



UNA UNIVERSIDAD CON ALMA DE GUERRERO

ANÁLISIS COMPARATIVO DE ALGORITMOS DE RECONOCIMIENTO DE IMÁGENES POR DESCRIPTORES DE COLOR PARA LA IDENTIFICACIÓN DE BILLETES

Tesis para optar por el Título de Ingeniero de Sistemas, que presenta el bachiller.

AUTOR

Jorge Luis Estela Zumaeta

ASESOR

Ing. Rosa América Cobeñas Sánchez

CHICLAYO - PERÚ 2016



Título de la tesis
**Análisis Comparativo De Algoritmos De Reconocimiento De Imágenes Por
Descriptores De Color Para La Identificación De Billetes**

Dr. Gutiérrez Gutiérrez, Jorge
Presidente del jurado de tesis

Ing. Vásquez Leyva, Oliver
Secretario del jurado de tesis

Ing. Cobeñas Sánchez, Rosa América
Vocal del jurado de tesis

Bach. Estela Zumaeta, Jorge Luis

DEDICATORIA

Primero agradecer a Dios por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mis padres, porque creyeron en mí, dándome ejemplos dignos de superación y entrega.

AGRADECIMIENTO

Al finalizar un trabajo tan arduo y lleno de dificultades como el desarrollo de un proyecto es inevitable que te asalte un muy humano egocentrismo que te lleva a concentrar la mayor parte del mérito en el aporte que has hecho. Sin embargo, el análisis objetivo te muestra inmediatamente que la magnitud de ese aporte hubiese sido imposible sin la participación de personas e instituciones que han facilitado las cosas para que este trabajo llegue a un feliz término. Por ello, es para mí un verdadero placer utilizar este espacio para ser justo y consecuente con ellas, expresándoles mis agradecimientos.

Debo agradecer de manera especial a la Mg. Ana María Guerrero Millones y a la Ing. Rosa América Cobeñas Sánchez por el asesoramiento respectivo para poder realizar esta tesis bajo sus asesorías. Su apoyo y confianza en este trabajo y su capacidad para guiar las ideas ha sido un aporte invaluable, no solamente en el desarrollo de esta tesis, sino también en mi formación como investigador. Las ideas propias, siempre enmarcadas en su orientación y rigurosidad, han sido la clave del buen trabajo que hemos realizado juntos, el cual no se puede concebir sin su siempre oportuna participación. Le agradezco también el haberme facilitado siempre los medios suficientes para llevar a cabo todas las actividades propuestas durante el desarrollo de esta tesis.

ÍNDICE

CAPÍTULO I:PROBLEMA DE INVESTIGACIÓN..... 14

 1.1.Situación Problemática..... 15

 1.2.Formulación del problema..... 18

 1.3.Justificación e importancia 18

 1.4.Objetivos 19

 1.5. Limitaciones del proyecto.....20

CAPÍTULO II:MARCOTEÓRICO21

 2.1.Antecedentes de Estudios22

 2.2.Estado del arte 24

 2.3. Base Teórico - Científicas26

CAPÍTULO III:MARCO METODOLÓGICO 48

 3.1.Tipo y diseño de la investigación..... 49

 3.2.Población y muestra 49

 3.3.Hipótesis 49

 3.4.Variables 50

 3.5.Operacionalización 51

 3.6.Métodos, técnicas e instrumentos de recolección de datos 52

 3.7.Procedimiento para la recolección de datos 53

 3.8.Análisis Estadísticos e Interpretación de los datos 53

 3.9.Principios éticos 54

 3.10.Criterios de rigor científico..... 54

CAPÍTULO IV:ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS..... 56

 4.1. Resultados en tablas y gráficos..... 67



4.1. Discusión de resultados	67
CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN	68
5.1. Adquisición de Imágenes.....	69
5.2. Pre-procesamiento	71
5.3. Extracción de Caracteres por Color	71
5.4. Clasificación	71
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES.....	78
6.1. Conclusiones.....	79
6.2. Recomendaciones.....	80
REFERENCIAS.....	84
ANEXOS	84

ÍNDICE DE FIGURAS

Figura 1: Billeto de S/. 10 Primera familiar	26
Figura 2: Billeto de S/. 20 Primera familiar	26
Figura 3: Billeto de S/. 50 Primera familiar	27
Figura 4: Billeto de S/. 100 Primera familiar	27
Figura 5: Billeto de S/. 200 Primera familiar	28
Figura 6: Billeto de S/.10 Nueva Familia	28
Figura 7: Billeto de S/.20 Nueva Familia	29
Figura 8: Billeto de S/.50 Nueva Familia	29
Figura 9: Billeto de S/.100 Nueva Familia	30
Figura 10: Billeto de S/.200 Nueva Familia	30
Figura 11: Cálculo Imagen Digital.....	31
Figura 12: Etapas del Procesamiento de Imágenes	32
Figura 13: Categoría de las características.....	34
Figura 14: Proceso de Filtrado	35
Figura 15: Filtro pasa-bajos: (a) en frecuencia, (b) en el dominio espacial	36
Figura 16: Máscara para filtro gaussiano de 3x3	36
Figura 17: Filtro pasa-alto: (a) en frecuencia, (b) en el dominio espacial.....	37
Figura 18: Máscara de 3x3 para un filtro pasa-alto simple.....	38
Figura 19: Máscara de 3x3 para un filtro pasa-alto con énfasis en las altas frecuencias.....	38
Figura 20: Modelo de color blanco y negro	40
Figura 21: Modelo de Color HSV	40
Figura 22: Modelo RGB	42
Figura 23: Imagen representada en el modelo YCbCr	43
Figura 24: Estructura de una red neuronal multicapa, con una capa de entrada, una capa de oculta y una capa de salida.....	45
Figura 25: Método Backpropagation	47
Figura 26: Características de billete de S/10.00.....	57
Figura 27: Características de billete de S/20.00.....	58
Figura 28: Características de billete de S/50.00.....	58
Figura 29: Características de billete de S/100.00	59
Figura 30: Características de billete de S/200.00	59
Figura 31: Billeto con filtro blanco y negro.....	61
Figura 32: Billeto con filtro HSV	62
Figura 33: Billeto con filtro rojo	62
Figura 34: Billeto con filtro verde.....	63
Figura 35: Billeto con filtro azul	63
Figura 36: Diagrama del entrenamiento.....	69
Figura 37: Máquina con Luz Ultravioleta Instrumento empleado para la captura de las imágenes digitales.....	71
Figura 38: Imagen con método por color Blanco y negro.....	73
Figura 39: Imagen con método por color HSV.....	73
Figura 40: Imagen con método por color RGB-Rojo.	74
Figura 41: Imagen con método por color RGB-Verde.....	75
Figura 42: Imagen con método por color RGB-Azul	75



Figura 43: Extracción Vector de Características de Histograma de Imagen 76
Figura 44: Diseño de la Red Neuronal..... 77
Figura 45: Clases Red Neuronal 77
Figura 46: Salidas Deseadas 77



ÍNDICE DE TABLAS

Tabla 1: Tiempos de procesamientos promedios de 72 imágenes, obtenidos con los métodos Blanco y negro, HSV, Rojo, Verde y Azul. Las cifras se muestran en milisegundos(ms).	60
Tabla 2: Tiempo de reconocimiento con backpropagation de cada filtro de color, obtenidos con los algoritmos de detección automática. Las cifras se muestran en milisegundos (ms).....	65
Tabla 3: Resultados del total de billetes por backpropagation-verde.....	65
Tabla 4: : Resultados del total de billetes por backpropagation-HSV.....	66
Tabla 5: Comparación para el uso del celular	70
Tabla 6: Comparación de algoritmos por color	72



RESUMEN

La falsificación de billete es una cuestión candente alrededor del mundo. Los falsificadores son cada vez más difíciles de rastrear debido a su rápida adopción y adaptación con una tecnología muy avanzada. Uno de los más eficaces métodos para detener la falsificación puede ser el uso generalizado de herramientas de detección de falsificaciones / software que son fáciles de conseguir y son eficientes en términos de precisión.

Este proyecto de investigación se realizó con el fin de automatizar el proceso de detección de billetes, para lograrlo se empleó la siguiente metodología: En primer lugar se capturó la imagen digital con un Smartphone huawei P8 lite, luego se aplicó los algoritmos por color que en este caso es RGB-Verde y se extrajeron las características por color RGB. Terminada la extracción se identificaron los patrones que son 4 (rojo, verde, azul y alfa), para pasar a la clasificación y se compararon los resultados sea verdadero o falso, obteniendo con éxito una precisión promedio de 63%.

Palabras clave: Reconocimiento de patrones, algoritmos por color, billetes, características por color



ABSTRACT

Counterfeit ticket is a burning issue around the world. Counterfeiters are increasingly difficult to track due to its rapid adoption and adaptation with highly advanced technology. One of the most effective methods to stop counterfeiting may be the widespread use of counterfeit detection tools / software that are easy to get and are efficient in terms of accuracy.

This research project was conducted in order to automate the detection process tickets, to achieve the following methodology was used: First the digital image was captured with a smartphone huawei P8 lite, then the algorithms was applied color in this case is RGB-Green and extracted by the features RGB color. After the extraction patterns that are 4 (red, green, blue, and alpha), moving to the classification and compare the results to be true or false, successfully obtaining an average accuracy of 63% were identified.

Keywords: Pattern recognition algorithms for color, notes, color characteristics



INTRODUCCION

En la última década se ha producido un aumento sin precedentes con respecto a la cantidad de contenidos disponibles debido principalmente al uso masivo de Internet y a la proliferación de dispositivos multimedia en el ámbito cotidiano tanto a nivel empresarial como personal. La necesidad de soluciones adecuadas es cada vez más demandada en distintas y tan variadas áreas como Internet, aplicaciones de usuario, TV, bibliotecas digitales, aplicaciones médicas, etc. y se requieren métodos de acceso y gestión de la información para hacerla disponible de una manera más eficiente.

Las técnicas mencionadas anteriormente tienen una característica en común: representan el contenido mediante valores numéricos, que componen las características o descriptores, y que extraen diferentes propiedades del contenido permitiendo así un tratamiento más objetivo y con independencia de la naturaleza del contenido. Con respecto al ámbito visual, dichas técnicas se basan, por lo general, en comparaciones visuales centradas en la eliminación de información redundante o la recuperación de segmentos de vídeo similares. Las comparaciones visuales se llevan a cabo haciendo uso de diferentes los descriptores visuales existentes.

Sin embargo, tales descriptores han sido evaluados normalmente centrándose en su rendimiento respecto de la identificación o recuperación de contenidos similares desde un punto de vista semántico y no focalizando en la capacidad para la detección de imágenes pertenecientes a una identificación de billetes. Este proyecto se centra en la evaluación de los descriptores de imagen respecto de este tipo de situaciones: la identificación de billetes ya sea por diferentes transformaciones tales como cambios de iluminación, variaciones de zoom, cambios del punto de vista del objetivo y movimiento del billete.

La cuestión principal en la que se ahonda en este proyecto es: ¿Qué características visuales resultan más precisas en cuanto a la representación del contenido y alcanzan un mayor rendimiento con respecto a la detección de imágenes? Esta cuestión es ampliamente investigada examinando el comportamiento de un conjunto representativo de los diferentes descriptores de imagen.



Esta tarea sobre cómo de bien se comportan las diferentes características de las imágenes está íntimamente relacionada con la cuestión sobre qué características pueden ser combinadas para obtener mejores resultados en tareas concretas. Tomando parte en esta última cuestión se elaboran diferentes métodos de combinación de características basados en la correlación de las características individuales. Para la evaluación de las diferentes características o descriptores se ha utilizado una base de datos propia de imágenes reales que ha sido creada a partir de un muestreo manual de diferentes colecciones de billetes y que conforma un buen punto de partida para evaluar el rendimiento de los descriptores seleccionados así como de las nuevas combinaciones desarrolladas.

CAPÍTULO I:

PROBLEMA DE

INVESTIGACIÓN

1.1. Situación Problemática

El delito de falsificación de billetes es uno de los grandes males que intentan combatir los distintos países. En la actualidad, estas falsificaciones son prácticamente iguales que los originales, haciendo cada vez más complejo distinguirlas. La tecnología empleada para la falsificación utiliza y desarrolla métodos que hacen prácticamente imposible distinguir a simple vista y manualmente los billetes falsos de los auténticos, por ello hay diversos tipos de investigaciones que aplican diversos algoritmos para que detectemos de una manera automática los billetes falsos.

Como contrapartida se ha desarrollado diversos algoritmos que permiten controlar en forma sumamente precisa la legitimidad de los billetes, utilizando una variada metodología de reconocimiento.

A nivel internacional

Arceo Moheno G, De los Santos Torres G, Rivera Reyes A1. (2012). Uso de agentes inteligentes en cajeros automáticos de la Comisión Federal de Electricidad en Villahermosa (México), se presentan los resultados obtenidos de la propuesta de un sistema mediante agentes inteligentes encargados de controlar el funcionamiento de estos cajeros. El sistema se desarrolló mediante la metodología Tropos y se utilizó la herramienta de programación JADE. Los resultados demostraron que se puede incrementar la disponibilidad de los CFEMáticos (centros de servicios de pagos, etc.) con la implementación de este sistema hasta en 99.86%, dado que los agentes inteligentes, monitorean siempre los procesos que realizan los usuarios, restableciendo de forma inmediata los errores que puedan presentarse brindando, de esta manera, una mejor atención a los clientes.



Vera, J. O. (2013.). Detección de billetes con OpenCV en Android. Argentina. Se desarrolló una aplicación en Android, se aplicaron los algoritmos SURF, SIFT y ORB, en esta primera parte se obtuvo como primera conclusión los dos algoritmos(SURF and ORB) se parecían. Se suelen confundir un billete de 100 por uno de 50. Tal vez aumentando la resolución de las imágenes de los billetes se logre mejores resultados y las características de las imágenes a detectar deben ser únicas y bien diferenciadas. Como decía más arriba, confundía billetes de 50 y 100 por ser similares, como trabajos futuros pasos a investigar es la del desarrollo de un servidor que albergue el programa principal de reconocimiento de billetes y que luego desde cualquier artefacto (PC, celular, tablet, reloj con webcam, anteojos Google, etc.) envíe la imagen a procesar vía Internet y devuelva el resultado como simple texto.

Para identificarlos, el MIT(Instituto tecnológico de Massachusetts – Estados Unidos) está desarrollando una aplicación móvil capaz de escanear los patrones de color. Tan solo haría falta incorporar una lente especial al teléfono para comprobar si el objeto en cuestión es falso o no.

A nivel nacional

En los principales bancos(BBVA, Banco de la Nación, Interbank, etc.) existen máquinas electrónicas para detectar, contar y reportar billetes. Comprobando para ello mediante diversas tecnologías (ultravioleta, magnética, infrarrojo) distintos mecanismos de seguridad del billete. El detector de billetes falsos es un aparato muy demandado por todos aquellos que manejan efectivo, debido a que cada vez existe una mayor calidad en las falsificaciones, lo que hace que sin ayuda de



detectores sea casi imposible distinguir los billetes falsos de los originales.

Guzmán Bárcena, Sheylla. (2012). Detector de Billetes Para Personas con Discapacidad Visual. Perú. Universidad Tecnológica del Perú encontramos una investigación que se llegaron a realizar diversos Métodos (Documento, proceso de escaneado, segmentación y normalización, cálculos de patrones, etc.) y algoritmos de Reconocimiento Óptico de Caracteres que han sido muchos los servicios que han introducido estos procesos para aumentar su rendimiento y otros que se basan completamente en estas tecnologías, donde los resultados no fueron favorables, porque el programa no es soportado por el celular ya que se pensó hacer la comparación con 4 patrones que se encuentran en la base de datos del programa como un gestor de base de datos, pese a insistir en colocar el programa en el celular.

A nivel local aún no existe alguna investigación para el beneficio de la sociedad por lo cual existe mayor grado de estafa en billetes, siendo los principales afectados las personas, los centros comerciales, medios de transporte, bodegas, cajeros, o cualquier centro que brinde algún servicio hacia la sociedad. Ya que cuentan con detectores no garantizados y en ocasiones se encuentran malogrados, o también por la falta de información de cómo poder detectarlos, por ese motivo se incrementa el movimiento de ello en nuestra localidad.

Por lo tanto, se concluye que no existen investigaciones que permitan detectar un billete falso con una certeza al 100% de exactitud.

1.2. Formulación del problema

¿Qué algoritmo de reconocimiento de imágenes permitirá identificar billetes de forma eficiente?

1.3. Justificación e importancia

1.3.1. Justificación

a) Justificación Tecnológica:

Se justifica en la medida de que va analizando todos los algoritmos relacionados a lo que es reconocimiento de imágenes por descriptores por color se adecue mejor a nuestra realidad y de esta manera agregar un gran valor en tecnología, para que pueda estar un paso más adelante de la innovación para cualquier cambio que surja y llegar a tener una mejor probabilidad de acierto para el beneficio y la calidad de vida de las personas en general.

Es por ello, que esta investigación se pretende realizar un análisis comparativo de algoritmos de procesamiento de imágenes por descriptores de color con el fin de obtener resultados más efectivos.

b) Justificación Económica:

En el ámbito económico gracias al reconocimiento de imágenes por color hacia los billetes, habrá un porcentaje mayor que beneficiara a las personas, aumentará el tiempo de respuesta y la calidad de la solución en la detección de billetes ya que al realizar esto podríamos disminuir costos, y generar un impacto económico.

c) Justificación Social:

En el ámbito social nos daremos cuenta, que habrá un mayor beneficio, para todas aquellas personas que en su debido momento han sido perjudicados por el gran movimiento de billetes falsos que existe en nuestra sociedad, ya que llevará varios motivos como la confianza.



1.3.2. Importancia

Varios sistemas están disponibles en nuestra sociedad para ayudar a la gente con el reconocimiento de billetes. Sondas dispositivos muy útiles para hacer que la gente libre de dificultades de billetes verdaderos y falsos, contando billetes, cambiar dinero o expendedoras de billetes. Mientras sensores y máquinas de escaneo pueden proporcionar información visual pero no han demostrado gran promesa en reconocimiento de billetes de banco, sus reconocimientos se limitan a específica el medio ambiente estándar.

Las técnicas de procesamiento de imagen le ayudarán a comprobar cada billete con sus respectivas características y de esta manera lograr compararlos y tener un mayor porcentaje de acierto. El procesamiento de imágenes digitales puede proporcionar una contribución útil para el proceso de comparación de cada billete.

1.4. Objetivos

1.4.1. Objetivo general

Comparar algoritmos de reconocimiento de imágenes por descriptores de color para la identificación de billetes.

1.4.2. Objetivos específicos

- a) Identificar las características claves que se tienen en cuenta para identificar billetes.
- b) Identificar los algoritmos de comparación para reconocimiento de imágenes por descriptores de color.
- c) Seleccionar los algoritmos de color para reconocer billetes.
- d) Implementar los algoritmos seleccionados para reconocer billetes
- e) Evaluar los algoritmos reconocimientos de imágenes



1.5. Limitaciones del Proyecto:

La investigación estará limitada para realizar las diferentes comparaciones de los algoritmos por descriptores de color para identificar cual es óptimo, no se puede ejercer por falta de información, costo y tiempo.

CAPÍTULO II:

MARCO

TEÓRICO

2.1. Antecedentes de Estudios

Vera, José Obdulio. 2013 - Argentina. Detección de Billetes con OpenCV en Android que se aplicaron los algoritmos SURF, SIFT y ORB, en esta primera parte se obtuvo como primera conclusión los dos algoritmos(SURF and ORB) se parecían. Se suelen confundir un billete de 100 por uno de 50. Tal vez aumentando la resolución de las imágenes de los billetes se logre mejores resultados y las características de las imágenes a detectar deben ser únicas y bien diferenciadas. Como decía más arriba, confundía billetes de 50 y 100 por ser similares, como trabajos futuros pasos a investigar es la del desarrollo de un servidor que albergue el programa principal de reconocimiento de billetes y que luego desde cualquier artefacto (PC, celular, tablet, reloj con webcam, anteojos Google, etc.) envíe la imagen a procesar vía Internet y devuelva el resultado como simple texto.

Para identificarlos, el MIT(Instituto tecnológico de Massachusetts – Estados Unidos) está desarrollando una aplicación móvil capaz de escanear los patrones de color. Tan solo haría falta incorporar una lente especial al teléfono para comprobar si el objeto en cuestión es falso o no.

Cristián Arriagada García / Diego AracenaPizarro. 2007. Escuela Universitariade Ingeniería Industrial, Informática y Sistemas - Chile. Detección y Reconocimiento de Señales de Tránsito Utilizando Matching de Chamferelprototipo desarrollado con técnicas de visión computacional, permite detectar y reconocer las señal éticas presentes en la vía e informar su naturaleza al conductor mediante una señal audible o proyección visual.

La investigación se centró principalmente en las fases de detección inicial; con el fin de contar con una heurística rápida, aprovechando la segmentación por color, con sus características de invariabilidad del sistema HSV (Brillo, Saturación, Valor) utilizando técnicas de transformación distancia y matching jerárquico de Chamfer.

Elena LazkanoOrtega-Basilio Sierra Araujo. 2010. Universidad del País Vasco / EuskalHerrikoUnibertsitatea Departamento de Ciencia de la Computación Inteligencia Artificial – España. Algoritmo de reconocimiento de formay color para una plataforma robótica, en esta investigación se lograron realizar 3 tipos de algoritmos RGB, YUV y HSL y teniendo como primer resultado la eficiencia obtenida con los filtros realizados con los distintos modelos de colores bastante parecida. Con el modelo RGB se obtiene una eficacia del 66.7%, con YUV el 72.2% y con HSL 74.4%, en el segundo resultado Se observa que la efectividad del sistema baja considerablemente cuando la luz del entorno en la que se ha realizado la prueba se considera inadecuada. Con una luminosidad adecuada se consigue una eficacia del 87.4% y en cambio cuando la luminosidad es considerada inadecuada la eficiencia cae hasta el 54.8%. En general la efectividad del sistema es de un 71.1%.

SheyllaGuzmán Bárcena. 2012. Universidad Tecnológica del Perú. Aplicación Móvil Para Personas con Discapacidad Visual Aplicando Diversos métodos(Documento, proceso de escaneado, segmentación y normalización, cálculos de patrones, etc.) y algoritmos de Reconocimiento Óptico de Caracteres que han sido muchos los servicios que han introducido estos procesos para aumentar su rendimiento y otros que se basan completamente en estas tecnologías, donde los resultados no fueron favorables, porque el programa no es



soportado por el celular ya que se pensó hacer la comparación con 4 patrones que se encuentran en la base de datos del programa como un gestor de base de datos, pese a insistir en colocar el programa en el celular.

2.2. Estado del arte

En 1914 se inició la circulación de los billetes en el Perú, ante la grave crisis por el pánico financiero internacional y local a raíz del inicio de la Gran Guerra, el Congreso y el gobierno peruano del coronel Óscar R. Benavides tomaron medidas económicas de emergencia. Así, mediante la Ley 1968 (22/08/1914) se autorizó a los bancos y a la Caja de Ahorros de la Beneficencia de Lima la emisión de cheques circulares (había resistencia a llamarlos billetes, aunque en la práctica lo eran) con valor de 1, 5 y 10 libras peruanas (la moneda de entonces) hasta por un millón 100 mil libras. Eran papeles garantizados por los activos y oro metálico, cédulas hipotecarias, fondos de oro en Inglaterra y Estados Unidos, bonos hipotecarios, créditos hipotecarios a favor y warrants de los almacenes fiscales del Estado peruano.

1933-1950(Emisión 1933 Ferrocarril) La primera emisión de billetes del Banco Central de reserva del Perú al amparo de la Ley N^o7137, fueron impresos por American Bank Note Company con denominaciones; S/5, S/10, S/50 y S/100.

1952-1968(Emisión 1952 Patria Sentada) A partir de 1946, comienzan a circular algunas denominaciones de esta emisión, y fue en 1952 que circula la emisión completa hasta 1968, al amparo de las leyes N^o 10535 y N^o13958. Fueron impresos por varias casas internacionales, con denominaciones; S/5, S/10, S/50, S/100 y S/500.



Posteriormente se emiten billetes de S/10.000(01/02/1979) y S/50.000. (05/11/1981). Impresos por varias casas internacionales

1985-1991(Emisión 1985 Intis) La llegada al Gobierno del Dr. Alan García Pérez ocasiona un cambio en el Sistema Monetario Nacional, denominación y diseño de los billetes.

Hasta esta fechas como no había mucha corrupción y estafa sobre los billetes no tenían seguridades que garantizaban la diferencia de un billete verdadero a un falso

1991(Primera Familia de Billetes), se sumaron algunas características principales para reconocerlas a simple vista como Marca de agua, hilo de seguridad, fibrillas, microimpresión debajo del nombre “Banco Central de Reserva del Perú”, etc.

2011(Nueva Familia de Billetes) nuevos billetes con renovados diseños, que incorporan elementos de seguridad de última generación de fácil reconocimiento por parte del público. En los billetes nuevos se mantienen las mismas figuras de los personajes principales, que son imágenes más grandes y con detalles más finos en los relieves que facilitan su reconocimiento, también encontramos marcas de aguas, fibrillas, hilos de seguridad y color para que de este modo sea menor las estafas de billetes y lo detecten con una mayor facilidad en el momento de la circulación de los billetes en el ámbito local.

2.3. Bases Teórico – Científicas

2.3.1. Billetes

Primera Familia de Billetes – 1991



Figura 1: Billeto de S/. 10 Primera familiar

En el anverso del billete de S/. 10 figura José Abelardo Quiñones Gonzales, héroe de la aviación militar del Perú y una vista de la Escuela de Oficiales de la Fuerza Aérea Peruana. En el reverso se aprecia el vuelo acrobático que realizó el día de su graduación.



Figura 2: Billeto de S/. 20 Primera familiar

En el anverso del billete de S/. 20 figura Raúl Porras Barrenechea, catedrático, diplomático y periodista, así como el patio principal de la Casona de San Marcos. El reverso presenta la fachada del Palacio de

Torre Tagle, sede de la Cancillería del Perú, donde ejerció el cargo de Ministro de Relaciones Exteriores.



Figura 3: Billeto de S/. 50 Primera familiar

En el anverso del billete de S/. 50 figuran imágenes del escritor Abraham Valdelomar Pinto y de la fachada del Palais Concert, centro de reunión habitual de los intelectuales de la época. En el reverso se muestra la laguna de Huacachina ubicada en Ica, ciudad natal del escritor.



Figura 4: Billeto de S/. 100 Primera familiar

En el anverso del billete de S/. 100 presenta la imagen del ilustre historiador Jorge Basadre Grohmann y una vista del Paseo Cívico de Tacna, su ciudad natal. En el reverso figura la fachada de la Biblioteca Nacional, la que se encargó de reconstruir luego de un incendio y de la cual fue director.



Figura 5: Billeto de S/. 200 Primera familiar

En el anverso del billete de S/. 200 se observa la imagen de Santa Rosa de Lima, Patrona de América, Indias y Filipinas y del pozo ubicado en el Santuario de Santa Rosa con siluetas estilizadas de golondrinas. En el reverso figura el interior del Convento de Santo Domingo donde descansan sus restos.

Nueva Familia de Billetes – 2011



Figura 6: Billeto de S/.10 Nueva Familia

En el anverso del billete de S/. 10 figura José Abelardo Quiñones Gonzales, héroe de la aviación militar del Perú y el vuelo acrobático que realizó el día de su graduación. En el reverso se aprecia la ciudadela de Machu Picchu en Cusco.



Figura 7: Billeto de S/.20 Nueva Familia

En el anverso del billete de S/. 20 se muestra a Raúl Porras Barrenechea, distinguido catedrático, diplomático y periodista y la fachada del Palacio de Torre Tagle, sede de la Cancillería del Perú. En el reverso figura una de las construcciones del Complejo Arqueológico de Chan Chan en La Libertad.



Figura 8: Billeto de S/.50 Nueva Familia

En el anverso del billete de S/. 50 figuran las imágenes del escritor Abraham Valdelomar Pinto y de la fachada del Palais Concert, centro de reunión habitual de los intelectuales de la época. En el reverso se observa las ruinas del Templo Nuevo de Chavín de Huantar en Ancash.



Figura 9: Billete de S/100 Nueva Familia

En el anverso del billete de S/. 100 se observa la imagen del ilustre historiador Jorge Basadre Grohmann y una vista del Paseo Cívico de Tacna, su ciudad natal. En el reverso figura una de las construcciones del Gran Pajatén ubicada en San Martín.



Figura 10: Billete de S/.200 Nueva Familia

En el anverso del billete de S/. 200 se presenta imágenes de Santa Rosa de Lima, Patrona de América, Indias y Filipinas y del pozo de su santuario. En el reverso figura la Ciudad Sagrada de Caral ubicada en Lima.

2.3.2. Reconocimiento de Imágenes

Imagen digital

Según Scarel (2010) se puede definir una imagen en escala de grises como una función bidimensional $f(x,y)$, donde x e y son coordenadas espaciales, y el valor de f en un par de coordenadas dado es denominado intensidad o nivel de gris. Cuando x , y y f son cantidades discretas, la imagen es llamada imagen digital. De este modo, esta puede ser representada como una matriz de $M \times N$ elementos llamados píxeles.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N - 1) \\ f(1,0) & f(1,1) & \dots & f(1, N - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M - 1,0) & f(M - 1,1) & \dots & f(M - 1, N - 1) \end{bmatrix}$$

Figura 11: Cálculo Imagen Digital

Para el tratamiento digital de imágenes (Gonzales y Woods, 1996), los valores de M (filas), N (columnas) y el número de niveles de gris permitidos por cada pixel (G) son resultados de potencias enteras de 2, relacionándose con el código binario; esto es:

$$N = 2^n \quad M = 2^k \quad G = 2^m$$

Donde n , k y m son números enteros; m representa la cantidad de bits a usar.

Empleando los diferentes elementos de la ecuación anterior se obtiene la cantidad de bits (b) que representan el peso en memoria de la imagen:

$$b = N \cdot M \cdot m$$

La resolución de una imagen (el grado de detalle discernible) depende directamente del muestreo y la cuantización. Mientras más grande sea la resolución, más se aproximará la imagen digitalizada a la imagen



real, pero esto conlleva al aumento del área de almacenamiento y peso en memoria, por lo que las necesidades de procesamiento crecen de forma proporcional con respecto a N , M y m ; que se traduce en una velocidad de procesamiento mucho menor.

Etapas del Procesamiento de Imágenes

El tratamiento digital de imágenes comprende hardware, software y recursos teóricos, donde debe existir un conocimiento previo sobre cómo será el resultado. González y Woods (1996) indican que este conocimiento no sólo guía la operación de cada módulo de procesamiento, sino que también ayuda en las operaciones de realimentación entre ellos a través de la base de conocimiento. Wainschenker (2011) menciona un diagrama (Ver figura N° 12) donde se pueden apreciar las diferentes etapas dentro del procesamiento de imágenes.

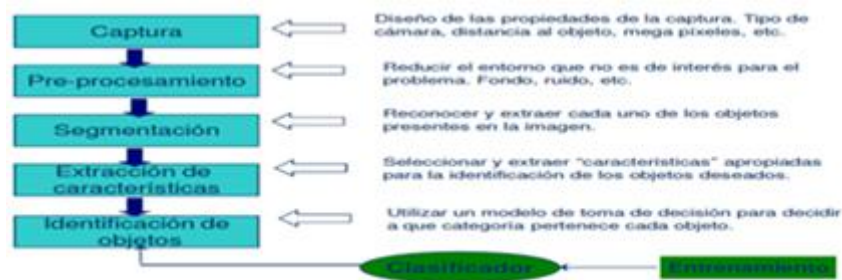


Figura 12: Etapas del Procesamiento de Imágenes

- **Captura: Adquisición de la imagen**

Para esta etapa se necesitan dos elementos: el primero es un dispositivo físico sensible a una determinada banda del espectro de energía electromagnética y que produzca una señal eléctrica de salida proporcional al nivel de energía detectado; el segundo, es un medio para convertir la señal de salida del sistema a forma digital.



Entre estos dispositivos para adquirir imágenes destacan las cámaras digitales, cámaras filmadoras, cámaras web, entre otros.

- **Pre-procesamiento de la imagen**

Su función es mejorar la imagen de forma que se aumenten las posibilidades de éxito en los procesos posteriores. Consiste en aplicar técnicas de mejoramiento del contraste, eliminar el ruido y aislar regiones cuya textura indica la probabilidad de información necesaria para el estudio mediante procedimientos que normalmente se expresan en forma de algoritmos.

- **Extracción de características**

Es el proceso de generar características que puedan ser usadas en la clasificación de los datos. En general, se busca el conjunto mínimo de características que permiten determinar de manera uniforme a qué clase pertenecen todos los objetos del universo de trabajo. La figura N° 13 muestra las categorías de las características: funciones o parámetros. Cuando se utilizan las características de función, la imagen es usualmente caracterizada en términos de una función del tiempo cuyos valores constituyen un conjunto de características. Cuando se utilizan características de parámetros, la imagen es caracterizada como un vector de elementos, cada elemento representa el valor de una característica. En general, las características de funciones tienen un mejor rendimiento que las de parámetros, pero consumen más tiempo.



Figura 13: Categoría de las características

Las características de parámetros se clasifican generalmente en dos categorías: global y local. La primera categoría concierne a la imagen completa, mientras que la segunda se refiere a características extraídas de partes específicas de la imagen. Dependiendo del nivel de detalle considerado, los parámetros locales pueden dividirse en parámetros de componente orientados, que son extraídos a nivel de cada componente, y parámetros orientados a pixel, que son extraídos a nivel de información de píxeles. Los parámetros extraídos a nivel de pixel pertenecen a características basadas en celdas, donde la imagen es dividida en regiones rectangulares y las características son evaluadas en cada región. (Impedovo y Pirlo, 2008).

Las características de parámetros locales, especialmente las extraídas a nivel de pixel, son las que comúnmente se utilizan en los Modelos Ocultos de Markov, pues asemejan la forma de la secuencia de observaciones requerida para crear dichos modelos.

Algunos ejemplos de este tipo de características son:

- Filtro

(Gonzales & Woods, 2008)El filtrar una imagen (f) consiste en aplicar una transformación (T) para obtener una nueva imagen (g) de forma que ciertas características son acentuadas o disminuidas:

$$g(x, y) = T[f(x, y)]$$



Podemos considerar que la señal (imagen) pasa a través de una caja o sistema (filtro) cuya salida es la imagen filtrada. (ver fig. 14).

De acuerdo a la teoría de sistemas, al pasar una señal por un sistema lineal, la salida es la convolución de la transformación del sistema (función de transferencia) con la señal de entrada:

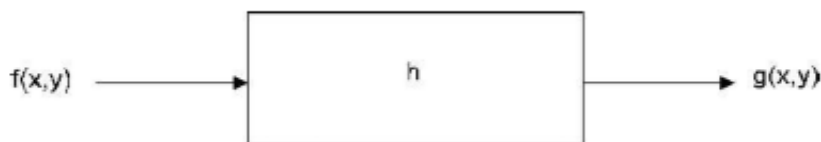


Figura 14: Proceso de Filtrado

$$g(x; y) = h(x; y) * f(x; y)$$

Por el teorema de la convolución, esto corresponde a la multiplicación en el dominio de la frecuencia:

$$G(u; v) = H(u; v)F(u; v)$$

Por esto, podemos pensar en dos formas básicas de filtrar una imagen, realizarlo en el dominio espacial, que implica una convolución, o en el dominio de la frecuencia, que implica solo multiplicación, pero dos transformaciones de Fourier (de espacio a frecuencia y viceversa). Ambos tipos de filtros han sido ampliamente estudiados y a continuación veremos sólo una introducción general y su aplicación en imágenes.

Filtros de suavizamiento: El objetivo de los filtros de suavizamiento es eliminar ruido o detalles pequeños que no sean de interés. Esto corresponde a un filtro pasa-bajo en el dominio de la frecuencia, es decir que se eliminan o reducen las altas frecuencias. En la figura 15 se muestra la respuesta de un filtro pasa-bajo en frecuencia (en una



dimensión) y la correspondiente respuesta que debe tener en el dominio espacial.

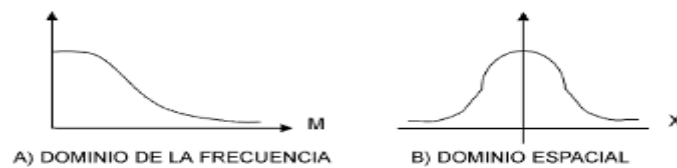


Figura 15: Filtro pasa-bajos: (a) en frecuencia, (b) en el dominio espacial

Existen varios tipos de filtros para suavizamiento, los más comunes son:

Promedio o media aritmética: Obtiene el promedio de los pixeles vecinos ($w = 1$); es decir, todos los valores de la máscara son 1.

Mediana: Substituye el valor del pixel central por el de la mediana de los valores contenidos en el vecindario.

Gaussiano: Aproximación a una distribución gaussiana en dos dimensiones.

Considerando una media igual a cero, la función de transformación de un filtro tipo gaussiano es:

$$T(x, y) = e^{-[(x^2 + y^2) / 2\pi\sigma^2]}$$

Para un máscara de 3x3 los valores de un filtro gaussiano "típico" se muestran en la figura 16 La cantidad de "suavizamiento" que realiza el filtro gaussiano se puede controlar variando la desviación estándar y el tamaño de la máscara.

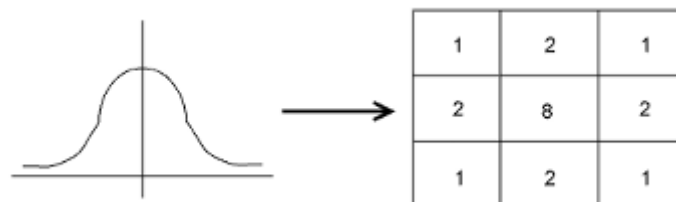


Figura 16: Máscara para filtro gaussiano de 3x3



El filtro Gaussiano, en general, da mejores resultados que un simple promedio o media y se argumenta que la vista humana hace un filtrado de este tipo. El filtro Gaussiano "normal" o isotrópico tiene la desventaja de suavizar las orillas o discontinuidades, generando que se emborronen.

Lo anterior genera problemas en las posteriores etapas de visión. El algoritmo de mediana es particularmente efectivo en imágenes con poco ruido. Su efectividad decrece drásticamente en imágenes ruidosas.

Filtros de acentuamiento: El objetivo de los filtros de acentuamiento es intensificar los detalles y cambios bruscos de intensidad mientras atenúa las bajas frecuencias. El resultado es un acentuamiento de las orillas (edgesharpening). Se conocen como filtros de pasa-alto porque dejan pasar las altas frecuencias y eliminan las bajas frecuencias, en forma inversa al filtro pasa-bajo. En la figura 17 se muestra como se reducen las bajas frecuencias y se mantienen las altas.

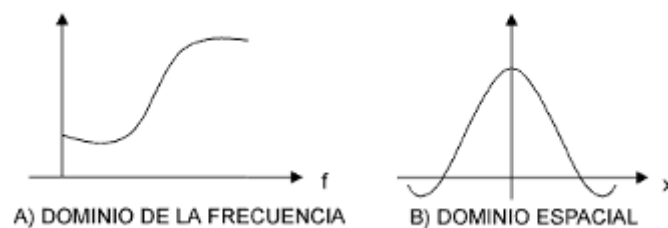


Figura 17: Filtro pasa-alto: (a) en frecuencia, (b) en el dominio espacial

Existen también varias formas de implementar este tipo de filtrado. Una forma típica de un filtro pasa-alto es una máscara del tipo de la figura 18. Para este filtro la suma de los pesos es cero, de forma que una región de intensidad constante resultará en un valor 0. Nótese que a diferencia del filtro de suavizamiento los pesos de los vecinos son

negativos, este efecto substractivo genera la acentuación de los cambios de intensidad.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 18: Máscara de 3x3 para un filtro pasa-alto simple

Filtro para énfasis de altas frecuencias: El filtrado de acentuamiento o pasa altos presenta sólo las discontinuidades, atenuando fuertemente las bajas frecuencias y haciendo que “desaparezcan” las regiones homogéneas. Un tipo de filtro que aún acentuando las altas frecuencias preserva las bajas es el filtro “énfasis de altas frecuencias” (highboost). Para obtener una imagen con énfasis de altas frecuencias (EA), se puede considerar que se multiplica la imagen original por una constante A, esta constante debe ser mayor que uno para que acentúe.

$$EA = (A) \text{ original PB}$$

Eso es equivalente a la siguiente expresión:

$$EA = (A - 1) \text{ original} + PA$$

En la práctica no es necesario hacer exactamente esta operación, sino se implementa haciendo la celda central del filtro pasa-alto:

$$w = 9A - 1$$

Como se ilustra en la figura 19

-1	-1	-1
-1	9A - 1	-1
-1	-1	-1

Figura 19: Máscara de 3x3 para un filtro pasa-alto con énfasis en las altas frecuencias.



- **Modelos de color Blanco y Negro:**

Primero la imagen debe convertirse en un formato más apropiado para extraer las características. Se consideró que en esta primera aproximación, que lo más relevante de las imágenes a tratar era su contorno, y para hallar su contorno no es necesario tener en cuenta el color de las imágenes. Por lo que es más conveniente convertir la imagen a una escala de grises. En este proyecto se trabajó con el modelo RGB. En este modelo se asigna una

intensidad a cada uno de los tres colores primarios de luz: rojo, verde y azul, de esta manera cada píxel en una fotografía se representa mediante un valor que identifique la intensidad de cada uno de estos tres colores

que, mezclados por adición, se acerque más al color verdadero del píxel, las cámaras digitales modernas utilizan valores entre 0 y 255 permitiendo de esta manera más de 16 millones de colores distintos. En una escala de grises en cambio, cada píxel es representado con un único valor entre 0 y 255, el cual representa que tan oscuro es el mismo. Para convertir un píxel de una imagen en color a uno en escala de grises se suele hacer un promedio ponderado de la intensidad de cada uno de los tres colores en donde a cada color se le asigna un peso. Al ser un promedio ponderado la suma de los tres pesos debe ser igual a 1. Los pesos utilizados por OpenCV son:

$$gris = 0,2989 * rojo + 0,5870 * verde + 0,1140 * azul$$

Estos pesos fueron calculados a partir de observaciones de la sensibilidad del ojo humano a cada uno de los tres colores.

El resultado de aplicar la anterior fórmula es el siguiente:





Figura 20: Modelo de color blanco y negro

- **Modelo de color HSV**

Este modelo de color es considerado más cercano que el modelo de color RGB en la forma en que las personas describen las sensaciones de color. El modelo de color HSV obtiene el nombre de las siglas en inglés (Hue, Saturation, Value) que en español significan Tonalidad, Saturación, Valor, respectivamente; es una transformación no lineal del modelo de color RGB, y los colores son una combinación de los tres valores: Tonalidad, Saturación o cantidad de color, y su valor. Estos valores son representados en un diagrama circular, que es mostrado en la Figura 21.



Figura 21: Modelo de Color HSV

Las tres magnitudes pueden tener los siguientes valores:

- Tonalidad: Es el tipo de color (por ejemplo: rojo, verde, o amarillo), que son representados como el valor del grado de un ángulo, cuyos valores posibles están en el rango de 0° a 360° (aunque para algunas aplicaciones estos valores son normalizados de 0% a 100%).



- Saturación: Es representado como la distancia del eje del brillo negro/blanco. Los posibles valores están en el rango de 0% a 100%.
- Valor: Representa la altura en el eje negro – blanco. Los posibles valores están en el rango de 0% a 100%. Donde 0 es siempre negro, dependiendo de la saturación, 100 podría ser blanco o un color más o menos saturado.

Para convertir del modelo de color RGB al HSV se usan las expresiones matemáticas (2) al (4), que son mostradas a continuación:

$$H = \arccos \frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{[(R-G)^2 + (R-B)(G-B)]}} \quad (2)$$

$$S = 1 - 3 \frac{\min(R,G,B)}{R+G+B} \quad (3)$$

$$V = \frac{1}{3}(R+G+B) \quad (4)$$

- **Modelo de color YUV**

El modelo de color YUV es usado en los sistemas PAL (Phase Alternation Line) y NTSC (National Television System Committee) de difusión televisiva, los cuales son los estándares en la mayoría de los países del mundo. YUV está definido en términos de una componente de luminancia (Y) y dos componentes de crominancia (U, V). Codifica una imagen o video color teniendo en cuenta la percepción humana, permitiendo usar un ancho de banda menor para las componentes de crominancia (U, V), disminuyendo los errores de transmisión o imperfecciones de compresión, que usando una representación RGB.

- **Modelo de color RGB**

La descripción RGB (de sus siglas en inglés Red, Green, Blue; "rojo, verde, azul") de un color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el



rojo, el verde y el azul. Es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores luz primarios. El modelo de color RGB no define por sí mismo lo que significa exactamente rojo, verde o azul, por lo que los mismos valores RGB pueden mostrar colores notablemente diferentes en diferentes dispositivos que usen este modelo de color. Aunque utilicen un mismo modelo de color, sus espacios de color pueden variar considerablemente.

Una imagen de color RGB (la más usada para la visión computacional) es representada por una matriz tridimensional $m \times n \times p$, donde m y n tienen la misma significación que para el caso de las imágenes de escala de grises mientras p representa el plano, que para RGB puede ser 1 para el rojo, 2 para el verde y 3 para el azul. La figura 22 muestra detalles de estos conceptos.

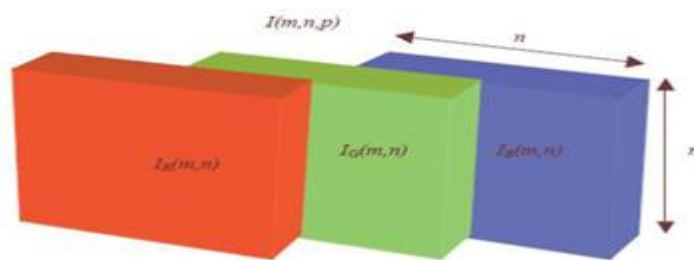


Figura 22: Modelo RGB

- **Modelo de color YCbCr**

Éste es un modelo de color usado en sistemas de video y fotografía digital, donde Y es la componente de luminancia y Cb y Cr son las componentes de crominancia azul y roja. YCbCr es el equivalente digital al modelo YUV (Sistema de TV). Concentra la mayor parte de la información de la imagen en la luminancia y menos en la crominancia. El resultado es que los elementos de YCbCr están menos



correlacionados y pueden ser codificados por separado. En el modelo YCbCr, cada cuadro se compone de una luminosidad única (blanco y negro) de la imagen, y dos componentes de crominancia. La información del Verde proviene de una combinación de los datos de crominancia y luminancia. YCbCr fue creado con el fin de reducir el ancho de banda de la señal de vídeo mediante la eliminación de la luminancia redundante en la señal de RGB (cada uno con su propio valor de luminancia). Esto también permite la reducción de la información de color sin afectar a la resolución real de la imagen. Este proceso se llama Sub muestreo de color. El paso de un sistema RGB a un YCbCr es obtenido mediante las ecuaciones:

$$Y = 0,3R + 0,6G + 0,1B$$

$$Cb = \frac{B - Y}{2} + 0.5$$

$$Cr = \frac{R - Y}{2} + 0.5$$

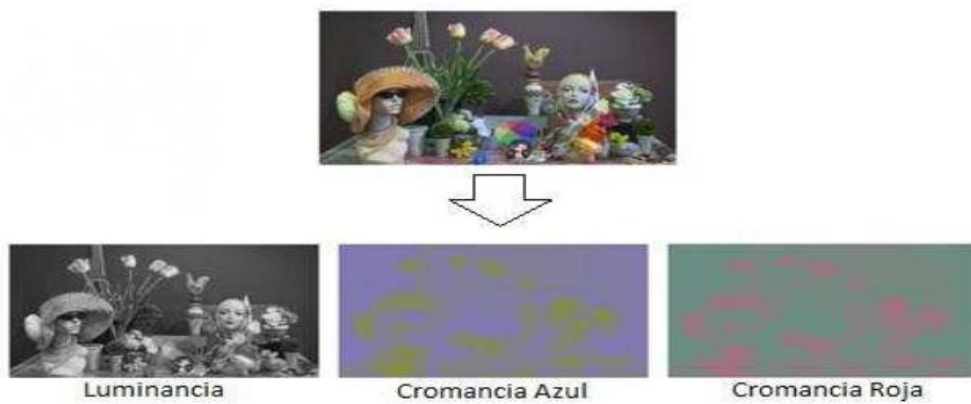


Figura 23: Imagen representada en el modelo YCbCr



2.3.3. Reconocimiento de Patrones

Detección de Objetos

(Szeliski, 2010) Si se nos da una imagen de analizar, como el retrato de grupo en la figura, podríamos tratar de aplicar un algoritmo de reconocimiento de todas las posibles sub-ventana en esta imagen.

Tales algoritmos son propensos a ser tanto lento y propenso a errores. En cambio, es más eficaz para construir Detectores de propósito especial, cuyo trabajo consiste en encontrar rápidamente probables regiones donde podrían ocurrir objetos particulares.

Características, Vectores de Características y Clasificadores

Descriptores en Imágenes

(Boullosa Gracia, 2011) Los descriptores visuales describen las características de los contenidos dispuestos en imágenes o en vídeos. Describen características elementales tales como la forma, el color, la textura o el movimiento, entre otros.

El contenido de una imagen está codificado digitalmente en el valor de cada una de las unidades mínimas de información que la componen llamadas pixels.

De esta manera los pixels representan el nexo de unión entre el contenido abstracto de sus valores y las características propias de una imagen que entendemos como relevantes para el humano. Es por ello que cualquier método de gestión de imágenes basado en su contenido deberá guardar algún tipo de relación o actuar sobre el valor de los mismos.

Redes Neuronales Artificiales

(Ponce Cruz, Inteligencia Artificial con Aplicaciones a la Ingeniería, 2010) Las redes neuronales artificiales son un método de resolver



problemas, de forma individual o combinadas con otros métodos, para aquellas tareas de clasificación, identificación, diagnóstico, optimización o predicción en las que el balance datos o conocimiento se inclina hacia los datos y donde, adicionalmente, puede haber la necesidad de aprendizaje en tiempo de ejecución y de cierta tolerancia a fallos. Está inspirado en el comportamiento biológico de las neuronas y en la estructura del cerebro.

Una neurona biológica es una célula especializada en procesar información. Está compuesta por el cuerpo de la célula (soma) y dos tipos de ramificaciones: el axón y las dendritas. La neurona recibe las señales (impulsos) de otras neuronas a través de sus dendritas y transmite señales generadas por el cuerpo de la célula a través del axón.

(Florez & Fernandez, 2008) La arquitectura de una red neuronal está conformada por múltiples unidades procesadoras elementales interconectadas, formando estas un sistema adaptivo que posee un algoritmo para ajustar sus pesos y alcanzar los requerimientos de desempeño del problema basado en muestras representativas.

Existen tres tipos de neuronas artificiales: unidades de entrada, de capa oculta y unidades de salida.

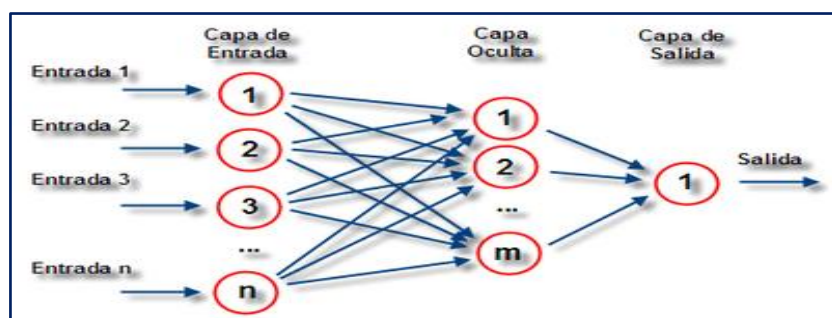


Figura 24: Estructura de una red neuronal multicapa, con una capa de entrada, una capa de oculta y una capa de salida.



Las neuronas de la capa de entrada reciben señales del entorno, del exterior, provenientes de sensores u otros medios de información. Las neuronas de la capa oculta reciben las señales recibidas dentro del sistema sin tener contacto con el exterior de la red. Es en esta capa donde ocurre el procesamiento de la información. Las neuronas envían la información procesada hacia el exterior.

Método Backpropagation.

El método backpropagation o también llamado propagación del error hacia atrás, este método está basado en la generalización de la regla delta. El funcionamiento de la red backpropagation consiste en el aprendizaje de un conjunto predefinido de pares de entradas-salidas dados como ejemplo: primero se aplica un patrón de entrada como estímulo para la primera capa de las neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado en las neuronas de salida con la salida que se desea obtener y se calcula un valor de error para cada neurona de salida. A continuación, estos errores se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aportación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que en la siguiente vez que se presente el mismo patrón, la salida esté más cercana a la deseada.

La importancia de la red backpropagation consiste en su capacidad de auto adaptar los pesos de las neuronas de las capas intermedias para aprender la relación que existe ente un conjunto de patrones de entrada y sus salidas correspondientes. Es importante la capacidad de generalización, facilidad de dar salidas satisfactorias a entradas que el



sistema no ha visto nunca en su fase de entrenamiento. La red debe encontrar una representación interna que le permita generar las salidas deseadas cuando se le dan entradas de entrenamiento, y que pueda aplicar, además, a entradas no presentadas durante la etapa de aprendizaje para clasificarlas.

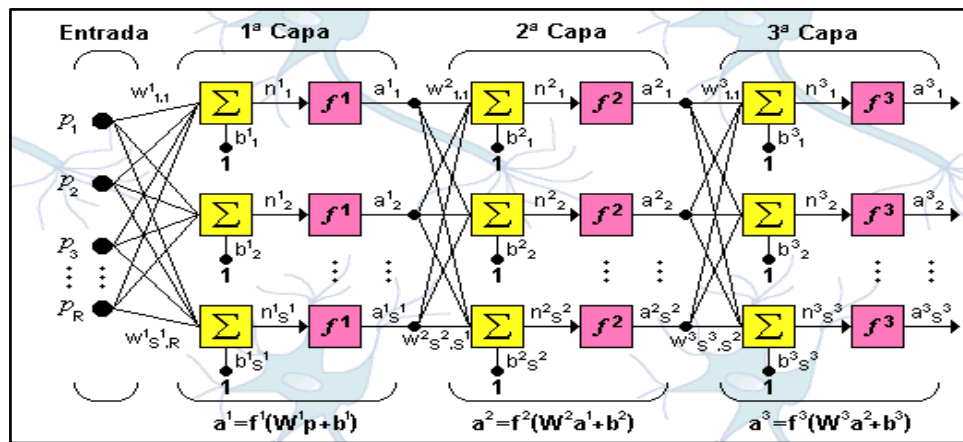


Figura 25: Método Backpropagation



CAPÍTULO III:

MARCO

METODOLÓGICO



3.1. Tipo y diseño de la investigación

La presente investigación es de tipo Tecnológica y su diseño Experimental.

3.2. Población y muestra

3.2.1. Población:

Nuestra población viene definidas bases de datos de 6000 imágenes de billetes digitales.

3.2.2. Muestra:

La muestra es la pequeña porción representativa y adecuada de la población, a partir de la cual el investigador va a obtener datos que son puntos de partida de las generalizaciones.

$$n = \frac{N * Z_a^2 * p * q}{d^2 * (N - 1) + Z_a^2 * p * q}$$

Donde:

N = Total de la Población

Z_a = 1.96 al cuadrado (si la seguridad es del 95%)

p = proporción esperada (en este caso 5% = 0.05)

q = 1 – p (en este caso 1 - 0.05 = 0.95)

d = precisión (5%)

Haciendo el cálculo con la fórmula, nuestra población trabajó con 72 billetes entre verdaderos y falsos.

3.3. Hipótesis

Si se comparan los algoritmos de reconocimiento de imágenes por descriptores de color se podrían precisar los requisitos para un algoritmo de identificación de billetes.



3.4. Variables

3.4.1. Independiente:

Reconocimiento de imágenes por descriptores de color

3.4.2. Dependiente:

Identificación de billetes.

3.5. Operacionalización

Variable Independiente	Dimensiones	Indicadores	Ítems
Reconocimiento de imágenes por de descriptores de color	Procesamiento de imágenes	Tiempo de Procesamiento de una imagen	Tiempo en milisegundos que lleva el procesamiento por imagen.
	Descriptores de color	Tiempo de extracción de características de una imagen	Tiempo en milisegundos que lleva a un descriptor obtener las características de una imagen
	Clasificadores	Tiempo de clasificación por imagen.	Tiempo en milisegundos lleva el procesamiento por imagen
	Imágenes digitales	Cantidad de imágenes utilizadas	Cantidad de imágenes de muestra para el entrenamiento y testeo del sistema.

Variable Dependiente	Dimensiones	Indicadores	Ítems
Identificación de billetes	Rendimiento del sistema de reconocimiento de patrones.	Imágenes clasificadas correctamente	Número de imágenes clasificadas correctamente
		Probabilidad de acierto global	Porcentaje calculado del número de aciertos sobre la cantidad de imágenes total
		Precisión	Proporción de muestras correctamente diagnosticadas del total de muestras evaluadas.
		$P = \frac{(\#Tadictos \cap \#Tmuestras)}{(\#Tmuestras)}$	
		Revocación	Proporción de muestras correctamente diagnosticadas para cada tipo de deficiencia billetes
		$R = \frac{(\#Tadictos \cap \#Tmuestras)}{(\#Tadictos)}$	
		Valor F	Permite corregir el error de la Distancia, en los casos en los que la revocación (R) y la precisión (P) se compensan, dando así, una medida de precisión al algoritmo.
		$F = 2 \cdot \frac{Precision \cdot Revocacion}{Precision + Revocacion}$	



3.6. Métodos, técnicas e instrumentos de recolección de datos

3.6.1. Método de investigación

El método de investigación que se ha empleado responde a la técnica e instrumentos de recolección de datos desde la perspectiva metodológica cuantitativa.

3.6.2. Técnicas de recolección de datos

Observación

Mediante esta técnica, se pretende observar los billetes a fin de detectar las posibles características que pueda tener. Esto nos ayuda a seleccionar las muestras más idóneas para el entrenamiento del sistema y obtener mejores resultados.

Juicio de expertos

Para la corroboración de datos sobre el diagnóstico de las características en una muestra, se recurrió a un profesional en billetes con conocimientos sobre características para la detección de billetes falsos. Se le entregó un formato de evaluación en el que señalará que tipo de características posee cada muestra. Esta información es importante para la clasificación de las muestras según qué tipo de deficiencia posee.

Entrenamiento y prueba del sistema

Las muestras fueron procesadas para obtener sus características utilizando distintos descriptores de color. Esta información fue almacenada en una base de datos. El sistema fue entrenado a partir de esta base para poder clasificar las siguientes muestras según el tipo de características que posea. Se recolectó información relevante sobre la eficiencia y eficacia de cada descriptor para obtener un diagnóstico en la etapa de prueba.



3.6.3. Instrumentos de recolección de datos

La guía de observación permitió determinar las características y reconocer los diferentes tipos de características de cada billete (Ver Anexo 1).

3.7. Procedimiento para la recolección de datos

En la observación se realizó visitas en las comisarías, y en los mismos bancos y se recolectó la mayor cantidad de muestras de distintos billetes con las características más resaltantes. Se requirió la ayuda de un especialista en la comisaría para distinguir las diferentes muestras de cada billete.

En primer lugar se seleccionó una muestra de billetes; luego utilizando el sistema, tomamos una imagen digital de cada billete, con dichas imágenes se entrenó una Red Neuronal para que reconozca acertadamente todas las características de la muestra. Luego, se procedió con las pruebas de reconocimiento con otros billetes y con los que no presentan dichas características, y si el software los reconoce acertadamente, por lo tanto se cumplió el objetivo.

3.8. Análisis Estadísticos e Interpretación de los datos

Para la evaluación de cada algoritmo y descriptor de color utilizado en cada prueba, se calculó las siguientes fórmulas estadísticas.

Precisión

Es la proporción de muestras correctamente diagnosticadas del total de muestras evaluadas.

$$P = \frac{|\#Taciertos \cap \#Tmuestras|}{|\#Tmuestras|}$$



Revocación

Es la proporción de muestras correctamente diagnosticadas para cada tipo de las características de los billetes.

$$R = \frac{|\#Taciertos \cap \#Tmuestras|}{|\#Taciertos|}$$

Valor-F

Nos permite corregir el error de la Distancia, en los casos en los que la revocación (R) y la precisión (P) se compensan, dando así, una medida de precisión al algoritmo.

$$F = 2 * \frac{\text{Precision} * \text{Revocacion}}{\text{Precision} + \text{Revocacion}}$$

3.9. Principios éticos

La presente propuesta de investigación se realizó siguiendo los criterios éticos que debe tener todo investigador que se tomarán en cuenta.

La expresión "principios éticos básicos" se refiere a aquellos conceptos generales que sirven como justificación básica para los diversos principios éticos y evaluaciones de las acciones humanas. Entre los principios básicos seguidos en el desarrollo de nuestra propuesta de investigación tenemos:

Responsabilidad social Rectitud

3.10. Criterios de rigor científico

La presente propuesta de investigación se realizó siguiendo los juicios científicos establecidos, estos permiten garantizar la calidad de la propuesta de investigación.

Así, seguimos la coherencia metodológica durante el desarrollo de la propuesta de la investigación, realización apropiada del muestreo de datos, los cuales son al azar para ser totalmente



imparcial en el recojo de datos. Los datos recolectados se basan en el color de los billetes, los cuales son tomados como datos principales.

CAPÍTULO IV:

ANÁLISIS E

INTERPRETACIÓN

DE LOS

RESULTADOS



En este capítulo se procedió a explicar los métodos (propuestos en investigaciones anteriormente desarrolladas) sobre detección de billetes, así como la interpretación de los resultados; seguidamente se presentará el método propuesto, junto a la interpretación de sus resultados. Se utilizaron un total de 72 imágenes de billetes falsos y verdaderos.

4.1. Resultados en tablas y gráficos.

4.1.1. Identificación de las características claves que se tienen en cuenta para identificar billetes:

En este proceso para identificar las características de cada billete, se colocó bajo un máquina o reflector que tenga luz UV y así detectar cada uno de sus partes resaltantes

NUEVOS BILLETES DE S/10,00



Figura 26: Características de billete de S/10.00

1. Fibrillas.
2. Hilo de seguridad.
3. Texto BCRP debajo de la cara del personaje de cada billete.
4. En cada esquina se notan claramente el número de cada billete en este caso 10 soles
5. En la parte trasera aparece 3 veces el texto BCRP10.
6. El aríbalo muestra dos colores fluorescentes impresos a registro.
7. El código se cada billete aparece de color más resaltante.

NUEVOS BILLETES DE S/. 20,00



Figura 27: Características de billete de S/20.00

1. Fibrillas.
2. Hilo de seguridad.
3. Texto BCRP debajo de la cara del personaje de cada billete.
4. En cada esquina se notan claramente el número de cada billete en este caso 20 soles.
5. En la parte trasera aparece 3 veces el texto BCRP20.
6. El Huaco Chimú muestra dos colores fluorescentes impresos a registro.
7. El código se cada billete aparece de color más resaltante.

NUEVOS BILLETES DE S/. 50,00



Figura 28: Características de billete de S/50.00

1. Fibrillas.
2. Hilo de seguridad.
3. Texto BCRP debajo de la cara del personaje de cada billete.
4. En cada esquina se notan claramente el número de cada billete en este caso 50.

5. En la parte trasera aparece 3 veces el texto BCRP50.
6. El Huaco Chavín muestra dos colores fluorescentes impresos a registro.
7. El código se cada billete aparece de color más resaltante.

NUEVOS BILLETES DE S/. 100,00



Figura 29: Características de billete de S/100.00

1. Fibrillas.
2. Hilo de seguridad.
3. Texto BCRP debajo de la cara del personaje de cada billete.
4. En cada esquina se notan claramente el número de cada billete en este caso 100.
5. En la parte trasera aparece 3 veces el texto BCRP100.
6. El ceramio muestra dos colores fluorescentes impresos a registro.
7. El código se cada billete aparece de color más resaltante.

NUEVOS BILLETES DE S/. 200,00



Figura 30: Características de billete de S/200.00

1. Fibrillas.
2. Hilo de seguridad.

3. Texto BCRP debajo de la cara del personaje de cada billete.
4. En cada esquina se notan claramente el número de cada billete en este caso 200.
5. En la parte trasera aparece 3 veces el texto BCRP200.
6. La imagen muestra dos colores fluorescentes impresos a registro.
7. El código se cada billete aparece de color más resaltante.

4.1.2. Determinación del Pre – Procesamiento de Imágenes

4.1.2.1. Métodos de descriptores por color:

Los siguientes resultados muestran cuanto tiempo se invierte en pre procesar las imágenes de los billetes verdaderos y falsos. La siguiente tabla nos muestra el resultado de tiempos de procesamiento obtenido en una imagen, correspondiente a los cuatro métodos implementados:

Métodos por color	Tiempo Promedio
Blanco y Negro	50''
HSV	50''
Rojo	54''
Verde	45''
Azul	62''

Tabla 1: Tiempos de procesamientos promedios de 72 imágenes, obtenidos con los métodos Blanco y negro, HSV, Rojo, Verde y Azul. Las cifras se muestran en milisegundos(ms).



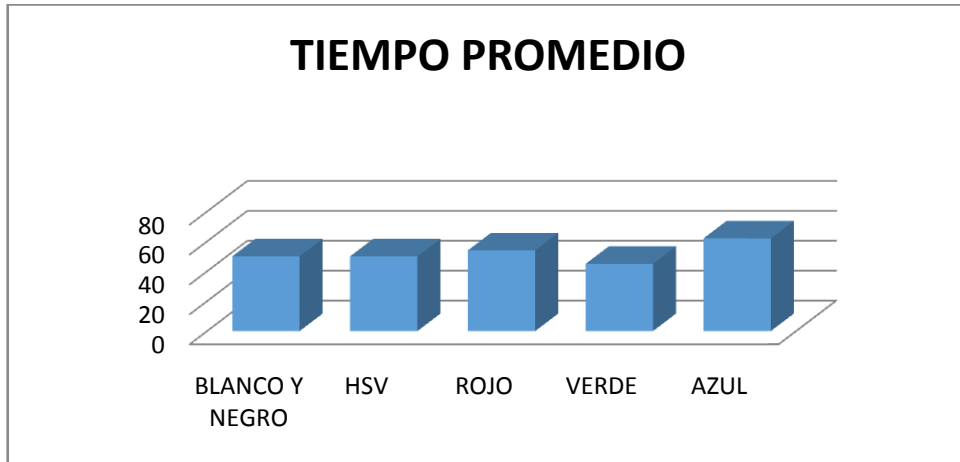


Grafico 1: Tiempos de procesamientos promedios de 72 imágenes, obtenidos con los métodos Blanco y negro, HSV, Rojo, Verde y Azul. Las cifras se muestran en milisegundos(ms).

a) Blanco y Negro: Primero la imagen debe convertirse en un formato más apropiado para extraer las características.



Figura 31: Billeto con filtro blanco y negro

b) HSV: El modelo HSV (Matiz, Saturación, Valor), también llamado HSB define un modelo de color en términos de sus componentes.

- Hue (H) es el tono de color (por ejemplo verde, rojo, morado, etc.)
- Saturation (S) es la intensidad de esta tonalidad. Cuanta menos saturación, más gris es el color.
- Value (V) es la luminosidad del color. En ocasiones también se escribe Luminosity(L)





Figura 32: Billeto con filtro HSV

c) RGB: La descripción RGB (de sus siglas en inglés Red, Green, Blue; "rojo, verde, azul") de un color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul.

- Red



Figura 33: Billeto con filtro rojo

En esta imagen observamos el método aplicado RGB, que es color rojo observamos que en algunas partes podemos llegar a la conclusión que refleja ciertos caracteres (Fibrillas, BCRP) explicados en el objetivo número uno, que se encuentra en la parte superior.



- Green



Figura 34: Billete con filtro verde

En esta imagen observamos el método aplicado RGB, que es color verde observamos que en algunas partes podemos llegar a la conclusión que refleja ciertos caracteres (Fibrillas, Código de cada billete, Numero 100) explicados en el objetivo número uno, que se encuentra en la parte superior.

- Blue



Figura 35: Billete con filtro azul

En esta imagen observamos el método aplicado RGB, que es color azul observamos que en algunas partes podemos llegar a la conclusión que refleja ciertos caracteres (Fibrillas) explicados en el objetivo número uno, que se encuentra en la parte superior.



4.1.2.2. Histograma de color:

En los campos procesamiento de imagen y fotografía, un histograma de color es una representación anterior de la distribución del color en una imagen. En las imágenes digitales, un histograma de color representa el número de píxeles que tienen colores en cada una de las listas fijas de rangos de colores, que se extienden sobre el espacio de color de la imagen, es decir, el conjunto de todos los posibles colores.

El histograma de color puede ser construido para cualquier tipo de espacio de color, aun cuando el término es usado más frecuentemente en espacios tridimensionales como RGB o HSV.

- Rojo, Verde, Azul

Sólo aparecen en capas de imágenes RGB. Muestran la distribución de los niveles de intensidad de los canales rojo, verde y azul respectivamente.

- Alfa

Muestra la distribución de los niveles de opacidad. Si la capa es completamente transparente (alfa = 0) o completamente opaca (alfa = 255).

4.1.2.3. Detección Automática

Los siguientes resultados muestran cuanto tiempo se invierte en reconocer las imágenes digitales de billetes falsos y verdaderos, además de el grado de precisión de backpropagation.

La siguiente tabla nos muestra el resultado de tiempos de reconocimiento obtenido en una imagen, correspondiente a los algoritmos de detección automática:

Algoritmo de Segmentación	Tiempo Promedio
Backpropagation-B/N	30''
Backpropagation-HSV	22''
Backpropagation-Rojo	25''
Backpropagation-Verde	20''
Backpropagation-Azul	23''

Tabla 2: Tiempo de reconocimiento con backpropagation de cada filtro de color, obtenidos con los algoritmos de detección automática. Las cifras se muestran en milisegundos (ms).

Backpropagation-Verde:

Backpropagation 72 Billetes	
Reconoció	42
No reconoció	30

Tabla 3: Resultados del total de billetes por backpropagation-verde

Precisión = Cantidad de muestras acertadas/cantidad de muestras acertadas + falsos positivos.

$$\text{Precisión} = 42/42+25 = 0.63$$

Revocación = Cantidad de muestras acertadas/cantidad de muestras



acertadas + falsos negativos.

$$\text{Revocación} = 42/42+17 = 0.71$$

Media = 2. Precisión X Revocación/ Precisión + Revocación

$$\text{Media} = 2 (0.63 \times 0.71) / 0.63 + 0.71 = 0.67$$

Backpropagation-HSV:

Backpropagation 72 Billetes	
Reconoció	34
No reconoció	38

Tabla 4: : Resultados del total de billetes por backpropagation-HSV

Precisión = Cantidad de muestras acertadas/cantidad de muestras acertadas + falsos positivos.

$$\text{Precisión} = 34/34+25 = 0.58$$

Revocación = Cantidad de muestras acertadas/cantidad de muestras acertadas + falsos negativos.

$$\text{Revocación} = 34/34+17 = 0.67$$

Media = 2. Precisión X Revocación/ Precisión + Revocación

$$\text{Media} = 2 (0.58 \times 0.67) / 0.58 + 0.67 = 0.62$$



4.2. Discusión de resultados

4.2.1. Discusión 1

Vero, J.O.(2013), en su investigación utiliza algoritmos SURF, SIFT y ORB que tuvo como conclusión una confusión en billetes de 50 y 100 ya que se parecían. En este proyecto de investigación hemos usado varios algoritmos de color(RGB, HSV, etc.) pero principalmente nos hemos basado en puntos claves y tener un área exacta de donde extraer los caracteres de color y evitar las confusiones de billetes.

4.2.2. Discusión 2

Guzmán Bárcena, Sheylla. (2012), en su investigación utiliza Proceso de escaneado, segmentación y normalización, cálculo de patrones, etc. En este proyecto de investigación se realizó la adquisición de imágenes de billetes, se aplicó métodos por color, histogramas de color RGB para la extracción de caracteres, y finalmente esos valores se entrenó en la red neuronal.

CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN

DIAGRAMA DEL ENTRENAMIENTO

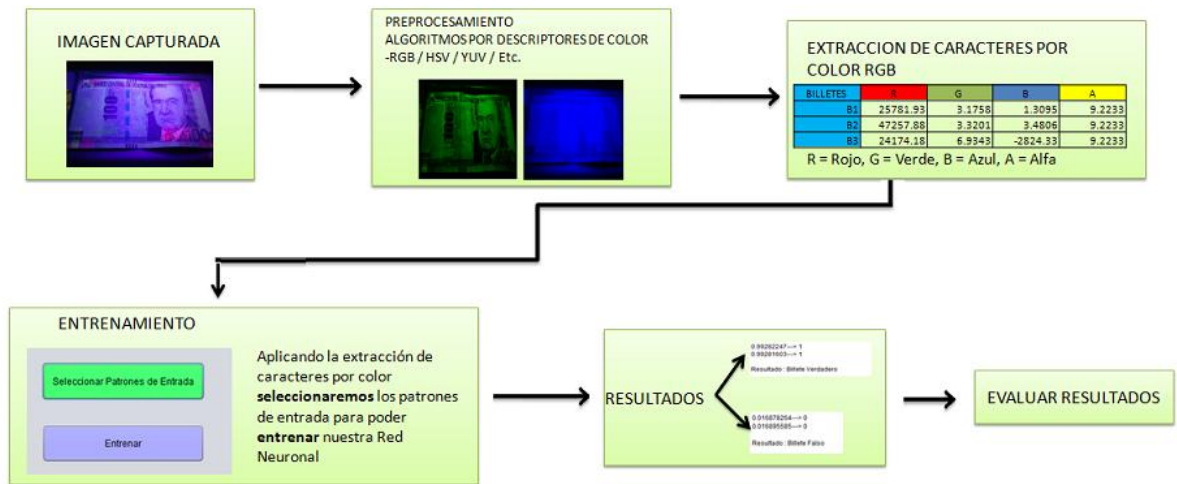


Figura 36: Diagrama del entrenamiento

Se recibe como entrada la imagen digital con un Smartphone huawei P8 lite, luego aplicamos los algoritmos por color que en este caso es RGB-Verde y se extraerán las características por color RGB. Terminada la extracción se identifican patrones que vienen hacer 4 entradas (rojo, verde, azul y alfa), se seleccionan los patrones de entradas para poder entrenar la Red Neuronal para luego pasar a la clasificación backpropagation y se compararan los resultados obteniendo 2 salidas verdadero o falso.

5.1. Adquisición de Imágenes

La adquisición de las imágenes es uno de los pasos más importantes en un sistema de visión artificial, ya que la calidad de las imágenes a analizar debe tener una calidad óptima, que no afecte al rendimiento del sistema.

A continuación, las imágenes son digitalizadas mediante la toma de fotografías con una cámara de celular Huawei P8 Lite de 13 Pixeles tomadas desde una máquina con luz ultravioleta(UV)




MODELO DE SMARTPHONE	IMAGEN	BENEFICIO	RESOLUCIÓN DE CÁMARA
HUAWEI P8 LITE		<p>En las imágenes reflejan:</p> <ul style="list-style-type: none"> - Código de seguridad - Fibrillas - En la esquina el numero del billete - Hilo de seguridad 	Tiene una resolución de 13mpx lo cual hace que sus principales características se noten mejor.
LG G2 MINI		<p>En las imágenes reflejan:</p> <ul style="list-style-type: none"> - Código de seguridad - Fibrillas - Hilo de seguridad 	Tiene una resolución de 8mpx lo cual las imágenes tomadas no reflejan claramente sus características del billete.
MOTOROLA G PRIMERA GENERACIÓN		<p>En las imágenes reflejan:</p> <ul style="list-style-type: none"> - Hilo de seguridad 	Tiene una resolución de 5mpx lo cual es mas bajo la claridad de sus características.

Tabla 5: Comparación para el uso del celular

El uso del Smartphone es porque lo tengo a mi disposición en cualquier momento que lo necesito para poder tomar las imágenes digitales de los billetes, y no descarto cualquier máquina para tomar fotos.

Si podemos ver el cuadro comparativo he tenido varios teléfonos pero cual mejor resalta sus características principales es cual especifique que viene hacer el Huawei P8 Lite con cámara de 13 Pixeles.





Figura 37: Máquina con Luz Ultravioleta Instrumento empleado para la captura de las imágenes digitales.

El uso de este detector de billetes, lo he implementado por que cuando lo colocamos a la luz UV resalta sus características principales omitidas por el BCRP, porque si solo ponemos el billete sin el detector no se notara absolutamente nada.

5.2. Pre-procesamiento

En la etapa de pre procesamiento se pretende corregir defectos que puedan producirse durante la captura y digitalización de las imágenes de los billetes. Si bien se ha tomado las precauciones para obtener una muestra lo más limpia posible.

ALGORITMOS	TIEMPO	USO	PRIORIDAD
RGB(Red, Green, Blue)	El tiempo de respuesta es el más apropiado, a comparación de otros métodos por color	Es el más usado por descriptores de color, ya que su valores son los colores rojo, verde y azul que encontramos un mayor reflejo de sus caracteres de los billetes	Se adecua a un porcentaje mayor que otros algoritmos, ya que también usaremos este método, para extraer los caracteres de cada billete.
HSV(<u>Matiz</u>, <u>Saturación</u>, <u>Valor</u>)	Su tiempo de ejecución es favorable para nuestra investigación	Refleja mejor la noción intuitiva de la saturación y la luminancia como dos parámetros independientes, y de esta manera no se logra ver muy bien los caracteres de los billetes.	
CMYK(Cyan, Magenta, Yellow y Key)	Es el tercer mejor algoritmo que tiene mejor tiempo de respuesta	Tiene un buen resultado con mayor contraste, pero a la hora de ejecutarlo algunos colores no se pueden apreciar bien como el color azul.	Su uso hace que algunas imperfecciones de compresión se oculten más eficientemente a la percepción humana que usando una representación RGB directa.
YUV o YCrCb	El tiempo de procesamiento de imágenes en YUV no es favorable para nuestra investigación.	Su uso se hace más en videos, que en imágenes.	
RYB	El tiempo de procesamiento de imágenes en RYB no es muy favorable para nuestra investigación.		

Tabla 6: Comparación de algoritmos por color



5.2.1. Blanco y Negro

Se aplicó el método por color Blanco y negro de OpenCV sobre la imagen original se obtuvo el siguiente resultado:



Figura 38: Imagen con método por color Blanco y negro.

```
Mat src = Funciones.cargar(ruta, false, false, 0);
Imgproc.cvtColor(src, src, Imgproc.COLOR_BGR2GRAY);
Funciones.mostrar(src, 0, 0, "Blanco y negro");
```

Método Blanco y negro, código completo en Anexo 2

5.2.2. HSV

Se aplicó el método por color HSV de OpenCV sobre la imagen original se obtuvo el siguiente resultado:



Figura 39: Imagen con método por color HSV.

```
File input = new File(ruta);
BufferedImage image = ImageIO.read(input);
byte[] data = ((DataBufferByte) image.getRaster().getDataBuffer()).getData();
Mat mat = new Mat(image.getHeight(), image.getWidth(), CvType.CV_8UC3);
mat.put(0, 0, data);

Mat mat1 = new Mat(image.getHeight(), image.getWidth(), CvType.CV_8UC3);
Imgproc.cvtColor(mat, mat1, Imgproc.COLOR_RGB2HSV);

byte[] data1 = new byte[mat1.rows() * mat1.cols() * (int) (mat1.elemSize())];
mat1.get(0, 0, data1);
BufferedImage image1 = new BufferedImage(mat1.cols(), mat1.rows(), 5);
image1.getRaster().setDataElements(0, 0, mat1.cols(), mat1.rows(), data1);
Funciones.mostrar(mat1, 0, 0, "HSV");
```

Método HSV, código completo en Anexo 3

5.2.3. RGB-Rojo

Se aplicó el método por color RGB-Rojo de OpenCV sobre la imagen original se obtuvo el siguiente resultado:



Figura 40: Imagen con método por color RGB-Rojo.

```
Canales cn = new Canales(ruta);
BufferedImage img = cn.getRedImage();
Mat xd = bufferedImageToMat(img);
Funciones.mostrar(xd, 0, 0, "Rojo");
```

Método Rojo, código completo en Anexo 4

5.2.4. RGB-Verde

Se aplicó el método por color RGB-Verde de OpenCV sobre la imagen original se obtuvo el siguiente resultado:



Figura 41: Imagen con método por color RGB-Verde.

```
Canales cn = new Canales(ruta);
BufferedImage img = cn.getGreenImage();
Mat xd = bufferedImageToMat(img);
Funciones.mostrar(xd, 0, 0, "Verde");
```

Método Verde, código completo en Anexo 5

5.2.5. RGB-Azul

Se aplicó el método por color RGB-Azul de OpenCV sobre la imagen original se obtuvo el siguiente resultado:



Figura 42: Imagen con método por color RGB-Azul

```
Canales cn = new Canales(ruta);
BufferedImage img = cn.getBlueImage();
Mat xd = bufferedImageToMat(img);
Funciones.mostrar(xd, 0, 0, "Azul");
```

Método Azul, código completo en Anexo 6

5.3. Extracción de Caracteres por color

5.3.1. Extracción Vector de Características de Histograma de Imagen

IMAGEN Nro	VECTOR DE CARACTERÍSTICAS			
	R	G	B	A
1	1156.0700000000000	330758.64000000000	18440.660000000000	43657.720000000000
	1045.8400000000000	291533.60000000000	14800.790000000000	36034.220000000000
	1252.0700000000000	373485.22000000000	11617.110000000000	56814.251.0;1.0
	1199.8000000000000	459196.28000000000	31196.200000000000	78208.580000000000
	1147.3600000000000	307832.28000000000	9121.5800000000000	30816.130000000000
2	2070.8700000000000	876155.42000000000	-71519.75;201148.20000000000	1.0;1.0
	1361.0400000000000	570788.49000000000	17234.200000000000	72651.649999999999
	2219.1300000000000	960859.25;-45612.79000000000	221479.160000000000	1.0;1.0
	2359.2300000000000	1098093.57000000000	-113864.120000000000	318223.870000000000
	3167.5900000000000	1858119.73000000000	-256968.660000000000	944137.189999999999
3	2739.1500000000000	1578172.41000000000	19316.1600000000000	642779.390000000000
	3140.8700000000000	2000336.39000000000	-105357.400000000000	796959.550000000000
	2234.5300000000000	795059.290000000000	-19247.3800000000000	189230.350000000000
	2099.3800000000000	77508.640000000000	-3020.22000000000000	131305.380000000000
	2506.0100000000000	784885.380000000000	-62654.970000000000	3400000000.1.0;1.0
4	2652.9700000000000	1512393.67000000000	-11276.3000000000000	382828.480000000000
	2393.2600000000000	1450278.31000000000	-23670.8600000000000	350211.080000000000
	5196.0200000000000	17634778.19000000000	5288561.600000000000	62424064.1300000000
	6168.5100000000000	55999581.96000000000	45797824.080000000000	92233720.3700000000
	4938.3300000000000	4412128.860000000000	-930797.430000000000	5275179.1700000000
5	3076.3100000000000	1776411.42000000000	-139980.560000000000	659534.340000000000
	3334.1300000000000	3962352;563195.430000000001	3660709.190000000000	1.0;1.0
	3246.8800000000000	4316779.330000000000	627900.8500000000000	4168047.200000000000
	5196.0200000000000	17634778.19000000000	5288561.600000000000	62424064.1300000000
	6828.5100000000000	55999581.96000000000	45797824.080000000000	92233720.3700000000

Figura 43: Extracción Vector de Características de Histograma de Imagen

```
protected static int[][] histogram(BufferedImage image, List<Color> channel) {
    Color auxColor;
    int histogramReturn[][] = new int[5][256];
    for (int i = 0; i < image.getWidth(); i++) {
        for (int j = 0; j < image.getHeight(); j++) {
            auxColor = new Color(image.getRGB(i, j));
            histogramReturn[0][auxColor.getRed()] += 1;
            histogramReturn[1][auxColor.getGreen()] += 1;
            histogramReturn[2][auxColor.getBlue()] += 1;
            histogramReturn[3][auxColor.getAlpha()] += 1;
            histogramReturn[4][calculateAverage(auxColor)] += 1;
        }
    }
    return histogramReturn;
}
```

Extracción Vector de Características de Histograma de Imagen, código completo Anexo 7

Se extrajo los caracteres por color RGB, como podemos ver en la Figura 43 para ver el número de vectores que usaremos para entrenar la red neuronal. En esta imagen se tomó solo como muestra 5 billetes, con sus respectivos valores:

- R = Rojo
- G= Verde
- B = Azul
- A = Alfa



5.4. Clasificación:

Para la clasificación de las muestras obtenidas en base a las características extraídas por los descriptores se ha escogido Redes Neuronales.

5.4.1. Red Neuronal Blackpropagation

Con una estructura de 1 capa de entrada (4 neuronas), 1 capa oculta (50 neuronas) y 1 capa de salida (2 neuronas) la cual se entrena con los patrones binarizados y las salidas deseadas de cada imagen.

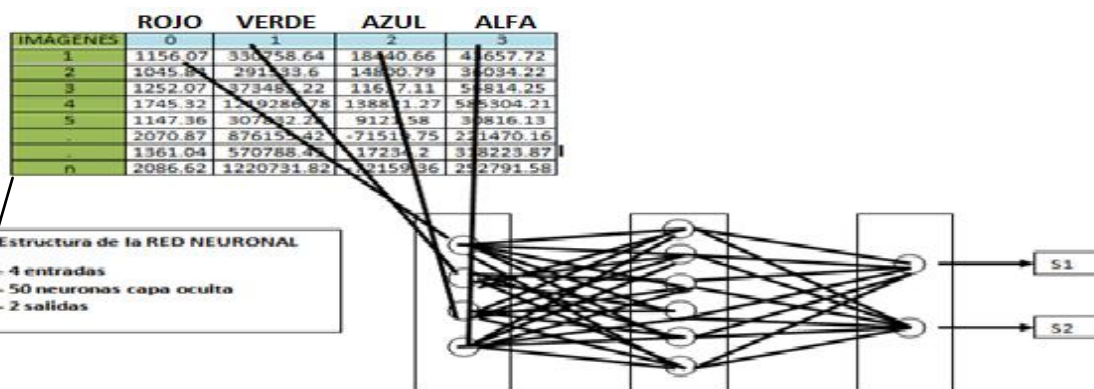


Figura 44: Diseño de la Red Neuronal

IMAGENES: BILLETES PARA LA EXTRACCIÓN DE CARACTERÍSTICAS

```

redNeuronal
├── AdminArrayDatosEntrenamiento.java
├── AdminArrayDeCapas.java
├── AdminArrayDeNeuronas.java
├── AdminArrayDeRedesNeuronales.java
├── AdminConexionesConNeuronas.java
├── AplicacionErrores.java
├── Capa.java
├── CreadorDeRedBackPropagacion.java
├── DatosDeEntrenamiento.java
├── NeuronArtificial.java
├── RedNeuronal.java
├── SomaDeNeurona.java
└── ZProcedimientos.java
    
```

Figura 45: Clases Red Neuronal

Clases Pertenecientes a la Red Neuronal Anexo 8

SALIDA DESEADA				DIAGNOSTICO
1	0	1	0	BILLETE VERDADERO
0	0	0	0	BILLETE FALSO

Figura 46: Salidas Deseadas



CAPÍTULO VI:

CONCLUSIONES Y

RECOMENDACIONES



6.1 Conclusiones

- El investigador logró realizar un pre-conocimiento de detectar billetes en 72 imágenes digitales. Para la identificación de las características de cada billete se concluyó que cuentan con varias formas de reconocer un billete falso como las técnicas TOQUE, MIRE Y GIRE, y también mediante la luz UV que en este proyecto resaltan 7 colores que son: Fibrillas, hilo de seguridad, texto BCRP debajo de la cara del personaje de cada billete, cada esquina se notan claramente el número de billete, en la parte trasera aparece 3 veces el texto BCRP, el símbolo de cada billete que lo representa muestra dos colores y por último el código que aparece en cada billete.
- Para el pre-procesamiento de la imagen se aplicaron los métodos por color Blanco y negro, HSV, RGB-Rojo, RGB-Verde y RGB-Azul, que posteriormente usaremos de estos 5 métodos el RGB-Verde, ya que resaltan sus principales características de cada billete y obtenemos un mejor tiempo de demora de pre-procesamiento de 15 milisegundos.
- Para la detección automática de imágenes se aplicó los algoritmos backpropagation-Verde y backpropagation-HSV donde se obtuvo un tiempo de demora de procesamiento de 20 milisegundos con precisión del 63% para el algoritmo backpropagation-Verde y un tiempo de 36 milisegundos con precisión del 58% para el algoritmo backpropagation-HSV, con error medio cuadrático 0.19235516.
- Se logró implementar el algoritmo de detección automática backpropagation en el lenguaje de programación Java con el IDE Netbeans 8.0.



6.2 Recomendaciones

- a) Al tomar las imágenes digitales deben tenerse algunas condiciones como por ejemplo tener un ambiente controlado (iluminación, distancia, temperatura fija, evitar el polvo) y que se capture con una cámara de alta resolución con el objetivo de no tener inconvenientes a la hora de hacer el procesamiento.

- b) Realizarlo en plataforma móvil, ya que de esta manera se podría realizar la identificación de billetes en tiempo real.

REFERENCIAS

- Ahmed Ali and Mirfa Manzoor. (2013). *Recognition System for Pakistani Paper Currency. COMSATS Institute of Information Technology, Abbottabad.*
- Arceo Moheno G, De los Santos Torres G, Rivera Reyes A1 (2012). *Uso de agentes inteligentes en cajeros automáticos de la Comisión Federal de Electricidad en Villahermosa (México)*
- Boullosa Gracia, O. (2011). *Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas.*
- Cardozo, J. A. (II-2014). *Aplicación Para La Identificación Automática De Billetes Falsos En Android. Bolivia: UNIVERSIDAD MAYOR DE SAN SIMÓN.*
- Faiz M. Hasanuzzaman, Xiaodong Yang, and YingLi Tian (2011). *Reconocimiento de billetes por SURF Características Robusto y eficaz basada en componentes. El City College de Nueva York, CUNY*
- GUZMÁN BÁRCENA, SHEYLLA. (2012). *DETECTOR DE BILLETES PARA PERSONAS CON DISCAPACIDAD VISUAL. Perú: UNIVERSIDAD TECNOLÓGICA DEL PERÚ.*
- Heredia Favieri, N. M. (2015). *Reconocimiento de Objetos en Imágenes. Argentina: Tesis para Licenciatura en Ciencias de la Computación.*
- Jarrett Chambers. (2012). *Digital Currency Forensics. Auckland University of Technology, School of Computing and Mathematical Sciences.*
- Matadamaz Ortiz, C. (2013). *Aplicacion de las redes neuronales pulsantes en el reconocimiento de patrones y analisis de imagenes. Mexico: Instituto Politecnico Nacional.*
- MIT. (2014). *Nanocristales que utilizan tu smartphone para detectar billetes falsos Instituto tecnológico de Massachusetts – Estados Unidos.*
- Ponce Cruz, P. (2010). *Inteligencia Artificial con Aplicaciones a la*



Ingeniería. Alfomega.

Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.

Vera, J. O. (2013.). Detección de billetes con OpenCV en Android. Argentina.

ANEXOS

ANEXO 1

FORMULARIO ACTA DE OBSERVACION

ACTA DE REUNION ____/____/____

Proyecto	ANÁLISIS COMPARATIVO DE ALGORITMOS DE RECONOCIMIENTO DE IMÁGENES POR DESCRIPTORES DE COLOR PARA LA IDENTIFICACIÓN DE BILLETES	Observador	Estela Zumaeta Jorge Luis
Lugar:		Situación:	
Objetivo de la observación	Determinar los tipos de billetes, la interpretación en cada una de ellas para obtener las normas para poder clasificar los distintos tipos de billetes entre falsos y verdaderos.		
Instrucciones	Las preguntas pueden variar al momento de la entrevista, dependiendo como fluye la conversación		
TEMA	PREGUNTAS		
Clasificación de las características principales en base a la interpretación de los billetes verdaderos y falsos.	¿Qué características se observan en los billetes?		
	¿Qué tipo de características observadas previamente?		
	¿Qué patrones determinan los caracteres de los billetes?		
Resumen	Gracias a la información recolectada se determinara las normas para clasificar los billetes, cuales son los problemas más frecuentes, sus respectivas características.		



ANEXO 2

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (ruta != "") {
        long tiempoInicio = System.currentTimeMillis();
        Thread t1 = new Thread();
        t1.start();

        Mat src = Funciones.cargar(ruta, false, false, 0);
        Imgproc.cvtColor(src, src, Imgproc.COLOR_BGR2GRAY);
        Funciones.mostrar(src, 0, 0, "Blanco y negro");

        long totalTiempo = System.currentTimeMillis() - tiempoInicio;
        System.out.println("El tiempo de demora es : " + totalTiempo + " miliseg");
    } else {
        JOptionPane.showMessageDialog(null, "SELECCIONE UNA IMAGEN", "Imagen", JOptionPane.WARNING_MESSAGE);
    }
}
```

Código completo filtro por color blanco y negro

ANEXO 3

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    if (ruta != "") {
        long tiempoInicio = System.currentTimeMillis();
        try {
            // TODO add your handling code here:
            File input = new File(ruta);
            BufferedImage image = ImageIO.read(input);
            byte[] data = ((DataBufferByte) image.getRaster().getDataBuffer()).getData();
            Mat mat = new Mat(image.getHeight(), image.getWidth(), CvType.CV_8UC3);
            mat.put(0, 0, data);

            Mat mat1 = new Mat(image.getHeight(), image.getWidth(), CvType.CV_8UC3);
            Imgproc.cvtColor(mat, mat1, Imgproc.COLOR_RGB2HSV);

            byte[] data1 = new byte[mat1.rows() * mat1.cols() * (int) (mat1.elemSize())];
            mat1.get(0, 0, data1);
            BufferedImage image1 = new BufferedImage(mat1.cols(), mat1.rows(), 5);
            image1.getRaster().setDataElements(0, 0, mat1.cols(), mat1.rows(), data1);
            Funciones.mostrar(mat1, 0, 0, "HSV");

            long totalTiempo = System.currentTimeMillis() - tiempoInicio;
            System.out.println("El tiempo de demora es : " + totalTiempo + " miliseg");
        } catch (Exception e) {
        }
    } else {
        JOptionPane.showMessageDialog(null, "SELECCIONE UNA IMAGEN", "Imagen", JOptionPane.WARNING_MESSAGE);
    }
}
```

Código completo filtro por colorHSV



ANEXO 4

```
private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (ruta != "") {
        long tiempoInicio = System.currentTimeMillis();
        try {
            Canales cn = new Canales(ruta);
            BufferedImage img = cn.getRedImage();
            Mat xd = bufferedImageToMat(img);
            Funciones.mostrar(xd, 0, 0, "Rojo");

            long totalTiempo = System.currentTimeMillis() - tiempoInicio;
            System.out.println("El tiempo de demora es : " + totalTiempo + " miliseg");
        } catch (IOException ex) {
            Logger.getLogger(frmPrincipal.class.getName()).log(Level.SEVERE, null, ex);
        }
    } else {
        JOptionPane.showMessageDialog(null, "SELECCIONE UNA IMAGEN", "Imagen", JOptionPane.WARNING_MESSAGE);
    }
}
}
```

Código completo filtro por color RGB-Rojo

ANEXO 5

```
private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (ruta != "") {
        long tiempoInicio = System.currentTimeMillis();
        try {
            Canales cn = new Canales(ruta);
            BufferedImage img = cn.getGreenImage();
            Mat xd = bufferedImageToMat(img);
            Funciones.mostrar(xd, 0, 0, "Verde");

            long totalTiempo = System.currentTimeMillis() - tiempoInicio;
            System.out.println("El tiempo de demora es : " + totalTiempo + " miliseg");
        } catch (IOException ex) {
            Logger.getLogger(frmPrincipal.class.getName()).log(Level.SEVERE, null, ex);
        }
    } else {
        JOptionPane.showMessageDialog(null, "SELECCIONE UNA IMAGEN", "Imagen", JOptionPane.WARNING_MESSAGE);
    }
}
}
```

Código completo filtro por color RGB-Verde



ANEXO 6

```
private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (ruta != "") {
        long tiempoInicio = System.currentTimeMillis();
        try {
            Canales cn = new Canales(ruta);
            BufferedImage img = cn.getBlueImage();
            Mat xd = bufferedImageToMat(img);
            Funciones.mostrar(xd, 0, 0, "Azul");

            long totalTiempo = System.currentTimeMillis() - tiempoInicio;
            System.out.println("El tiempo de demora es : " + totalTiempo + " miliseg");
        } catch (IOException ex) {
            Logger.getLogger(frmPrincipal.class.getName()).log(Level.SEVERE, null, ex);
        }
    } else {
        JOptionPane.showMessageDialog(null, "SELECCIONE UNA IMAGEN", "Imagen", JOptionPane.WARNING_MESSAGE);
    }
}
}
```

Código completo filtro por color RGB-Azul

ANEXO 7

```
protected static int[][] histogram(BufferedImage image, listCanales chanel) {
    Color auxColor;
    int histogramReturn[][] = new int[5][256];
    for (int i = 0; i < image.getWidth(); i++) {
        for (int j = 0; j < image.getHeight(); j++) {
            auxColor = new Color(image.getRGB(i, j));
            histogramReturn[0][auxColor.getRed()] += 1;
            histogramReturn[1][auxColor.getGreen()] += 1;
            histogramReturn[2][auxColor.getBlue()] += 1;
            histogramReturn[3][auxColor.getAlpha()] += 1;
            histogramReturn[4][calculateAverage(auxColor)] += 1;
        }
    }
    return histogramReturn;
}

public int[] histogramRed(BufferedImage image) {
    int[] histRed;
    histRed = copyHistogram(this.histogram(image, listCanales.red), 0);
    calcularMaxMinValor(histRed);
    return histRed;
}

public int[] histogramGreen(BufferedImage image) {
    int[] histGreen;
    histGreen = copyHistogram(this.histogram(image, listCanales.green), 1);
    calcularMaxMinValor(histGreen);
    return histGreen;
}

public int[] histogramBlue(BufferedImage image) {
    int[] histBlue;
    histBlue = copyHistogram(this.histogram(image, listCanales.blue), 2);
    calcularMaxMinValor(histBlue);
    return histBlue;
}

public int[] histogramAlpha(BufferedImage image) {
    int[] histAlpha;
    histAlpha = copyHistogram(this.histogram(image, listCanales.alpha), 3);
    calcularMaxMinValor(histAlpha);
    return histAlpha;
}

public int[] histogramGrayscale(BufferedImage image) {
    int[] histGrayscale;
    histGrayscale = copyHistogram(this.histogram(image, listCanales.grayscale), 4);
    calcularMaxMinValor(histGrayscale);
    return histGrayscale;
}
}
```

Extracción Vector de Características de Histograma de Imagen



ANEXO 8

```
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.Formatter;
import java.util.Iterator;

public class RedNeuronal {

    private AdminArrayDeCapas capasRed = new AdminArrayDeCapas();

    public AdminArrayDeCapas capas() {...3 lines }

    public Capa CapaDeEntrada() throws Exception {...6 lines }

    public Capa CapaDeSalida() throws Exception {...6 lines }

    public void ConectarDosNeuronas(NeuronaArtificial NeuronasOrigen, NeuronasArtificial NeuronasDestino, float Peso) thro

    public void ConectarDosNeuronas(NeuronaArtificial NeuronasOrigen, NeuronasArtificial NeuronasDestino) throws Exception

    public void ConectarCapas(Capa Capa1, Capa Capa2) throws Exception {...19 lines }

    public void ConectarCapas() throws Exception {...10 lines }

    public void EntrenarRed(DatosDeEntrenamiento tablaEntrenamiento) throws Exception {...98 lines }

    public ArrayList EjecutarRed(ArrayList Entradas) throws Exception {...41 lines }

    public ArrayList ObtenerSalidaRed() throws Exception {...10 lines }

}
```

```
public class AdminArrayDatosEntrenamiento extends java.util.ArrayList
{
    public final DatosDeEntrenamiento Add(DatosDeEntrenamiento objTipoEntrenamiento)
    {...4 lines }

    public final DatosDeEntrenamiento Add()
    {...5 lines }

    public final void Insert(int index, DatosDeEntrenamiento objTipoEntrenamiento)
    {...3 lines }

    public final void Remove(DatosDeEntrenamiento objTipoEntrenamiento)
    {...3 lines }

    public final DatosDeEntrenamiento getItem(int index)
    {...3 lines }

    public final void setItem(int index, DatosDeEntrenamiento Valor)
    {...3 lines }

}
```

Código que permite administrar los datos de entrenamiento de la red



```
public class AdminArrayDeCapas extends java.util.ArrayList
{
    public final Capa Add(Capa objTipoCapa)
    {...4 lines }

    public final Capa Add()
    {...5 lines }

    public final void Insert(int index, Capa objTipoCapa)
    {...3 lines }

    public final void Remove(Capa obj)
    {...3 lines }

    public final Capa getItem(int index)
    {...3 lines }
    public final void setItem(int index, Capa Valor)
    {...3 lines }
}
```

Código que permite administrar las capas de la red neuronal

```
public class AdminArrayDeNeuronas extends java.util.ArrayList<NeuronaArtificial>
{
    public final NeuronaArtificial Add(NeuronaArtificial objTipoNeurona)
    {...4 lines }

    public final NeuronaArtificial Add()
    {...7 lines }

    public final void Insert(int index, NeuronaArtificial objTipoNeurona)
    {...3 lines }

    public final void Remove(NeuronaArtificial objTipoNeurona)
    {...3 lines }

    public final NeuronaArtificial getItem(int index)
    {...3 lines }
    public final void setItem(int index, NeuronaArtificial Valor)
    {...3 lines }
}
```

Código que permite administrar las neuronas en cada capa

```
public class AdminConeccionesConNeuronas extends java.util.Hashtable {
    public final NeuronaArtificial Add(NeuronaArtificial entrada, float peso) {...4 lines }
    public final NeuronaArtificial Add(NeuronaArtificial entrada) {...4 lines }
    public final void Remove(NeuronaArtificial obj) {...3 lines }
    public final java.util.Collection Neurona() {...3 lines }
    public final float getpeso(NeuronaArtificial objetoTipoNeurona) {...3 lines }
    public final void setpeso(NeuronaArtificial objetoTipoNeurona, float valor) {...4 lines }

    public class AdminPesoConeccion {
        public float pesoConeccion;
        public AdminPesoConeccion(float peso) {...3 lines }
    }
}
```

Código que permite conectar las neuronas de la red




```
public class AplicacErrores extends RuntimeException {
    public AplicacErrores(String Mensaje, Exception exp) {
        super(Mensaje, exp);
    }
}
```

Código que permite mostrar los errores de la aplicación

```
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Random;

public final class ZProcedimientos {
    public static final double TASA_APRENDIZAJE = 0.5;

    public static float ObtenerValorAleatorio() {
        Random rnd = new Random();
        return (float) (rnd.nextFloat() * 2 - 1);
    }
}
```

Código que permite mostrar la tasa de aprendizaje de la aplicación

```
public class SomaDeNeurona {
    public float CalcularDelta(float salida, float FactorDeError) {...8 lines }

    public float FuncionActivacion(float valor) {...8 lines }

    public float CalcularNETI(AdminConeccionesConNeuronas NeuronasConectadasDeAtras, float bias) {...16 lines }
    float tasa_aprendizaje = new Float(0.5);

    public final float CalcularNuevoUmbral(float Umbral, float delta) {...9 lines }

    public final void ActualizarPesos(AdminConeccionesConNeuronas NeuronasConectadasDeAtras, float ValError) {...12 lines }
}
```

Código que permite calcular la neto, la función de activación , actualizar los pesos y calcular el nuevo umbral

```
public class CreadorDeRedBackPropagacion {
    public final RedNeuronal CrearRedNeuronal(java.util.ArrayList CantidadCapasConNeuronas) throws Exception {...22 lines }

    public final RedNeuronal CrearRedNeuronal(long NeuronasEntrada, long NeuronasSalida) throws Exception {...10 lines }
}
```

Código que permite crear la red neuronal



```

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.Formatter;
import java.util.Iterator;

public class RedNeuronal {

    private AdminArrayDeCapas capasRed = new AdminArrayDeCapas();

    public AdminArrayDeCapas capas() { ...3 lines }

    public Capa CapaDeEntrada() throws Exception { ...6 lines }

    public Capa CapaDeSalida() throws Exception { ...6 lines }

    public void ConectarDosNeuronas(NeuronaArtificial NeuronaOrigen, NeuronaArtificial NeuronaDestino, float Peso) throws Exception { ...8 lines }

    public void ConectarDosNeuronas(NeuronaArtificial NeuronaOrigen, NeuronaArtificial NeuronaDestino) throws Exception { ...11 lines }

    public void ConectarCapas(Capa Capa1, Capa Capa2) throws Exception { ...19 lines }

    public void ConectarCapas() throws Exception { ...10 lines }

    public void EntrenarRed(DatosDeEntrenamiento tablaEntrenamiento) throws Exception { ...98 lines }

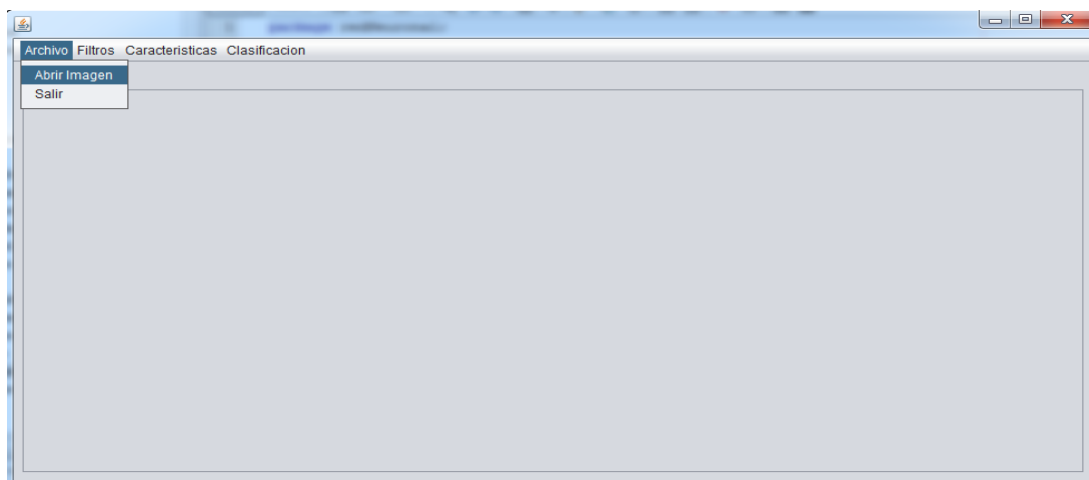
    public ArrayList EjecutarRed(ArrayList Entradas) throws Exception { ...41 lines }

    public ArrayList ObtenerSalidaRed() throws Exception { ...10 lines }
}
    
```

Código que permite crear la red neuronal, ahí están los métodos para conectar las capas, entrenar red, ejecutar la red y obtener las salidas



APLICACIÓN



Abrimos la imagen en la aplicación



De esa manera se ve reflejada en la aplicación la imagen abierta





Aplicamos diferentes filtros por color para ver cual nos conviene mejor

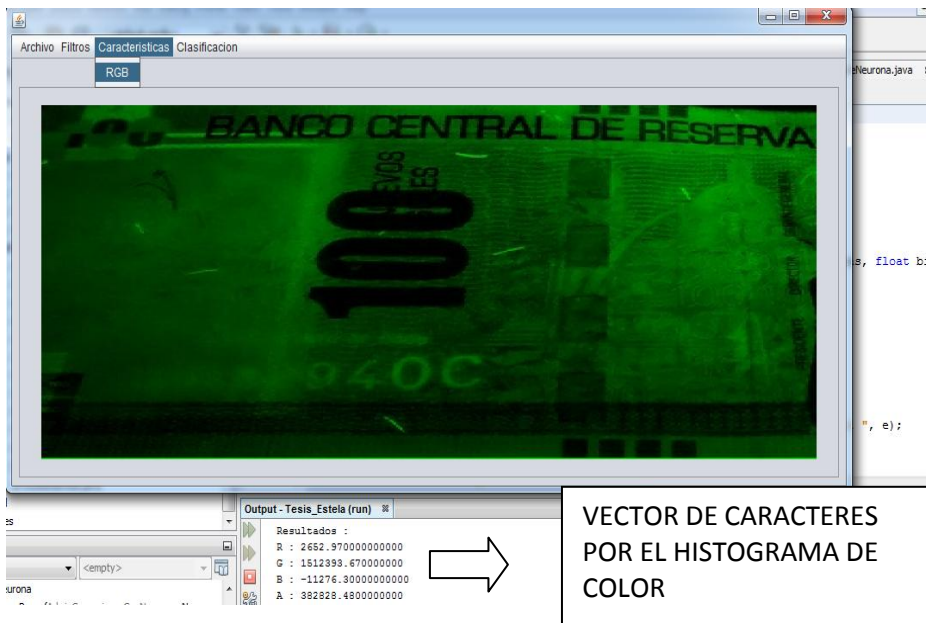


En este proyecto hemos escogido RGB-Verde que refleja las características principales del billete.



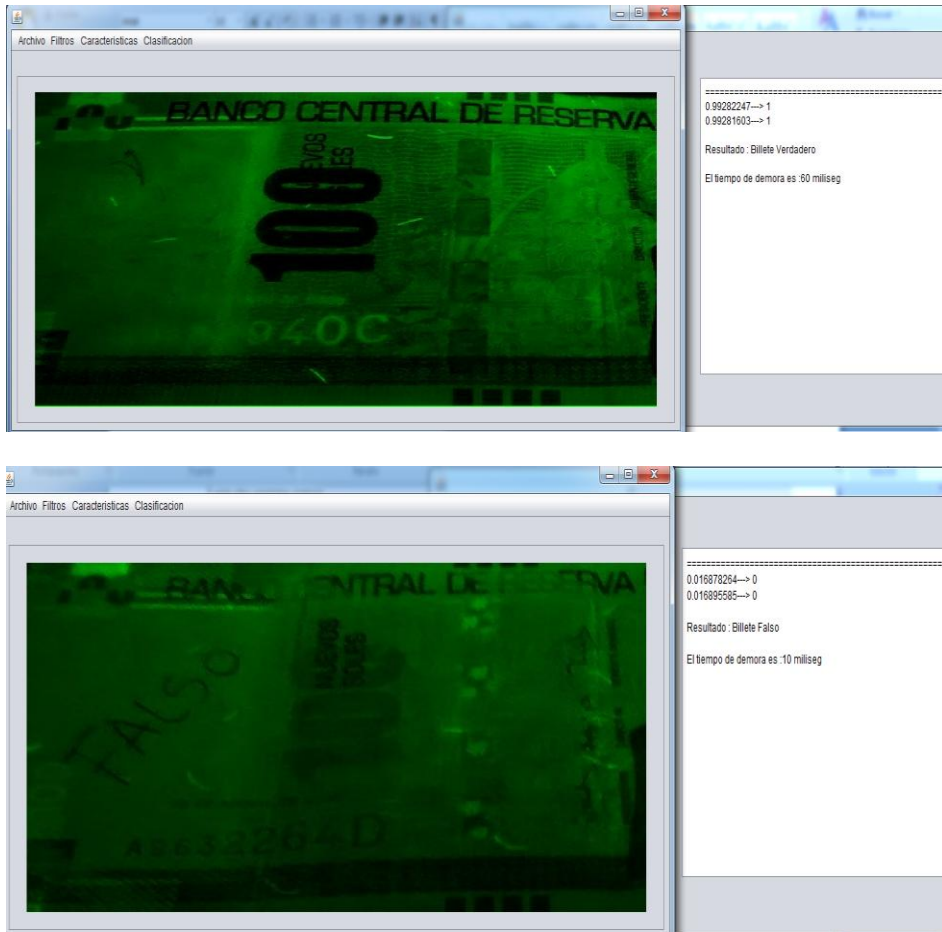


Una vez aplicada el filtro por color, extraemos la extracción de característica



Observamos el vector de características una vez aplicado el paso anterior.





Aplicamos el entrenamiento y dará como resultado 2 salidas, si el billete es verdadero o falso.



