a=FI1*10;
b=FI2*100;
c=FI3*1000;
d=FI4*10000;
e=FI5*100000;
f=FI6*1000000;
g=FI7*10000000;
mh=[a b c d e f g];
mhnt=mh'/37.425157319999997;
%%
Y = sim( net, mhnt );

if(abs(Y)>=0.05 & abs(Y)< 0.145)
    marron=1
    set(handles.text5,'String','MARRÓN')
    audiomarron='colormarron.mp4';
    in_play=audiomarron;
    [y,Fs]=audioread(in_play);
    handles.player=audioplayer(y,Fs);
    play(handles.player)
    guidata(hObject, handles);
elseif(abs(Y)>=0.145 & abs(Y)< 0.245)
    amarillo=1
    set(handles.text5,'String','AMARILLO')
    audioamarillo='coloramarillo.mp4';
    in_play=audioamarillo;
    [y,Fs]=audioread(in_play);
    handles.player=audioplayer(y,Fs);
    play(handles.player)
    guidata(hObject, handles);
elseif(abs(Y)>=0.245 & abs(Y)< 0.345)
    amarillocla=1
    set(handles.text5,'String','AMARILLO CLARO')
    audioamarillocla='coloramarillocla.mp4';
    in_play=audioamarillocla;
    [y,Fs]=audioread(in_play);
    handles.player=audioplayer(y,Fs);
    play(handles.player)
    guidata(hObject, handles);
elseif(abs(Y)>=0.345 & abs(Y)< 0.445)
    violeta=1
    set(handles.text5,'String','VIOLETA')
    audiovioleta='colorvioleta.mp4';

```

```

in_play=audiovioleta;
[y,Fs]=audioread(in_play);
handles.player=audioplayer(y,Fs);
play(handles.player)
guidata(hObject, handles);
elseif(abs(Y)>=0.445 & abs(Y)< 0.545)
rojo=1
%Resultado en forma textual
set(handles.text5,'String','ROJO')
%Resultado en forma auditiva
audiorojo='colorrojo.mp4';
in_play=audiorojo;
[y,Fs]=audioread(in_play);
handles.player=audioplayer(y,Fs);
play(handles.player)
guidata(hObject, handles);
elseif(abs(Y)>=0.545 & abs(Y)< 0.645)
azul=1
set(handles.text5,'String','AZUL')
audioazul='colorazul.mp4';
in_play=audioazul;
[y,Fs]=audioread(in_play);
handles.player=audioplayer(y,Fs);
play(handles.player)
guidata(hObject, handles);
elseif(abs(Y)>=0.645 & abs(Y)< 0.745)
verde=1
set(handles.text5,'String','VERDE')
audioverde='colorverde.mp4';
in_play=audioverde;
[y,Fs]=audioread(in_play);
handles.player=audioplayer(y,Fs);
play(handles.player)
guidata(hObject, handles);
elseif(abs(Y)>=0.745 & abs(Y)< 0.845)
naranja=1
set(handles.text5,'String','NARANJA')
audionaranja='colornaranjao.mp4';
in_play=audionaranja;
[y,Fs]=audioread(in_play);
handles.player=audioplayer(y,Fs);
play(handles.player)
guidata(hObject, handles);
elseif(abs(Y)>=0.845 & abs(Y)<= 0.94)
celeste=1
set(handles.text5,'String','CELESTE')
audiocelste='colorcelste.mp4';
in_play=audiocelste;
[y,Fs]=audioread(in_play);
handles.player=audioplayer(y,Fs);
play(handles.player)
guidata(hObject, handles);
elseif(abs(Y)>0.945 & abs(Y)< 1)
rosa=1
set(handles.text5,'String','ROSADO')
audiorosa='colorrosa.mp4';
in_play=audiorosa;
[y,Fs]=audioread(in_play);
handles.player=audioplayer(y,Fs);
play(handles.player)
guidata(hObject, handles);

```

```
elseif(abs(Y)>0.95 & abs(Y)< 0.05)
    set(handles.text5,'String','Intente nuevamente')
end
```

```
%%
```