DESARROLLO DE UN SOFTWARE PARA DETERMINAR EL RESULTADO OBTENIDO POR UNOS DADOS EN EL JUEGO CRAPS MEDIANTE PROCESAMIENTO DE IMÁGENES Y TECNICAS DE VISION ARTIFICIAL PARA PERSONAS NO VIDENTES O CON BAJA CAPACIDAD VISUAL

autor
YERSICA PAOLA CARRILLO PEÑALOZA.

Director
M.Sc. LUIS ENRIQUE MENDOZA

Codirector
Ing. NICOLÁS HERNÁNDEZ DÍAZ

PROGRAMA DE INGENIERÍA ELECTRÓNICA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y TELECOMUNICACIONES
FACULTAD DE INGENIERÍAS Y ARQUITECTURA

RESUMEN

El reconocimiento de patrones es un área de investigación sobresaliente en relación a la visión artificial, donde se han propuesto diferentes métodos en los últimos 50 años. En esta propuesta se pretende desarrollar un software para identificar el resultado de dos dados de seis caras usados en el juego Craps con el propósito de servir de ayuda a personas con discapacidad visual.

Para ello se estructura dicho software en cuatro etapas, la primera consiste en la captura de imágenes mediante un dispositivo con cámara digital conectada a la web por dirección IP para enviar información hacia el host; la segunda etapa corresponde al procesamiento realizado a la imagen capturada, debido a que es necesario establecer un tamaño de imagen estándar se procede a redimensionar y ecualizar la imagen digitalizada, la tercera etapa busca segmentar el objeto de estudio mediante técnicas de visión artificial como: eliminación de objetos, área de los objetos, medir las propiedades de la región y segmentación en el espacio de color L*a*b* usando la aproximación del vecino más cercano, etc. para identificar la posición de los dados y extraerlos como recortes, la cuarta etapa presenta el diseño y la clasificación de los recortes mediante CNN, la quinta etapa describe la metodología para la predicción el resultado clasificado, y por último, para la sexta etapa, consiste en tomar el resultado e indicarlo al usuario a través de un altavoz y una pantalla.

Los resultados del sistema se identifican en las seis etapas mencionadas, mediante una aplicación de escritorio en Matlab intuitiva y accesible, especialmente dirigida a personas con discapacidad visual.

ABSTRACT

Pattern recognition is an outstanding area of research in computer vision, where different methods have been proposed in the last 50 years. In this proposal we intend to develop a software to identify the result of two six-sided dice used in the game Craps in order to help visually impaired people.

For this purpose, the software is structured in four stages, the first one consists of capturing images through a device with a digital camera connected to the web by IP address to send information to the host; the second stage corresponds to the processing of the captured image, because it is necessary to establish a standard image size we proceed to resize and equalize the digitized image, the third stage seeks to segment the object of study by artificial vision techniques such as: object elimination, object area, measuring the properties of the region and segmentation in L*a*b* color space using nearest neighbor approximation, etc. to identify the position of the dice and extract them as clippings, the fourth stage presents the design and classification of the clippings using CNN, the fifth stage describes the methodology for predicting the classified result, and finally, for the sixth stage, it consists of taking the result and indicating it to the user through a speaker and a display.

The results of the system are identified in the six stages mentioned above, by means of an intuitive and accessible Matlab desktop application, especially aimed at visually impaired people.

Contenido

Figuras

Tablas

Agradecimientos

Hoy, de todo corazón, quiero expresar mis más sinceros agradecimientos:

A Dios primeramente por ser mi mayor ejemplo de vida.

A mis padres, Gonzalo y María Eufemia por ser mi soporte y guías durante toda mi formación profesional.

A mis hermanos Tatiana y Maicol por su eterna amistad y amor.

Gracias a Nicolas por aparecer en mi vida, por ser la luz que ilumina mi sendero y me mantiene fuerte ante cualquier adversidad.

Agradezco también a todos mis docentes, compañeros y demás personas que de una manera u otra estuvieron conmigo a lo largo de este proceso.

A la Universidad de Pamplona por mi formación como profesional.

1

1. INTRODUCCIÓN

El ser Humano adquiere información del medio ambiente a través de los sentidos, siendo uno de los más importantes el sentido de la vista[1]. El deterioro o ausencia de este sentido genera una desventaja del individuo en su desenvolvimiento diario. Según la ley colombiana 1680 de noviembre del 2013 por la cual se garantiza a las personas ciegas y con baja visión, el acceso a la información, a las comunicaciones, al conocimiento y a las tecnologías de la información y de las comunicaciones [2], se estima que Colombia tiene una prevalencia de ceguera de 0,7% lo cual implica según el censo del 2020, 1.948.332 de personas con discapacidad visual [3].

En Colombia es uno de los juegos más populares como el juego Crap [4]. Por las características del juego, este no puede ser ejercido por personas no videntes, generando un problema de exclusión y cohíbe a invidentes o personas con baja capacidad visual el acceso a este tipo de recreación o interacción social.

Los algoritmos de reconocimiento de patrones se han usado en ciertos trabajos. Un primer artículo [5] se publicó en el año 2017, presenta un acercamiento a la identificación de los naipes y el conteo de fichas en un entorno de juego de póker, usando técnicas de visión por computadora. Para la identificación de los naipes, haciendo uso de Template Matching, que es un patrón de reconocimiento que permite detectar la presencia de un objeto específicos en una imagen.

En un segundo articulo [6] se publicó en el año 2017, se desarrolla un sistema de reconocimiento automático de una carta de baraja ubicada sobre una mesa, el cual es asistir

a personas no videntes en el juego de azar denominado “cuarenta”. Este dispositivo informa al usuario de las cartas en juego, vía audio, para el desarrollo se usó el algoritmo de k-vecinos más cercanos entrenado con un conjunto de caracteres alfanuméricos. El conjunto de prueba consta de fotografías tomadas en condiciones controladas de iluminación, con las cartas ubicadas en orientación arbitraria. El algoritmo fue implementado en un sistema embebido Raspberry Pi 3, y obtuvo un tiempo de respuesta de 5 segundos, incluida la salida a audio.

En un tercer trabajo [7] se publicó en el año 2020, se entrenó a un modelo usando tensorflow para detectar el palo y el número de naipes dada su imagen. se creó un conjunto de datos tomando fotografías de cada tarjeta, luego se realizó el procesamiento y el aumento de datos usando las funciones de rotación, zoom, brillo y corte. El conjunto de datos final consta de 39000 imágenes en escala de grises de 180x180 que forma una red neuronal convolucional para clasificar las imágenes de la tarjeta.

En un cuarto trabajo [8] se publicó en el año 2012, Se realiza una serie de tareas de localización de objetos mediante visión artificial, los objetos son estáticos y no se mueven. En este trabajo se pretende abordar el problema en cuanto a los objetos están en movimiento utilizando como sistema sensorial una cámara y como método de seguimiento del objeto técnicas de predicción con el objetivo de obtener un sistema más rápido y robusto. Se analizan los problemas de predecir la posición de un objeto sobre una secuencia de imágenes utilizando splines cúbicos y se plantea una solución para que la trayectoria de salida sea más suave.

En un quinto trabajo [9] se publicó en el año 2014, consiste en la identificación de fichas de dominó por un sistema de forma automática a partir de imágenes tomadas por una cámara digital. Para ello se ha implementado un algoritmo en Matlab, el cual recibe un archivo de imagen (típicamente jpg, tiff…etc.) en el que aparecen fichas de dominó sobre un tapete color verde. El sistema es capaz de identificar cada una de las fichas tanto si se encuentran separadas como si están colocadas juntas unas a otras. Además, se han implementado técnicas para recuperar partes de la imagen perdidas debido a la iluminación.

En un sexto trabajo [10] se publicó en el año 1995 , se describe un sistema de visión artificial típico en una aplicación inusual, la inspección visual automatizada de las mesas de juego de un casino. El sistema de visión artificial, tiene como objetivo automatizar las tareas de detección y clasificación de las puntuaciones de los dados en las mesas del juego 'Banca Francesa' (que significa Banca Francesa) en Casinos. El sistema se basa en el análisis en línea

de las imágenes captadas por una cámara CCD (Charge Coupled Device o, en español, Dispositivo de Carga Acoplada) monocroma colocada sobre las mesas de juego, con el fin de extraer información relevante sobre la puntuación indicada por los dados. Se desarrollaron e implementaron algoritmos de procesamiento de imágenes para la detección automática de lanzamientos y la clasificación de dados en tiempo real.

En un séptimo trabajo [11] se publicó en el año 2008, se propone un novedoso método de identificación basado en un sistema de visión artificial para reconocer la puntuación de los dados. El sistema emplea técnicas de procesamiento de imágenes y el algoritmo de agrupamiento gris no supervisado modificado (MUGCA) para estimar la ubicación de cada dado e identificar el número de punto de forma precisa y eficaz. Los algoritmos propuestos sustituyen al reconocimiento manual. De los resultados experimentales, se encuentra que este sistema es excelente debido a sus buenas capacidades que incluyen flexibilidad, alta velocidad y alta precisión.

En un octavo trabajo [12] se publicó en el año 2019, Las técnicas de visión por computadora juegan un papel importante en la extracción de información significativa de las imágenes. Un proceso de extracción, análisis y comprensión de la información de las imágenes puede lograrse mediante un proceso automatizado que utiliza visión por computadora y técnicas de aprendizaje automático. clasificación de etiquetas múltiples que se experimenta en un conjunto de datos que contiene 25000 imágenes de flores de 102 especias diferentes. Se extraen características básicas y morfológicas que incluyen color, tamaño, textura, tipo de pétalo, número de pétalos, flor de disco, corona, estivación de flor y clase de flor para aumentar la precisión de la clasificación. Se aplican varios clasificadores en el conjunto de características extraídas y se discute su rendimiento. El resultado de MKL - SVM con clasificación de etiquetas múltiples es muy prometedor con un 76,92% como tasa de precisión.

Un último trabajo [13] se publicó en el año 2021, se describe una forma de usar un sistema de visión de grado industrial para contar los valores de un par de dados después que se colocan a la vista de una cámara, para ello hacen uso de un software comercial denominado y su modulo Blob Count, en un entorno completamente controlado.

Los trabajos anteriormente citados han desarrollado algoritmos de reconocimiento de patrones utilizando la interfaz de Matlab, se ilustra un método de bajo coste computacional para la adquisición de datos, no están relacionados con el problema de las personas no

videntes y su inclusión en juegos de mesa. Este trabajo busca abordar la problemática de este grupo y ofrecer una solución para la misma.

los siguientes objetivos son propuestos:

1.1. Objetivo General

Implementar un software que permita mostrar el resultado obtenido por unos dados en el juego Craps mediante técnicas de visión artificial y procesamiento de imágenes para personas invidentes o con baja capacidad visual.

1.2. Objetivos Específicos

Los objetivos específicos de este Proyecto de Fin de Carrera mismos que son necesarios para poder cumplir con el objetivo general, se resumen en los siguientes puntos:

- Crear un ambiente como símil al juego Craps para realizar los respectivos ensayos y la adquisición de datos (Imágenes del espacio de trabajo).
- Desarrollar un algoritmo para capturar, procesar y analizar los datos de interés presentes en una imagen mediante técnicas de visión artificial, para obtención de resultados y transmisión audible o visual.
- Implementar un algoritmo que permita predecir el resultado de los dados en el lanzamiento.
- Diseñar una interfaz de usuario que permita mayor interacción.

1.3. Acotaciones

El proyecto está localizado en la ciudad de Pamplona Norte de Santander, Colombia; el desarrollo de un software (de bajo costo y accesible a la comunidad interesada) permite determinar y predecir los resultados de dos dados usados en el juego Craps mediante el reconocimiento de patrones haciendo uso de técnicas de visión artificial y procesamiento de imágenes.

En general, el objetivo principal del proyecto es la inclusión de personas invidentes o con baja capacidad visual en el juego Craps, esto en conformidad a la ley 1680 Nov 2013, al brindar una herramienta que hace uso de (altavoz y pantalla) para compartir el resultado obtenido.

El sumario de este trabajo es el siguiente: el capítulo 2 presenta la descripción de los preliminares, el capítulo 3 está dedicado a describir el ambiente y adquisición de datos de trabajo con la finalidad de conocer más el entorno de trabajo, el capítulo 4 presenta la captura, procesamiento y análisis de los datos de interés, el capítulo 5 está dedicado a la clasificación del resultado de los dados, el capítulo 6 presenta la implementación de un algoritmo que permite predecir el resultado de los dados en el juego, el capítulo 7 presenta el desarrollo de la interfaz que permite al usuario interactuar con el programa, Las conclusiones y trabajo a futuro son presentadas en el Capítulo 8.

2

2. PRELIMINARES

2.1. Matemática Preliminar

En este capítulo la matemática preliminar usada en el desarrollo de este trabajo la cual incluye una descripción general de filtros locales, filtros globales, ecualización, operaciones básicas, operaciones morfológicas y el procedimiento que se ha usado en técnicas de aprendizaje profundo, teoría básica de CNN para lograr parámetros que entrenaran en la red y así obtener un modelo neuronal para clasificar las seis clases posibles.

2.2. Adquisición de Imágenes

La primera etapa, dentro de un proceso de visión computacional es la etapa de adquisición, se adquiere la imagen sea la más adecuada posible para que se pueda continuar las siguientes etapas.

Una correcta adquisición de la imagen es un paso muy importante para que el proceso de reconocimiento tenga éxito. Dentro de esta etapa existen múltiples factores que afectan correctamente al proceso de captura de la imagen, formando fundamentalmente por: el sistema hardware de visión artificial (cámara, dispositivo de adquisición de imágenes, ordenador y software MATLAB), el entorno y posición de los elementos (la iluminación, posición del tapete y posición correcta de la cámara, etc.)[14].

2.2.1. ***Dispositivo móvil para la adquisición de imágenes***

La elección de la cámara depende de múltiples factores que se detallan a continuación: para obtener la imagen a color siendo de una cámara web o dispositivo móvil, una de las ventajas principales de estos dispositivos son la imagen, ya que esta necesita tiempo de procesado y

procesos en tiempo real, la velocidad de cálculo es un parámetro importante a considerar, realmente esta proporciona una información más completa que puede servir en la etapa de segmentación y extracción de patrones; uno de los parámetros más importantes es la resolución, en cuanto más resolución tiene la cámara, mejor podemos distinguir los detalles más pequeños de los objetos.

Estas son las ventajas de estos dispositivos, pero hay una gran diferencia en cuanto al costo de las cámaras digitales, ya que estas son más costosas en el mercado y diferencia de un dispositivo móvil son más asequible para el jugador.

Teniendo una idea más detallada para la elección de la cámara y dispositivo móvil se usó el celular MOTO G7 POWER para la captura de imágenes y el celular IPHONE X, el cual se usó para la predicción del resultado, como en la Tabla 1 se muestras comparativa de características principales del dispositivos.[15]

CLASE	MOTO G7 POWER	IPHONE X
RESOLUCIÓN DE CAMARA TRASERA	1.512 x 720 píxeles	2.436 x 1.125 píxeles
PROCESADOR	Snapdragon 632 de ocho núcleos de 1.8 GHz	Apple A11 Bionic
ALMACENAMIENTO	32/64 GB	64/256 GB
BATERIA	5.000 mAh con Turbo Charge	2.716 mAh
VIDEO A CAMARA LENTA	1080p a 60 o 30 f/s	1080p a 120 o 240 f/s

Tabla 1. Características de los dispositivos de adquisición

Para la calidad de la imagen existen técnicas de filtrado que sirven para eliminar el ruido, existen cámaras que permiten la conexión desde un dispositivo de adquisición hacia el ordenador como Wi-Fi utiliza el Protocolo de Internet (IP), en este caso para el celular moto G7 Power como la aplicación de Webcam IP [16] y para el celular iPhone X como la aplicación de EpocCam [16] son para comunicarse entre los dispositivos de punto final (end points) y la red LAN. Se crea una conexión Wi-Fi mediante un enrutador inalámbrico que está conectado a la red y permite que los dispositivos accedan a Internet [14].

El modelo TCP/IP (Transmission Control Protocol/Internet Protocol) es un tipo de descripción de protocolos de red. Como ésta, TCP/IP Dicho modelo hace posible que otros dispositivos estén controlados entre sí y establece los preceptos sobre los cuales un equipo puede

comunicarse con una red, además de especificar el formateado de los datos. Representa todas las reglas de comunicación para Internet y concede una dirección IP a cada equipo para enrutar paquetes de datos. Además, usa una técnica de direcciones y detecta errores en las transmisiones de datos, y se asegura que los paquetes sean entregados en el mismo orden de envío y evitar la pérdida de datos durante la transmisión [17].

2.2.2. ***Fotogramas por segundo***

Un fotograma o frame es cada una de las imágenes que forman un vídeo. Se expresan con las siglas fps y en hercios (Hz). Se usan en campos de vídeo, el cine o gráficos por ordenador. Podemos hacer una grabación en diferentes fotogramas por segundo, dependiendo del tipo de vídeo que sea, se tiene dos tipos de fotogramas, el primer es progresivo consiste que cada fotograma muestra todas las líneas que forman la imagen. El formato progresivo se expresa con una p (1080p) y el segundo es, entrelazado cada fotograma sólo muestra la mitad de las líneas de una imagen, por lo tanto, se necesitan dos fotogramas para mostrar la imagen completa. En imágenes estáticas la calidad está bien, pero si las imágenes contienen un movimiento rápido producen unas líneas como si fueran persianas. El formato entrelazado se expresa con una i (1080i).[18]

Los FPS son básicamente, el número de imágenes unidas que necesitamos para crear un segundo de vídeo. El número de imágenes que entren en un segundo, será el número de fotogramas por segundo que tenga el vídeo como se observa la Figura 1. No solo existen los fotogramas por segundo que hemos visto hasta ahora, sino también a 20, 30, 50, 60 ,100, 120 y 240 fps. [19]

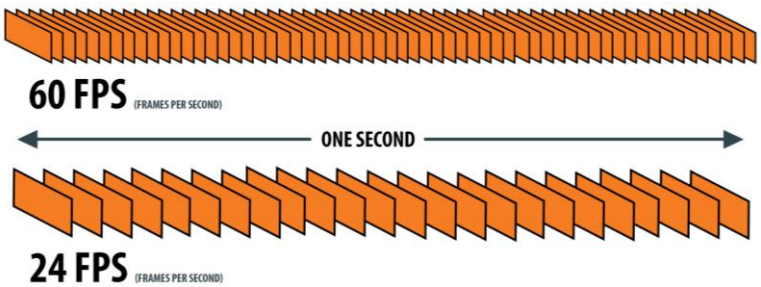


Figura 1. ejemplo representativo Frames por segundo (FPS)

2.2.3. *El host y el software*

2.2.3.1. Selección del tipo de Host

Actualmente se puede llegar a concluir que un proceso de visión artificial requiere gran cantidad de cálculo y sobre todo si se quiere que el sistema actúe en tiempo real. En el mercado existen máquinas de muy diversa índole diferenciándose especialmente por el procesador o procesadores que llevan.

Dentro de la gama de sistemas que se pueden encontrar ordenadores que hay en el mercado están basadas en el uso de los procesadores de la familia INTEL encontrando placas compuestas por procesadores 80386 hasta placas de última generación de procesadores Intel Rocket Lake-S de 11ª generación bajo el socket LGA1200 [20].

Características del host usado: Existen ordenadores basados en la plataforma de INTEL como la familia de computadores como Dell Laptop Inspiron 14" 3467 y un Fabricante del Procesador Intel® Core™ Modelo del procesador i5-7200U de séptima generación (memoria caché de 3 MB, hasta 3,10 GHz) de ordenador, gran cantidad de información del software y de los sistemas conocidos por muchas personas, multitud de herramientas de programación de bajo precio, fácil mantenimiento, etc. [21]

2.2.3.2. Selección del lenguaje y entorno de programación

Los programas de visión artificial y el procesamiento de imágenes están desarrollados por las siguientes partes: entorno visual de pruebas, procesado, leguajes de programación.
Generalmente, los programas de visión artificial suelen tener un entorno donde se puede capturar la imagen, procesarla con diferentes algoritmos y determinar, por ejemplo, que funciones se van utilizar. Además, suelen tener un lenguaje de programación propia, que permite la realización de subrutinas que no existan en sus propias librerías de desarrollo [14].

Por otro lado, es necesario investigar los diferentes entornos de programación como: Eclipse, Netbeans, Visual Studio, Java, Ruby, Python, PHP, SQL, Objective-C, C++ y JavaScript y Matlab, se toman dos entornos de programación como Python y Matlab, en la Tabla 2 se mencionan las ventajas y desventajas y a partir de esto seleccionar cual se adapta más al proyecto.

	MATLAB	PYTHON
Ventajas	• Amplio soporte matemático • Alta precisión • Amplio soporte matemático además de posibilidad de uso de precisión extendida en los cálculos. • Amplio soporte de funciones ya desarrolladas. • Rápido prototipo • El script es corto, pero está altamente integrado con todos los paquetes. • Visualización óptima de gráficos y cuadros interactivos • Fácil de administrar el soporte de múltiples subprocesos y la recolección de basura. • El mejor depurador.	• Desarrollo de un extremo a otro para la ejecución (algunos paquetes de corredores permiten la ejecución) • Paquetes de código abierto (Pandas, Numpy, scipy) • Paquetes comerciales (zipline, pybacktest, pyalgotrade) • Es un lenguaje extremadamente productivo. • Tiene licencia de código abierto y está disponible para los usuarios de forma gratuita. • Cuenta con una amplia comunidad activa de desarrollo. • Presenta una fácil integración con otros lenguajes de programación.
Desventajas	• No se puede ejecutar, debe convertirse a otro idioma. • Problemas eventuales de velocidad • Es difícil detectar desviaciones en el sistema comercial (está construido para simulaciones matemáticas y de ingeniería), por lo que pueden ser necesarias pruebas exhaustivas. • No es posible desarrollar una aplicación separada.	• Ejecución más lenta que en los lenguajes no interpretados. • Sus funciones dinámicas pueden causar errores en tiempo de ejecución. • Posee un consumo ineficiente de memoria para realizar algunas tareas. • Python no adopta un entorno de desarrollo predeterminado (IDE). -Anaconda, un paquete de Python muy apreciado, incluye dos IDE totalmente diferentes. -Spyder y JupyterLab funcionan con la misma eficacia que el IDE de MATLAB.

Tabla 2. ventajas y desventajas de Software como Python y Matlab [23]

Finalmente, el entorno de programación escogido es el software de Matlab en su versión R2017b cuyas siglas en inglés provienen de MATrix LABoratory (laboratorio de matrices). Es necesario observar la cantidad de subrutinas de que constan de las librerías para cada etapa del proceso de visión artificial: (filtrados, ecualización por histogramas, transformaciones geométricas, etc.)[24].

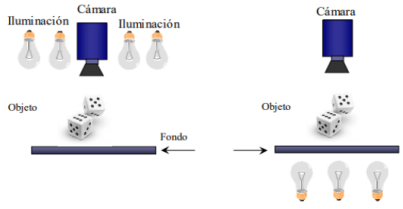
GUIDE

La interfaz gráfica de usuario, conocida también como GUIDE (Graphical User Interface Development Enviroment) es un entorno de programación visual disponible en MATLAB que permite diseñar interfaces gráficas y ejecutar programas, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Esta herramienta es muy útil cuando se tiene un programa MATLAB que necesita un ingreso continuo de datos o la interacción activa con el usuario. Con GUIDE es posible crear de forma sencilla ventanas y diálogos que contengan los componentes básicos, como son etiquetas, campos de texto, botones, contenedores, entre otros [24].

2.2.4. *Técnicas de iluminación*

Una vez escogido los elementos del sistema de visión artificial, se pasan a estudiar los elementos correspondientes al contexto. Un entorno debidamente controlado es imprescindible para obtener unas condiciones de partida optimas que aseguran una perfecta adquisición. Dentro del entorno aparecen como parte fundamental que influyen en la calidad de la imagen buscada: la iluminación, el fondo, la posición de la cámara, etc. [14].

La iluminación se realiza de forma correcta, dada la importancia que tiene. Existen fundamentalmente dos formas de iluminación. La iluminación frontal de la Figura 2 (a), donde se permite distinguir los detalles y la iluminación que incide directamente sobre el objeto, y así detallar más su forma, permitiendo extraer más parámetros de cada objeto como su color, detalles internos, etc., que permiten una mejor segmentación, ya sea verticalmente, horizontalmente, de forma oblicuo o de forma difusa. La iluminación retroiluminada Figura 2 (b), se ilumina una pantalla de forma que se busca la detección de contorno simplificados del objeto a modo de sombra este objeto puede estar delante o detrás de la pantalla, la etapa de segmentación que se debe efectuar posteriormente a la captura [14].



(a) (b)

Figura 2. Tipos de iluminación: (a) frontal, (b) retroiluminación [14]

2.3. PREPROCESADO

Toda imagen que se adquiere por medios ópticos, electrónicos la cual sufre ciertas medidas como los efectos de la degradación que se manifiestan en forma de ruido, perdida en la imagen provocadas por el ruido del enfoque y captura de la cámara, movimiento de la misma o perturbaciones aleatorias. Los algoritmos de procesado permiten modificar la imagen, mejorar la intensidad o el constante, etc.; proceso que permite mejorar el resultado final de la imagen capturada ya que estos procesos requieren una cantidad de tiempo de procesado, existen características de la imagen que en muchos casos es conveniente mejorar como el contraste, brillo, niveles de grises, eliminar brillos, aumentar los bordes, mejorar las texturas, etc. [14].

2.3.1. ***Operaciones Básicas***

Existen gran cantidad de operaciones básicas utilizadas en visión artificial.

2.3.1.1. Operaciones Lógicas

Son expresiones matemáticas cuyo resultado es un valor booleano (verdadero o falso). Los factores que intervienen en una operación lógica solo pueden ser ciertos o falsos [25].

AND: & Funciona con dos operadores, el resultado es verdadero si ambos son verdaderos, en otro caso el resultado es falso (0).

OR : I Funciona con dos operandos. El resultado es verdadero si alguno de los dos es verdadero, en otro caso (los dos son falsos) el resultado es falso (0).

NOT: ~ Funciona con un operando. La negación del operando, es decir, verdadero (1) si el operando es falso, y falso (cero) si el operando es verdadero.

2.3.1.2. Umbralización binaria

La umbralización es una técnica de segmentación, los objetos presentes en una imagen están formados de píxeles de intensidad homogénea. De esta manera, cada pixel es comparado con un umbral prefijado: si el valor de la intensidad del pixel es mayor, este es considerado de una determinada categoría, sin embargo, si el valor de la intensidad es menor, el pixel es de una región diferente. Bajo estas circunstancias, la calidad de la segmentación dependerá de la elección de un valor apropiado del umbral como en la ecuación 1. [26].

El resultado de la condición será si el valor da pixel es mayo que el valor umbral, se le asigna el valor 1, caso contrario se le asigna valor 0.

$$B(i,j) = (A(i,j) \geq umbral \quad (1)$$

Dónde:

- ✓ B es la matriz binaria
- ✓ A es la matriz a binarizar
- ✓ I- enésima filas de la matriz
- ✓ j- enésima las columnas de matriz

umbral es la frontera de binarización como un valor escalar

2.3.2. ***Valor medio***

suponiendo que se tienen MxN valores de una matriz. Se denotan los valores de la matriz n la posición (i, j) por $x_{1,1}, x_{1,2}, .., x_{i,j}$ siendo el valor de la primera posición de la matriz $x_{1,1}$, $x_{1,2}$ el segundo valor de la segunda posición de la matriz y así sucesivamente, por lo tanto, una generalidad para determinar la media aritmética o media o promedio, denotada por x, viene dada por la fórmula.

$$\bar{p} = \frac{\sum_{i=1}^{NxM} p(i)}{NxM} \quad (2)$$

La media anterior representa el centro del conjunto de datos. Si se supone que los valores $x_{1,1}, \ldots, x_{i,j}$. se trazan en una barra delgada en forma de línea recta y las bolas que representan el se colocan en las posiciones de estos puntos y la barra se corta en los dos puntos extremos, entonces el valor x es el centro de gravedad de la barra. los valores de las observaciones se encontrarán alrededor de la media. esta es la interpretación geométrica de la media.[27]

2.4. TRANSFORMACIONES GEOMÉTRICAS DE IMÁGENES

Las transformaciones Geométricas modifican la relación espacial entre píxeles, su uso fundamental se aplica en la reconstrucción de imágenes deformadas. En términos del procesamiento de imágenes digitales una transformación geométrica consiste de dos operaciones básicas: [28]

1. Una transformación espacial que define la reubicación de los píxeles en el plano imagen.
2. Interpolación de los niveles de grises, los cuales tienen que ver con la asignación de los valores de intensidad de los píxeles en la imagen transformada.

2.4.1. ***Algoritmos de rotación o giro***

Los algoritmos de giro son generalmente lo más complejos y por lo tanto los más costosos en tiempo de proceso. Debido a esto, solo se utilizan cuando es posible obtener una posición de giro que simplifique más posteriores procesos. [14]

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} cos\,\theta & -sen\,\theta \\ sen\,\theta & cos\,\theta \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} \quad (3)$$

Por lo tanto, la versión girada de la imagen principal será:

$$IMB(i',j') = IMB(i\cos\theta, j\ \ sen\theta, i\ \ sen\theta + j\ cos\theta) = IMA(i,j) \quad (4)$$

Dónde:

- ✓ B es la matriz de los datos de A rotados un ángulo θ
- ✓ A es la matriz a rotar
- ✓ i - enésima filas de la matriz
- ✓ j - enésima columnas de matriz
- ✓ θ el ángulo a rotar (grados)
- ✓ i' y j' las coordenadas del nuevo punto

Después serán necesaria realizar una interpolación, si se tiene una imagen formada por múltiples rectángulos y cuadrados situados de forma perpendicular la imagen esta girada, es muy útil en los ejes de (x,y) de la imagen, estos rectángulos o líneas serán más rápidos y eficaces. [14]

2.4.2. ***Espejos***

Algoritmos que realizan un espejo respecto de la horizontal, vertical o diagonal. Si la imagen es de dimensiones MxN [14].

$$IMB(i,j) = IMA(N-i,j) \text{ (Espejo vertical)} \quad (5)$$
$$IMB(i,j) = IMA(i,M-j) \text{ (Espejo horizontal)} \quad (6)$$

Espejos se pueden realizar de múltiples formas, dependiendo de los ejes para realizarlo.[14]

- ✓ B es la matriz a procesar
- ✓ A es la matriz a procesar
- ✓ i - enésima filas de la matriz
- ✓ j - enésima las columnas de matriz
- ✓ M son las filas de la matriz
- ✓ N son las columnas de matriz

2.5. MEJORA Y FILTRADO DE IMÁGENES

2.5.1. ***Filtrado***

El filtrado es el conjunto de técnicas englobadas dentro del preprocesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella [29].

1. Suavizar la imagen: reducir la cantidad de variaciones de intensidad entre píxeles vecinos.
2. Eliminar ruido: eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
3. Realzar bordes: destacar los bordes que se localizan en una imagen.
4. Detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad.

2.5.2. ***Normalización***

Normalización consiste en aplicar una transformación de luminosidad para alargar la adquisición de la imagen se ha habituado con un rango menor, por ejemplo, si se captura una imagen con 128 niveles de grises y se tiene la posibilidad de utilizar 256, el desarrollo de normalización consistiría en ampliar el rango de 128 a 256 niveles. Puede parecer que es una

operación al pixel, hay que darse cuenta que para alcanzar los rangos máximos y mínimo, primero hay que registrar todos los puntos para determinar el mínimo y máximo valor. [14]

$$IMB(i,j) = K \cdot \frac{IMA(i,j) - K_{min}}{K_{max} - K_{min}} \quad (7)$$

Dónde:

- ✓ B es la matriz a normalizar
- ✓ A es la matriz a normalizar
- ✓ i - enésima las filas de la matriz
- ✓ j - enésima las columnas de matriz
- ✓ K es la transformación resultante
- ✓ K_{min} son los rangos mínimos
- ✓ K_{max} son los rangos máximos

En la ecuación 7, es aplica a los datos entregados a la red para entrenamiento clasificación de la red neuronal convolucional con la finalidad de evitar sobreajuste o errores de clasificación.

2.5.3. ***Operaciones Morfológicas***

La morfología matemática es una herramienta muy utilizada en el procesamiento de imágenes proporciona una serie de importantes operaciones de procesamiento de imágenes, que incluyen erosión, dilatación, apertura y cierre. Todos estos operadores morfológicos toman dos datos como entrada. Una es la imagen de entrada, que puede ser binaria o en escala de grises para la mayoría de los operadores. Pueden simplificar los datos de una imagen la cual se encarga de cambiar la forma y la estructura del objeto que conforman dicha imagen, Además, permiten modificar estas formas para separar los objetos unos de otros, El otro es el elemento estructurante. Es esto lo que determina los detalles precisos del efecto del operador en la imagen[14].

2.5.3.1. Dilatación Binaria

La operación de dilatación hace que un objeto aumente de tamaño. La medida en que crece depende de la naturaleza y la forma del elemento estructurante. La dilatación de una imagen A (conjunto) por el elemento estructurante B se define en la ecuación 8.

$$A \oplus B = \left\{ z \,\middle|\, (\hat{B})_z \cap A \neq \emptyset \right\} \quad (8)$$

Tengamos en cuenta que, para la intersección sólo consideramos los píxeles negros de A y B.

El primer elemento de la dilatación, A, está asociado con la imagen que se está procesando y el segundo recibe el nombre de elemento estructural, la forma que actúa sobre A en la dilatación para producir $A \oplus B$. [30]

- Selecciona el pixel de la imagen original a tratar.
- Busca el mayor de los píxeles de la vecindad, incluido el central, definidos por la forma y tamaño del elemento estructural.
- Sustituye el valor del píxel por el máximo valor.

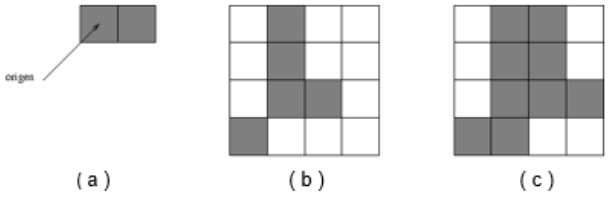


Figura 3. Ejemplo de dilatación. (a) Elemento estructural, B. (b) Imagen, A. (c) Resultado de la dilatación $A \oplus B$. [30]

2.5.3.2. Erosión Binaria

La operación de erosión es el complemento de la operación de dilatación en contexto con el efecto de la operación. Es decir, la operación de erosión hace que el objeto pierda su tamaño. La erosión de una imagen A por un elemento estructurante B se define como dos conjuntos A y B de Z^2 la erosión, denotada por $A \ominus B$, se define como:

$$A \ominus B = \{z \,|(B)_z \subseteq A\} \quad (9)$$

Tengamos en cuenta que, para la condición $|(B)_z \subseteq A$, sólo consideramos los píxeles negros de A y B. [30]

Dónde:

- la erosión es la operación morfológica dual de la dilatación.
- La erosión se concibe usualmente como una reducción de la imagen

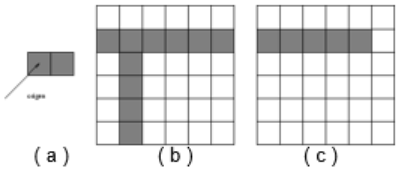


Figura 4. Ejemplo de erosión. (a) Elemento estructural, B. (b) Imagen, A. (c) Resultado de la erosión $A \ominus B$. [30]

2.5.3.3. Apertura

Es una combinación de una erosión seguida de una dilatación siempre con el mismo elemento estructural. Su efecto es eliminar elementos salientes, conexiones finas o suavizar los contornos de un objeto. [30]

La apertura de A por un elemento estructural K se define como:

$$A \cdot K = (A \ominus K) \oplus K \quad (10)$$

Establece que la apertura de A por K es simplemente la erosión de A por K, seguido de la dilatación del resultado por K.

Si A no cambia con la apertura con K, diremos que A es abierto respecto a K.

2.5.3.4. Cierre

Consiste en la conexión de una dilatación seguida de una erosión. Su efecto es el de unir dos objetos separados por un estrecho margen, eliminar pequeños agujeros manteniendo el tamaño del objeto, rellenar huecos existentes en el contorno o suavizar el contorno. es la operación dual del opening por lo que su definición es:

$$X \cdot B = (X \oplus \bar{B}) \ominus B \quad (11)$$

En la Figura 5 se muestran dos ejemplos sencillos cierre (derecha). [30]

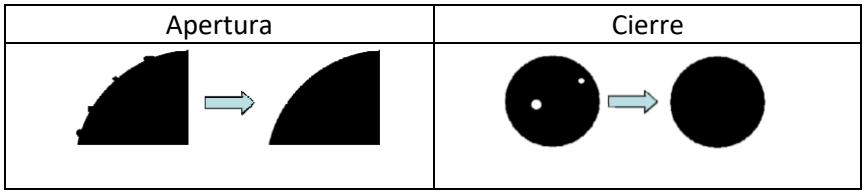


Figura 5. Ejemplos de operaciones de apertura (izquierda) y cierre (derecha) [30].

2.5.3.5. Elemento estructurante morfológico

Un elemento estructurante es una matriz que identifica el píxel en la imagen que se está procesando y define la vecindad utilizada en el procesamiento de cada píxel. Por lo general, elige un elemento de estructura del mismo tamaño y forma que los objetos que desea procesar en la imagen de entrada. Por ejemplo, para buscar líneas en una imagen, cree un elemento de estructura lineal. [31].

Hay dos tipos de elementos estructurantes: planos y no planos. Un elemento estructurante plano es una vecindad con valores binarios, ya sea bidimensional o multidimensional, en la que los píxeles verdaderos se incluyen en el cálculo morfológico y los píxeles falsos no. El píxel central del elemento estructurante, llamado origen, identifica el píxel de la imagen que se está procesando. Utilice la función strel para crear un elemento de estructuración plano. Puede utilizar elementos de estructuración planos con imágenes binarias y en escala de grises. La siguiente Figura 6 un elemento estructurador plano. [30]

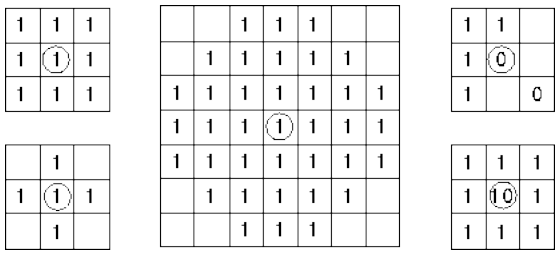


Figura 6. Representación de los elementos estructurantes. [30]

2.5.4. ***Tipos de elementos estructurantes***

A continuación, se describen cada una de los elementos estructurantes haciendo del lenguaje de programación de Matlab.

2.5.4.1. Elemento estructurante diamante

SE = strel ('diamante', r) crea un elemento estructurante en forma de diamante, donde r especifica la distancia desde el origen del elemento estructurante hasta los puntos del diamante [31].

2.5.4.2. Elemento estructurante disco

SE = strel ('disco', r, n) un elemento de estructuración en forma de disco, donde r especifica el radio en específico del elemento de estructura de línea utilizados para aproximar la forma

del disco. Las operaciones morfológicas que utilizan aproximaciones de disco se ejecutan mucho más rápido cuando el elemento estructurante utiliza aproximaciones [31].

2.5.4.3. Elemento estructurante Octágono

SE = strel ('octágono', r) crea un elemento de estructura octogonal, donde r especifica la distancia desde el origen del elemento de estructura a los lados del octágono, medida a lo largo de los ejes horizontal y vertical. r debe ser un múltiplo no negativo de 3 [31].

2.5.4.4. Elemento estructurante lineal

SE = strel ('linea', grados) crea un elemento de estructuración lineal que es simétrico con respecto al centro vecino, con longitud aproximada y ángulo [31].

2.5.4.5. Elemento estructurante rectángulo

crea un elemento de estructuración plano, con forma de rectángulo, donde MN especifica el tamaño. MN debe ser un vector de dos elementos de enteros no negativos. El primer elemento de MN es el número de filas en la vecindad del elemento estructurante; el segundo elemento es el número de columnas [31].

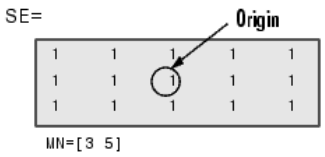


Figura 7. Elemento estructurante rectangular [31].

SE = strel ('rectángulo', [m n]) crea un elemento de estructura rectangular de tamaño [m n].

2.5.4.6. Elemento estructurante cuadrado

SE = strel ('cuadrado', w) crea un elemento de estructuración cuadrado cuyo ancho es w píxeles [31].

2.5.4.7. Elemento estructurante cubo

SE = strel ('cube', w) crea un elemento estructurante cúbico 3-D cuyo ancho es w píxeles [31].

2.5.4.8. Elemento estructurante cuboide

SE = strel ('cuboide', [m n p]) crea un elemento estructurante cuboidal 3-D de tamaño [m n p] [31].

2.5.4.9. Elemento estructurante esfera

SE = strel ('esfera', r) crea un elemento estructurante esférico 3-D cuyo radio es r píxeles [31].

2.5.5. ***Rellena regiones y agujeros de la imagen***

infill (BW"= imfill (BW, locations)), realiza una operación de relleno por inundación en los píxeles de fondo de la imagen binaria de entrada, comenzando desde los puntos especificados en las ubicaciones [31].

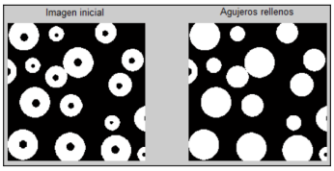


Figura 8. Relleno de agujeros [31].

2.6. PROPIEDADES DE IMAGEN Y REGIÓN

Las regiones de imagen, también llamadas objetos, componentes conectados o manchas, tienen propiedades como área, centro de masa, orientación y cuadro delimitador. Para calcular estas propiedades (y muchas más), puede utilizar la aplicación Image Region Analyzer o la función regionprops.

También puede medir valores de píxeles de píxeles individuales, a lo largo de una ruta en una imagen o agregados en una imagen completa.

2.6.1. ***Eliminar objetos pequeños de la imagen binaria***

Elimina todas las componentes conectadas que tienen menos píxeles de la imagen binaria como en la Figura 9 de la izquierda produciendo una nueva imagen binaria como se muestra en la Figura 9 de la parte derecha [31].

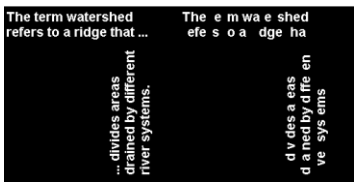


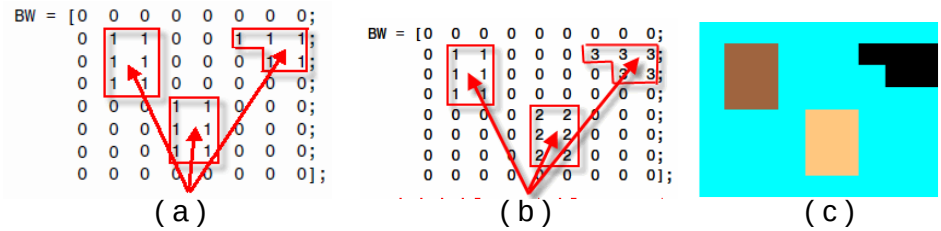
Figura 9. Eliminar objetos pequeños de la imagen binaria

2.6.2. *Etiqueta los componentes conectados en una imagen binaria 2-D*

Un componente conectado de una imagen binaria es un conjunto de píxeles que forman un grupo conectado. Por ejemplo, la siguienteFigura 10 (a), binaria tiene tres componentes conectados. [31].

Especifica una conectividad de los píxeles continuos que forman parte del mismo objeto si ambos están conectados a lo largo de la dirección horizontal o vertical. Se etiqueta los componentes conectados en el proceso de identificar los componentes conectados de la Figura 10 (b) y asignar a cada uno una etiqueta única.

Para visualizar los componentes conectados de la etiqueta que identifica cada objeto en la matriz de etiquetas se asigna a un color diferente en el mapa de colores. Como en la Figura 10(c) [31]



(a) (b) (c)

Figura 10. (a) Imagen binaria, (b) Componentes Etiquetados, (c) Mapa de colores de los componentes etiquetados.

2.6.3. *Área*

Número real de píxeles en la región, devuelto como escalar. Para encontrar el equivalente al área de un volumen, se usa la propiedad de 'Volumen' como en la Figura 11 la cual se tienen dos áreas diferentes. [31]

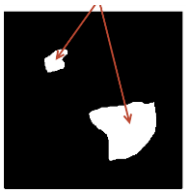


Figura 11. Área del objeto

2.6.4. ***Argumentos***

2.6.4.1. Medir las propiedades de las regiones de la imagen

El contorno delimitador se presenta en la Figura 12 de objeto permite encuadrar dentro de una región un objetos de interés dadas sus coordenadas (x,y,w,h), donde (x,y) corresponde al punto superior izquierdo, w lo ancho del encuadre y h lo alto.[31].

Figura 12. Cuadros delimitadores alineados con el eje

2.6.4.2. Centroide

Centro de la región, devuelto como una matriz. El primer elemento de Centroide es la coordenada horizontal (o coordenada x) del centro de masa. El segundo elemento es la coordenada vertical (o coordenada y). Todos los demás elementos de Centroide están en orden de dimensión. Después, calculamos cuáles son los centroides en Matlab como en la Figura 13 (a), y finalmente cuando mostramos los centroides usando código de Matlab como se puede apreciar en la Figura 13 (b) [31]

```
img = logical(...
   [0  0  0  0  0  1  1  1  0  0;
    0  1  0  1  0  0  1  1  0  0;
    0  1  1  1  0  0  0  0  0  0;
    0  0  0  0  0  0  0  0  0  1;
    0  0  0  0  0  0  0  0  1  1;
    0  0  1  1  1  1  0  0  1  1]);
s = regionprops(img, 'Centroid');
```

(a)

```
>> disp(cat(1,s.Centroid))
    3.0000    2.6000
    4.5000    6.0000
    7.2000    1.4000
    9.6000    5.2000
```

(b)

Figura 13. (a) Matriz a determinar los centroides, (b) Centroides hallados

2.7. SEGMENTACIÓN

Segmenta una imagen digital representar dividirla en zonas disjuntas e individualizadas. Es decir, consiste en distinguir los diversos objetos y donde se localiza el fondo, que puede ser más o menos difícil de la imagen.

Al final de la etapa de segmentación, se tiene que conocer perfectamente los objetos que hay para quitar las características particulares de cada uno de ellos. Además, cada $pixel$ de la imagen tiene que tener una marca que los defina, de forma que simplemente por agrupación de puntos con misma característica, se puede definir la lista de objetos.

2.7.1. *Segmentación basada en color mediante el espacio de color L*a*b**

En esta sección se va a mostrar de forma resumida los principales modelos del color existentes. Se deja para el final CIE L*a*b* por ser el estudiado en este proyecto.

Antes de comenzar es necesario explicar en qué consiste un modelo del color. Se trata de la representación matemática que permite definir los colores de forma abstracta. Está basada en un sistema de coordenadas, con base de N vectores, cuya combinación lineal genera todo el espacio de color. Los distintos modelos tratan de englobar el mayor número de colores visibles al ojo humano, aunque habrá modelos que no tengan esta finalidad porque no les sea conveniente para su posterior aplicación.

Una de las clasificaciones que se puede realizar es en función del número de dimensiones empleadas.

DIMENSIÓN	MODELO COLOR
1 dimensión	Escala de grises
2 dimensiones	Sub-espacio RG
3 dimensiones	RGB, CMY, HSI, YUV, CIE XYZ, CIE L*a*b*
4 dimensiones	CMYK

Tabla 3. Clasificación de algunos modelos del color según su dimensión [32].

El objetivo de investigar de estos modelos es el de identificar los espacios de color.

Modelo de Color	Características
Modelo RGB	El espacio de color como un sistema cartesiano donde las bases de referencia son los colores como: rojo (Red (255, 0, 0)), verde (Green (0, 255, 0)) y azul (Blue (0, 0, 255)), expresados como vectores ortogonales. - Aplicación de este modelo RGB se aprecia en sistemas emisores o receptores de luz
Modelo	El espacio de color CMY define el color como un sistema cartesiano, donde la base de referencia son los colores cian (Cyan), magenta (Magenta) y amarillo (Yellow), expresados como vectores ortogonales.

	CMY	Éste se basa en la síntesis sustractiva cuya representación de un color se fundamenta en la no absorción de la luz, donde la suma de los tres colores bases da como resultado el negro y la ausencia implica el blanco.
	CMYK	El modelo CMYK está basado en el modelo CMY donde se incluye como cuarto color el negro (K). De esta forma, se resuelve el problema práctico que tiene el modelo CMY con respecto a la generación del color negro.
Modelo	YUV	El modelo YUV define un espacio de color en términos de una componente de luminancia y dos componentes de crominancia. El valor Y representará la luminancia que corresponde con la información en blanco y negro de una imagen, y los valores U y V representará la crominancia, es decir, la información de color de la imagen.
	YIQ	El modelo YIQ fue creada por NTSC (National Television System Committee) y ésta se basó en las especificaciones del modelo YUV. El modelo YIQ especifica la Y con la componente de luminosidad y las I y Q con la cromaticidad.
Modelo	HSI	El modelo HSI define el color en las componentes de matiz, saturación e intensidad. Tiene dos características importantes que hacen que el modelo HSI sea una herramienta ideal para desarrollar algoritmos de procesamiento de imágenes basada en la percepción del color del sistema humano. - La componente de intensidad (I) está desacoplada de la información cromática contenida en la imagen.
	HSV	El modelo HSV es muy parecido al modelo HSI, donde se cambia la intensidad por el valor de color máximo de las componentes RGB. - El color asociado con los términos sombras, tintes y tonos se representan en un plano seccional-transversal del cono hexagonal HSV
Modelo CIE XYZ		El modelo CIE XYZ es un modelo matemático que incluye todas las gamas de colores reproducibles, porque el modelo RGB existente no era capaz de representar todos los posibles colores. El espacio tridimensional del color en dos dimensiones artificiales de cromaticidad (x,y). Los tres primarios imaginarios (X, Y, Z) con los que se caracterizan espectralmente son los que representan el color.
Modelo CIE L*u*v*		El modelo de color L * u * v * fue creado por CIE (Comisión Internacional de Iluminación) en 1976 como parte del intento de alcanzar la uniformidad en la percepción del color. Es utilizado con frecuencia en aplicaciones gráficas que usan luces de colores.

	Se explicará de forma breve el significado de cada componente: - L*es la luminosidad del color: Su rango de valores abarca entre 0 y 100, cuando vale 0 representa el color negro y cuando vale 100 simboliza el color blanco. - U* representa la gama de colores entre el verde y rojo: Su rango de valores comprende normalmente desde el -100 al 100. Los valores negativos representan colores cercanos al verde y los positivos colores cercanos al rojo. - V * representa la gama de colores entre el amarillo y azul: Su rango de valores comprende normalmente desde el -100 al 100. Los valores negativos representan colores cercanos al amarillo y los positivos colores cercanos al azul.
Modelo CIE L*a*b*	El CIE propuso el espacio de color CIELAB, también llamado espacio de color CIE 1976 (CIE 1976 color space) como una aproximación a un espacio de color uniforme. [33] Los tres parámetros que componen el modelo son: - L* representa la luminosidad: El rango de valores estará comprendido entre el 0% (negro) y el 100% (blanco). - a* representa la posición del color entre rojo y verde: Su rango de valores comprende desde el -120 al 120. Los valores negativos indican colores cercanos al verde, mientras que los positivos se acercan al rojo. - b* representa la posición del color entre amarillo y azul: Su rango de valores comprende desde el -120 al 120. Los valores negativos indican colores cercanos al azul, mientras que los positivos se encuentran cerca del amarillo.

Tabla 4. Modelos de color [32]

El espacio de color es una transformación matemática del espacio XYZ en el cual se fija un blanco de informe y cuyos valores de triestímulo son (Xn, Yn, Zn). Ese blanco de referencia puede ser, por ejemplo, una fuente luminosa, el iluminante al que se haya adaptado el observador, un difusor perfecto o el color neutro más reflectante o transmisor de un medio de reproducción.

En el sistema CIELAB, los colores deben verse sobre un fondo que vaya de blanco a gris medio por un observador adaptado a un iluminante que no sea demasiado distinto a la luz natural del medio día [33].

1. Los tres ejes del sistema CIELAB se indican con los nombres L*a*b*. Representan, respectivamente Luminosidad, tonalidad de rojo a verde y tonalidad de amarillo a azul.
2. Esa especificación de los colores no es fácilmente interpretable en términos de dimensiones psicofísicas de percepción del color; es decir, brillo, tono y coloración.
3. El sistema XYZ y los diagramas de cromaticidad asociados no son perceptualmente uniformes.

El espacio sRGB fue definido por HP y Microsoft en 1998 con el concepto en mente de que visualizar fotografías a través de Internet (por lo que están comprimidas para ahorrar espacio), Lo malo de este espacio de color es que su rango es demasiado pequeño para monitores de gama alta. Adobe sRGB tiene colores más apagados como "azul / rojo / verde" y perfiles de diferentes colores. Sin embargo, sRGB es algo limitado en cuanto a la gama de colores. En consecuencia, puede parecer que las imágenes tienen tonos más sutiles que no se muestran [33].

coordenadas cromáticas (X, Y, Z): Desde que se estableció el observador patrón CIE 1931, con sus valores triestímulo X,Y, Z se han ido introduciendo muchas coordenadas colorimétricas a partir de ellos por expresiones más o menos complejas. Tales coordenadas sitúan cada color en un espacio definitivo, que tendrá tantas extensiones como número de coordenadas se precisen. Surgen así los denominados espacios de color, habitualmente de tres dimensiones. Por lo tanto, una vez designadas unas coordenadas colorimétricas a emplear queda determinado un espacio de color [34].

A. Generalidad de la conversión

Para obtener una imagen en espacio de color L*a*b* se deben hacer unas conversiones como: sRGB a coordenadas cromáticas (X, Y, Z) y de coordenadas cromáticas (X, Y, Z) a el espacio de color de interes L*a*b*.

B. Conversión de sRGB a coordenadas cromáticas (X, Y, Z)

Para un R, G o B, a $a = 0.055$ y $\gamma = 2.4$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.41224 & 0.357579 & 0.180464 \\ 0.212656 & 0.71518 & 0.072186 \\ 0.019332 & 0.119193 & 0.950444 \end{bmatrix} \begin{bmatrix} g(R/255) \\ g(G/255) \\ g(B/255) \end{bmatrix} \quad (12)$$

Donde:

$$g(K) = \left(\frac{K+a}{1+a}\right)^{\Upsilon} para\ K > 0.04045 \qquad (13)$$
$$si\ no, g(K) = \frac{K}{12.92}$$

C. Coordenadas cromáticas (X, Y, Z) a L*a*b*

$$L^* = 116f\left(Y/Y_n\right) - 16 \qquad (14)$$

$$a^* = 500\left[f\left(X/X_n\right) - f\left(Y/Y_n\right)\right] - vv \qquad (15)$$

$$b^* = 200\left[f\left(Y/Y_n\right) - f\left(Z/Z_n\right)\right] \qquad (16)$$

Donde

$$f(t) = \begin{cases} t^{1/3} & Para\ t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & Para\ otro\ valor \end{cases} \qquad (17)$$

Donde X_n, Y_n, Z_n son los valores triestimulo XYZ del punto de blanco de referencia. Para un $Observador = 2°, Iluminación = D65,$

$$X_n = 95{,}047$$
$$Y_n = 100.000 \qquad (18)$$
$$Z_n = 108.883$$

La división de la función f(t) en dos dominios se hace prevenir una pendiente infinita en t=0. Se asume que la función f(t) sea lineal para u tiempo $t = t_0$ y que coincide con la raíz cubica de t para $t = t_0$, tanto en valor como en pendiente. En otras palabras, esto se puede expresar como:

$$t_0^{1/3} = at_0 + b \qquad (coincicencia\ en\ valor) \quad (19)$$

$$\frac{1}{3}t_0^{-2/3} = a \qquad (coincidencia\ en\ pendiente) \quad (20)$$

El valor de b fue ajustado a 16/116. Las anteriores ecuaciones pueden ser resueltas para *a* y t_0.

$$a = \frac{1}{3}\delta^{-2} \qquad = 7.787037\ldots \quad (21)$$

$$t_0 = \delta^3 \qquad = 0.008856\ldots \quad (22)$$

Donde $\sigma = {}^{6}/_{29}$. Nótese que ${}^{16}/_{116} = {}^{2\delta}/_{3}$

2.7.2. ***Promedio o media de los elementos de la matriz***

B = mean2 (A) calcula la media es el valor promedio de un conjunto de datos numéricos, calculada como la suma del conjunto de valores dividida entre el número total de valores en la matriz A, La media [14].

2.7.3. ***Distancia euclidiana***

La distancia euclidiana entre dos puntos en el plano o en el espacio tridimensional mide la longitud de un segmento que conecta los dos puntos. Es la forma más obvia de representarla distancia entre dos puntos. El Teorema de Pitágoras se puede utilizar para calcular la distancia entre dos puntos, por lo tanto. Si los puntos (x1, y1) y (x2, y2) están en un espacio bidimensional, entonces la distancia euclidiana entre ellos es[35].

$$\sqrt{(x2 - x1)^2 + (y2 - y1)^2} \quad (23)$$

Una generalidad de la ecuación 23 para una matriz, puede ser reescrita como se observa en la ecuación 24.

$$d^2_{(i,j)} = \sum_{k=1}^{k} (X_{ik} - X_{jk})^2 \quad (24)$$

2.7.4. *Elementos mínimos de una matriz*

M = min(A), Si A es una matriz multidimensional, entonces min (A) opera a lo largo de la primera dimensión de la matriz cuyo tamaño no es igual a 1, tratando los elementos como vectores. El tamaño de esta dimensión se convierte en 1, mientras que los tamaños de todas las demás dimensiones siguen siendo los mismos. Si A es una matriz vacía con la primera dimensión 0, entonces min (A) devuelve una matriz vacía con el mismo tamaño que A [14].

$$\left(\mu_k, Z_k^i\right) = argmin_{\left(Z_k^i, \mu_k\right)} J \quad (25)$$

2.8. APRENDIZAJE PROFUNDO

El aprendizaje profundo o deep learning es un tipo de aprendizaje automático que enseña al ordenador a realizar algo natural para el ser humano: aprender a partir de la experiencia. Consigue llevar a cabo tareas de clasificación a partir de imágenes, textos o sonidos directamente, es decir, a partir de los datos sin depender de ecuaciones o modelos establecidos [31].

El aprendizaje profundo es especialmente adecuado para aplicaciones de identificación como pueden ser el reconocimiento facial, la traducción de texto o el reconocimiento de voz. Este tipo de aprendizaje se llama “profundo” por el número de capas que tiene la red neuronal. Mientras una red neuronal tradicional solo constituye de 2 o 3 capas, una red de aprendizaje profundo puede tener cientos de ellas.

Lo que hace que el aprendizaje profundo sea una tecnología de vanguardia es su exactitud. Dicho grado de precisión es posible especialmente gracias a tres cosas. Una de ellas es el fácil acceso a conjuntos masivos de datos etiquetados, como ImageNet o PASCAL VoC. Otra, que el poder de la computación se halla en aumento, ya que existen GPU de alto rendimiento las cuales aceleran el entrenamiento de datos, sometiendo el tiempo de entrenamiento de semanas a tan solo horas. Por último, el hecho de que se tenga acceso a modelos preentrenados, como puede ser AlexNet, facilita la creación de nuevas redes neuronales a

partir de estos modelos. AlexNet ha sido entrenada con más de un millón de imágenes en alta resolución para reconocer mil objetos diferentes [36].

2.9. REDES NEURONALES

Las redes neuronales están compuestas de elementos simples operando en paralelo. Se usan para llevar a cabo de tareas complicadas en muchas materias, como el reconocimiento de patrones, identificación, clasificación, lenguaje, visión y control de sistemas.

En la Figura 14, se observa la imagen general de una red neuronal de topología clásica, la cual está compuesta por un conjunto organizado de capas y los pesos de las neuronas son representados por las flechas que las interconectan entre sí. En primer lugar, todas la ANN cuentan con una capa de entrada de los datos, la cual recibe las características xn del patrón a clasificar; seguido a esto se encuentran las capas ocultas, el número de estas capas además del número de las neuronas depende del entrenamiento y de la aplicación para la cual es usada la ANN. Finalmente, se tiene la capa de clasificación, la cual arrojará la etiqueta final del dato clasificado; es importante aclarar que la etiquetas que recibe la red neuronal son etiquetas binarias y que el número de neuronas de la capa de clasificación depende del número de clases en las que están divididos los datos.[27]

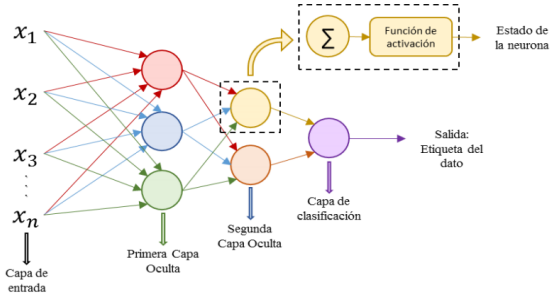


Figura 14. Red neuronal artificial de topología clásica, explicación de la topología general y de la neurona como unidad.

Una red neuronal de aprendizaje profundo consta de múltiples capas de proceso no lineales: una capa de entrada, varias capas ocultas y una capa de salida. Las capas están interconectadas a través de neuronas, y cada capa oculta usa como entrada la salida de la capa anterior. En la Figura 15 se esquematiza la arquitectura de la red [37].

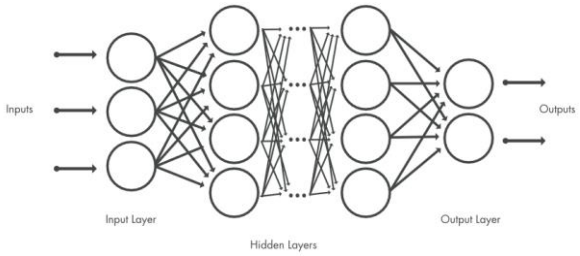


Figura 15. Representación gráfica esquematizada de una red neuronal[38].

Para entrenar una red neuronal lo principal es seleccionar un conjunto de imágenes suficientemente grande como para que la red aprenda. Si por ejemplo quisiéramos hacer la clasificación de cuatro objetos diferentes, deberíamos conseguir imágenes en las que aparezca una de las cuatro categorías diferentes de objetos que hayamos seleccionado.

El objetivo es que la red sea capaz de reconocer automáticamente alguno de esos objetos en una imagen. Las imágenes que servirán para entrenar la red deben estar etiquetadas, ya que la red comprenderá las características específicas del objeto en cuestión al asociarlas con la categoría a la que pertenece. En cada capa de la red neuronal se toman datos de la anterior, que son transformados y pasados a la siguiente, así que la red se vuelve cada vez más compleja. La red aprende directamente a partir de los datos, por lo que no podremos influenciar sobre las características que esté aprendiendo [38].

2.9.1. ***Aumento de datos (Data Augmentation)***

Data augmentation es la generación artificial de datos son técnicas que se utilizan para aumentar la cantidad de datos agregando copias ligeramente modificadas de datos ya existentes o datos sintéticos recién creados a partir de datos existentes, Esto nos permite aumentar tanto en tamaño como en diversidad nuestro set de datos de entrenamiento. Esta técnica se convirtió en un estándar de regularización, y también para mejorar la performance y combatir el sobreajuste en CNN [31].

2.10. REDES NEURONALES CONVOLUCIONALES

Las redes neuronales convolucionales se usan a menudo para técnicas de aprendizaje profundo, especialmente para las que usan imágenes como datos de entrada. Otra característica de una red neuronal convolucional, que la diferencia del resto, es que sus neuronas producen resultados correlacionados, mientras que las de otros tipos de redes no comparten ninguna conexión, produciendo resultados independientes. Una red neuronal

convolucional consta de distintos tipos de capas: capas convolucionales, capas de agrupación máxima o de acumulación media y capas completamente conectadas [39].

La arquitectura de una red neuronal convolucional puede varían los tipos y número de capas incluidas, y se seleccionan en función precisa de una respuesta categórica o continua, Por lo general, la red constará de una capa de entrada, una de salida, diversas capas e intermedias. Dichas capas pueden dividirse en dos grupos, las capas de detección de funciones y las de clasificación.

Entre las capas de detección de funciones hallamos tres tipos según la operación que realicen: convolución. La primera coloca las imágenes de acceso a través de un conjunto de filtros convolucionales, cada uno de los cuales activa ciertas características de las imágenes. Las de agrupación simplifican la salida ya que realizan un muestreo descendente no lineal para reducir el número de medidas sobre los que la red necesita aprender. Por último, permiten que el entrenamiento sea más rápido y efectivo de valores negativos a cero y manteniendo valores positivos. Estas tres sistematizaciones se repiten en decenas o cientos de capas, con cada una de ellas la red aprende a detectar diferentes características [40].

Las redes neuronales convolucionales se inspiran en la estructura biológica de una corteza visual, que contiene disposiciones de células simples y complejas. Se encuentra que estas celdas se activan en función de las subregiones de un campo visual. Estas subregiones se denominan campos receptivos. Inspiradas en los hallazgos de este estudio, las neuronas en una capa convolucional se conectan a las subregiones de las capas antes de esa capa en lugar de estar completamente conectadas como en otros tipos de redes neuronales. Las neuronas no responden a las áreas fuera de estas subregiones en la imagen.

Estas subregiones pueden superponerse, por lo tanto, las neuronas de una ConvNet producen resultados correlacionados espacialmente, mientras que, en otros tipos de redes neuronales, las neuronas no comparten ninguna conexión y producen resultados independientes.

Además, en una red neuronal con neuronas completamente conectadas, la cantidad de parámetros puede aumentar rápidamente a medida que aumenta el tamaño de la entrada. Una red neuronal convolucional reduce el número de pesos con el número reducido de conexiones, pesos compartidos y reducción de resolución.

Una ConvNet consta de varias capas, como capas convolucionales, capas de agrupación máxima o agrupación promedio y capas completamente conectadas.[40][41]

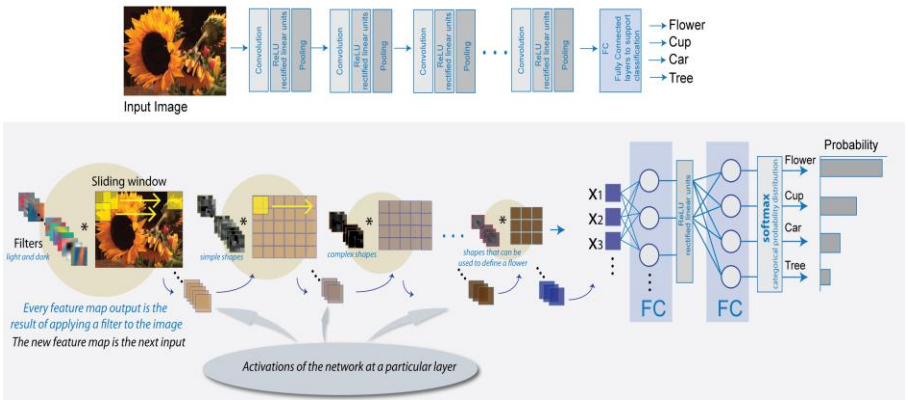


Figura 16. Redes neuronales convolucionales

2.10.1. ***Convoluciones***

La capa convolucional de las CNN es el factor principal de estas, debido a que permite que se reciba el patrón puro a clasificar y no características extraídas del mismo.

Como su nombre lo indica, la capa convolucional cuenta con una operación llamada convolución. La convolución es la operación del cálculo del área bajo la curva del producto entre dos funciones, una de estas rotada en torno al origen como se ve en la ecuación 26, y en función de un tiempo continuo τ.

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \quad (26)$$

Contrario a lo anterior, para las aplicaciones de aprendizaje de máquina, la convolución se realiza en tiempo discreto y generalmente sobre arreglos multidimensionales; para el caso de las CNN los arreglos son: segmentos de píxeles de la imagen de entrada $\big(l(m.n)\big)y\ el\ kernel\ (K(i-m, j-n))$, que es el parámetro adaptado por el algoritmo de aprendizaje.

Lo anterior, significa que teniendo en cuenta la ecuación 26 se puede hacer el cálculo de la convolución entre los arreglos como una sumatoria del desplazamiento del uno sobre el otro, sin embargo, no puede ser infinita debido a que los arreglos tienen la obligatoriedad de que son de dimensión finita.

$$S(k,l) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \quad (27)$$

De esta manera, en la ecuación 27, se muestra la convolución en dos dimensiones para el procesamiento de datos en las CNN. Para este caso m y n son las dimensiones de la imagen filas y columnas, respectivamente, y las variables i y j, indican el corrimiento del kernel en filas y columnas sobre la misma [27]

2.10.2. ***Filtros espaciales (kernels) aplicarlo en CNN***

Recorre toda la imagen de pixel a pixel, y realiza alguna operación aritmética con un numero de pixeles vecinos. Al conjunto de estos vecinos se les denomina ventana, la cual podrá ser de 3x3, 5x5,... Son las operaciones que se realizan directamente sobre los pixeles. Se define una matriz de coeficientes del filtro (mascara, tamaño mxn) los cuales constituirán pesos ponderados por los que se multiplicarán los valores originales de pixeles Convolución es el tratamiento de una matriz por otra que se llama "kernel".

En la Figura 17 se puede ver que dicho kernel se mueve a través de la imagen original, se define una ventana móvil que contiene un arreglo de factores ponderales. Estos arreglos se definen como operadores o kernels, cuyo tamaño es normalmente el de un número impar de pixeles (3x3, 5x5, 7x7, etc) [14].

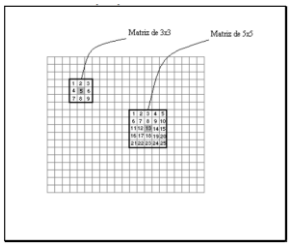


Figura 17. Representación de dos tipos de ventana para filtrado de imágenes.[14]

Por un lado, la propia imagen a la que aspiramos aplicar la convolución (en forma de matriz, claro), y otra matriz más pequeña llamada kernel. Es el efecto de la convolución es otra matriz de las mismas dimensiones que la imagen original. Para deducir la matriz de convolución vamos desplazando el kernel por cada uno de los elementos de la matriz principal y hacemos la suma de los productos de cada uno de los elementos de ambas matrices.

2.10.3. *Capa de normalización por lotes (BatchNormalization)*

Una capa de normalización por lotes normaliza un mini lote de datos en todas las investigaciones para cada canal de forma independiente. Para acelerar el entrenamiento de la red neuronal convolucional y comprimir la sensibilidad a la inicialización de la red, usa capas de normalización por lotes entre capas convolucionales y no linealidades, como las capas ReLU.

Para una capa con entradas d-dimensionales x = (x1, x2, ... xd) obtenemos la normalización con la siguiente fórmula (con la expectativa de la desviación estándar, una media cero, calculadas sobre un lote B) aplicamos la siguiente normalización [31].

$$\hat{x}^{(k)} = \frac{x^{(k)} - E\left[x^{(k)}\right]}{\sqrt{VAR[x^{(k)}]}} \quad (28)$$

2.10.4. *Capa de agrupación máxima (Max-Pooling)*

Una capa de agrupación máxima realiza un muestreo reducido partiendo la entrada en regiones de agrupación rectangulares y calculando el máximo de cada región.

La agrupación máxima fraccionada es un algoritmo de agrupación desarrollado que utiliza una relación de agrupación fraccionaria a diferencia de los enfoques generales de agrupación máxima, donde la agrupación generalmente se realiza en una analogía de enteros. Una relación de agrupación fraccionada permite una mejor escala de las imágenes y nos permite traer un mayor número de capas convolucionales para aprender mejor la imagen a diferentes escalas. El uso de series pseudoaleatorias agrega aleatoriedad a la operación de agrupación y accede a aprender características más sólidas para la clasificación [31].

Dada una matriz A podemos definir el proceso de Max-Pooling con una amplitud k y un stride p como la matriz P(i,j) tal que:

$$P(i,j) = \underset{n,m\,=\,1,\ldots,k}{máx} A[(i-1)p+m, (j-1)p+n] \quad (29)$$

Si fuera más necesarias más filas o columnas para el desarrollo del algoritmo se pueden incluir más filas y columnas mediante el método de zero-padding [42].

2.10.5. ***Capa de abandono (Dropoutlayer)***

El abandono es una técnica que se maneja para mejorar el sobreajuste en las redes neuronales; debe utilizar el abandono inmediato con otras técnicas como la regularización L2, durante el entrenamiento, la mitad de las neuronas de una capa en particular se desactiva. Esto mejora la generalización porque obliga a la capa a aprender con diferentes neuronas.

Estas capas de exclusión normalmente son algunos modelos de aprendizaje profundo usan Dropout en las capas completamente conectadas, pero también es posible usar Dropout después de las capas de agrupación máxima, creando algún tipo de aumento de ruido de imagen [43].

2.10.6. ***Capa de probabilidad para todas las clases (softmaxLayer)***

Una capa softmax a la entrada se utiliza para la clasificación de múltiples categorías. En general, para una tarea de clasificación de categoría N, el número de neuronas en la capa softmax es N. Desde el punto de vista de la teoría probabilística, la salida de la neurona m-ésima en la capa softmax es la probabilidad condicional del caso de prueba actual perteneciente a la categoría m-ésima [44].

2.10.1. ***Capa completamente conectada (fully connected layer)***

Es una capa totalmente conectada en una red convolucional son prácticamente un perceptrón multicapa (generalmente un MLP de dos o tres capas) que tiene como objetivo mapear el $m_1^{(l-1)} \times m_2^{(l-1)} \times m_3^{(l-1)}$, activación a partir de la combinación de capas diferentes anteriores en una distribución de probabilidad de clase cada una de las salidas en la capa anterior está conectada con un peso a cada una de las neuronas de la capa. [31].

La capa completamente conectada de un vector, se define como:

Si $l-1$ es una capa completamente conectada;

$$\Upsilon_i^t = f(z_i^{(l)}) \quad con\ z_i^{(l)} = \sum_{j=1}^{m_1^{(l-1)}} w_{i\,j}^{(l)} \gamma_{i\,j}^{(l-1)} \qquad (30)$$

$$\Upsilon_i^t = f(z_i^{(l)}) \quad con\ z_i^{(l)} = \sum_{j=1}^{m_1^{(l-1)}} \sum_{r=1}^{m_2^{(l-1)}} \sum_{s=1}^{m_3^{(l-1)}} w_{i,j,r,s}^{(l)} \left(Y_i^{(l-1)}\right)_{r,,s} \quad (31)$$

El objetivo de la estructura completamente conectada es ajustar los parámetros de peso $w_{i,j}^{(l)}$ o $w_{i,j,r,s}^{(l)}$ para crear una representación estocástica de verosimilitud de cada clase basada en los mapas de activación generados por la concatenación de capas convolucionales, de no linealidad, rectificación y agrupación. Las capas individuales completamente conectadas operan de manera idéntica a las capas del perceptrón multicapa con la única excepción de la capa de entrada[45]

2.10.2. ***Capa de salida de clasificación (ClassificationLayer)***

Una capa de clasificación computa la pérdida de entropía cruzada para trabajos de clasificación y clasificación contenida con clases recíprocamente excluyente. La capa deduce el número de clases del tamaño de salida de la capa anterior. Por ejemplo, para especificar el número de clases K de la red, incluye una capa totalmente conectada con tamaño de salida K y una capa softmax antes de la capa de clasificación [31].

2.11. Neural Network Toolbox en MatLab

Para realizar la red neuronal convolucional se ha utilizado Neural Network Toolbox en MatLab. Esta proporciona comandos simples de MatLab para crear e interconectar las capas de una red neuronal profunda.

El entrenamiento de un modelo de aprendizaje profundo puede tomar horas, días o semanas, dependiendo del tamaño de los datos y la cantidad de potencia de procesamiento que tenga disponible. Para evitar la necesidad de utilizar tantos datos y activar la creación de la red neuronal, se usa una técnica popular como transfer learning. El aprendizaje de transferencia, toma una red reentrenada y la usa como punto de partida para aprender la nueva tarea que se requiera. Adaptar una red reentrenada a nuestras necesidades es además de más rápido, más posible que construir y preparar de cero una nueva red. La red destinada ya ha aprendido un amplio conjunto de tipos que puede utilizar en el nuevo entrenamiento con nuestro conjunto de datos [45].

Actualmente existen tres opciones de computación, y es conveniente para el modelo que se esté ejecutando también es una pieza clave en el trabajo. Estas iniciativas son: basada en GPU y basada en la nube, es la opción más simple y disponible, recomendado solo para ejemplos simples utilizando una red destinada. El uso de una GPU reduce el tiempo de preparación de la red de días a horas. El cálculo de la GPU basado en la nube no necesita que se compre ni configure el hardware [45].

2.12. Red neuronal de regresión generalizada (GRNN)

La red neuronal de regresión generalizadas, se utiliza a menudo para la aproximación de funciones. Se ha demostrado que, dado un número suficiente de neuronas ocultas, los GRNN pueden aproximarse a una función continua con una precisión arbitraria [46]. GRNN se puede considerar como una base radial normalizada Red de función (RBF) que tiene una capa de base radial y una capa lineal especial. Estas unidades RBF se denominan kernels. y suelen ser funciones de densidad de probabilidad como la Gaussiano.

Los pesos ocultos a la salida son solo el objetivo valores, por lo que la salida es simplemente un promedio ponderado de los valores objetivo de casos de entrenamiento cercanos al caso de entrada dado. La primera capa es como una red RBF con tantas neuronas ya que hay vectores de entrada / destino. Elegir el parámetro de dispersión de la capa de base radial determina el ancho de un área en el espacio de entrada, a la que cada neurona responde. La propagación debe ser lo suficientemente grande como para que las neuronas responden fuertemente a regiones superpuestas del espacio de entrada.

Si la propagación es pequeña, el RBF es muy pronunciado, por lo que la neurona con el vector de peso más cercano a la entrada tendrá una producción mucho mayor que otras neuronas. La red tiende a responder con el vector objetivo asociado con el vector de entrada de diseño más cercano. A medida que la propagación se hace más grande, La pendiente de RBF se vuelve más suave y varias neuronas pueden responder a un vector de entrada. La red entonces actúa como si estuviera tomando un promedio ponderado entre los vectores objetivo cuyo diseño los vectores de entrada son los más cercanos al nuevo vector de entrada. [47]

2.12.1. ***Arquitectura de red de función de base radial***

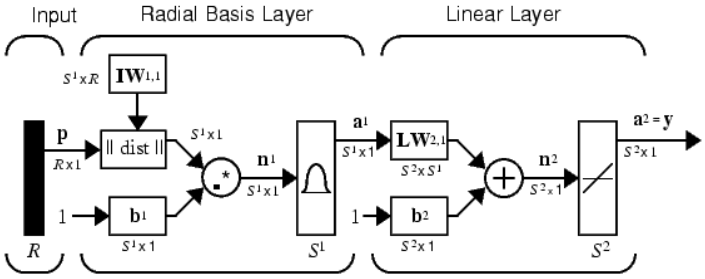


Figura 18. Arquitectura de RBF [46]

$$a_i^1 = radbas\ (\|I_i W^{1,1} - p\| b_i^1) \quad (32)$$

$$a^2 = purelin\ (LW^{2,1} a^1 + b^2) \quad (33)$$

a_i^1 es i elemento de a^1 donde $I_i W^{1,1}$ es un vector hecho de la i^{th} fila de $IW^{1,1}$ [46]

Donde:

R= número de elementos en el vector de entrada

S^1= número de neuronas en la capa 1

S^2= número de neuronas en la capa 2

2.12.2. ***Arquitectura de red neuronal de regresión generalizada***

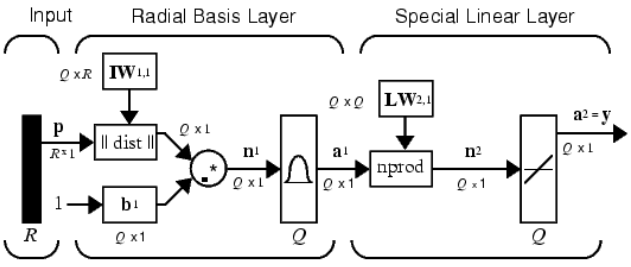


Figura 19. Estructura GRNN [46]

$$a_i^1 = radbas\ (\|I_i W^{1,1} - p\| b_i^1) \quad (34)$$

$$a^2 = purelin\ (n^2) \quad (35)$$

a_i^1 es i^{th} elemento de a^1 donde $I_i W^{1,1}$ es un vector hecho de la i^{th} fila de $IW^{1,1}$[46]

Donde:

R= N° de elementos en el vector de entrada

Q= N° de neuronas en la capa 1

Q= N° de neuronas en la capa 2

Q = número de pares de entrada / destino

2.12.3. ***Matriz de confusión***

La Matriz de Confusión, también llamada tabla de contingencia es una herramienta que nos muestra el desempeño de un algoritmo de clasificación, describiendo cómo se distribuyen los valores reales y nuestras predicciones, y también son métricas más intuitivas y sencillas que se utiliza para encontrar la precisión y exactitud del modelo. Se utiliza para el problema de clasificación donde la salida puede ser de dos o más tipos de clases.

Figura 20. Matriz de confusión

La fila Positive son aquellos elementos que han sido clasificados como positivos, y la fila Negative son los que han sido clasificados como negativos. Por otro lado, la columna Positive

son los que, en realidad, son positivos, y la columna Negative los que, en realidad, son negativos (téngase en cuenta que los conceptos de "positivo" o "negativo" son completamente arbitrarios).[48]

3

3. ELEMENTOS DEL AMBIENTE DE TRABAJO, ENSAYOS Y ADQUSICIÓN DE DATOS

En este capítulo se describirán los diferentes ambientes de trabajo desarrollados y los distintos elementos que componen cada ambiente, esto con la finalidad de proporcionar al lector una perspectiva más específica del espacio de trabajo.

3.1. Elementos del ambiente de trabajo

Los elementos usados para crear el ambiente como símil al juego Craps con la finalidad de realizar los respectivos ensayos y la adquisición de datos (Imágenes del espacio de trabajo) son:

3.1.1. ***Trípode***

El mecanismo usado para sostener el dispositivo Android con cámara corresponde a un trípode marca Beston modelo 3110A con las siguientes características:

- ✓ Hecho a base de aluminio, Duradero y Resistente
- ✓ Cabeza L : Plástico
- ✓ Altura máxima: 105 cm
- ✓ Altura mínima: 35 cm
- ✓ Peso: 580 g
- ✓ Patas ajustables ideales para ubicar en superficies irregulares
- ✓ Plataforma vertical de 90°
- ✓ Nivelador horizontal para unas mejores fotos y vídeos.

- ✓ Piernas de liberación rápida con seguros antideslizantes en pies plásticos ajustables.

El mecanismo utilizado dentro del entorno de trabajo se muestra en la Figura 21.

Figura 21. Trípode usado.

3.1.2. ***Tapete***

Para poder representar el juego Craps se usó un tipo de papel fotográfico mate y así obtener el entorno de trabajo; Las dimensiones de este tapete son de 90x45cm, como se observa en la Figura 22.

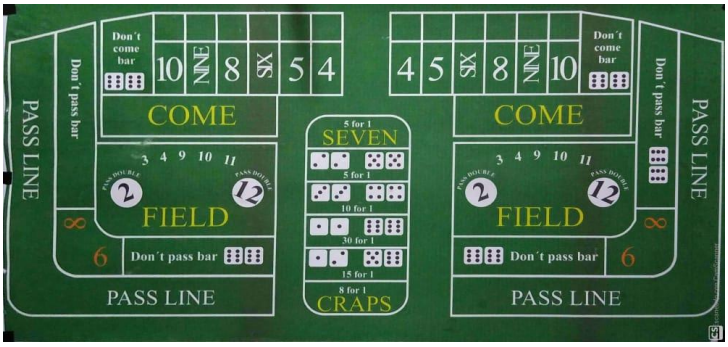


Figura 22. Tapete en el entorno de trabajo

3.1.3. ***Luz***

La luz usada para iluminar el entorno de trabajo es un tubo led tipo lineal de marca ENERLUX, tiene las siguientes características:

- ✓ Referencia : E-TLT818G13P
- ✓ 18W Lumenes: 1600 Lm +15%
- ✓ Vida útil: 30.000 horas
- ✓ Voltaje: 100-240v 50/60Hz
- ✓ Temperatura de color: 6000K – 6500K

Un ejemplo del tubo led utilizado dentro del entorno de trabajo corresponde al observar en la Figura 23.

Figura 23. Tubo led 18W [49].

3.1.4. ***Dispositivo Android y IOS con cámara***

El dispositivo Android usado para adquirir las imágenes fue Moto G7 POWER ya que permite mayor alcance de adquisición de imágenes; su resolución de lente o de cámara trasera es de mayor alcance a comparación con otros dispositivos, tiene las siguientes características:

- ✓ PROCESADOR: Snapdragon 632
- ✓ CÁMARA TRASERA: 12 megapíxeles f/2.0, PDAF, flash LED
- ✓ BATERÍA: 5.000 mAh con Turbo Charge
- ✓ SOFTWARE: Android 9 Pie

Un ejemplo del dispositivo Android con cámara, como en este caso es Motorola G7 Power utilizado dentro del entorno de trabajo para la adquisición de imágenes corresponde al observar en la Figura 24.

Figura 24. Motorola G7 Power

El dispositivo IOS usado para adquirir videos fue IPHONE X ya que permite mayor resolución de lente o de cámara trasera es de mayor alcance a comparación con otros dispositivos en cámara lenta (1080p a 120 o 240 FPS) estos nos permiten mayor uso para la predicción del resultado del juego, tiene las siguientes características:

- ✓ RESOLUCIÓN DE CAMARA TRASERA: 2.436 x 1.125 píxeles
- ✓ PROCESADOR: Apple A11 Bionic
- ✓ ALMACENAMIENTO: 64/256 GB
- ✓ CÁMARA TRASERA: 12 megapíxeles f/2.0, PDAF, flash LED
- ✓ BATERÍA: 2.716 mAh
- ✓ VIDEO A CAMARA LENTA: 2.716 mAh

Un ejemplo del dispositivo IOS, como en este caso es IPHONE X utilizado dentro del entorno de trabajo para la predicción del resultado corresponde al observar en la Figura 25.

Figura 25. IPHONE X

3.2. Ambiente de trabajo

El total de ambientes probados durante el desarrollo del proyecto son tres, de los cuales el ultimo es usado para ser aplicado en el proyecto (Obsérvese Figura 26). Con la finalidad de explicar al lector la decisión de trabajar con el ultimo ambiente acá expuesto, se presentan a continuación argumentos que relacionan cada ambiente con sus ventajas y desventajas.

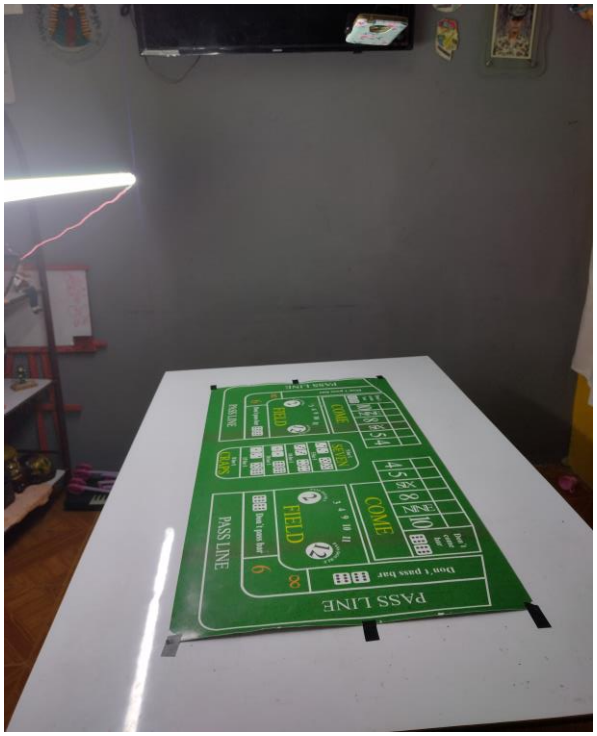


Figura 26. Ambiente aplicado durante el proyecto.

3.2.1. ***Primer ambiente de trabajo***

En este primer entorno de trabajo se obtuvo inconvenientes como la posición del trípode sobre la mesa como se puede ver en la Figura 27, ya que la disposición de este no era la más adecuada, porque obstruía la visualización del entorno de trabajo por lo tanto no se pudo adecuar para el dispositivo móvil con sistema operativo Android y se dificultaba más para la adquisición de imágenes.

Figura 27. Primer entorno de trabajo

Una de las dificultades del primer entorno de trabajo se puede ver en la Figura 28, como se nombraba anteriormente la posición del trípode no era la más adecuada ya que obstruía la zona del juego y por lo tanto al momento de adquirir la imagen y hacer la segmentación se pierde parte de la información, como se puede ver en la Figura 28 a) hay un objeto, el cual era una de las patas del trípode y en la Figura 28 b) se observa un dado y una de las patas del trípode, la posición del trípode en espacio de trabajo no era la más adecuada.

Figura 28. Problemas de segmentación del primer entorno de trabajo

3.2.2. ***Segundo ambiente de trabajo***

En este segundo entorno de trabajo se tuvo dificultades ya que se cambió de lugar de trabajo, uno de ellos es la ubicación del tapete para dar solución a este problema se ubicó el tapete sobre una superficie plana, pero la posición de la luz esta se reflejaba sobre el tapete y no permite una buena segmentación del resultado obtenido por el dispositivo móvil con sistema operativo Android como se puede ver en la Figura 29.

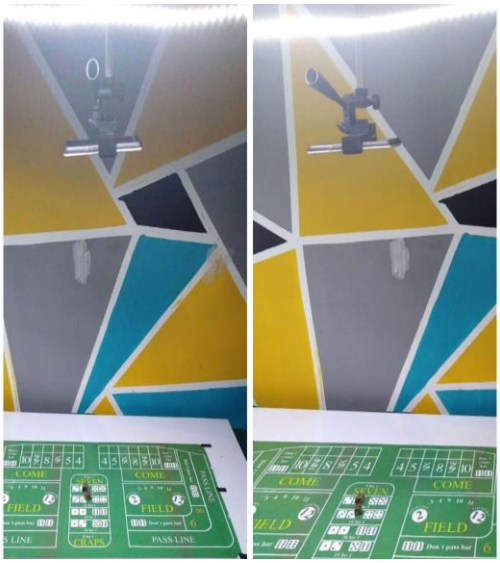


Figura 29. Segundo entorno de trabajo

Una de las dificultades del segundo entorno de trabajo se puede ver en la Figura 30, como se nombraba anteriormente la posición de la luz no era la más adecuada ya que esta incidía sobre la zona de trabajo, y dificultaba más el proceso al hacer la segmentación de la imagen.

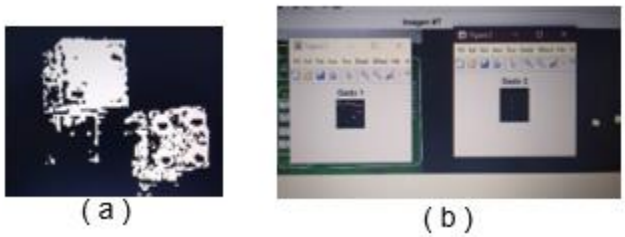


Figura 30. Problemas de segmentación del segundo entorno de trabajo a) deformación del objeto, b) baja definición en el recorte

3.2.3. ***Tercer ambiente de trabajo***

En este tercer y último entorno de trabajo se solucionaron todos los inconvenientes de los dos anteriores, como la posición del trípode, en este caso el trípode se suspendió sobre el tapete y así poder sujetar nuestro dispositivo móvil con sistema operativo Android, el tapete se ubicó sobre una superficie plana, igualmente la posición de la luz esta se suspendió a un costado del entorno de trabajo y así asegurando que la luz no reflejara sobre el tapete como se puede ver en la Figura 31.

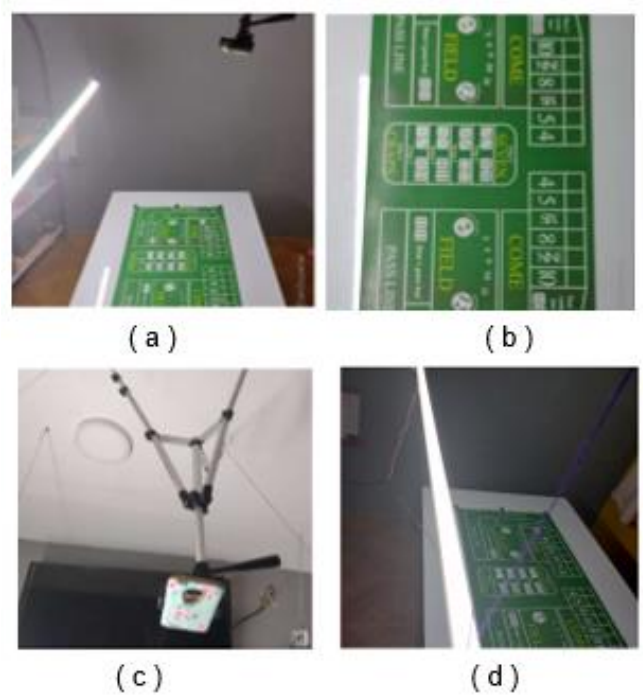


Figura 31. Tercer entorno de trabajo a) superficie de trabajo, b) luz del entorno de trabajo, c) trípode encargado de sostener el dispositivo de adquisición, d) entorno de trabajo

Solucionado los inconvenientes anteriormente mencionados, este fue seleccionado como el entorno de trabajo y la segmentación final se puede ver en la Figura 32.

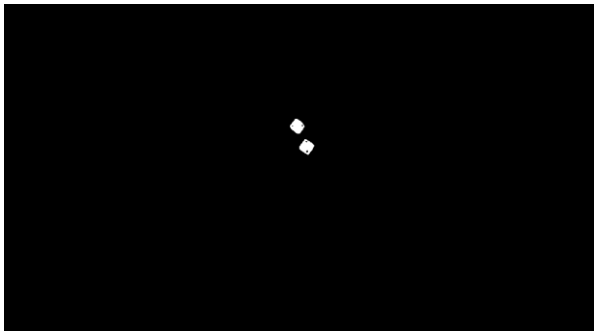


Figura 32 Segmentación de dados del tercer entorno de trabajo

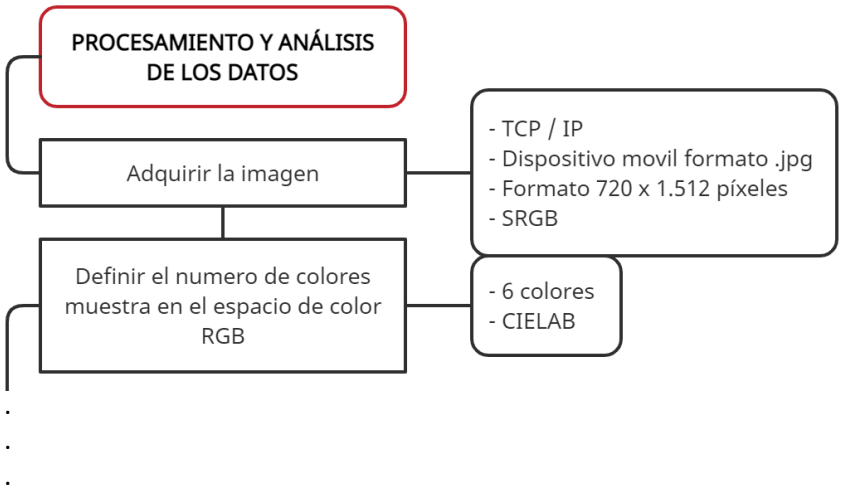
4

4. CAPTURA, PROCESAMIENTO Y ANÁLISIS DE LOS DATOS DE INTERÉS

Esta fase está dedicada a describir el proceso que se realizó para identificar los objetos por su campo de color, inmersos en la imagen, esto, con la finalidad de detectar la posición en coordenadas (x,y) de los dados en la imagen para su respectiva extracción o recorte. Este capítulo está dividido en dos secciones, la recolección de datos recolectados.

4.1. Diagrama de flujo de la metodología usada

La Figura 33 presenta un bosquejo del algoritmo desarrollado para la segmentación de objetos de una imagen mediante su representación en el espacio de color en l*a*b*.



. . .

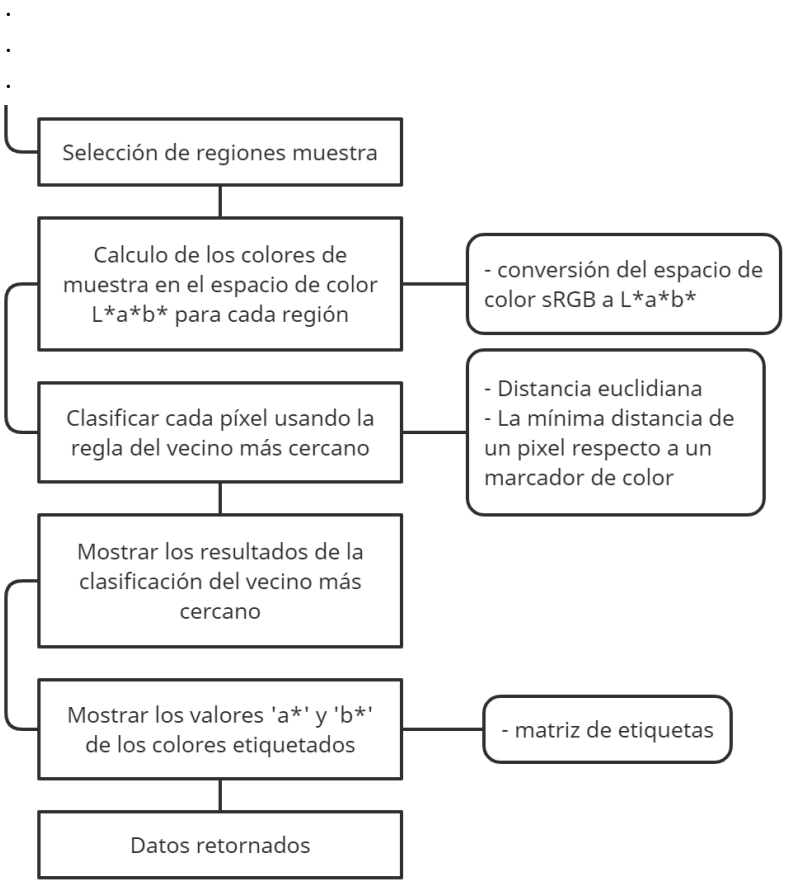


Figura 33. Metodología del capítulo 4

4.2. Recolección de datos

Las imágenes del ambiente de trabajo se adquirieron mediante una rutina de 1200 iteraciones por sesión (3600 en 3) utilizando comunicación TCP/IP con el servidor de cámara web inicializado en un celular con sistema operativo Android a través de la aplicación IP Webcam; esta rutina se encontraba condicionada al momento de realizar el cambio entre una iteración y otra, dado que dependía crucialmente de una entrada manual (Presione una tecla) descrita como una bandera que representaba el estado final del lanzamiento de los dados.

No fue una tarea fácil el proceso de adquisición de imágenes dado que hubo problemas al momento de segmentarlas por color (revise "Conjunto de pasos para procesamiento y análisis"); esto se debía principalmente a que con el conjunto de imágenes, las cuales constan

de alrededor de 3600 imágenes divididas en tres sesiones distintas, donde en cada sesión se variaba en el tipo luz usada (ya sea esta artificial o ambiente), se cambiaba la posición de la luz y se ubicaba la cámara respecto al tapete de diferente forma, no se lograba alcanzar una segmentación adecuada de los datos de interés, por lo tanto se optó por uniformizar el color de los dados, solucionando así el problema de segmentación; Una muestra del tipo de imagen problema es ilustrada en la Figura 34.

Figura 34. Imagen tipo problema.

Y una muestra del tipo de imagen solución es ilustrada en la Figura 35,

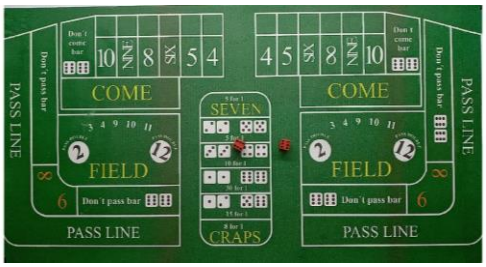


Figura 35. Imagen tipo solución.

La recolección de datos, aparentemente la tarea más sencilla del trabajo, fue una de las más agotadoras. Aunque en una primera instancia parecía que no sería necesario mucho esfuerzo para realizarla, resultó convirtiéndose en un desafío que claramente fue solucionado a cabalidad.

Cuando se inició el trabajo, el objetivo era recolectar pocas imágenes que describieran las posibles posiciones de los dados, inicialmente tomé un total de 40, sin embargo, en el proceso de segmentación desarrollado de manera heurística se evidencio que era necesario tomar un conjunto mayor de imágenes en todos los posibles esquemas de posicionamiento de los

dados y en lo posible con variaciones finas de iluminación, contraste y color; esta metodología fue aplicada, y el resultado es una pequeña base de datos con imágenes de tipo solución con un total de 1200 imágenes.

4.3. Conjunto de pasos para procesamiento y análisis de los datos

A continuación, se muestra cómo identificar los diferentes colores sobre el tapete analizando el espacio de color L*a*b*, precisando claro está, en el color de los dados, finalmente se presenta un ejemplo del tipo de variable de retorno tras ejecutar este conjunto de pasos.

4.3.1. ***Paso 1: Adquirir la imagen***

Inicialmente se lee una imagen mediante comandos de Matlab a través de una solitud TCP/IP al servidor cámara inicializado en el dispositivo Android, esta imagen se encuentra en formato .jpg y es procesada por Matlab como una imagen RGB tipo Uint8. Una muestra del tipo de imagen extraída mediante Matlab es apreciada en la Figura 36.

Figura 36. Imagen RGB capturada.

4.3.2. ***Paso 2: Definir el número de colores muestra en el espacio de color RGB***

Si se observa con detenimiento en la Figura 36 se pueden apreciar 6 colores principales en la imagen: el color de fondo (verde), el blanco, el negro, el naranja, el amarillo y el rojo. Ciertamente identificar cada color de forma visual es una tarea muy sencilla para un ser humano, de modo que se toma ventaja de tal conocimiento explicito para cuantificar estas diferencias visuales mediante el espacio de color L*a*b* (también conocido como CIELAB o CIE L*a*b*). El espacio de color L*a*b* se deriva de los valores triestímulos CIE XYZ. El espacio L*a*b* consiste en una luminosidad 'L*' o capa de brillo, una capa de cromaticidad 'a*' que indica dónde cae el color a lo largo del eje rojo-verde y una capa de cromaticidad 'b*' que indica dónde cae el color a lo largo el eje azul-amarillo.

4.3.3. ***Paso 3: Selección de regiones muestra***

Para realizar este paso, se trata la imagen de la Figura 37 mediante un bucle de tamaño igual al número de colores previstos para ser clasificados, en este caso un total de 6; de modo que, una pequeña región de muestra se extrae en cada iteración y es etiquetada según el color que corresponda.

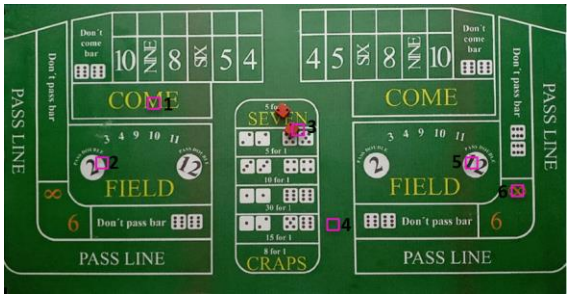


Figura 37. Regiones muestreadas para los diferentes colores 1. amarillo, 2. blanco, 3. rojo, 4.verde, 5. negro y 6. naranja de la Figura 36.

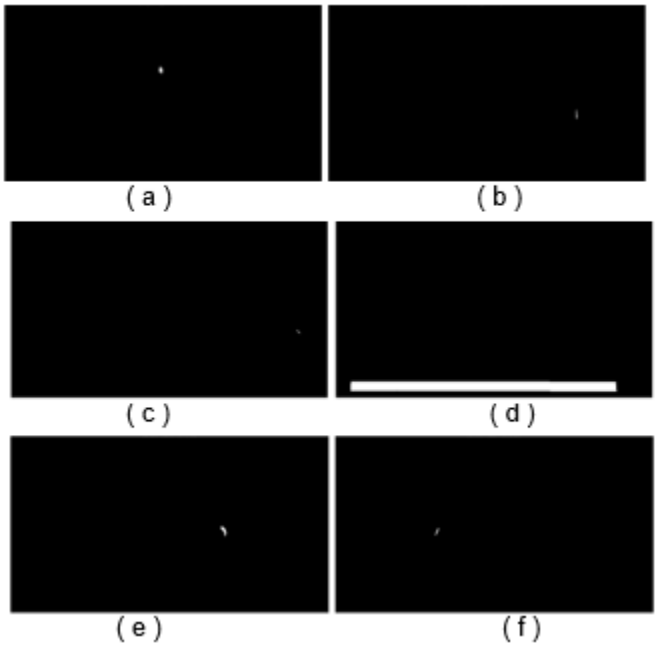


Figura 38. Regiones muestreadas para los diferentes colores a) rojo, b) amarillo, c) naranja, d) verde, e) blanco y f) negro de la Figura 37.

Un ejemplo de las seis muestras obtenidas para una misma imagen es ilustrado en la Figura 38, las cuales explican los colores de interés según la región polinomial trazada por color.

4.3.4. *Paso 4: Calculo de los colores de muestra en el espacio de color L*a*b* para cada región*

Para realizar el cálculo de los colores muestra o marcadores de color, se realiza la conversión del espacio de color sRGB al L*a*b* de la imagen de la Figura 36, luego se toman las componentes a*b* descritas por las regiones o coordenadas trazadas previamente y se calcula el valor medio de cada región extraída. Estos valores sirven como marcadores de color en el espacio 'a*b*', la Tabla 5 presenta los marcadores de color obtenidos, resultado de los cálculos realizados.

```
T =

  6×3 table

    Colores        a          b
    _______    ________    ______

    'red'        47.777    38.144
    'yellow'    -16.327    59.754
    'orange'     22.507    46.539
    'green'     -29.077    24.455
    'white'    -0.73269    2.7916
    'black'     3.2322    2.4652
```

Tabla 5. Marcadores de color en el espacio L*a*b*

4.3.5. *Paso 5: Clasificar cada píxel usando la regla del vecino más cercano*

Dado que cada marcador de color ahora tiene un valor 'a*' y un 'b*'. Se procede a clasificar cada píxel en la imagen original convertida al espacio de color L*a*b*, para ello se calcula la distancia euclidiana de cada píxel de la imagen respecto a cada marcador de color. La mínima distancia de un pixel existente entre un marcador de color específico determina cuanta semejanza existe entre ambos datos. Por ejemplo, si la distancia entre un píxel y el marcador de color rojo es la más pequeña, el píxel se etiquetaría como un píxel rojo, lo cual favorece la segmentación de los dos objetos de interés que son los dados.

4.3.6. *Paso 6: Mostrar los resultados de la clasificación del vecino más cercano*

La matriz de etiquetas obtenida tras aplicar el mínimo entre cada pixel y los diferentes marcadores de color presentes en la imagen original, permite desarrollar la segmentación deseada por color de objeto, tal como se observa en la Figura 39.

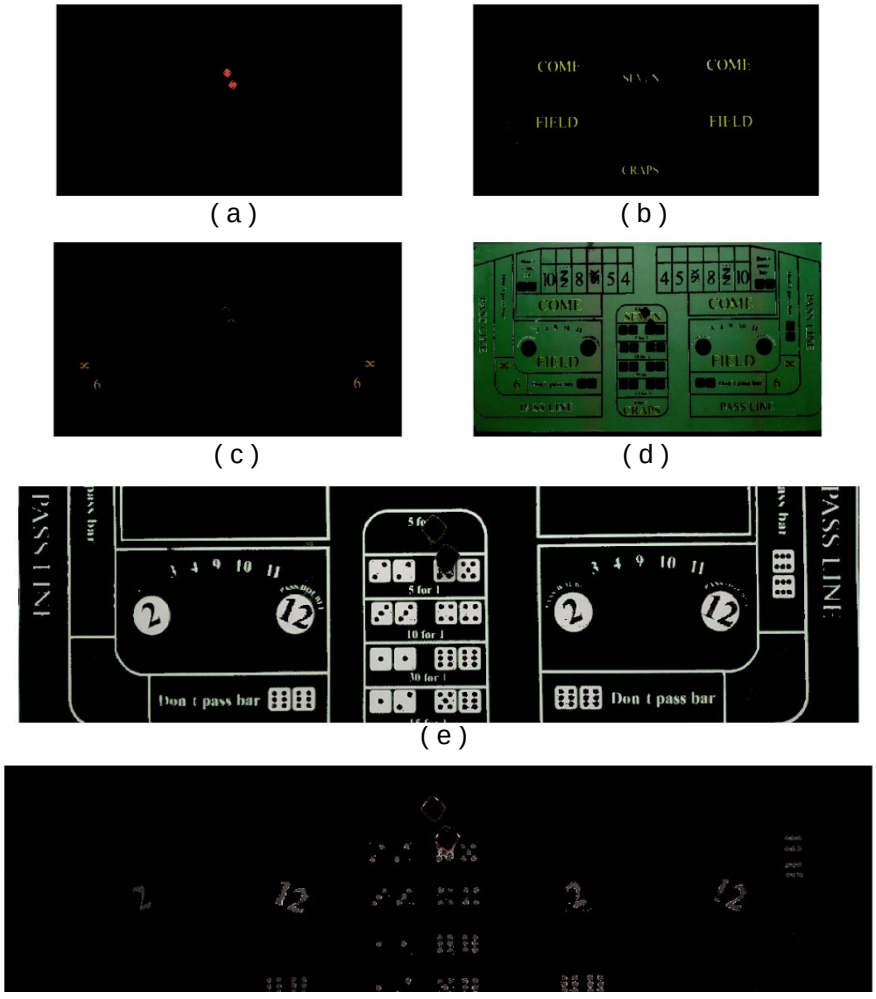


Figura 39. Segmentación de objetos por color, a) rojo, b) amarillo, c) naranja, d) verde, e) blanco y f) negro, respectivamente.

4.3.7. ***Paso 7: Mostrar los valores 'a*' y 'b*' de los colores etiquetados.***

La matriz de etiquetas a clasificar para representación en el espacio de color L*a*b* es mostrada en la Tabla 6, donde para rojo el color asignado a nivel de plot es rojo, para amarillo el color asignado a nivel de plot es amarillo, para naranja el color asignado a nivel de plot es azul oscuro, para verde el color asignado a nivel de plot es verde, para blanco el color asignado a nivel de plot es azul celeste y para negro el color asignado a nivel de plot es negro.

```
T1 =

  6×2 table

    Colores     Etiqueta_color
    _______     ______________

    'red'       'r'
    'yellow'    'y'
    'orange'    'b'
    'green'     'g'
    'white'     'c'
    'black'     'k'
```

Tabla 6. Relación color - etiqueta

Como se puede observar en la Figura 39 la clasificación del vecino más cercano es muy efectiva, dado que separaron las diferentes poblaciones de colores con un porcentaje de error observable muy pequeño.

Una representación en el espacio trazable 'a*' y 'b*' de los píxeles que se clasificaron se ilustra en la Figura 40 en colores separados y de acuerdo a las etiquetas de color de la Tabla 6.

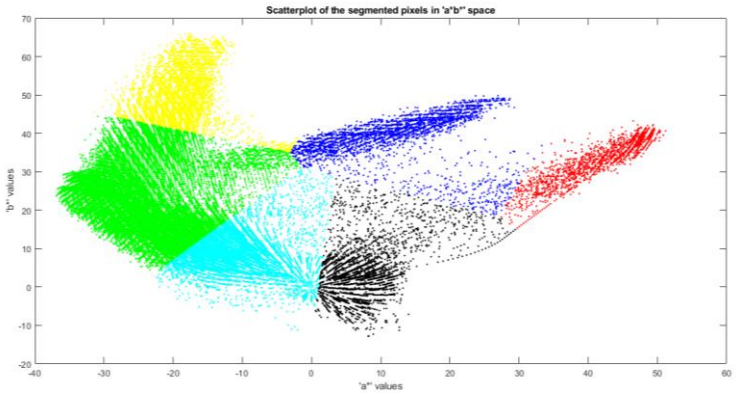


Figura 40. Segmentación de color obtenida para la Figura 36 en el espacio 'a*' y 'b*'

Si observa a detalle la Figura 40 son apreciables los diferentes agrupamientos de color o clúster obtenidos según marcador de color definido o centroide para el caso de estudio que se presenta en la Figura 36.

4.3.8. ***Paso 8: Zona de interés.***

Este procedimiento retorna únicamente una imagen binaria que representa el posicionamiento de los dados la cual es ilustrada en la Figura 41.

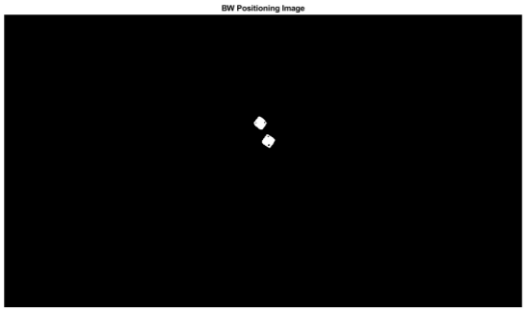


Figura 41. Imagen de posicionamiento BW del objeto de interés

Mediante los pasos anteriores se garantiza la detección de la posición de los dados en el tapete o superficie de trabajo con un porcentaje de efectividad aproximado a 100%, para la base de datos usada de 1200 imágenes, esto facilita la extracción o recorte de los dados mediante una comparación entre una imagen binaria como la de la Figura 41 que representa el posicionamiento y la imagen original en RGB.

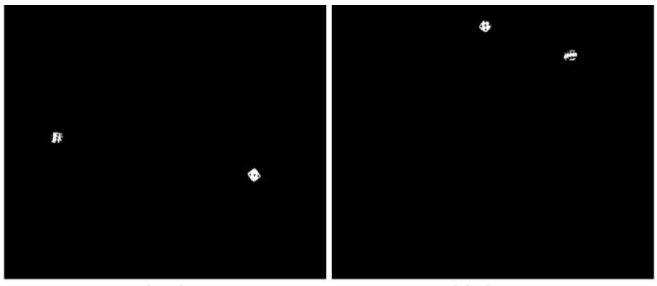
5

5. CLASIFICACIÓN DEL RESULTADO DE LOS DADOS

Esta fase está dedicada a describir la metodología aplicada para identificar el resultado obtenido por un jugador, tras lanzar los dados sobre el ambiente símil al juego Craps. Este capítulo está dividido en tres secciones, la primera corresponde al tratamiento que se le realiza a la imagen obtenida del capítulo CAPTURA, PROCESAMIENTO Y ANÁLISIS DE LOS DATOS DE INTERÉS, la segunda explica la construcción y configuración de la red utilizada para la clasificación del resultado de los dados y la última describe el proceso de entramiento y los resultados obtenidos.

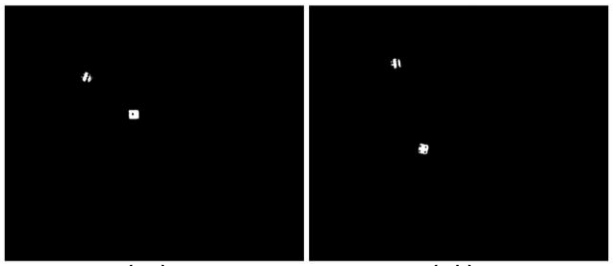
5.1. Postratamiento de la imagen binaria y recortes

Si se observan las imágenes presentes en la Figura 43 todas comparten un mismo problema de interpretación y es que para cualquiera de ellas al momento de verificar el conteo de números de objetos presentes se pueden apreciar un total de tres cuando realmente el número de dados presentes es dos, esto supone un problema (Para efectos del documento este problema recibe el nombre de Error de Segmentación 1).



(a) (b)

Figura 42. Tipos de imágenes binarias del Error de segmentación 1.



(c) (d)

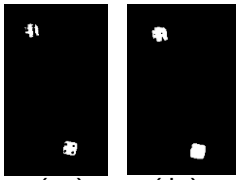
Figura 43. Continuación de la Figura 42, Tipos de imágenes binarias del Error de segmentación 1.

Por lo tanto, se hace extremadamente necesario garantizar que solo se perciban dos objetos después de preprocesar las imágenes mediante el inciso Conjunto de pasos para procesamiento y análisis de los datos.

Dentro del procesamiento de imágenes existen técnicas que permiten, sin deformar la información en la imagen binaria, agrupar o desagrupar objetos cercanos entre sí, este tipo de técnicas son aplicadas para evitar el (Error de segmentación 1), cabe aclarar que ciertos procesos de los enumerados a continuación han sido configurados de manera heurística,

1. Operación de llenado por inundación en píxeles de fondo,
2. Cierre morfológico con elemento estructurante tipo rectángulo de 7x7,
3. Remoción de pequeños objetos con menos de 30 pixeles,
4. Dilatación de los objetos en la imagen con elemento estructurante tipo rectángulo de 5x5.

Al finalizar todas las operaciones enumeradas, se soluciona por completo el Error de segmentación 1, tal como se observa en la Figura 44.



(a) (b)

Figura 44. Primera solución (Error segmentación 1).

Aunque en efecto el (Error de segmentación 1) fue solucionado, se presenta un segundo problema cuando los dos objetos de interés se encuentran muy cercanos o alcanzan a

conectarse entre sí formando un único objeto), tal y como se observa en la ,(Para efectos del documento este problema recibe el nombre de Error de Segmentación 2).

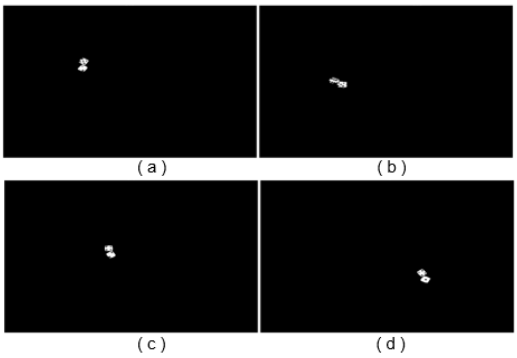


Figura 45. Tipos de imágenes binarias del Error de segmentación 2.

A razón de esta situación, se ve en la necesidad de considerar una segunda etapa que garantice que se perciban mínimo dos objetos después de solucionar el Error de segmentación 1, para ello se dispone de la siguiente metodología.

1. Adquisición de la imagen binaria obtenida mediante el inciso Conjunto de pasos para procesamiento y análisis de los datos.
2. Operación de llenado por inundación en píxeles de fondo,
3. Apertura morfológica con elemento estructurante tipo rectángulo de (gr x gr) a través de un bucle con gr en rango 1 A 18,
4. Remoción de pequeños objetos con menos de 30 pixeles,
5. Dilatación de los objetos en la imagen con elemento estructurante tipo rectángulo de 5x5.

Al finalizar todas las operaciones enumeradas, se soluciona por completo el Error de segmentación 2, tal como se observa en la Figura 46.

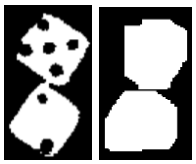


Figura 46. Segunda solución (Error segmentación 2).

Es observable que a través de la solución propuesta en la Figura 46 se separan en dos objetos este conjunto de elementos interconectados; no obstante, el proceso de apertura morfológica para casos como el observado en la Figura 47, es demasiado rudo al realizar la segmentación; situación que se ve reflejada al momento de extraer en la imagen RGB la zona de interés descrita por la imagen binaria (Para efectos del documento este problema recibe el nombre de Error de Extracción 1)

(a) (b)

Figura 47. Caso, Error de extracción 1.

Este último error presentado, puede ser solucionado modificando el constructor de propiedades de los objetos presentes en una imagen binaria tras ser etiquetados y caracterizados mediante, su centroide, área y cuadro delimitador de contorno, para esto solo basta con identificar las dimensiones de cada objeto y mediante un umbral, en este caso, obtenido de manera heurística de tamaño menor a 35, redimensionar el cuadro delimitador de contorno.

Un ejemplo de lo que sucede es descrito por la Figura 48, donde si se observa a detalle, en la imagen de la izquierda el cuadro delimitador de contorno (CDC) al no ser modificado este omite información importante del dado 1, mientras que el CDC modificado conserva la información requerida del dado 1.

(a) (b)

Figura 48. Solución (Error de extracción 1), a) CDC sin modificar, b) CDC modificada.

Aplicando cada solución según el tipo de error ya sea de segmentación o extracción es posible afirmar que en efecto se cuenta con las coordenadas delimitadoras de objeto que permitirán realizar el recorte de la zona de interés que corresponde a la localización de los dados en la imagen RGB, un conjunto de ejemplos de los tipos de recortes realizados puede ser verificados en la Figura 49.

Figura 49. Estilo de los recortes obtenidos.

Definido el proceso secuencial de tratamiento de las imágenes para el recorte de los datos de interés, se realiza el proceso de clasificación y almacenamiento de los recortes de forma manual, en esta etapa de la base de datos construida inicialmente (un total de 1200 imágenes) se tomó el 40% para ser etiquetadas una por una de manera supervisada mediante una entrada por teclado, construyendo de esta manera un vector de 1000 etiquetas y una variable tipo 4D (filas, columnas, profundidad, numero de imágenes) de 1000 dimensiones que contiene en cada posición de una imagen, todo esto con la intención de entrenar las arquitecturas de redes CNN descritas a continuación.

El mismo procedimiento fue realizado con el 60% de los datos restantes, obteniendo por lo tanto un vector de 1400 etiquetas y una variable tipo 4D (filas, columnas, profundidad, numero de imágenes) de 1400 dimensiones, que se usaran como datos de testeo validación.

5.1.1. *Selección del tamaño de foto estándar*

Las fotos deberán ser redimensionadas antes de ser ingresadas al clasificador, tanto para entrenamiento como testeo, de esta manera se garantiza que todas las imágenes tengan la misma forma.

Hay muchas maneras de lograr esto, aunque la más común es una simple operación de cambio de tamaño que estirará y deformará la relación de aspecto de cada imagen y la forzará a adoptar la nueva forma. Para esto se cargan todas las fotos, se observa la distribución de los anchos y alturas de cada una y por último se define un nuevo tamaño de foto que refleje mejor lo que es más probable que se observe durante la puesta en práctica del clasificador.

De modo que, partiendo de la premisa de que por lo general los recortes se encuentran en un rango entre los 40 y 70 pixeles para cualquiera de sus dos ejes se selecciona un tamaño de imagen de **28x28** considerando que, entre más pequeña sea el tipo de imagen de entrada a la CNN más rápido es posible entrenar el modelo de CNN.

5.2. Creación y configuración de la red

Durante el desarrollo del proyecto se probaron diferentes enfoques con la finalidad de implementar un sistema de clasificación multiclase, sin embargo técnicas de baja complejidad como filtros pasa alta, filtros adaptativos, ecualización del histograma y filtros HSV, no fueron muy adecuados para abordar el problema base del proyecto el cual a groso modo consiste en identificar 6 posibles clases o caras de un dado que se encuentra sometido a variaciones finas de luminosidad, contraste e intensidad y sobre todo bajo perspectivas observables no estandarizadas, es decir rotaciones, deformaciones, escalamientos, espejos en ambos ejes y reflexión lumínica, un ejemplo del tipo de situación aquí descrita es observada en la Figura 50.

Figura 50. Muestra del número 6 y sus diferentes variaciones.

En función del comportamiento de los datos de entrenamiento registrados, se elige el uso de CNN (Redes Neuronales Convolucionales) como la técnica de inteligencia artificial que ha de asegurar una clasificación robusta de un conjunto de datos multiclase.

Una vez identificados los datos de entrenamiento y la técnica de inteligencia artificial para clasificación a usar, se procede a configurar las distintas redes que se evaluaran a lo largo del proyecto, para ello se proponen 3 arquitecturas.

Las arquitecturas propuestas a continuación están configuradas con los siguientes parámetros. Tamaño de la imagen en altura (H=28), ancho (W=28) y canales de profundidad; el número de filtros que se pueden aprender (F=$[32, 64, 128]$); el tamaño del lote (B) (por defecto 128), el tamaño del filtro (S=[3 3]) y por último el número de clases deseadas como parámetro (C=6).

5.2.1. ***Arquitectura CNN #1***

En esta subsección, se presenta en la Figura 51 un modelo de red neuronal convolucional de referencia para el conjunto de datos de las caras de dos dados.

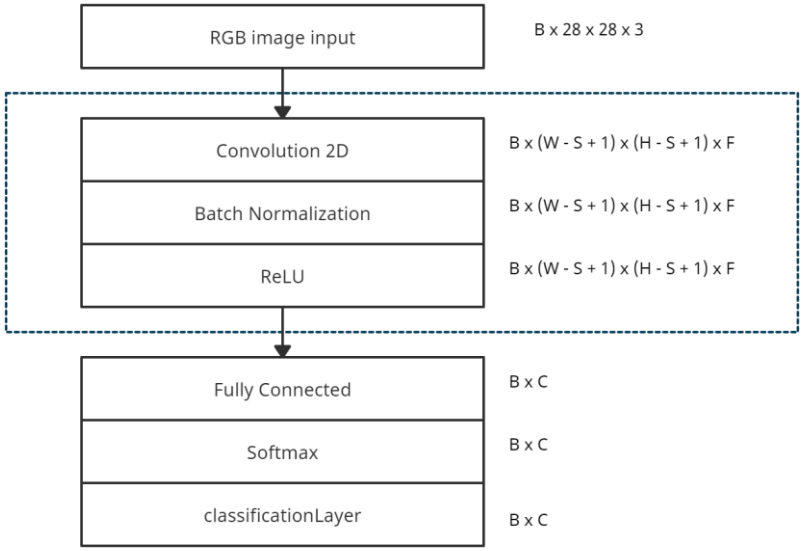


Figura 51. Primera arquitectura CNN configurada $F = 32$.

Este modelo se considera dentro del proyecto como base de referencia, ya que representa un rendimiento mínimo comparable con otras arquitecturas acá dispuestas, por lo tanto, se hace uso del mismo como base de estudio y mejora.

Dado que en efecto no existe una metodología clara en la manera que se debe construir una CNN, se verifica en la literatura que arquitecturas son similares a la ilustrada en la Figura 51, encontrando que un buen punto de partida son los principios arquitectónicos generales de los modelos VGG y esto se debe a que lograron el máximo rendimiento en la competencia ILSVRC 2014 y porque la estructura modular de la arquitectura es fácil de entender e implementar (Para obtener más detalles sobre el modelo VGG, consulte el documento [50]). La arquitectura VGG implica apilar capas convolucionales con pequeños filtros de 3 × 3 seguidos de una capa de agrupación máxima. Juntas, estas capas forman un bloque, y estos bloques se pueden repetir donde el número de filtros en cada bloque aumenta con la profundidad de la red, como $F = [32, 64, 128, 256]$ para los primeros cuatro bloques del modelo. El Padding se utiliza en las capas convolucionales para garantizar que las formas de alto y ancho de los mapas de características de salida coincidan con las entradas.

5.2.2. ***Arquitectura CNN #2***

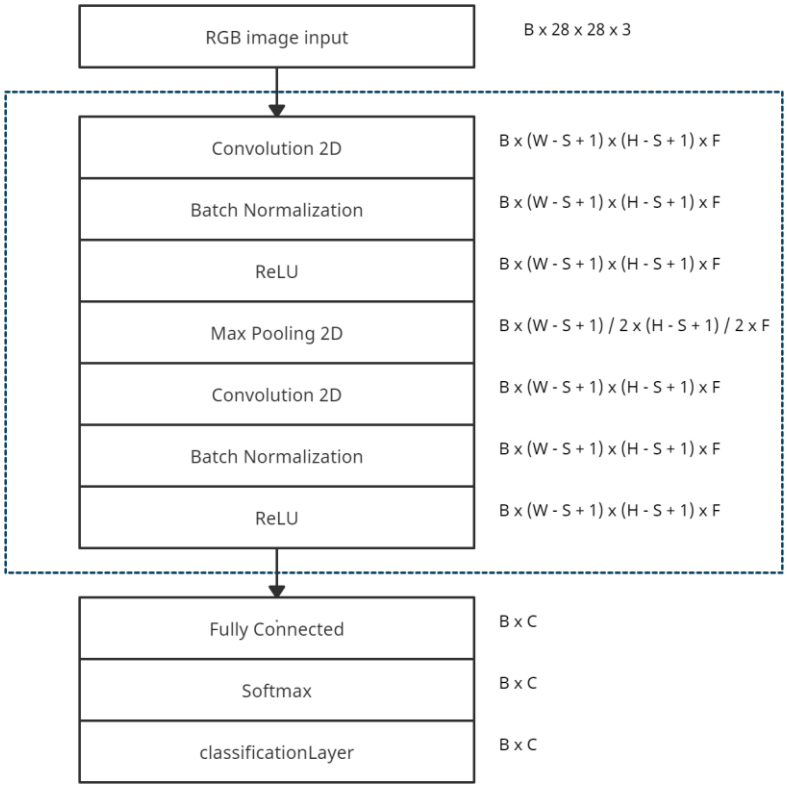


Figura 52. Segunda arquitectura CNN configurada con F=[Bloque VGG1 = 32, Bloque VGG2 = 64].

Para todas las arquitecturas de CNN mostradas durante el documento se usa la función de activación ReLU y se inicializa el peso He, practicas muy recomendadas al momento de procesar imágenes; además, dado que la clasificación es multiclase, es decir 6 posibles tipos de caras, se procede a utilizar la función de activación Softmax en la capa de salida, tal como se observa en la Figura 52.

Tras realizar algunas pruebas con la versión de la Figura 52 que contiene un bloque del modelo VGG más que el de la Figura 51, los resultados obtenidos no eran muy concluyentes, así que se aumentaron las capas convolucionales de la red tal como se prevé en una arquitectura VGG

5.2.3. ***Arquitectura CNN #3***

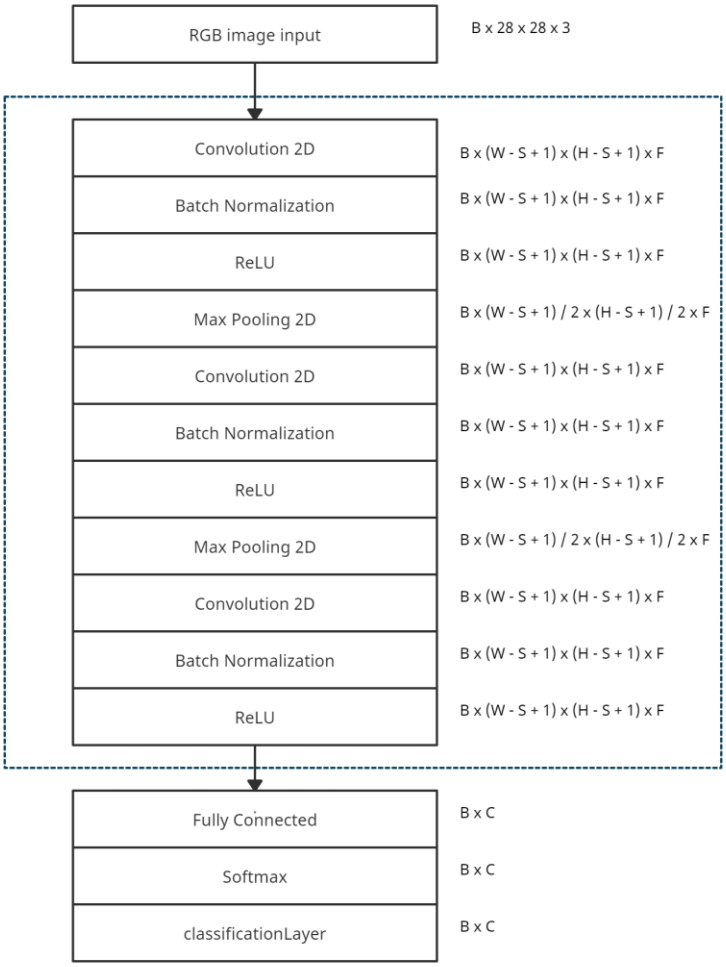


Figura 53. Tercera arquitectura CNN configurada con $F = [Bloque\ VGG1 = 32, Bloque\ VGG2 = 64,\ Bloque\ VGG3 = 128]$.

La diferencia presente en la arquitectura de la Figura 53 respecto a la de la Figura 51 es que se pasa de considerar un bloque VGG a considerar 3. Los resultados mejoraron, pero aun así se buscó hacerla más eficiente. Para esto se ocuparon técnicas como Dropout y DataAugmentation.

5.3. Entrenamiento y resultados

Los resultados obtenidos con las tres arquitecturas descritas anteriormente y las posibles mejoras mencionadas para la última arquitectura se encuentran en esta sección.

5.3.1. ***Conjunto de datos***

Las arquitecturas descritas anteriormente se han probado con la base de datos de 1200 imágenes recortadas y procesadas mediante la sección Postratamiento de la imagen binaria y recortes, de esta base de datos se tomaron el 40% es decir 500 imágenes para el proceso de entrenamiento de cualquiera de las tres arquitecturas y el 60% restante es decir 700 imágenes son usadas para probar cada arquitectura y verificar los criterios de evaluación de la siguiente sección, recuérdese que por cada imagen tomada de la base de datos se obtienen dos recortes que representan el resultado de cada dado.

5.3.2. ***Criterios de evaluación***

Para evaluar los resultados en función de cada arquitectura, se han tenido en cuenta dos criterios de evaluación uno general y otro especifico según clase, el primero consiste en el porcentaje de precisión general de la arquitectura de estudio para todas las clases, su ecuación es,

$$Precisión\ General = \frac{Casos\ Clasificados\ correctamente}{Casos\ evaluados} \times 100 \quad (36)$$

El segundo corresponde a la precisión particular según clase,

$$Precisión\ Particular\ \#n = \frac{Casos\ Clasificados\ correctamente\ sc_n}{Casos\ evaluados\ segun\ clase\ n} \times 100 \quad (37)$$

5.3.3. ***Resultados experimentales***

En este apartado se exponen las opciones de entrenamiento utilizadas con cada arquitectura, los resultados obtenidos y algunas gráficas del proceso de entrenamiento para cada arquitectura usada, así mismo las posibles mejoras y sus resultados.

Las opciones de entrenamiento configuradas para todas y cada una de las arquitecturas son las siguientes,

```
opts = trainingOptions('sgdm', ...
    'MaxEpochs',45, ...
    'Shuffle','every-epoch', ...
    'InitialLearnRate',1e-3, ...
    'Plots','training-progress', ...
    'Verbose',false, ...
    'ValidationFrequency',25, ...
    'ValidationData',{XValidation,YValidation});
```

Los resultados obtenidos para la arquitectura de la Figura 51 son los que aparecen en la Tabla 5.

Clase	Cara #1	Cara #2	Cara #3	Cara #4	Cara #5	Cara #6
Precisión (%) 92.0714	93.5780	91.0714	84.0909	90.3509	92.5094	97.5610
# imágenes T= 1400	218	224	176	228	267	287

Tabla 7. Resultados obtenidos al usar la arquitectura de la Figura 51.

La Figura 54 ilustra el histórico de entrenamiento mediado por la arquitectura de la Figura 51 usando los datos de entrenamiento y testeo previstos, aunque en efecto esta arquitectura es capaz de clasificar alrededor del 100% de los datos de entrenamiento, al momento de clasificar los datos de testeo alcanza una precisión máxima de 92.07%, lo cual no es muy conveniente, para el sistema que se desea desarrollar, sin embargo es un excelente punto de partida.

La manera en la que debe interpretar el monitor de entrenamiento y testeo de la red (véase un ejemplo en la Figura 54), consiste en lo siguiente: en el eje de abscisas sé encuentran las iteraciones, en el eje de ordenadas se localiza el porcentaje de acierto, es decir, la precisión. La línea azul corresponde al histórico del entrenamiento "suavizado" (training smoothed), la línea celeste con el entrenamiento de la red en bruto (training) y la discontinua con la validación de la red (validation). Cada vez que aparece un punto en la línea discontinua, la red realiza una validación preliminar de la precisión de la red, y de esta manera es posible verificar cuan adecuada es la arquitectura usada para un procesamiento particular.

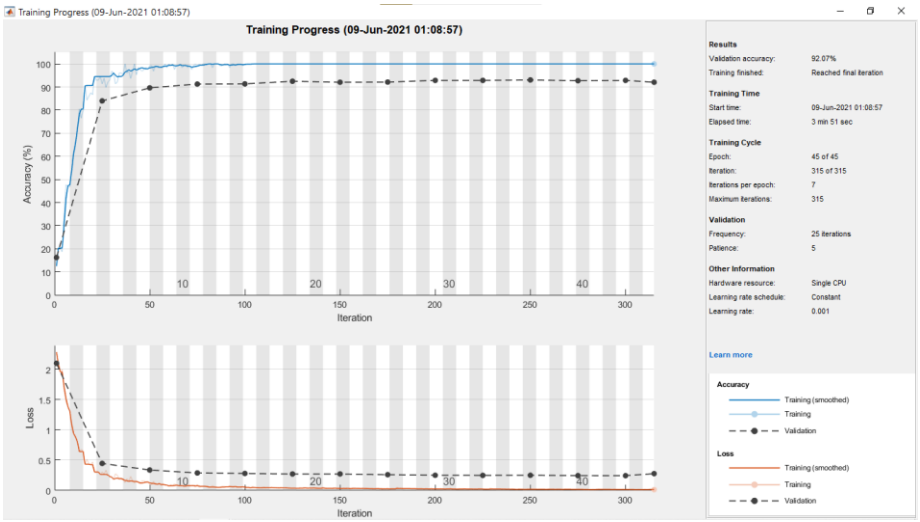


Figura 54. Monitor de entrenamiento para la arquitectura de la Figura 51.

Al probar la arquitectura de la Figura 52, se observó una mejora en los resultados. Estos pueden ser observados en la Tabla 7.

Clase	Cara #1	Cara #2	Cara #3	Cara #4	Cara #5	Cara #6
Precisión (%) 96.2857	96.7890	97.7679	93.1818	94.7368	95.1311	98.9547
# imágenes T= 1400	218	224	176	228	267	287

Tabla 8. Resultados obtenidos al usar la arquitectura de la Figura 52.

La Figura 55 Figura 54 ilustra el histórico de entrenamiento mediado por la arquitectura de la Figura 52 usando los datos de entrenamiento y testeo previstos, aunque en efecto esta arquitectura es capaz de clasificar alrededor del 100% de los datos de entrenamiento, al momento de clasificar los datos de testeo alcanza una precisión máxima de 96.2857%, lo cual no es muy conveniente para el sistema que se desea desarrollar, no obstante, nos entrega una idea general de la relación número de capas convoluciones sobre la precisión alcanzada por la red, para este caso una diferencia 4.2143%. respecto a la arquitectura evaluada inicialmente.

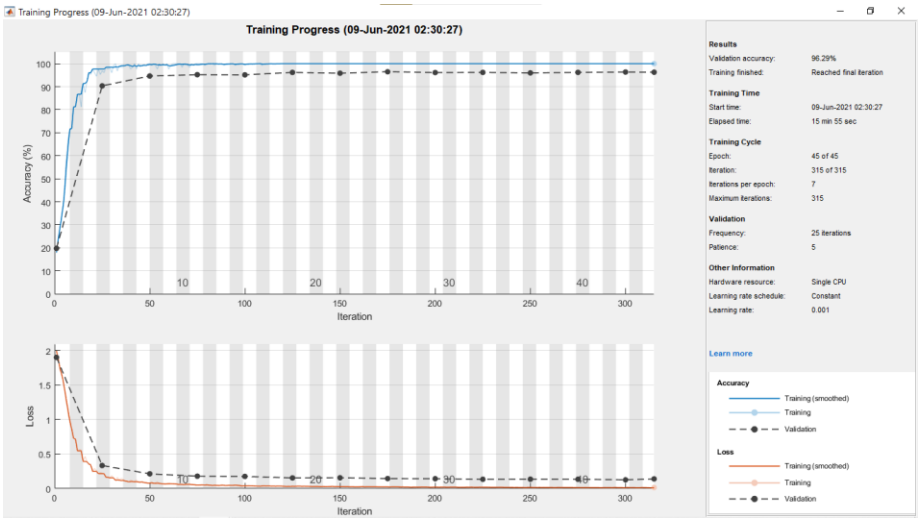


Figura 55. Monitor de entrenamiento para la arquitectura de la Figura 52.

Al probar la arquitectura de la Figura 53, se observó que respecto a la arquitectura anterior esta arquitectura presenta una mejora del 2.1429%. Los resultados pueden ser observados en la Tabla 7.

Clase	Cara #1	Cara #2	Cara #3	Cara #4	Cara #5	Cara #6
Precisión (%) 98.4286	98.6239	99.1071	96.5909	97.3684	98.1273	100
# imágenes T= 1400	218	224	176	228	267	287

Tabla 9. Resultados obtenidos al usar la arquitectura de la Figura 53

La Figura 56 ilustra el histórico de entrenamiento mediado por la arquitectura de la Figura 53 usando los datos de entrenamiento y testeo previstos, aunque en efecto esta arquitectura es capaz de clasificar alrededor del 100% de los datos de entrenamiento, al momento de clasificar los datos de testeo alcanza una precisión máxima de 98.4286%, lo cual no es muy conveniente para el sistema que se desea desarrollar, no obstante, se encuentra muy próximo al objetivo.

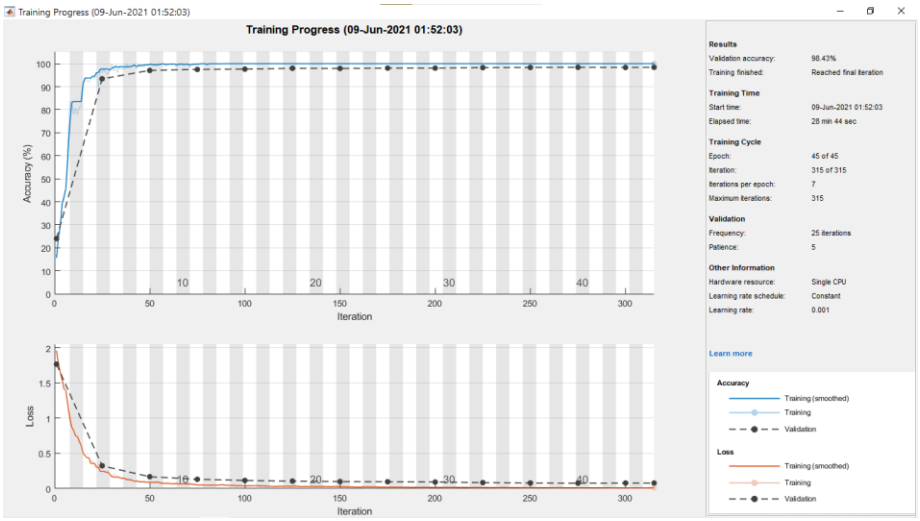


Figura 56. Monitor de entrenamiento para la arquitectura de la Figura 53.

Tal como se puede inferir, tras probar todas y cada una de las arquitecturas de la sección Creación y configuración de la red el porcentaje de precisión alcanzado no es el más satisfactorio, por lo tanto, se propone el uso de técnicas como DropOut, DataAugmentatión y filtrado escalonado, con la arquitectura de la Figura 53, la cual fue entre todas las arquitecturas la que obtuvo el mayor número de ciertos.

Para ello se configura el módulo de DataAugmentation over training de la siguiente manera,

```
augmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandYReflection',true, ...
    'RandRotation',[0 360]);
```

Y además se agrega previo a la capa FullyConnected un DropOut del 50% mediante la siguiente línea,

```
dropoutLayer(0.5)
```

y se agrega un cuarto bloque VGG con numero de filtros de 256 de manera escalonada, es decir, se aplica un bloque VGG con numero de filtros de $256/2$ seguido de otro con las mismas características, un bosquejo de la arquitectura de la Figura 53 modificada es ilustrado en la Figura 58 con DropOut.

Modificando de esta forma la arquitectura de la Figura 53; los resultados posentrenamiento pueden ser verificados en la Tabla 10.

Clase	Cara #1	Cara #2	Cara #3	Cara #4	Cara #5	Cara #6
Precisión (%) 98.7143	99.0826	99.5536	97.1591	98.6842	97.3783	100
# imágenes T= 1400	218	224	176	228	267	287

Tabla 10. Resultados obtenidos al usar la arquitectura de la Figura 58 con DropOut.

La Figura 57 ilustra el histórico de entrenamiento mediado por la arquitectura de la Figura 58 con DropOut usando los datos de entrenamiento y testeo previstos, aunque en efecto esta arquitectura es capaz de clasificar alrededor del 100% de los datos de entrenamiento, al momento de clasificar los datos de testeo alcanza una precisión máxima de 98.7143%, lo cual sigue sin ser lo deseado para con el sistema que se desea desarrollar, no obstante, se encuentra muy próximo al objetivo y presenta una mejora del 0.2857% respecto a la arquitectura anterior.

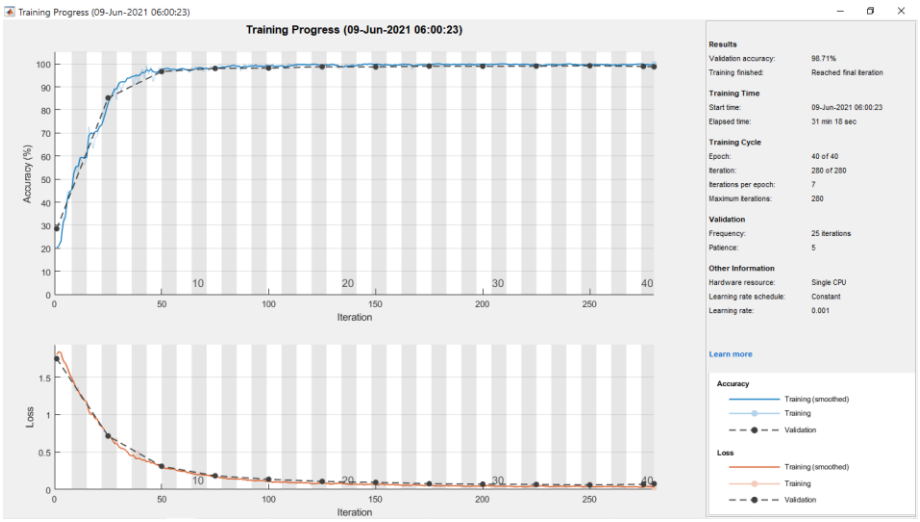


Figura 57. Monitor de entrenamiento para la arquitectura de la Figura 58 con DropOut.

Finalmente, aunque se probó la arquitectura Figura 58 con DropOut los resultados siguen sin ser los deseados, por lo tanto, se procede a no aplicar DropOut en la arquitectura de la Figura 53 quedando la arquitectura con sus otras dos mejoras tal como se aprecia en la Figura 58.

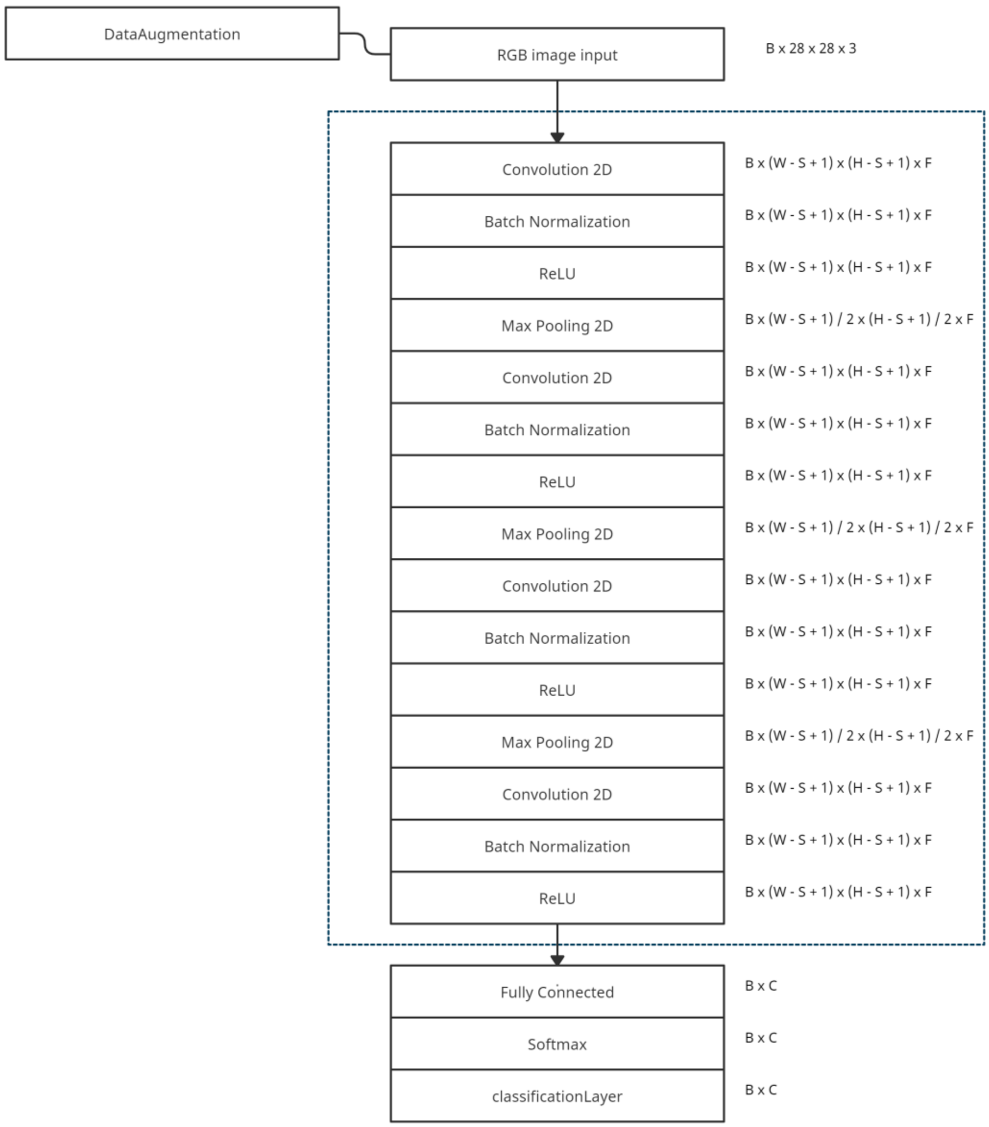


Figura 58. Quinta arquitectura CNN configurada con F=[Bloque VGG1 = 32,Bloque VGG2 = 64,Bloque VGG3=128 y Bloque VGG4=128].

Al evaluar la arquitectura de la Figura 58 se obtienen los resultados de testeo, los cuales pueden ser verificados en la Tabla 11.

Clase	Cara #1	Cara #2	Cara #3	Cara #4	Cara #5	Cara #6
Precisión (%) 99	99.5413	99.5536	98.2955	98.6842	97.7528	100
# imágenes T= 1400	218	224	176	228	267	287

Tabla 11. Resultados obtenidos al usar la arquitectura de la Figura 58.

La Figura 59 ilustra el histórico de entrenamiento mediado por la arquitectura de la Figura 58 usando los datos de entrenamiento y testeo previstos, aunque en efecto esta arquitectura es capaz de clasificar alrededor del 100% de los datos de entrenamiento, al momento de clasificar los datos de testeo alcanza una precisión máxima de 99%, lo cual aunque sigue sin ser lo deseado para con el sistema que se desea desarrollar, una tolerancia del 1% es aceptable, considerando que 1% de $1400\,imágenes$ son $14\,imagenes$ clasificadas erroneamente.

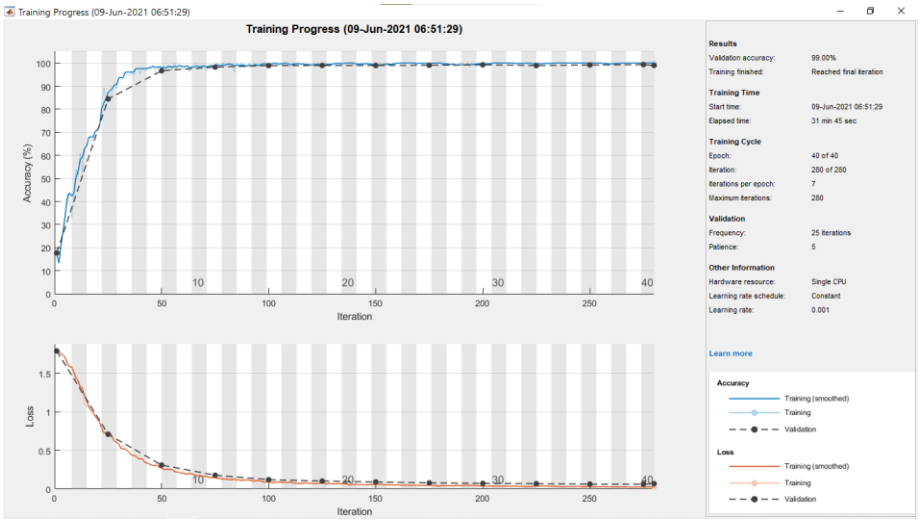


Figura 59. Monitor de entrenamiento para la arquitectura de la Figura 58.

La matriz de confusión tras evaluar los datos de testeo es mostrada en la Figura 69 de la cual se puede analizar.

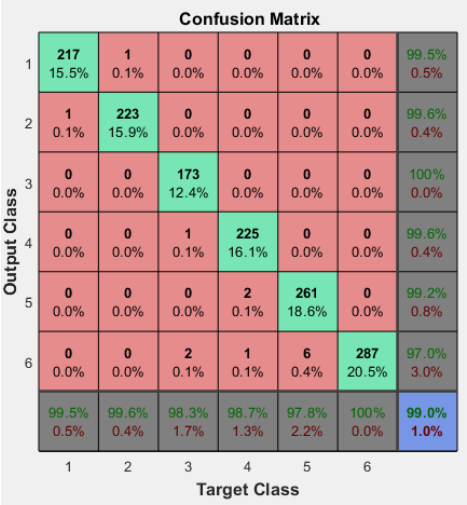


Figura 60 Matriz de confusión de la CNN de testeo.

Las matrices de confusión para el entrenamiento y testeo del proceso general se pueden ver en la Figura 60. Se observa que la salida es precisa ya que el número de respuestas correctas que se indican en los cuadrados verdes [los cuadrados con índices (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)] son medios y el número de respuestas incorrectas está representado en los cuadrados rojos [cuadrados con índices (1,2), (1,3), (1,4), (1,5), (1,6), (2,1), (2,3), (2,4), (2,5), (2,6), (3,1), (3,2), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,5), (4,6), (5,1), (5,2), (5,3), (5,4), (5,6), (6,1), (6,2), (6,3), (6,4), (6,5)]] son bajos. Las precisiones generales de la Figura 60, indicados en los cuadrados azules de la parte inferior derecha [cuadrados con índice (7, 7)] son altas, lo que justifica que la fiabilidad del sistema, para la matriz de testeo de la base de datos de 1200 imágenes es de 99% y un error del 1%.

Se puede inferir que el modelo de red neuronal convolucional (CNN) desarrollado en este proyecto es un modelo preciso y confiable.

6

6. PREDICCIÓN DEL RESULTADO DE LOS DADOS

Esta fase está dedicada a describir el proceso que se aplicó para predecir el resultado que ha de obtener un jugador tras lanzar los dados sobre el ambiente de trabajo. Este capítulo está dividido en tres secciones, la primera corresponde al procedimiento para obtener el conjunto de imágenes que describen una trayectoria, la segunda explica la segmentación y clasificación del conjunto de imágenes obtenidas, la tercera explica el tipo de red utilizada para la predicción y su respectivo entrenamiento, y la última presenta los resultados obtenidos.

6.1. Procedimiento de captura de la trayectoria

Para obtener la trayectoria que describen los dados tras ser lanzados, es necesario hacer uso de tecnología de captura con alta taza de frames por segundo (fps) y con buena resolución, capaz de grabar los cambios rápidos que se presentan en la rotación de las 6 posibles caras para cada dado; un dispositivo con este tipo de prestaciones, es utilizado en el proyecto, el cual corresponde a un Iphone X apto para grabar a 240 fps y 720p.

Mediante esta tecnología se grabaron y almacenaron una serie de lanzamientos consecutivos en un único archivo de video, para luego separar manualmente cada lanzamiento en distintos archivos de video, construyendo así un base de datos de 50 lanzamientos, donde para cada lanzamiento se garantizó que desde el primer instante de tiempo se observara la rotación de los dados sobre el ambiente de trabajo.

Construida la base de datos se procede a determinar los tiempos máximos de grabación presentes en cada video, obteniendo que el tiempo relativo a utilizar para adquisición de datos muestra de cada video es el promedio de los datos que se presentan en la Tabla 12.

4:32	4:34	5:21	4:54	4:51	4:17	5:10	5:26	5:48	5:06
5:23	4:57	4:66	4:00	5:68	4:70	5:39	5:14	5:08	4:00
4:75	5:25	4:06	5:67	4:53	5:57	5:34	4:34	4:46	4:45
4:56	4:45	4:54	4:39	4:35	5:54	4:26	5:43	4:35	4:39
4:45	4:25	5:25	4:52	5:36	4:15	5:24	4:35	5:32	4:36

Tabla 12. Tiempo en segundos que tarda el lanzamiento desde su punto inicial hasta su punto final

La ecuación aplicada para obtener el promedio del tiempo relativo es:

$$P = \frac{238.3700\ s}{50} = 4.7674\ s \quad (38)$$

Un ejemplo de lo que se proyecta con este tiempo relativo es ilustrado en la Figura 61, donde para un lanzamiento el cual tiene un tiempo de duración de 5.48s solo se toma como información útil la zona de color azul, mientras que los datos en el tiempo de la zona de color amarillo son desechados, ya que en dicha zona es el tiempo final y en esta sección se desea predecir el resultado de los dados.

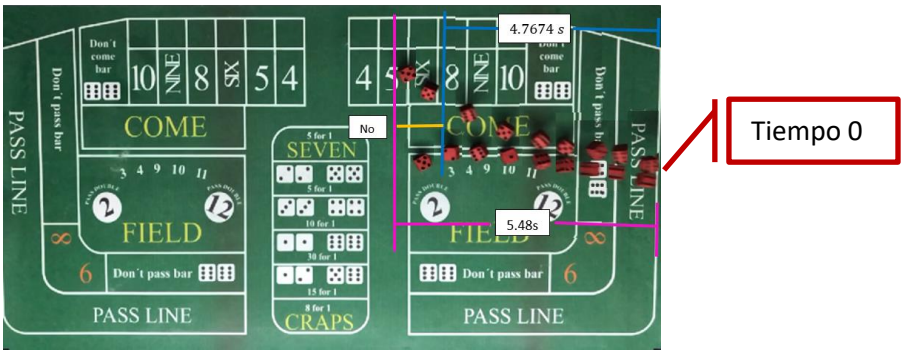


Figura 61. Tiempo recorrido de un lanzamiento

Aunque la región descrita por el tiempo representado por el color azul en la Figura 61 contiene la información considerada como útil, se puede observar que desde el tiempo cero hasta casi la mitad de dicho tiempo las caras de los dados dentro de la región de color cian de la Figura 62 se encuentran borrosas, por lo tanto, se construye un último rango de tiempo como se observa en la Figura 62 con el color verde; este procedimiento se realiza para cada lanzamiento teniendo en cuenta del mismo tiempo promedio.

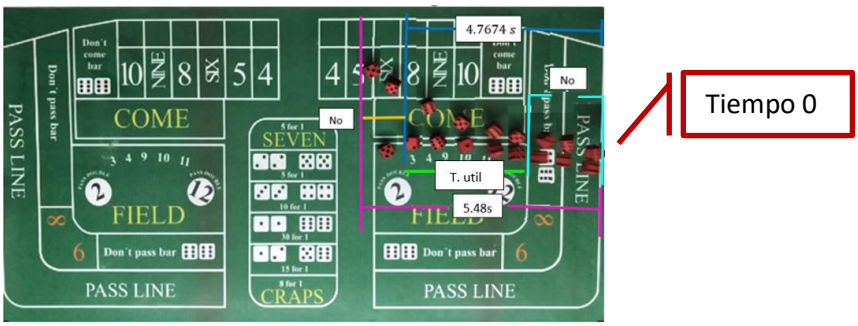


Figura 62. Tiempo útil de un lanzamiento

Finalmente se obtiene una nueva base de datos con archivos de video que responde al tiempo descrito por la línea de color verde, los datos en esta región deben descomponerse en frames, para ello se lee cada video en la base de datos y se procesa mediante una línea de código que me convierte un video en un conjunto de imágenes; de modo que una tercera base de datos es obtenida, la cual contiene un total de 50 conjuntos de imágenes donde cada conjunto es en efecto un lanzamiento con dos trayectorias, lo cual en total son 100 lanzamientos distintos, procesado según los tiempos definidos en la Figura 62.

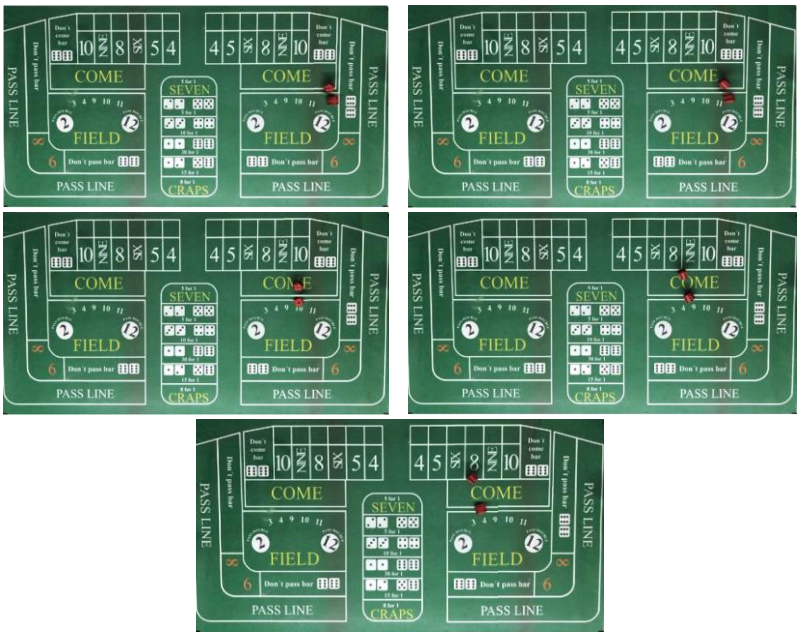


Figura 63. Sucesión de datos de un lanzamiento de la Figura 62

Se puede observar en la Figura 63, un dato de los 50 en la base de datos que describen la información que es útil en el proyecto para un lanzamiento.

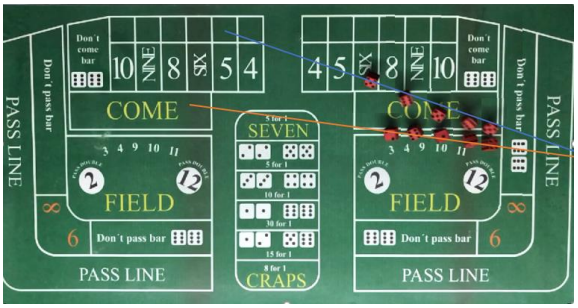


Figura 64. Trayectoria del dado 1 y dado 2

La Figura 64 resumen un conjunto de datos útiles y el bosquejo de la trayectoria que recorre cada dado, donde se presenta la primera trayectoria de color naranja del dado 1 y la segunda trayectoria de color azul del dado 2; es este tipo de conjunto de datos el cual es usado en la siguiente sección.

6.2. Segmentación y clasificación del conjunto de imágenes

6.2.1. ***Segmentación de un conjunto de datos***

Se toma un conjunto de datos de los 50 en la base de datos, se trata una a una las imágenes para un máximo de 6 iteraciones, en el conjunto de datos y se le aplica el proceso de segmentación el cual consiste inicialmente en definir el número de colores muestra en el espacio de color sRGB como se observa la Figura 36 se pueden apreciar 6 colores principales en la imagen como: verde, blanco, negro, naranja, amarillo y rojo. Luego se hace la selección de regiones muestra para ello se trata la Figura 37, mediante un bucle de tamaño igual al número de colores previstos para ser clasificados, en este caso un total de 6; una pequeña región de muestra se extrae en cada iteración y es etiquetada según el color que corresponda, luego se realiza el cálculo de marcadores de color por cada región y la conversión del espacio de color sRGB al L*a*b* de la Figura 36, como se observa en la Tabla 5 presenta los marcadores de color obtenidos, y los resultados de segmentación de objetos de la clasificación del vecino más cercano como se ilustra en la Figura 39, y partiendo de la zona de interés de la imagen binaria que representa el posicionamiento de los dados la cual es apreciable en la Figura 41.

6.2.2. *Clasificación de un conjunto de datos*

Las dimensiones de todos los recortes para los datos en el conjunto de datos siguen siendo de **28x28,** por lo tanto, se mantiene las dimensiones de las imágenes presentes durante la trayectoria, lo cual se hace la segmentación y detección de la posición delimitadoras del objeto que permite realizar el recorte de la zona de interés que corresponde a la localización de los dados en la imagen RGB, como se observa la Figura 65 y Figura 66 es un ejemplo de los tipos de recortes realizados de los dados en cada trayectoria.

Figura 65. Recorte de la trayectoria del dado 1

Figura 66. Recorte de la trayectoria del dado 2

Se presentan las trayectorias de los recortes de las imágenes tras ser etiquetados y caracterizados mediante, su centroide, área y cuadro delimitador de contorno, como se muestra en las Figura 67 y Figura 68, si se observa a detalle la imagen del cuadro delimitador de contorno (CDC) etiquetado como 1 y 2 esta información es importante para clasificar.

Figura 67 Cuadro delimitador de la trayectoria del dado 1

Figura 68 Cuadro delimitador de la trayectoria del dado 2

Para poder clasificar los dados de los recortes de la Figura 67 y Figura 68 se hace uso del modelo de red obtenido tras aplicar la arquitectura de la Figura 58, con ellos se reduce la dimensionalidad de las Figura 67 y Figura 68 quedando que,

4	6	1	6	2

Tabla 13 Clasificación de Figura 67 usando el modelo de la red (CNN)

5	6	5	6	6

Tabla 14 Clasificación de Figura 68 usando el modelo de la red (CNN)

La Tabla 13 y Tabla 14 corresponden a la clasificación resultante de la trayectoria que sigue cada dado, son estos vectores los que se usan para entrenar la red de regresión generalizada de la siguiente manera

$$a = \begin{vmatrix} 4 & 5 & v_{3.1} & v_{4.1} & \dots & v_{n.1} \\ 6 & 6 & v_{3.2} & v_{4.2} & \dots & v_{n.2} \\ 1 & 5 & v_{3.3} & v_{4.3} & \dots & v_{n.3} \\ 6 & 6 & v_{3.4} & v_{4.4} & \dots & v_{n.4} \\ 2 & 6 & v_{3.5} & v_{4.5} & \dots & v_{n.5} \end{vmatrix} \quad (39)$$

$$T = \begin{vmatrix} 4 \\ 6 \\ t_3 \\ t_4 \\ \dots \\ t_n \end{vmatrix} \quad (40)$$

La matriz (a) corresponde a la estructura de datos que se le entrega a la red, donde cada columna representa un conjunto de datos o conjunto de imágenes, para un lanzamiento son cada dos trayectorias, de modo que, para entrenar la red de los 50 conjuntos de datos solo se usaron 30 datos, en esta etapa de la base de datos construida inicialmente (un total de 60 trayectorias)

Debido a que el entrenamiento es supervisado es necesario entregarle a la red los objetivos según conjunto de datos, para ello se define el vector (T) que tiene tantas filas como conjunto de datos usados para entrenamiento.

Los 20 conjuntos de datos (un total de 40 trayectorias)restantes son usados para testear la red mediante la siguiente matriz (b).

$$b = \begin{vmatrix} v_{1.1} & v_{2.1} & v_{3.1} & v_{4.1} & \cdots & v_{20.1} \\ v_{1.2} & v_{2.2} & v_{3.2} & v_{4.2} & \cdots & v_{20.2} \\ v_{1.3} & v_{2.3} & v_{3.3} & v_{4.3} & \cdots & v_{20.3} \\ v_{1.4} & v_{2.4} & v_{3.4} & v_{4.4} & \cdots & v_{20.4} \\ v_{1.5} & v_{2.5} & v_{3.5} & v_{4.5} & \cdots & v_{20.5} \end{vmatrix} = \begin{vmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{vmatrix} \quad (41)$$

En efecto, en este caso no se le entrega a la red ya entrenada los objetivos según conjunto de datos, con la intensión de que la red **prediga** el valor final resultante tras los dados culminen su trayectoria, por este motivo se considera que las variantes de (R) en la ecuación 41 no se conocen.

La matriz de confusión tras evaluar los datos de testeo es mostrada en la Figura 69 de la cual se puede analizar.

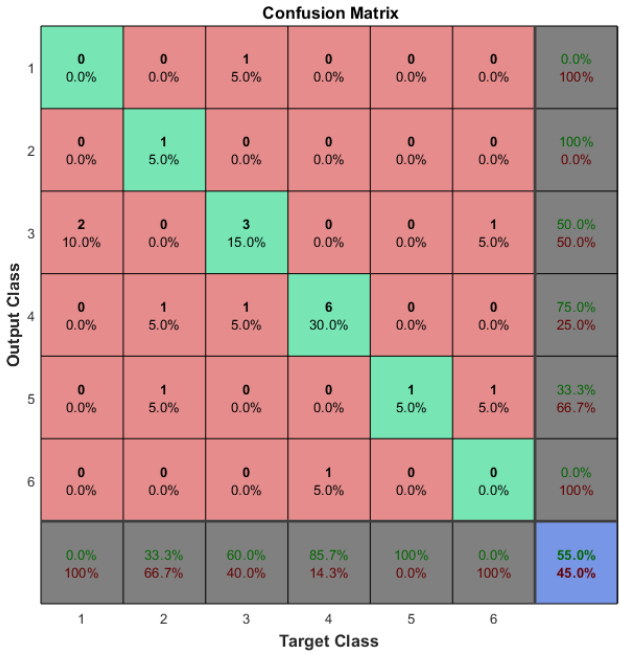


Figura 69 Matriz de confusión de la GRNN de testeo

Las matrices de confusión para el testeo del proceso general se pueden ver en la Figura 69. Las filas corresponden a la clase predicha y las columnas corresponden a la clase verdadera, se observa que la salida es precisa ya que el número de respuestas correctas que se indican en los cuadrados verdes [los cuadrados con índices (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)] son

medios y el número de respuestas incorrectas está representado en los cuadrados rojos [cuadrados con índices (1,2), (1,3), (1,4), (1,5), (1,6), (2,1), (2,3), (2,4), (2,5), (2,6), (3,1), (3,2), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,5), (4,6), (5,1), (5,2), (5,3), (5,4), (5,6), (6,1), (6,2), (6,3), (6,4), (6,5)]] son bajos, tanto el número de observaciones como el porcentaje del número total de observaciones se muestran en cada celda. La fila en la parte inferior del gráfico muestra los porcentajes de todos los ejemplos pertenecientes a cada clase que están clasificados correcta e incorrectamente. La precisión general de la Figura 69 indicada en el cuadrado azul de la parte inferior derecha [cuadrados con índice (7, 7)] es medio, lo que justifica que la fiabilidad del sistema de la base de datos de 40 imágenes es del 55% y un 45 % de error.

La dispersión de los datos es ilustrada tanto de testeo como de predicción son ilustrados en la Figura 70, de modo que si el marcador es un círculo azul o una cruz roja significa que no hubo coincidencia sin embargo que el marcador es un círculo azul con una cruz roja significa que hubo un 100% de coincidencia.

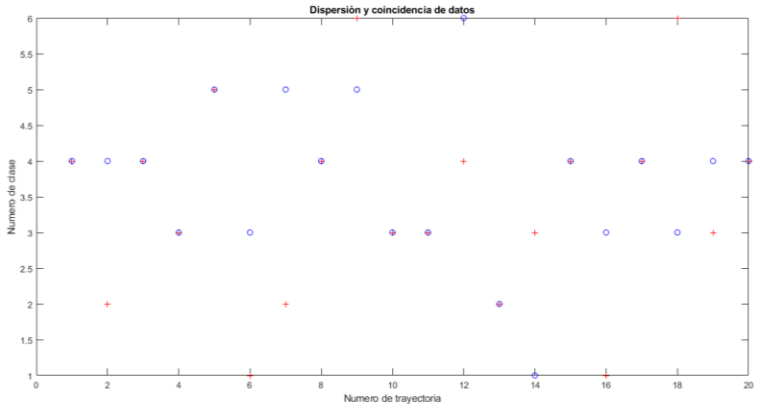


Figura 70 Dispersión de los datos

La exactitud se evalúa de la siguiente manera:

$$E = \frac{11}{20} = 0.55 \quad (42)$$

Obteniendo que, para la ecuación 43 tuvo un total del 55% de exactitud en la predicción.

7

7. DISEÑO INTERFAZ DE USUARIO Y RESULTADOS

Para el desarrollo de la interfaz de usuario, se dispuso del entorno GUIDE de Matlab, el cual provee mediante Widgets opciones simples para dar formato a la interfaz que se desea, en este capítulo se presenta el proceso llevado a cabo para el diseño y puesta en marcha de la interfaz.

7.1. Diseño

El entorno de diseño y sus respectivas herramientas son ilustradas en la Figura 71, el diseño de la aplicación solo ocupa 5 herramientas, estas son: Push Button, Estatic Text, Axes y Panel.

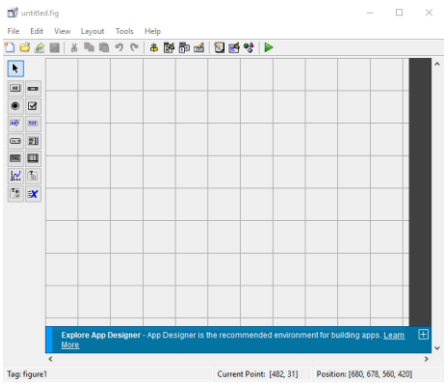


Figura 71 Entorno de diseño GUIDE.

Dado que la forma más adecuada de realizar un diseño en GUIDE, es empezar desde la capa inferior hasta la más alta, de modo que ningún objeto se encuentre debajo de un objeto en el que no deba estar, se procede a identificar y seleccionar cada Widget de la caja de herramientas previstas por el GUIDE. Inicialmente se hace uso de un primer Axes, este es destinado para establecer una imagen de fondo y recibe la etiqueta axes8 por medio del entorno Inspector el cual se inicializa dando doble clic sobre el objeto, como se puede apreciar en la Figura 72.

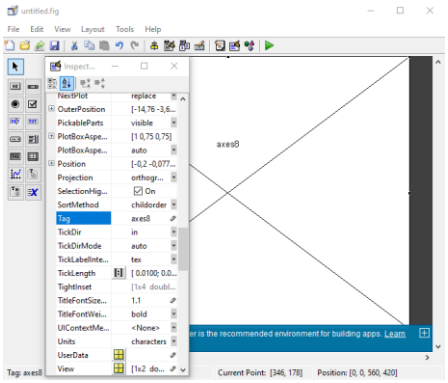


Figura 72 Herramienta Axes, variable axes8.

Para la segunda capa de diseño se toma un segundo Axes el cual es destinado para plotear la imagen obtenida como resultado tras detectar que los dados han sido lanzados, a este axes se le asignara la etiqueta imagen; La siguiente herramienta a utilizar en la misma capa corresponde a un Estatic Text el cual contendrá el título del axes, para ello se requieren modificar las propiedades BackgroudColor, FontSize, FontWeight y String, en este último se asigna el texto Captura Tomada, como se puede ver en la Figura 73.

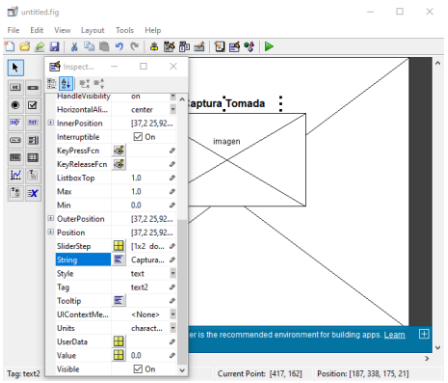


Figura 73 Texto estático Captura Tomada.

En la misma capa se utiliza la herramienta Panel y se modifican las siguientes características en el Inspector, BackgroudColor, FontSize, ShadowColor y Title, a este último se le asigna el texto Panel De Acceso. Para la tercera capa se agrega un Panel y Push Button, al primer item se le modifican las propiedades, BackgroundColor, ShadowColor y Title, a este último se le pone la palabra Resultado y al segundo String, FontSize y Tag, a este último al igual que el anterior al String se le coloca la palabra Iniciar, el Push Button será el encargado de iniciar toda la secuencia de ejecución del programa principal, tal como se aprecia en la Figura 74.

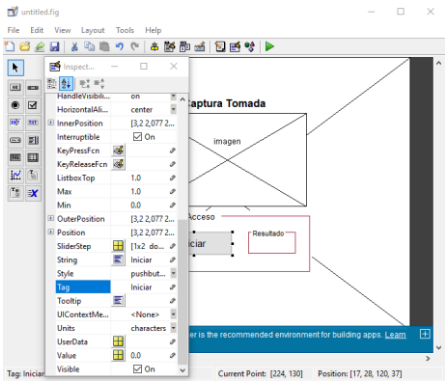


Figura 74. Push Button Iniciar.

Para la cuarta y última capa se agrega un Estatic Text, este mostrara el resultado obtenido después de agitar los dados, para ello se modifican las propiedades, Background, se borra el apartado String y se escribe en Tag se asigna la palabra resultado.

Figura 75 Diseño final de la interfaz GUIDE.

Luego de asignar la ubicación de los objetos que conformaran la interfaz basada en Guide de Matlab y configurar las propiedades de cada elemento con las cuales el usuario ha de interactuar, se procede a guardar el diseño de la Figura 75.

7.2. Resultados

Tras ingresar dentro del GUIDE diseñado los algoritmos desarrollados de identificación y clasificación, se procede a realizar varios ensayos ingresando al sistema datos que desconoce por completo y revisando el error relativo en la clasificación final del sistema.

$$e_R = \frac{|V_E - V_V|}{V_V} * 100 \quad (43)$$

Obteniendo que, para un total de 120 lanzamientos continuos el error relativo es:

$$e_R = \frac{|119 - 120|}{120} * 100 = 0.83\% \quad (44)$$

Un ejemplo de lo observado a través de la interfaz desarrollada es apreciable en la Figura 76, cual muestra el valor obtenido y una imagen del tapete, un botón de inicio, botón donde se muestran los resultados de la clasificación, botón donde se muestra el resultado de la predicción y también reproduce mediante un altavoz el valor obtenido.

Figura 76 Interfaz final en tiempo real.

8

8. CONCLUSIONES Y TRABAJO FUTURO

Lo expuesto anteriormente permite concluir que es posible incluir a personas no videntes o con baja capacidad visual en juegos con características similares al del juego Craps, por lo tanto, se motiva al lector o interesados a usar la metodología expuesta en este trabajo como herramienta para la identificación del resultado de un par dados es este tipo de ambientes.

Se alcanzo a cabalidad la creación de un ambiente como símil al juego Craps para realizar los respectivos ensayos y la adquisición de datos, sin embargo, es bueno mencionar que, alrededor del 70% o más del trabajo depende de este tipo de espacio, su iluminación, la resolución del dispositivo que adquiere las imágenes y el grado de estabilidad de la zona de trabajo son los factores que garantizan la calidad de los datos recolectados, por lo tanto se

puede afirmar que entre mejor desarrollados sean estos tres factores proporcionalmente mejores resultados serán obtenidos.

El algoritmo desarrollado para capturar, procesar y analizar los datos de interés presentes en una imagen hace uso de técnicas de visión artificial como filtros, operaciones morfológicas, conversiones de espacio color y clustering, logrando así diseñar un sistema robusto, de bajo consumo computacional, intuitivo y amigable con el usuario y principalmente facilitando la segmentación de los datos de interés y reduciendo la dimensionalidad de los datos de entrada en su salida.

Se hace uso del aprendizaje profundo para realizar clasificación multiclase a través de las redes convolucionales. Como bien es sabido, el entrenamiento de este tipo de redes requiere un alto consumo computacional, por lo tanto, maquinas con bajos recursos tardan demasiado tiempo en ajustar un modelo que satisfaga los datos de entrada, en este trabajo alrededor de entre 10 a 40 minutos para las arquitecturas propuestas; no obstante, tras obtener tal modelo el proceso de clasificación es menos costoso y bastante rápido el uso de este tipo de herramienta.

El algoritmo implementado es la red neuronal de regresión generalizada (GRNN) para predecir el resultado de los dados en el lanzamiento, se hizo mediante la preclasificación de las imágenes descompuestas que describen la trayectoria de los dados, en efecto en este trabajo es de mayor complejidad. Ya que en los dispositivos Andorid se requieren mínimo 240 fotogramas por segundo a una resolución de cámara HD de 720p que proporciona imágenes de 1280 x 720 píxeles (921.600 píxeles).

La interfaz de usuario diseñada es muy intuitiva y minimalista, esto se debe a que básicamente lo que se buscaba presentar al usuario es una versión objetiva en lo que se refiere al uso de este Software en campo, por lo tanto, solo basta con presionar un botón dentro de la interfaz para inicializar todo el sistema y otro para finalizarlo.

La comunicación usada para la adquisición de datos y su puesta en marcha en tiempo real para la clasificación multiclase prevista, es muy estable y eficaz exceptuando la latencia natural del sistema, sin embargo, los retardos que se introducen al momento de dar un resultado para un caso de estudio específico, se deben a las características del host que desarrolla el procesamiento.

Evidentemente lo resultados obtenidos finalmente con el proyecto no han sido por completo los esperados, considerando que el sistema presenta un error natural del 1% en relación a la base de datos construida, sin embargo, es una tolerancia aceptable para este tipo de prototipo considerando que es un primer acercamiento a un posible desarrollo e implementación final.

Cabe destacar que no fue posible realizar las pruebas con una GPU. Todas las pruebas se realizaron con ordenadores y servidores que contaban con una única CPU. Esto hizo tedioso el proceso de entrenamiento y evaluación de las redes. Realizar pruebas para variar y probar distintas configuraciones llevaba mucho tiempo.

Se le agrego al algoritmo la capacidad de predecir el resultado de los dados antes de que terminen de rotar sobre el tapete del juego Craps con ciertas limitaciones en el tiempo de respuesta, para ello se usó un dispositivo capaz de capturar más de 240 fps a una resolución mínima de 1080p.

8.1 Futuras posibles ampliaciones del trabajo

Evidentemente, los resultados son bastante buenos para ser usados en un entorno real del juego Craps. sin embargo, la clasificación no es totalmente fidedigna. Para lograrlo, una posibilidad es mejorar la base de datos, es decir adquirir una base de datos por el orden de los 10 miles con características iguales o similares a la base de datos usada en este documento para incrementar tanto el número de datos de entrenamiento como de testeo.

Además, una vez se alcanzará una clasificación multiclase fidedigna, podría ampliarse la capacidad del algoritmo. Por ejemplo, podría sistematizarse en conjunto con un dispositivo de desplazamiento cartesianos para tras localizar la posición de los dados desplazar el dispositivo Android con cámara al tal ubicación y así adquirir mejores imágenes del estado de los dados tras ser lanzados.

9

9. REFERENCIAS

[1] " El sentido más importante (la vista)." https://www.elobservador.com.uy/nota/la-vista-es-el-sentido-mas-importante-20151304550 (accessed Jun. 22, 2021).

[2] "LEY 1680 NOV 2013 - Discapacidad Colombia." https://discapacidadcolombia.com/index.php/legislacion/122-ley-1680-nov-2013 (accessed Jun. 09, 2021).

[3] P. López-Jaramillo, P. A. Camacho López, D. Gómez-Arbeláez, J. José, and R. Serrano, "La ceguera y la discapacidad visual son una prioridad de la salud pública en Colombia." Accessed: Jun. 22, 2021. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/.

[4] "Craps | scj." https://www.scj.gob.cl/funcionamiento-de-los-juegos/juegos-de-dados/craps (accessed Jun. 22, 2021).

[5] Carrera, "'RECONOCIMIENTO AUTOMÁTICO DE CARTAS DE BARAJAS UBICADAS SOBRE UNA MESA'UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO."

[6] H. Ortega, R. Tufiño, and J. Estévez, "Hacia la construcción de un dispositivo de asistencia para personas no videntes en el juego de cuarenta," *Enfoque UTE*, vol. 8, no. 4, pp. 27–40, Sep. 2017, doi: 10.29019/enfoqueute.v8n4.170.

[7] A. Rohlfing-Das, "Image Classification for Playing Cards." https://medium.com/swlh/image-classification-for-playing-cards-26d660f3149e (accessed Mar. 23, 2021).

[8] A. Sánchez and C. Ramos, "SEGUIMIENTO VISUAL DE OBJETOS UTILIZANDO TÉCNICAS DE PREDICCIÓN."

[9] E. Politecnica Superior, D. Y. RECONOCIMIENTO DE ELEMENTOS EN MESAS DE PÓKER ONLINE Manuel Jorge Gómez Campo Tutor, Á. García Martín Ponente, and J. María Martínez Sánchez, "UNIVERSIDAD AUTONOMA DE MADRID TRABAJO FIN DE GRADO," 2018.

[10] B. A. B. Correia, J. A. Silva, F. D. Carvalho, R. Guilherme, F. C. Rodrigues, and A. M. de Silva Ferreira, "Automated detection and classification of dice," in *Machine Vision Applications in Industrial Inspection III*, Mar. 1995, vol. 2423, pp. 196–202, doi: 10.1117/12.205506.

[11] K. Y. Huang, "An auto-recognizing system for dice games using a modified unsupervised grey clustering algorithm," *Sensors*, vol. 8, no. 2, pp. 1212–1221, 2008, doi: 10.3390/s8021212.

[12] I. Patel and S. Patel, "Flower identification and classification using computer vision and machine learning techniques," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 277–285, Aug. 2019, doi: 10.35940/ijeat.E7555.088619.

[13] "Automated Counting Dice." http://www.roborealm.com/tutorial/Counting_Dice/index.php (accessed May 26, 2021).

[14] "TÉCNICAS Y ALGORITMOS BÁSICOS | Enhanced Reader." moz-extension://6d56dbec-85dd-4be5-844d-22b6dbedda95/enhanced-reader.html?openApp&pdf=https%3A%2F%2Fpublicaciones.unirioja.es%2Fcatalogo%2Fonline%2FVisionArtificial.pdf (accessed May 17, 2021).

[15] "Especificaciones técnicas: moto G7 power | Asistencia de T-Mobile." https://es.t-

mobile.com/support/devices/android/moto-g7-power/tech-specs-moto-g7-power (accessed May 30, 2021).

[16] "Cómo usar la cámara de tu iPhone y Android como webcam." https://blogthinkbig.com/apps-para-convertir-tu-iphone-y-android-en-una-webcam (accessed Jun. 29, 2021).

[17] "5.1 Protocolo TCP/IP (Transmission Control Protocol/Internet Protocol)." https://upanama.e-ducativa.com/archivos/repositorio/6000/6126/html/51_proto.htm (accessed May 17, 2021).

[18] "Cómo utilizar los fotogramas en vídeo (frame rate) - CasanovaFotoBlog." https://www.casanovafoto.com/blog/2014/10/frame-rate/ (accessed Jun. 29, 2021).

[19] "Cómo Usar Los FPS - Fotogramas Por Segundo - 24, 25, 30, 60, 100, 120 Fps." https://www.editalo.pro/videoedicion/fps/ (accessed Jun. 29, 2021).

[20] "Los procesadores Intel Alder Lake-S se lanzarán al mercado en 2021." https://hardzone.es/noticias/procesadores/intel-alder-lake-lanzamiento-2021/ (accessed Jun. 23, 2021).

[21] "Laptop Inspiron 3467 de 14 pulgadas con pantalla HD antirreflejo | Dell Colombia." https://www.dell.com/co/p/inspiron-14-3467-laptop/pd (accessed May 30, 2021).

[22] "D-Link DSS-100E-18P: Análisis switch videovigilancia Long-Range PoE." https://www.redeszone.net/analisis/switches/d-link-dss-100e-18p/ (accessed Jun. 23, 2021).

[23] "Comparación de los cuatro principales lenguajes de programación de aprendizaje automático: R, Python, MATLAB, Octave - programador clic." https://programmerclick.com/article/64751476429/ (accessed Jul. 01, 2021).

[24] N. Jagadeesh and C. M. Patil, "Iris recognition system development using MATLAB," in *Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017*, Feb. 2018, vol. 2018-Janua, pp. 348–353, doi: 10.1109/ICCMC.2017.8282706.

[25] V. B. Macías, "Herramientas computacionales para la matemática MATLAB: Funciones lógicas y estructuras de control," 2012.

[26] M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella, "Computer vision for

assistive technologies," *Comput. Vis. Image Underst.*, vol. 154, pp. 1–15, Jan. 2017, doi: 10.1016/j.cviu.2016.09.001.

[27] M. Alejandra and S. Lizarazo, "SISTEMA DIGITAL DE RECONOCIMIENTO DE PATRONES MEDIANTE REDES NEURONALES CONVOLUCIONALES."

[28] S. Barreto, "TRANSFORMACIONES GEOMÉTRICAS SOBRE IMÁGENES DIGITALES."

[29] D. Gallardo, R. • Jesús, G. Martín, • Carlos, and P. Lombardo, "DETECCIÓN Y EMBORRONADO FACIAL."

[30] nombre Ruiz Fernández and L. Ángel, "Aplicación de filtros morfológicos en imágenes."

[31] I. The MathWorks, "MATLAB R2017b." 2017.

[32] F. Ponce, R. Tutora, and I. Fondón García, "Proyecto Fin de Carrera Ingeniería de Telecomunicación Evaluación de distancia de color perceptiva en el espacio de color CIE L*a*b* para la segmentación de imágenes de color."

[33] M. Boscarol, "El espacio de color L*a*b* | Imagen digital." http://www.gusgsm.com/el_espacio_de_color_lab (accessed Jun. 01, 2021).

[34] E. T. Hernández, "La medida práctica del color-1." https://docplayer.es/32013432-Capitulo-4-la-medida-practica-del-color-en-el-capitulo-anterior-se-han-abordado-los-aspectos-basicos-sobre-los-que-se.html (accessed Jun. 01, 2021).

[35] P. E. Danielsson, "Euclidean distance mapping," *Comput. Graph. Image Process.*, vol. 14, no. 3, pp. 227–248, Nov. 1980, doi: 10.1016/0146-664X(80)90054-4.

[36] Garófano Alcaraz Carmen, "ANÁLISIS DE IMÁGENES DIGITALES MEDIANTE TÉCNICAS DE APRENDIZAJE PROFUNDO PARA LA DETECCION DE MELANOMAS." moz-extension://6d56dbec-85dd-4be5-844d-22b6dbedda95/enhanced-reader.html?openApp&pdf=https%3A%2F%2Friuma.uma.es%2Fxmlui%2Fbitstream%2Fhandle%2F10630%2F16965%2FAlcaraz%2520Garfano%25C2%25B4CarmenMemoria.pdf%3Fsequence%3D1%26isAllowed%3Dy (accessed May 24, 2021).

[37] "Introducing Deep Learning with MATLAB - MATLAB & Simulink." https://www.mathworks.com/campaigns/offers/next/deep-learning-ebook.html (accessed May 24, 2021).

[38] "MathWorks Introducing Deep Learning with MATLAB." Accessed: May 24, 2021. [Online]. Available: https://it.mathworks.com/campaigns/offers/deep-learning-with-

matlab.html.

[39] "Diseño, entrenamiento y análisis de redes de deep learning." https://es.mathworks.com/products/deep-learning.html (accessed May 24, 2021).

[40] "Learn About Convolutional Neural Networks - MATLAB & Simulink - MathWorks Italia." https://it.mathworks.com/help/deeplearning/ug/introduction-to-convolutional-neural-networks.html (accessed May 24, 2021).

[41] "Learn About Convolutional Neural Networks - MATLAB & Simulink - MathWorks América Latina." https://la.mathworks.com/help/deeplearning/ug/introduction-to-convolutional-neural-networks.html (accessed Jun. 16, 2021).

[42] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6354 LNCS, no. PART 3, pp. 92–101, doi: 10.1007/978-3-642-15825-4_10.

[43] "Dropout Layer - Artificial Inteligence." https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/dropout_layer (accessed May 29, 2021).

[44] B. Yuan, "Efficient hardware architecture of softmax layer in deep neural network," in *International System on Chip Conference*, Jul. 2016, vol. 0, pp. 323–326, doi: 10.1109/SOCC.2016.7905501.

[45] M. Hudson, B. Martin, T. Hagan, and H. B. Demuth, "Neural Network Toolbox™ 7 User's Guide," 1992. Accessed: May 24, 2021. [Online]. Available: www.mathworks.com.

[46] "Architecture Radial Basis Networks (Neural Network Toolbox) ." http://matrix.etseq.urv.es/manuals/matlab/toolbox/nnet/radial14.html (accessed Jun. 30, 2021).

[47] Z. L. Wang and H. H. Sheng, "Rainfall prediction using generalized regression neural network: Case study Zhengzhou," in *Proceedings - 2010 International Conference on Computational and Information Sciences, ICCIS 2010*, 2010, pp. 1265–1268, doi: 10.1109/ICCIS.2010.312.

[48] "Matriz de confusión | Interactive Chaos."

https://interactivechaos.com/es/manual/tutorial-de-machine-learning/matriz-de-confusion (accessed Jul. 03, 2021).

[49] LUMINA, "Tubo LED T8 en vidrio," *BIOLED*, 2019. https://bioled.com.co/producto/tubo-led-t8-vidrio/.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2015, Accessed: Jun. 08, 2021. [Online]. Available: http://www.robots.ox.ac.uk/.