



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**RECONOCIMIENTO DE PLACAS VEHICULARES
MEDIANTE VISION COMPUTACIONAL PARA
MEJORAR EL ACCESO A UN PARQUEADERO**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
DE SISTEMAS**

Autor:

Bach. Perez Silva Edwin Wildor

ORCID: <https://orcid.org/0000-0003-3345-8774>

Asesor:

Mg. David Enrique Bances Saavedra

ORCID: <https://orcid.org/0000-0002-7164-8918>

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2022

APROBACIÓN DEL JURADO

TITULO

Bach, Edwin Wildor Pérez Silva

Autor

Mg, Bances Saavedra David Enrique

Asesor

Dr. Vásquez Leyva Oliver
Presidente de Jurado

Mg, Chirinos Mundaca Carlos Alberto
Secretario de Jurado

Mg, Bances Saavedra David Enrique

Vocal de Jurado

Dedicatorias

El presente proyecto de investigación se lo dedico esencialmente a mis padres María Eva Silva Estela y Rómulo Pérez Barboza ellos constituyen los pilares fundamentales en mi vida cotidiana y en mi formación personal, aquellos que me han educado con principios y valores para ser personas integras ante la sociedad. A mi esposa Ninfa Flores Mayta, y a mis hijos Alexandra Nicolle Pérez Flores y Albert Ivan Pérez Flores por motivarme para seguir superándome en los distintos aspectos de mi vida.

Edwin Wildor Pérez Silva.

Agradecimiento

El agradecimiento en primer lugar a Dios, por darme la vida, la salud y la sabiduría para lograr mis objetivos propuestos. Además, agradecerle infinitamente a mi familia y amigos, que me han ayudado en mi desarrollo personal y por su apoyo incondicional para poder cumplir mis metas trazadas.

Al mismo tiempo agradecer a nuestra casa de estudios la Universidad Señor de Sipán, quien me dio la oportunidad para mi formación universitaria y me brindó los conocimientos necesarios para ser profesionales competitivos en un mundo integrado. Finalmente agradecer a todos los docentes de la carrera de Ingeniería de Sistemas por brindarnos su amistad, sus conocimientos aprendidos a través de una enseñanza de calidad y además del apoyo incondicional.

Edwin Wildor Pérez Silva.

Resumen

En el sector automotriz del Perú se utilizan diversas formas para controlar el ingreso de vehículos a parqueaderos tales como: tickets de control, fotografía del vehículo, encargo de la tarjeta de propiedad, software a medida o libre. Sin embargo, el uso de software es en porcentaje mínimo con respecto al total de parqueaderos que existen en el país; por tanto, ocurren distintos problemas de seguridad tales como pérdida del vehículo, demoras a la salida de un parqueadero, pérdida de documentos.

La presente investigación tuvo como objetivo esencial el desarrollo de un sistema de reconocimiento de placas vehiculares mediante visión computacional. En primer lugar, realizamos el análisis de diversas técnicas aplicados en reconocimiento de placas vehiculares, luego seleccionamos la data que serían imágenes o secuencia de video donde se observe el ingreso de vehículos, se utilizó algoritmos de obtención de contornos, umbralización y segmentación, para el entrenamiento y aprendizaje se implementó un algoritmo KNN, finalmente aplicamos un algoritmo de OCR para mostrar los caracteres que se encuentran en la placa.

El rendimiento del sistema se midió con relación a la precisión 89.9% en caracteres numéricos, 80.5% en reconocimiento de letras, exactitud 36.4%, recall 46.9%. Llegando a la conclusión, dado las diferencias mínimas en la precisión en trabajos según la literatura revisada, podemos decir que el algoritmo diseñado fue efectivo.

Palabras claves:

Visión artificial, Algoritmos KNN, Procesamiento de imágenes, Reconocimiento de placas.

Abstrac

In the automotive sector in Peru, various forms are used to control the entry of vehicles to parking lots such as: control tickets, vehicle photography, ordering the property card, custom or free software, However, the use of software is a minimum percentage with respect to the total parking spaces that exist in the country; therefore there are different security problems such as loss of the vehicle, delays when leaving a parking lot, loss of documents.

The main objective of this research was the development of a system for recognizing license plates by means of computer vision. First of all, we carry out the analysis of various techniques applied in the recognition of license plates, then we select the data that would be images or video sequence where the entry of vehicles is observed, algorithms for obtaining contours, thresholding and segmentation were used, for training and learning a KNN algorithm was implemented, finally we apply an OCR algorithm to display the characters found on the plate.

The performance of the system was measured in relation to precision 89.9% in numerical characters, 80.5% in letter recognition, accuracy 36.4%, recall 46.9%. Coming to the conclusion, given the minimal differences in precision in works according to the literature reviewed, we can say that the designed algorithm was effective.

Keyword: Machine vision, Algorithms KNN, Image processing, Plate recognition.

Indices

I. INTRODUCCIÓN.....	10
1.1. Realidad Problemática.....	10
1.2. Trabajos Previos.	13
1.3. Teorías Relacionadas con el Tema.....	20
1.4. Formulación del Problema.	47
1.5. Justificación e Importancia del estudio.	47
1.6. Hipótesis.....	48
1.7. Objetivos.....	48
1.7.1. Objetivo general.....	48
1.7.2. Objetivos específicos.....	49
II. MATERIAL Y MÉTODO.....	50
2. 1. Tipo y Diseño de Investigación.	50
2. 1. Población y Muestra	50
2.4. Técnicas e Instrumentos de Recolección de Datos, Validez y confiabilidad.....	53
2.5. Procedimiento de análisis de datos	54
2.5.1. Análisis Estadístico e Interpretación de los datos	55
2.6. Criterios Éticos.....	57
2.7. Criterios de Rigor Científico.....	57
III. RESULTADOS.....	58
3.1. Resultados en tablas y figuras.....	58
3.1.1. Tiempo de respuesta	58
3.3. Aporte Práctico	66
Evaluar el rendimiento del sistema	86
IV. CONCLUSIONES Y RECOMENDACIONES.....	88
4.1. Conclusiones:	88
4.2. Recomendaciones.....	90
Referencias.....	91
Anexos.....	95

Indices de tablas

Tabla 1: Matriz de operacionalización de las variables de la investigación.	52
Tabla 2: Porcentaje de consumo de CPU.....	59
Tabla 3: Porcentaje de consumo de memoria RAM.	60
Tabla 4: Exactitud del algoritmo kNN.....	61
Tabla 5: Precisión del algoritmo kNN.....	62
Tabla 6: Recall del algoritmo kNN.....	63
Tabla 7: Personal involucrado en el desarrollo del sistema.....	68
Tabla 8: Personal involucrado en el desarrollo del sistema.....	68
Tabla 9: Captura de imagen.....	69
Tabla 10: Preprocesamiento de la imagen.....	69
Tabla 11: Localización de la placa del vehículo	69
Tabla 12: Segmentación de caracteres	70
Tabla 13: Reconocimiento de caracteres de la placa vehicular	70
Tabla 15: Elementos que componen arquitectura del sistema	73

Índice de Figuras

Figura 1: Representación general del proceso de imágenes.....	21
Figura 2: Representación de visión general por computador.....	22
Figura 3: Reconocimiento de caracteres en base a su codificación radial... 22	22
Figura 4: Un sistema de visión en su arquitectura.....	23
Figura 5: Operación puntual.....	25
Figura 6: Función de transformación	25
Figura 7: Presenta un modelo de una imagen o retrato que ha logrado binarizar	27
Figura 8: Modelo de operaciones exactas	28
Figura 9. Filtro gaussiano con diferentes radios.....	29
Figura 10. Imagen resultante (derecha) del algoritmo	32
Figura 11. Imagen resultante (derecha) del algoritmo	33
Figura 12. Imagen resultante (derecha) del algoritmo	33
Figura 13. Imagen resultante (derecha) del algoritmo	34
Figura 14. Imagen resultante (derecha) del algoritmo	34
Figura 15: Conseguimos un perro dálmata reconocer	35

Figura 16: El sistema visual en automático completa	36
Figura 17: Modelo de discontinuidades	37
Figura 18: El Sobel de operadores, prestar atención del incluido	38
Figura 19: Localización con los operadores de orillas de Sobel	38
Figura 20: Modelo de retrato con significativas de las regiones	39
Figura 21: Fraccionamiento que se da por histograma	40
Figura 22: El ejemplo de la controlada clasificación	42
Figura 23: Ejemplo de representación del ciclo de diseño de un clasificador	44
Figura 24: Paso 1: La aleatoria inicialización de los centros	45
Figura 25: Paso 2: En la asignación de puntos a ejes en función de los recorridos.....	46
Figura 26 Paso 3: El centro se traslada atravesando a ser	47
Figura 27: Algoritmo K-means	47
Figura 28: Porcentaje de consumo de CPU	59
Figura 28: Porcentaje de consumo de memoria RAM.....	60
Figura 29: Exactitud del algoritmo kNN	61
Figura 30: Precisión del algoritmo kNN	62
Figura 31: Recall del algoritmo kNN	63
Figura 32: Vista lógica de los componentes de la arquitectura del sistema desarrollado.....	71
Figura 33. Vista física de los componentes de la arquitectura del sistema desarrollado.....	72
Figura 34: Distribución del contenido de la placa de rodaje,	75
Figura 35: Dataset implementado con imágenes de placas vehiculares.	76
Figura 36: Conversión de imagen a formato HSV	77
Figura 37: Maximizar contraste.....	78
Figura 38: Desenfocado Gaussiano	79
Figura 39: Imagen binarizada por umbralización	80
Figura 40: Segmentación de la placa de la imagen.....	82
Figura 41: Segmentación de los caracteres de la placa vehicular.....	84
Figura 42. Plantilla de fuente de muestra.....	85
Figura 43: Reconocimiento e interpretación de caracteres con kNN.....	85

I. INTRODUCCIÓN

1.1. Realidad Problemática

Tenemos que en los últimos tiempos especialmente en las ciudades de la costa peruana se ha generado la necesidad de encontrar un espacio para el cuidado vehicular, por la realidad que vivimos se ha vuelto algo habitual, requerimos los servicios o lugares de estacionamientos controlados para evitar hurtos o daños a los vehículos; consecuentemente el tiempo que se demora para ingresar a los estacionamientos generan más emisión de CO2 contaminando el medio ambiente (Guerrero Ibañez, Acosta Mendoza, & Contreras Castillo, 2015), utilizando la tecnología actual como los sistemas inteligentes basados en el reconocimiento de placas de los vehículos pueden ser una elección para neutralizar este problema, reconocer un vehículo de manera más rápida por que le permiten identificar (Simon Mandi, Shibwabo, & Mutua Raphael, 2017).

Los sistemas de transporte inteligentes (ITS) son un campo de estudio muy diverso que se ha transformado en un área de beneficio popular para varios investigadores, se ocupa de la aplicabilidad de la tecnología en las comunicaciones y en la información para acrecentar el beneficio de los sistemas de transporte. Como tal, tanto los viajeros como los automovilistas pueden atravesar las carreteras con mejor movilidad, viajes más seguros y mayor comodidad, los algoritmos para la automatización de procesos manuales pueden tener numerosas aplicaciones que incluyen, entre otras, sistemas de gestión de estacionamiento, sistemas de cobro de peajes electrónicos, comunicaciones entre vehículos y sistemas de vigilancia vial. (Quiros, y otros, 2017).

Se conoce que hasta el año de dos mil quince el parque automotriz paso los cinco millones de coches, en la metrópoli de Lima la generalidad de coches concentrados, como resultante, la congestión y la escasez de espacios para estacionar, se ha transformado en un problema que tiende a agudizarse, en diversos casos el diseño de los estacionamientos no es el apropiado, creando el desorden y grandes esperas en las colas, no solo

para ingresar a un estacionamiento, sino también para encontrar el parqueo disponible (CEPLAN, 2021).

Actualmente existen parqueaderos que no cuentan con un sistema que permita realizar el reconocimiento de las placas de los vehículos para garantizar un acceso rápido y seguro en lugares donde se requiera ciertas medidas de seguridad; ya que todo esto gestionado por una personas lo cual está sujeto al error humano.

Existen estudios que explican todos los beneficios del uso de los diferentes sistemas de visión computacional en el reconocimiento de placas vehiculares, entre los trabajos afines más importantes para el presente estudio son los siguientes.

En la investigación Un enfoque basado en kNN para la visión artificial del reconocimiento de caracteres de los números de matrícula, Varios estudios han utilizado la clasificación del vecino k-más cercano para realizar el reconocimiento de patrones, en presente estudio el problema que enfrentó fue la comparación de caracteres existentes con los métodos de reconocimiento, como tenemos las redes neuronales artificiales (ANN), la precisión, además el tiempo medio de procesamiento. La investigación propone automatizar la placa proceso de reconocimiento mediante la instalación de una cámara IP en una carretera y el análisis de la transmisión de video para capturar los vehículos a lo largo de esa carretera. Al obtener la imagen de la placa, se computaron los contornos dentro de la placa y se evaluó si eran caracteres válidos según su tamaño. Tras la validación, las placas se segmentaron en los contornos detectados. Cada contorno se clasifica mediante un algoritmo kNN. Las simulaciones muestran que el sistema logró una precisión del 87,43% para el reconocimiento de placas. Dado que la diferencia en las precisiones de los algoritmos kNN y ANN era insignificante, se puede decir que el algoritmo diseñado estaba a la par con técnicas precisas (Quiros, y otros, 2017).

Realizaron la investigación Diseño de sistema de reconocimiento de matrículas basado en aprendizaje automático, el problema principal de la

detección de matrículas es detectar y extraer el área de la matrícula de las imágenes recopiladas del vehículo, incluida principalmente la ubicación de la matrícula y el juicio de la misma, para el reconocimiento de las matrículas es el reconocimiento de caracteres del área de matrículas adquiridas. La segmentación de caracteres es necesaria antes de reconocimiento de caracteres. El reconocimiento de caracteres adopta una red neuronal convolucional (CNN). Los resultados efectivos manifiestan que el método de ubicación de la placa y el juicio de la placa tiene una alta confiabilidad, y el reconocimiento de caracteres de la placa tiene una alta precisión. (Zhang, Yu, Su, & Li, 2019).

1.2. Trabajos Previos.

(Hung Lin & Sin Sie, Two-Stage License Plate Recognition System Using Deep learning, 2019), Realizo un estudio, Sistema de reconocimiento de matrículas de dos etapas mediante aprendizaje profundo, el estudio propone una nueva metodología de dos etapas basada en la tecnología de aprendizaje profundo con desafíos en diferentes entornos como: ángulos de disparo y personajes de matrículas borrosos en vehículos en movimiento, otros problemas incluyen condiciones complejas de la carretera, letreros publicitarios, señales de tráfico e indicadores de nombres de carreteras, el sistema de reconocimiento de matrículas aplicado al monitor de intersección se divide en cuatro pasos, incluida la detección de vehículos, detección de matrículas, segmentación de caracteres y reconocimiento de caracteres. Este método puede enfocar el reconocimiento de matrículas en el propio vehículo, pero este paso complicado hará que el sistema de reconocimiento de matrículas sea menos eficiente. Los resultados experimentales muestran que la metodología alcanza el 98,23% de la tasa de detección de matrículas y el 97,38% de la tasa de reconocimiento de caracteres. El rendimiento de la metodología jerárquica es de unos 25 fps. Esta metodología muestra la superioridad tanto en precisión como en rendimiento en comparación con los sistemas tradicionales de reconocimiento de matrículas.

(Satsangi, Yadav, & Sewak Sudhish, 2018) En su artículo License Plate Recognition, presenta una comparación basada en consideraciones como exactitud promedio, precisión y recuperación entre algoritmos según valores de umbral, reconocimiento de caracteres. Podemos hacer nuestro reconocimiento de matrículas en tres pasos secuenciales que involucran la detección, así como la captura de una imagen del vehículo seguida de la detección y extracción de la matrícula y finalmente usando la técnica de segmentación de imágenes para obtener los caracteres individuales. En este trabajo se realizó el reconocimiento de matrículas

mediante el algoritmo de Viola Jones. Comparamos la salida obtenida de esto con las salidas de la tecnología de umbral y OCR. Entre las tres competencias probadas, llegamos a la conclusión de que el enfoque de aprendizaje automático que hizo uso del algoritmo de Viola Jones no capacitado fue el más eficiente.

(Zainal Abedin, Chandra Nath, Dhar, Deb, & Shahadat Hossain, 2017) realizó un estudio, del sistema está implementado en el entorno Python OpenCV para imágenes de matrículas de automóviles fuera de línea que se toman en diferentes iluminaciones, escenarios de carreteras y autos de colores. El rendimiento del sistema se evalúa en términos de tasa de detección, tasa de segmentación, tasa de reconocimiento y tiempo de ejecución. La detección se basa en el procesamiento morfológico y suavizado gaussiano, umbral adaptativo y propiedades de contorno de filtrado de los caracteres en la parte inferior de la placa. El reconocimiento se lleva a cabo mediante el modelo de aprendizaje profundo de redes neuronales de convolución multicapa (CNN). Los resultados de la simulación revelan que el enfoque propuesto tiene un rendimiento significativo en términos de precisión de detección, segmentación y eficiencia en el tiempo de ejecución. El algoritmo muestra un buen rendimiento en el procesamiento de varios tipos de imágenes como lluviosas, rotadas, diferentes iluminaciones y bajo contraste. El algoritmo logra una tasa de éxito del 93% para la detección de LP, el 98% para la segmentación de caracteres y el 98% para el reconocimiento y el tiempo de ejecución es de 0,11 segundos, lo cual es muy importante para la implementación en tiempo real. La detección de la placa se basa en las propiedades de contorno de los caracteres de la parte inferior de la placa y el reconocimiento se realizan mediante un enfoque de aprendizaje profundo denominado red neuronal convolucional.

(Agbeyangi, Alashiri, & Otunuga, 2020). Realizó la investigación, Automatic Identification of Vehicle Plate Number using Raspberry Pi. En donde menciona que, en muchos países, los formatos de las placas con

licencia a menudo difieren, pero las técnicas de reconocimiento automático pueden ser las mismas (detección, segmentación y reconocimiento de caracteres). De las tres técnicas clave de reconocimiento automático, la tarea más crucial es detectar la matrícula y su falla afectará en gran medida la precisión del reconocimiento. La imagen de la placa del vehículo se captura con la ayuda de la cámara Pi de 2MP interconectada y se almacena en una tarjeta de memoria SD para su preprocesamiento y reconocimiento. Una vez realizado el preprocesamiento mediante el inicio de OpenCV, los caracteres de la placa se reconocen mediante el reconocimiento de caracteres en forma óptico (OCR) y se muestran los caracteres correspondientes que se encuentran en la placa. Cada etapa del proceso de desarrollo se probó y evaluó en relación con la configuración existente. Las etapas involucradas son pre-implementación y post-implementación.

- Pruebas previas a la implementación. - La prueba previa a la implementación se llevó a cabo con los componentes para garantizar que cada componente se especificara.
- Pruebas posteriores a la implementación. - La prueba se realizó para garantizar que los componentes del sistema funcionen correctamente. Realizamos la prueba utilizando diferentes números de vehículos en diversas condiciones para mostrar la efectividad de nuestro sistema, la elaboración del sistema de reconocimiento de placas de vehículos muestra cómo se puede aplicar el uso de Open CV y OCR en la extracción de caracteres y el reconocimiento de placas de vehículos. Aunque, este diseño es solo una prueba del concepto (prototipo) y, por lo tanto, incluye el paso inicial en un estudio que tiene el potencial de expandirse en el futuro.

(Krishna Thangallapally , Maripeddi, & Krishna Banoth , 2018) realizaron un estudio, Sistema de seguridad electrónica para el seguimiento del número de vehículos en el estacionamiento, en este artículo, en este trabajo los desafíos enfrentados durante todo el proceso son: Imágenes borrosas, Matrículas rotas para el reconocimiento y la detección de las matrículas de los vehículos usando las técnicas de procesamiento de imágenes. Hay seis pasos principales en el reconocimiento y la detección de la matrícula del vehículo

El primer paso es capturar la imagen del vehículo usando una cámara. El siguiente paso es el procesamiento. En este paso, convertiremos la imagen RGB en la retrato en escala que debe ser en grises y eliminaremos el ruido utilizando el filtrado de la mediana y, finalmente, usaremos el método de umbral adaptativo, la imagen para convertir necesariamente en escala de grises en la imagen binaria. Luego, el siguiente paso es extraer la región de la placa de matrícula de la imagen del umbral. El siguiente paso es separar cada carácter de la matrícula, que es la Paso vital para todo el proceso. Si la segmentación no se realiza correctamente, existe el riesgo de obtener resultados inexactos. Cada carácter se reconoce mediante el algoritmo KNN. Primero, entrenaremos nuestro algoritmo con diferentes tipos de caracteres y luego los personajes son reconocidos usando el algoritmo KNN. Estos datos de los diferentes nodos se envían al Arduino que actúa como servidor principal. Los datos en Arduino incluyen detalles sobre la puerta a través de la cual el vehículo ingresó a una localidad en particular y sus tiempos de entrada / salida. Cuando accedemos al servidor utilizando el navegador web, se mostrarán los detalles del vehículo.

Los resultados demuestran que la precisión de detección y reconocimiento es optima. Para el sistema propuesto anteriormente, fue realizado en Python utilizando la biblioteca Opencv3 utilizada para aplicaciones de procesamiento de imágenes.

(Dhar, Zainal Abedin, Karim, Tuj Johora², & Shahadat Hossain, Bangladeshi License Plate Recognition Using, 2019) Realizaron un estudio, Reconocimiento de matrículas de Bangladesh mediante el clasificador Adaboost, para desarrollar una gestión inteligente del tráfico, como la supervisión de vehículos, LPR es un componente clave. Sin embargo, debido a la diversidad de diseños y caracteres de las placas, la solución universal no es posible, por lo tanto, esta investigación se centraliza en el desarrollo del algoritmo para el reconocimiento de matrículas de Bangladesh mediante el uso de procesamiento de imágenes y modelo de aprendizaje automático con buena precisión de detección y reconocimiento. Este algoritmo se ejecuta en tres pasos: detección de la placa con verificación de forma, corrección de inclinación y reconocimiento del número. Para la detección, se aplican el espacio de color RGB, el filtrado medio, la binarización, el análisis morfológico y las propiedades de la región para el filtrado. Para descartar un objeto ruidoso, La verificación de la forma se realiza a través de distancias robustas a los vectores de los bordes. Antes de la segmentación de caracteres, se aplica la corrección de inclinación horizontal. Luego, los caracteres se extraen utilizando parámetros de cuadro delimitador de la placa extraída. Finalmente, el reconocimiento se implementa mediante la combinación de características de gradiente orientado a histograma (HOG) y patrón binario local (LBP) y el clasificador de refuerzo adaptativo (Adaboost) se usa para categorizar los caracteres. El resultado experimental muestra que la precisión de detección y reconocimiento es digna de mención, el algoritmo muestra una tasa de éxito del 97,3% en la clasificación de personajes el reconocimiento se implementa mediante la combinación de características de gradiente orientado a histograma (HOG) y el local del binario patrón (LBP) y el clasificador de refuerzo adaptativo (Adaboost) se usa para categorizar los caracteres.

(Hung Lin, Sin Lin, & Chen Liu, An Efficient License Plate Recognition System Using Convolution Neural Networks, 2018) Realizó la investigación, aunque el sistema de reconocimiento de matrículas se ha aplicado con éxito al sistema de estacionamiento inteligente controlado por el medio ambiente, todavía enfrenta muchos desafíos en el sistema de vigilancia, como tráfico congestionado con múltiples placas, letreros y anuncios ambiguos, placas inclinadas, así como imágenes oscuras tomadas con mal tiempo y de noche. Para resolver estos problemas, proponemos una metodología jerárquica eficiente para el sistema de reconocimiento de matrículas que primero detecta los vehículos mediante técnicas de aprendizaje profundo y luego recupera las matrículas de los vehículos detectados para reducir los falsos positivos en la detección de matrículas. Luego, en la etapa final, proponemos una red neuronal de convolución LPR (LPRCNN) para mejorar el reconocimiento de caracteres de imágenes borrosas y oscuras. Los resultados experimentales muestran que la metodología logra 96,12% de la tasa de detección de vehículos y 94,23% de la tasa de detección de placas. Con LPRCNN, logramos un 99,2% de precisión en el reconocimiento de caracteres. Esta metodología muestra la superioridad tanto en precisión como en rendimiento en comparación con los sistemas tradicionales de reconocimiento de matrículas.

(Atikuzzaman, Asaduzzaman, & Zahidul Islam, Vehicle Number Plate Detection and Categorization Using CNNs, 2019) realizó un estudio: Detección y categorización de matrículas de vehículos mediante CNN, nuestro método propuesto para detectar y reconocer matrículas en tiempo real que está especialmente diseñado para trabajar en videos capturados por una cámara. Es un enfoque distinto que se compone de tres fases principales, como la detección de placas, la segmentación de letras de clase y el reconocimiento. Estas fases se completan mediante la adopción de un clasificador basado en funciones HAAR para detectar matrículas, un extractor de letras de clase con un método propuesto y una red neuronal de convolución para reconocer letras de clase. Nuestro método dado logró resultados cautivadores en nuestro conjunto de datos

recopilados. 91,38% con aproximadamente 30 fotogramas / segundo. Evaluamos el rendimiento de nuestro sistema de detección de matrículas con 390 imágenes de prueba y obtenemos una precisión del 96,92% y la segmentación de letras de clase ha alcanzado el 94,61% con el mismo tamaño de datos. Logramos una tasa de reconocimiento general exitosa del 90,90% con desempeño en tiempo real. Hemos presentado una combinación en cascada con el modelo CNN para un sistema que permita el reconocimiento de letras en su respectiva clase que minimiza el tiempo de cálculo al tiempo que logra una alta tasa de precisión. Debido a que tiene características que extraen capacidades de aprendizaje profundo, CNN funciona mejor que las redes neuronales tradicionales.

(Chen, 2017) realizó un estudio: Chinese License Plate Identification Based On Android Platform, en la actualidad, el sistema de reconocimiento de matrículas se limita a identificar la foto proveniente de una cámara fija, en la gestión del estacionamiento, estacionamiento ilegal, aplicación móvil de la policía de tránsito, entre otros, podemos usar un teléfono inteligente para identificar el número de placa. Presentamos un método basado en la plataforma Android, la identificación de la placa de matrícula utiliza OpenCV en la plataforma Android. El método se basa en la información de color, que está influenciada por las diferencias de color de diferentes fondos y condiciones de luz. Morfología matemática para determinar la matrícula, después de analizar estos métodos, el algoritmo de ubicación de la placa se realiza mediante combinación de segmentación de color y detección de bordes, después de varias pruebas experimentales, el efecto de reconocimiento general es satisfactorio, en la inclinación de la placa de matrícula o en condiciones de poca luz, es necesario mejorar la eficacia del reconocimiento. Además, a diferencia de otros sistemas de reconocimiento de matrículas, todo el sistema se basa en la plataforma Android.

(Taufiq Musaddid, Bejo, & Hidayat, 2019) Realizo un estudio: Mejora de la segmentación de caracteres para el algoritmo de reconocimiento de matrículas de Indonesia mediante CNN, la presente investigación enfrenta uno de los procesos desafiantes es segmentar los caracteres de la placa detectada. Los caracteres segmentados se extraen para realizar el reconocimiento. Por lo tanto, el rendimiento de la segmentación de personajes afecta el resultado final. En este método propuesto, CNN se utiliza para distinguir la región de caracteres y no caracteres. Para alimentar regiones a CNN, se aplica la técnica de ventana deslizante. Los cuadros delimitadores finales finalmente se refinan para aumentar la precisión. Y el modelo desarrollado se probó en 130 imágenes de matrículas de vehículos de Indonesia que contienen 982 caracteres en total y arrojó un 87,06% de precisión. A partir de la implementación del método, se concluye que se necesita un refinamiento del cuadro delimitador para mejorar la precisión de la segmentación.

1.3. Teorías Relacionadas con el Tema.

A efectos de este estudio, el enfoque teórico científico se dirige hacia las técnicas utilizadas para el suavizado de la imagen (pre-procesamiento), binarización (segmentación) y patrones (reconocimiento de objetos individuales).

1.3.1. Visión artificial.

La visión es la lumbrera al universo de diversos cuerpos, la función primordial es localizar y reconocer cosas en los lugares mediante el proceso de los retratos, el enfoque computacional es la tesis de estos procesos, para construir y concebir aparatos con capacidades equivalentes. (Sucar, Visión Computacional, 2017), coexisten diversas tesis de visión, éstas entre las que logramos indicar las sucesivas, visión es estar al tanto de dónde y que hacer, por medio de la vista, visión es rescatar de los sentidos de la vista la información y las pertenencias válidas que se toman exterior del mundo, visión la causa de uno que se

origina a partir de las representaciones pictóricas del exterior del mundo, una representación que solamente útil para el espectador, según esta idea efectiva que se tiene sobre visión computacional de Marr nos dice que se tiene tres aspectos significativos que se debe tener muy presentes como la visión es un asunto o proceso de computadora, también la representación a lograr, estar en manos del espectador y por ultimo nos dice que es obligatorio excluir la información que no le sea útil, es el paso de la reducción de información.

En la actualidad hay un área muy relacionada a la de visión de computadora que es la de proceso de representaciones pictóricas, sin embargo, uno y otro campo tienen mucho en común, el objetivo al final es desigual, se debe tener en cuenta que el objetivo del procesamiento de representaciones pictóricas o imágenes es la de mejorar principalmente la calidad de ellas mismas, para la futura interpretación o uso, a continuación, tenemos unos ejemplos:

- Los defectos remover,
- Problemas por movimiento o desenfoque que debes de remover,
- Se debe mejorar algunas propiedades como contraste, color, estructura y otros.
- También el añadir falsos colores a retratos monocromáticos.

En la figura 1 tenemos una muestra la orientación de procesamiento de representaciones pictóricas, con el cual se logra un retrato mejorado para su futuro procesamiento o paráfrasis por otra persona o investigador.

El meta de la visión por computadora es extirpar particularidades de un retrato para su interpretación y representación por el computador, como, por ejemplo:

- establecer la ubicación y prototipo de objetos en la imagen,



Figura 1: Representación general del proceso de imágenes. Tomado de visión artificial. Fuente: (Sucar & Gómez, Visión Computacional, pág. 2). Se construye una tridimensional graffa de un objeto,

- Después se analiza el objeto para establecer su calidad,
- Se procede a descomponer el objeto o la imagen en partes.

Se puede decir, que en visión la búsqueda está en conseguir cuadros ventajosos para cada tarea a ejecutar, que demandará cambiar algunas propiedades, ver figura 2.

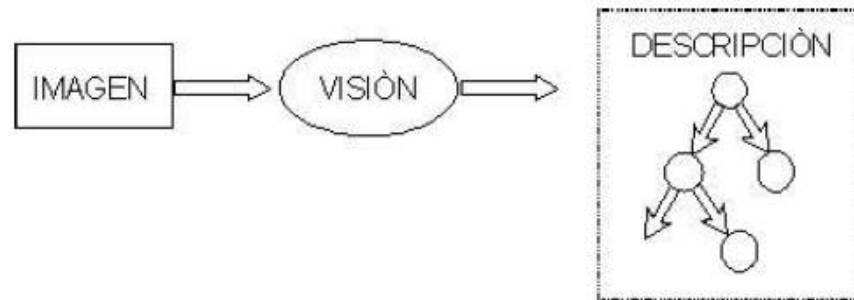


Figura 2: Representación de visión general por computador, se puede apreciar que la imagen de entrada pasa por un proceso para extraer los atributos, logrando obtener como resultado una representación analizada de la imagen. Fuente: (Sucar & Gómez, Visión Computacional, pág. 2).

La figura 3: enseña la disconformidad entre los procesos de retratos y visión; observe que el retrato demuestra ciertas representaciones significativas, como son las cifras, que anticipadamente fueron revelados, la justificación de esta visión del sistema se perfecciona con un paradigma para la afirmación de los patrones, se puede decir con el estar al tanto que números y letras tiene la placa.



Figura 3: Reconocimiento de caracteres en base a su codificación radial. Tomado de "Visión Artificial" Fuente: (Sucar & Gómez, Visión Computacional, pág. 3).

1.3.2. Elementos de un Sistema de Visión

1.3.2.1. Dispositivos para visión

Hay otros terminales para la obtención de retratos, dicho retrato o imagen son almacenadas y digitalizadas en la base de datos que tiene el computador, logrando tenerla en el computador en momentos desde el propio terminal la captura, el retrato o la imagen para logra que sea procesada. (Sucar, Visión Computacional, 2017)

Debemos tener en cuenta que para la toma de la imagen o del retrato se pide del terminal material que tienen que ser muy sensible a una definitiva necesaria banda para la toma del espectro electromagnético adecuado, el terminal origina o emite una señal automática conveniente para al nivel de energía adecuado y que pueda ser detectado, por la cual es digitalizada consecutivamente. (Sucar, Visión Computacional, 2017)

1.3.2.2. Arquitectura de un sistema de visión

Los sistemas característicos de visión que se dan por computador, conjuntamente de un terminal de captura de imagen, debe tener por lo menos otros cuatro componentes o elementos en un terminal de transformación del formato analógico para el formato digital A/D, un elemento de procesamiento, una memoria de video y un pantalla, en la figura 4 se presenta la básica arquitectura educado para un sistema de visión. (Sucar, Visión Computacional, 2017)

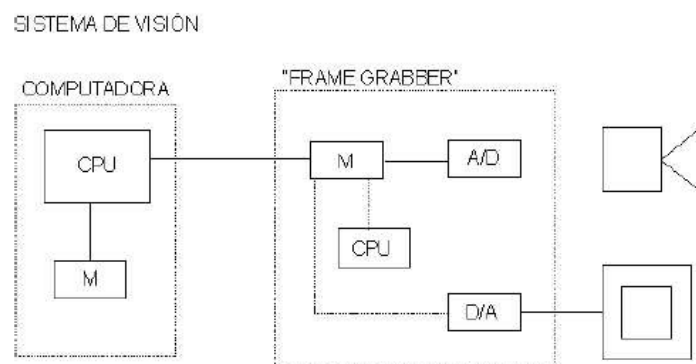


Figura 4: Un sistema de visión en su arquitectura de. Fuente: (Sucar, Visión Computacional, 2017).

1.3.3. Mejoramiento de la imagen

La meta de visión de nivel bajo o proceso adelantado es hacer metamorfosis claramente sobre el retrato o la imagen para conseguir toda la investigación de las participaciones ópticas de las cosas que están en la imagen y que permitan obtener el mejor provecho para los subsiguientes niveles o etapas de visión, las importantes propiedades que se tiene en cuenta y son significativos para conseguir de un retrato o imagen, son como orillas o discontinuidades, de la textura, profundidad, color y gradiente. (Sucar, Visión Computacional, 2017)

De la misma forma, conseguimos recapacitar que, del original retrato o imagen, se logrará obtener siempre una nueva, por cada particularidad que se extirpe del retrato o la pintura lo que Marr denomina el Primal sketch - citadas como imágenes interiores, se enseña en la figura. (Sucar, Visión Computacional, 2017).

Anterior para la obtención de las peculiaridades en diversas veces, obligatorio mejorar el retrato o la imagen para sobresalir todos los aspectos que se desean y excluir los no se desean, cortes principalmente el ruido, esta labor se presenta mucho usualmente con los procesamiento de retratos o imágenes, esto es un panorama de acción muy extenso, pero nos centraremos especialmente en 03 tipos de tecnologías utilizadas más continuamente en la etapa del preprocesamiento como las exactas operaciones, eualización, depurado por histograma (Sucar, Visión Computacional, 2017).

1.3.3.1. Operaciones puntuales

En la puntual operación se transfigura un retrato o imagen de ingreso a un retrato o imagen de resultado, la imagen de representación que cada pixel de salida, sólo dependerá de su pixel correspondiente del retrato o la imagen de ingreso, como se enseña en la figura 5 (Sucar, Visión Computacional, 2017).

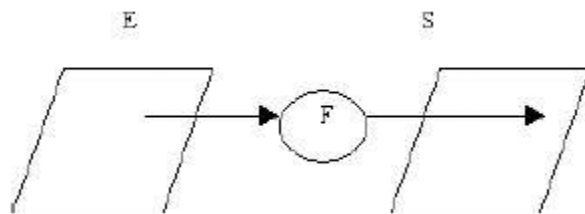


Figura 5: Operación puntual. Fuente: (Sucar, Visión Computacional, 2017).

Un ejercicio puntual se puede expresar matemáticamente como:

$$S[x, y] = f(E[x, y])$$

Detalle de para E es retrato (imagen) del ingreso donde el S es el retrato (imagen) de resultado, para la función f detalla de la identificación de los paralelismos de gris del ingreso a los paralelismos de gris del resultado, la gráfica en que se convierta el retrato o la imagen estar en manos de esta función, debemos entender que la función, puede aclarar descriptivamente como lo enseña en la figura 6, como se muestra el punteo de la línea a cuarentaicinco grados en la imagen muestra cómo se transforma por el que cada pixel del resultado, es necesariamente igual al de identidad o entrada. (Sucar, Visión Computacional, 2017)

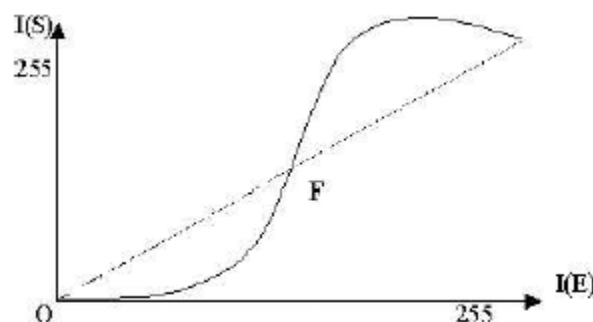


Figura 6: Función de transformación. Fuente: (Sucar, Visión Computacional, 2017).

1.3.4. Binarización por umbral

La labor de Binarización, de forma básica como mínimo, es una propia puntual operación, donde conseguir un retrato binario se crea una metamorfosis o transformación no-lineal del retrato de entrada, para conseguir una imagen o retrato de resultado obtenido en que cada pixel consigue necesariamente uno de los dos valores tomar como 0 y 1, que son el blanco y negro, donde 0 y 255, etc. Hacia esto, se adquiere un valor de umbral T , de forma que: (Sucar, Visión Computacional, 2017)

$$S[x, y] = 1, E[x, y] > T$$

$$S[x, y] = 0, E[x, y] \leq T$$

Debemos tener en cuenta que esta habilidad se puede emplear como natural forma de retirar el objeto de interés de la parte del retrato o imagen que no es usada, como muestra, el objeto de interés asume u obtiene el valor 1 y el resto o los demás 0, el inconveniente que se presenta es así que establecer el umbral, tenemos la muestra, en la figura 7 porque no es viable que se establezca cuál es el valor inmejorable, para apartar los caracteres de la base, una de las formas para determinar instintivamente este valor de umbral es manipulando su histograma por segmentación o histograma de tonos de grises, como se distinguirá más adelante. (Sucar, Visión Computacional, 2017)

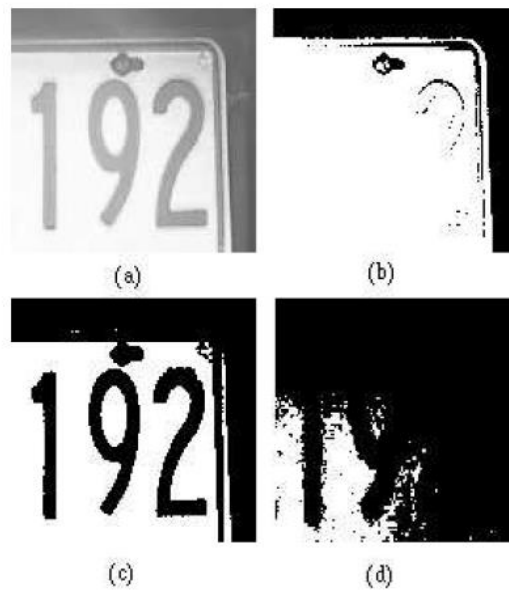


Figura 7: Presenta un modelo de una imagen o retrato que ha logrado binarizar, todos los pixeles con sus productos mínimos, para eso el umbral se muestran en negro con 0, para el caso inverso los pixeles se muestran en blanco 255. Fuente: (Sucar, Visión Computacional, 2017).

1.3.4.1. Transformaciones de intensidad

La evolución de rigor reside en lograr identificar los productos de toda su intensidad para cada uno de los pixeles, tomando para otros productos de acuerdo a innegable la función para la metamorfosis, las funciones para metamorfosis consiguen presentarse de dos tipos como no-lineales o lineales, (Sucar, Visión Computacional, 2017), debemos tener en cuenta que para las metamorfosis lineales, se posee una función o relación rectilíneo de los pixels con los valores de rigor para la imagen de salida en proporción para la imagen original, dentro de las características lineales de metamorfosis más usados frecuentemente son como conseguir el negativo del retrato o de la imagen, disminuir o para aumentar el brillo o la intensidad de la misma, con el ampliación de contraste. (Sucar, Visión Computacional, 2017)

1.3.5. Aumento lineal del contraste

Mediante la utilización del valor de la intensidad máximo y mínimo en un retrato o imagen, se podrá generalmente su contraste aumentar, teniendo como la idea primordial de transportar el valor mínimo siempre a 0 y el máximo a 255, teniendo en cuenta en retratos o imágenes monocromáticas de 0 a 255, la transformación forma que se alejen las intensidades que estén conforme a cierto factor o pendiente, tenemos que el factor o pendiente para esta extensión lineal de discrepancia es como lo indica la formula (Sucar, Visión Computacional, 2017).

$$c(x, y) = \left(\frac{I(x, y) - \min}{\max - \min} * 255 \right)$$

Tenemos que para $I(x, y)$ es el retrato o la imagen para proceder a su proceso y $C(x, y)$ es el retrato o imagen que con ampliación para la lineal del discrepancia, podemos comprobar fácilmente que donde $I(x, y)$ en mínimo, $C(x, y)$ su resultante es cero, también cero es el numerador, donde $I(x, y)$ en máximo, $C(x, y)$ en consecuencia es 255. (Sucar, Visión Computacional, 2017).

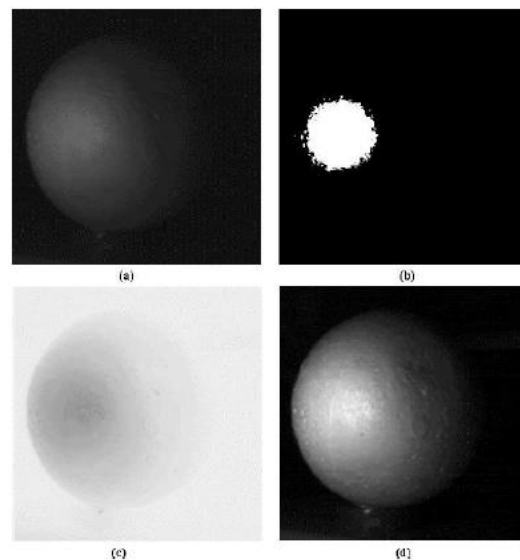


Figura 8: Modelo de operaciones exactas donde la imagen original es (a), la binarización de 80 de umbral es (b), el negativo original de la

imagen es (c), el aumento de contraste lineal es (d). Fuente: (Sucar, Visión Computacional, 2017).

1.3.6. Pre-procesamiento digital de la imagen.

Una de las operaciones necesarias antes de procesar una imagen es eliminar el ruido y las sombras, para ello es necesario aplicar un filtro. La técnica del filtrado busca la modificación o mejora de la imagen mediante un algoritmo que considera el valor de los pixeles vecinos (Esqueda, 2005). Para el logro de esta mejora visual se utilizan los denominados filtros de suavizado los cuales tienen un efecto de desenfoco que desdibuja la imagen y hace que parezca más borrosa perdiendo definición (Olaya, 2014) .



Figura 9. Filtro gaussiano con diferentes radios. Tomado de Las operaciones de la imagen sobre el histograma y *filtrado*. Fuente: (Aranda, 2016) (Edits.), *Métodos y conceptos en visión por computadora*, p. 39.

Uno de estos filtros es el gaussiano, el cual consiste en una máscara que imita a la campana de Gauss, empleado para corregir los desajustes de luces producidas de la cámara por el sensor, en el tipo este de filtrado se da más peso al pixel central, un ejemplo de su implementación se presenta en la Figura 9.

1.3.7. El método otsu en la segmentación de la imagen digital.

Otsu es un método que sirve para segmentar y aislar partes de la imagen, separando el fondo de los elementos que se muestran en primer plano. A través de un procedimiento estadístico y el empleo de histogramas bimodales divide en tramos según el valor del nivel digital de los píxeles por medio del cálculo de los valores mínimos y máximos del umbral, de forma automática, sin requerir otra información que no sea la imagen binarizada (Górriz, 2018).

Esto supone, el pre-procesamiento de la imagen a color ya binarizada y convertida a escala de grises para, desagregar el fondo dejándolo en color blanco y el objeto de interés en color negro. Esta técnica exige que la imagen sea de fondo uniforme, no difuso, sin muchas sombras, sin variaciones de luminosidad que impidan la distinción entre imagen y fondo. De encontrarse estos problemas, será necesario aplicar primero un suavizado y, si aún no se corrige, la imagen no podrá ser procesada por este método sino por los métodos adaptativos nickelback y souvola.

Para su implementación, el algoritmo que lo ejecuta hace uso del histograma bimodal calculando el umbral global que será aplicado a toda la imagen a partir del cociente de la varianza, es decir, de la distancia que hay entre los valores de grises dentro de cada segmento que será el valor del umbral mínimo; y, el valor del umbral máximo que corresponde al resultado de la dispersión entre segmentos diferentes. De esta manera, el histograma será una representación gráfica de ambos segmentos, imagen y fondo, con dos picos claros de color que muestran el valor del umbral óptimo.

Al ser aplicado, se toma un fragmento de la imagen que es de interés y se le aplica el algoritmo, seguidamente, se desplaza a otro segmento repitiendo el proceso hasta completar toda la imagen basándose en la probabilidad de ocurrencia de los píxeles desde el punto cero hasta el umbral. En definitiva, la selección del umbral es clave y puede

seleccionarse en base a las características de la imagen o, partiendo de atributos previos conocidos, del objeto a definir. Para profundizar más, a continuación, se aborda la herramienta histograma.

1.3.8. Operaciones morfológicas para el análisis de la imagen digital.

Durante el pre-procesamiento, la segmentación o la extracción de características puede ocurrir el análisis de la imagen binaria, con niveles de grises o a color; éste se efectúa basándose en las propiedades de la forma, o sea, su morfología, las simplifica y preserva su forma (Cancelas, 2016). Se distinguen dos operaciones básicas: erosión y dilatación; de ellas proceden la apertura que resulta en un suavizado de los contornos y el cierre. Todas ellas se utilizan, según sea el caso, para reducir, aumentar, suavizar, eliminar, rellenar elementos que no forman parte del fondo de la imagen, su salida es una imagen binaria.

1.3.8.1. Dilatación.

La dilatación trata del crecimiento progresivo de los elementos de la imagen que significa, muchas veces, pérdida en los detalles de la forma; además, rellena los espacios que existen entre dos elementos llegando, en ocasiones, a fusionar objetos diferentes que están cercanos (Cancelas, 2016). Este algoritmo une los valores agregando píxeles blancos en los bordes, un ejemplo se observa en la Figura 5, la imagen se engrosa, aumenta por los bordes; en el recorrido pixel a pixel busca el pixel blanco y al encontrarlo realiza la unión con el color blanco. En el caso de una imagen con niveles de grises el resultado es una imagen más clara que la original porque los objetos brillantes crecen y aumenta el brillo, en tanto, los oscuros pierden contraste tendiendo a desaparecer.

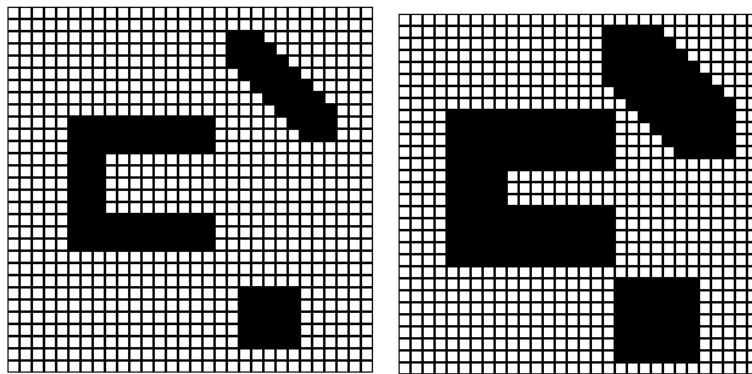


Figura 10. Imagen resultante (derecha) del algoritmo de dilatación a partir de la imagen original (izquierda). Adaptado de “Sistemas de información geográfica”. Fuente: (Oyala, 2014)p. 446-447.

1.3.8.2. Erosión.

La erosión se utiliza para eliminar el ruido digital, es decir, las variaciones indeseadas de brillo o color; también, para quitar los elementos más diminutos dejando solo los de mayor volumen; y, para separar objetos al deshacer los pixeles que los relacionan. Esto se logra al disminuir el tamaño de los elementos de la imagen y agrandar los espacios que existen entre (Cancelas, 2016).

Al ejecutar este algoritmo se eliminan pixeles blancos en los bordes, ver la Figura 6; se difumina y es más oscura. En el caso de una imagen con niveles de grises el resultado es una imagen más oscura que la original porque, a lo inverso de la dilatación, los objetos brillantes disminuyen junto con el brillo estando propensos a disiparse, no así, los oscuros ganan contraste.

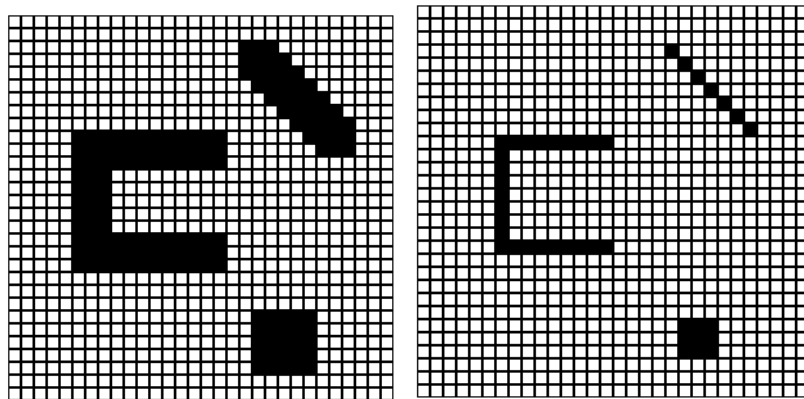


Figura 11. Imagen resultante (derecha) del algoritmo de erosión a partir de la imagen original (izquierda). Adaptado de “Sistemas de información geográfica”. Fuente: (Oyala, 2014), p. 446-447.

1.3.8.3. Apertura y cierre.

El fin de las transformaciones de apertura y cierre es mantener el tamaño del objeto. En primer lugar, la apertura consiste en erosionar y, luego, dilatar la imagen; al erosionar desaparece el ruido, disminuye la imagen; al dilatar la imagen aumenta y los detalles que no fueron quitados al ser erosionados se resaltan (Cancelas, 2016). Este algoritmo tiene como entrada una imagen original que resulta en una imagen dilatada y, finalmente, en una imagen binaria suavizada en sus contornos eliminando los píxeles aislados, ver Figura 12. Si al hacer el recorrido se encuentra un hueco en la parte interna del objeto del tamaño de un píxel lo abre, caso contrario, lo borra.

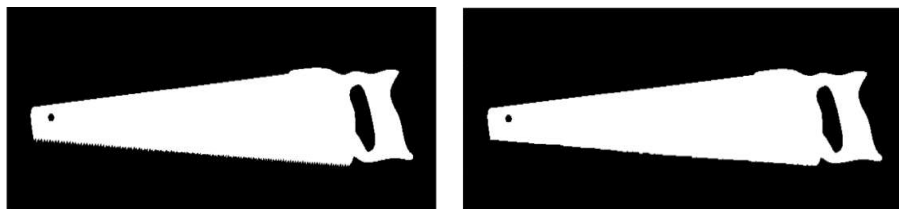


Figura 12. Imagen resultante (derecha) del algoritmo de apertura a partir de la imagen original (izquierda). Tomado de “Procesamiento morfológico”. Fuente: (Cancelas, 2016), Métodos y conceptos en visión por computador, p. 84.

Así mismo, el cierre reside en dilatar y, después, erosionar la imagen; como se explicó al dilatar la imagen crece y al erosionar regresa a un tamaño equivalente al original (Cancelas, 2016). Con este algoritmo se rellenan las hendiduras, ver Figura 13, se eliminan detalles oscuros, también quitan las rugosidades huecas de los bordes.

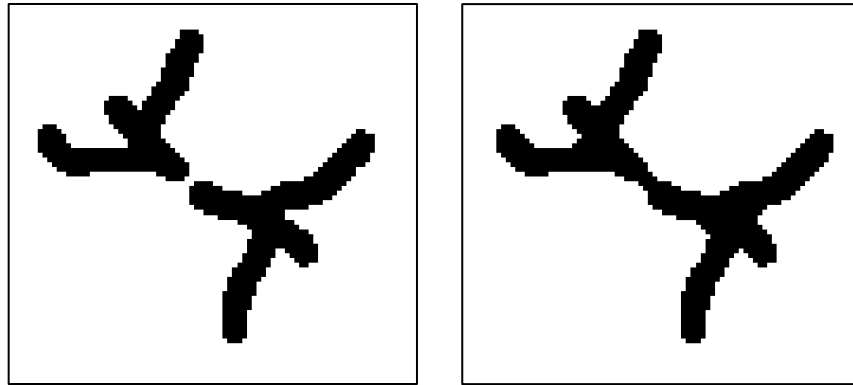


Figura 13. Imagen resultante (derecha) del algoritmo de cierre a partir de la imagen original (izquierda). Tomado de "Sistemas de información geográfica". Fuente: (Oyala, 2014), p. 449.

1.3.8.4. Transformación top-hat.

Después de aplicada la umbralización, cuando el fondo no permite destacar los elementos de menor tamaño se emplea la transformación top-hat para resaltarlos; bien sea, realzando los elementos claros con el top-hat de blancos o acentuando los elementos oscuros con el top-hat de negros (Cancelas, 2016). Con este algoritmo se separan las estructuras oscuras suprimidas por la operación de cierre, contraponiéndola en relación al fondo, ver Figura 14.

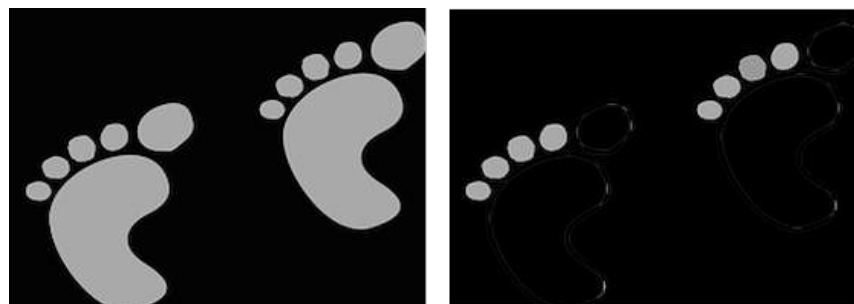


Figura 14. Imagen resultante (derecha) del algoritmo de apertura a partir de la imagen original (izquierda). Tomado de "Procesamiento morfológico". Fuente: (Cancelas, 2016) (García M. A., 2016) Métodos y Conceptos en visión por computador, p. 95.

1.3.9. Detección de orillas

Muchas experimentaciones psicofisiológicas han expuesto que los sistemas de visual de la persona maneja una extensa escala de fuentes de la información, cortes realizados de las longitudes, simetrías, sombras, intensidades, curvaturas y color, de lo anterior, como las diferenciaciones en orillas o intensidad, se refieren entre el aumento significativos, si un retrato o imagen escasea de la información de la textura, sombras o tridimensional, logramos mostrarse conforme el esencia manejando sus orillas o silueta (Sucar, Visión Computacional, 2017).



Figura 15: Conseguimos un perro dálmata reconocer aun que si la imagen escasea de sombras de información, textura o tridimensional. Fuente: (Sucar, Visión Computacional, 2017).

El sistema visual primario procesa la información de orillas, donde se hallan especializadas células que manifiestan todas las discontinuidades (Sucar, Visión Computacional, 2017), la visión de las personas maneja todas las orillas de una diferenciada manera, utilizando y agrupando la práctica visual incluso lograr mostrarse de acuerdo a los objetos más complejos que en las líneas, tenemos como objetos y rostros geométricos (Sucar, Visión Computacional, 2017). El subsistema en biológica de la visión casualmente perfecciona orillas donde están, implícitos u ocluidos, al parecer, los bordes personales de Kanizsa, figura 15, son ejemplo adonde el sistema visual perfecciona los orillas y modifica necesariamente las intensidades, lo expresar completando con

figuras regulares y aparecen más brillantes (Sucar, Visión Computacional, 2017).

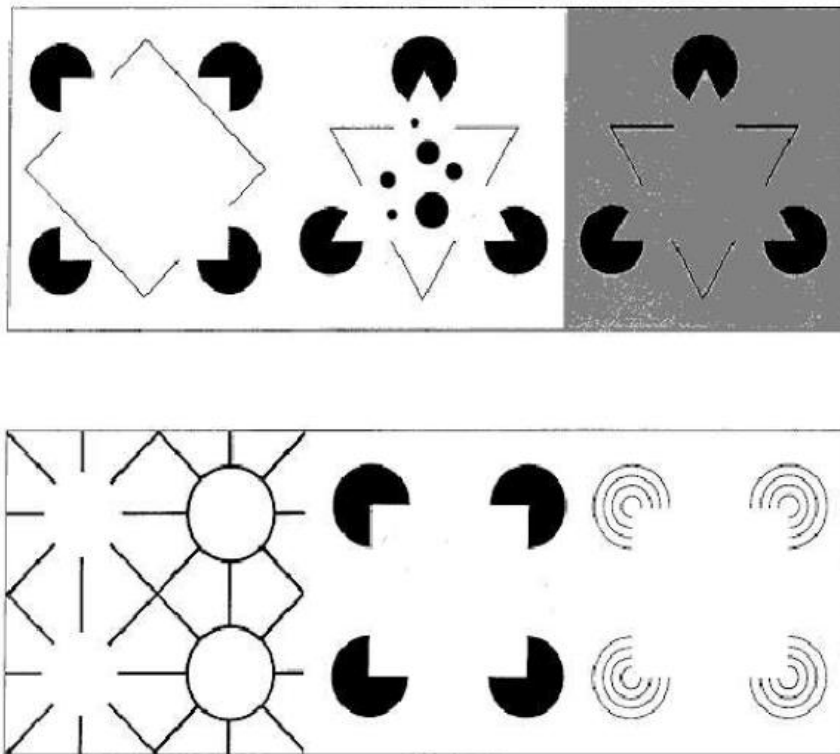


Figura 16: El sistema visual en automático completa, las imágenes añadiendo lo faltantes de las orillas, como en los subjetivos contornos de Kanizsa. *Fuente:* (Sucar, Visión Computacional, 2017).

El modo más habitual para descubrir bordes, es usar el tipo de algún diferencial o derivada, normalmente aplicado en un vecindad pequeña, la derivada nos admite computar entre un punto las variaciones y su vecindario, distinguiendo el retrato o la imagen a manera una función, implica en un alrededor una des continuidad para contenido de la función, es decir en adonde la función posee el valor de derivada o gradiente alta (Sucar, Visión Computacional, 2017).

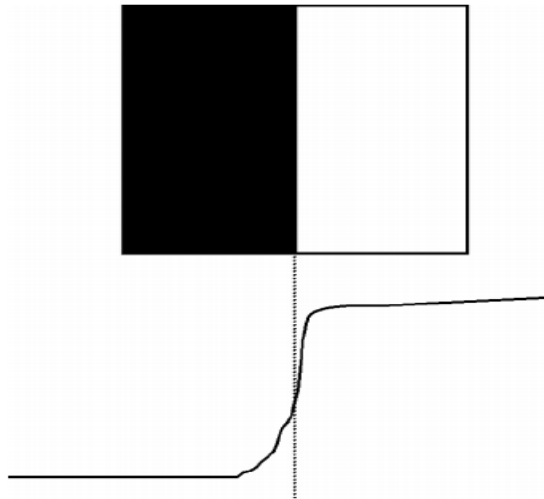


Figura 17: Modelo de discontinuidades, en la imagen de arriba se muestra la discontinuidad de intensidad entre el pedazo izquierda y derecha, en la imagen que está en la parte inferior se grafica toda la intensidad de una incisión horizontal de la gráfica de solo una línea, del que se puede ver el agudo gradiente en su parte conveniente en la discontinuidad. Fuente: (Sucar, *Visión Computacional*, 2017)

1.3.9.1. Laplaciano de una Gaussian

Por la última etapa de los años 70s, David Marr experimentó con la visión de los cuadrúpedos e imaginó una teoría que completaba experimentalmente, todo lo que se conoce sobre biológica de la visión, su revelador de bordes se fundamentó en la subalternas laplaciano o derivadas de una Gaussiana, la función en el Laplaciano de dos variables se define como:

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figura 18: El Sobel de operadores, prestar atención del incluido suavizamiento a los operadores de Prewitt. Fuente: (Sucar, Visión Computacional, 2017)



Figura 19: Localización con los operadores de orillas de Sobel. La original Imagen (a), el absoluto valor del gradiente plano (b), el dominante valor del gradiente vertical (c), La dimensión del gradiente (d), donde se sistematizaron las magnitudes para optimizar el despliegue. fuente: (Sucar, Visión Computacional, 2017).

1.3.10. Segmentación

Al dividir el retrato o imagen en componentes reveladoras es un paso sustancial en visión de una computadora para obtener el objeto reconocido, todo el proceso se reconoce como la segmentación, el modelo de una la imagen de segmentar, se realiza mediante la determinación al incluir a todos los bordes, el doble de esta dificultad, es establecer todas las regiones; como se dice, los segmentos o las pedazos que se logren seleccionar como las unidades reveladoras, esto refuerza a la obtención una traducción más maciza de la información pero de bajo horizonte, llaqué en vez de millones o miles de pixeles, se puede lograr llegar a verdaderamente a decenas de regiones, con las particularidades más habituales para segmentar o delimitar los territorios, son como darle el rigor de los pixeles, color, textura, profundidad y gradiente concerniente. (Sucar, Visión Computacional, 2017).



Figura 20: Modelo de retrato con significativas de las regiones, cada uno de los huevos de desigual color pertenece a solo una región, también se tiene terceras dos regiones que pertenecen al canasto y el fondo. Fuente: (Sucar, Visión Computacional, 2017).

1.3.10.1. Segmentación por histograma

El fraccionamiento por thresholding o histograma, se toma como una habilidad global basada, primeramente en posesionarse, que solo hay un objeto sobre una base o base uniforme, con ese argumento esto se tiene en cuenta en dos zonas de la imagen, luego para fragmentar, luego se considera como soporte del histograma de las intensidades, se puede adjudicarse si hay necesariamente 02 regiones, se debe tener dos cimas en cada histograma, en aquel momento es necesario tomar el valle del mínimo de los dos y se debe considerar la partición necesaria entre las 02 partes, de la grafía para todos los pixeles, que pertenezcan a un costado del histograma, se asume necesariamente una región y la otra parte la zona, como se enseña en siguiente figura 21. (Sucar, Visión Computacional, 2017).

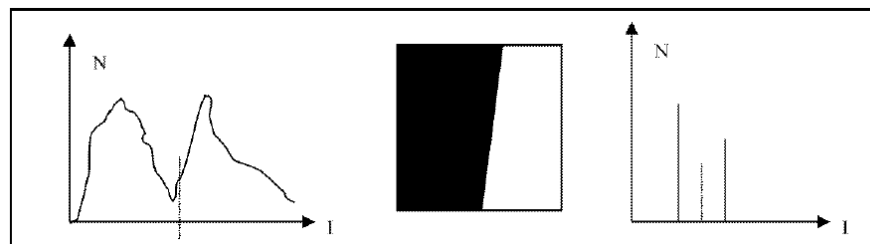


Figura 21: Fraccionamiento que se da por histograma, de la parte de la izquierda se presenta un histograma tópic de la imagen con 02 zonas llamados bimodal, la imagen o figura del centro es una muestra de una ideal imagen, que presenta 02 regiones, para la cual origina el histograma de la imagen del lado derecho, en uno y otro histograma se muestra con la punteada línea del apartamiento del histograma, que pertenece a las 02 zonas de la imagen. Fuente: (Sucar, Visión Computacional, 2017).

1.3.11. Clasificación y reconocimiento de patrones

Los progresos tecnológicos de los últimos tiempos han hecho viable computarizar numerosos trabajos que anticipadamente requerían de un aumento significativo del tiempo repetitivo del trabajo y manual, optimizando la prontitud y comprimiendo traspiés en esas labores

substancialmente mecánicas. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016)

1.3.11.1. Fundamentos del Reconocimiento de Patrones

Para realizar reconocimiento bien hecho de todos los modelos radica fundamentalmente las etiquetas asignar a los objetos señalados debidamente a qué variedad conciernen, dichos objetos al ser personificados por un grupo de características o moderadas, a las que describiremos como propiedades o particularidades, la labor de reconocer involucra obligatoriamente unos procesamientos de amaestramiento, iniciando necesariamente del conjunto de objetos. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

Existen dos magnas tipologías de dificultades de reconocimiento de los patrones como no controlado y controlado, tenemos que en el amaestramiento supervisado se instala los datos en un conjunto adyacente con sus respectivos protocolos, identificando a qué variedad deben pertenecer, por cada ejemplo o dato, el objetivo principal es obtener un modelo archivador en este caso, que realice el marcar de forma automática desconocidos datos, que no se usaron en el arreglo del modelo de la codificación, este procedimiento es acreditado como adiestramiento de la creación o clasificador del tipo, debemos tener en cuenta que si los protocolos no poseen un valor prudente, debido que al tomar valores continuos, entonces, hablaremos una retrocesión, la afirmación óptico de representaciones sería una muestra de codificación controlada, mientras que la evaluación de los años de una persona a partir del fisonomía establecería un dificultad de retrocesión. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

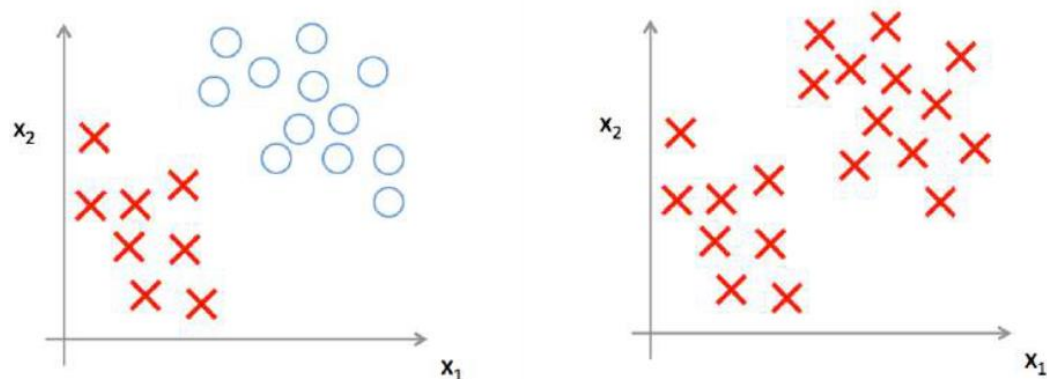


Figura 22: El ejemplo de la controlada clasificación (a) y no controlada (b). Los datos por 02 características se personifican x_1 y x_2 . Fuente: (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

1.3.11.2. Ciclo de diseño de un clasificador

Como primer lugar, los datos del conjunto con que se va a efectuar el amaestramiento debe ser característico de los contextos en ellas que va a manipular el registrador, si todavía no instalamos ese conjunto de datos, es forzoso definir los tipos que logren ser distinguidos para la dificultad y diseñar un ensayo para la toma de medidas que sean necesarias, las restricciones en la toma de datos universalmente comienzan dadas por la cavidad que un propósito sea financiado, la posibilidad de acceso a las muestras, la disposición necesaria de tiempo etc. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

En algunos casos, se ejecutan con medidas, como por muestra las tomadas de imágenes por una dispositivo de video, tienen que ingresar a un proceso para lograr, a partir de estas, tipologías distinguidas para el sistema de reconocer los modelos, para esto se consigue incluir ordenamientos de normalización, depurado, fraccionamiento de representaciones pictóricas u producción de descriptores de las representaciones pictóricas que logren, por ejemplo, que no se transformen, a la escala, a la rotación etc. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

Las características o los descriptores, logrados normalmente no son igual de relevantes en su totalidad, algunos pueden serlo uno por uno en correspondencia con otros, pero otros obtienen ser irrelevantes porque solo forman ruido, para los sistemas de pueden realizar el reconocimiento en materia, partiendo de eso se aplica la técnica de seleccionar las características o elaborar nuevas características a partir de las que se tiene y puede optimizar la calidad de la descripción. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

Cuando se selecciona un tipo de clasificación, para su adiestramiento y la valoración del propio, forma el núcleo céntrico del progreso de un sistema de poder reconocer los patrones, se tiene en la muestra en la imagen siguiente, en la cual se muestra las líneas punteadas, teniendo en cuenta el ciclo para el diseño del clasificador, se puede obstruir en desiguales lugares, se puede decidir usar similar modelo de codificación con desiguales cuantificaciones, utilizar otro modelo de codificación o bien, utilizar desiguales metodologías de extracción y selección de tipologías. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

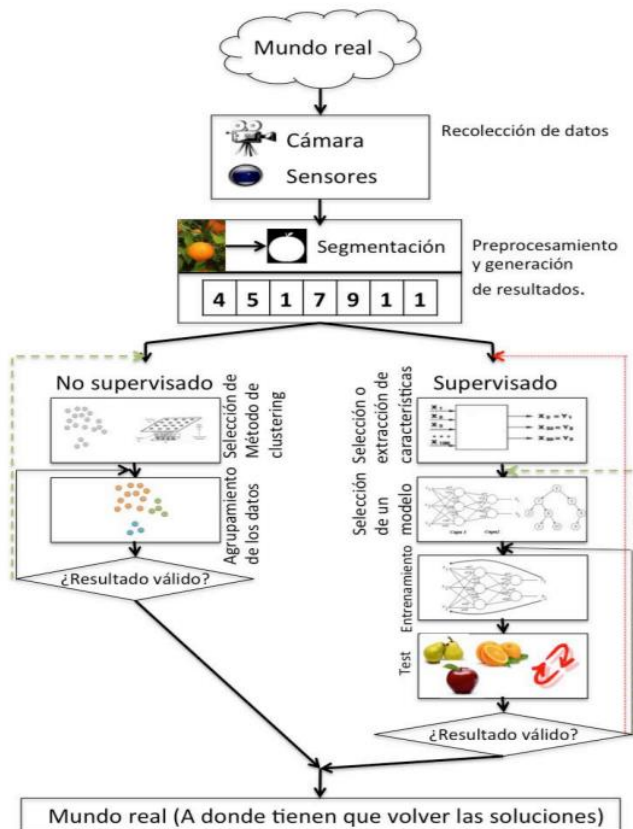


Figura 23: Ejemplo de representación del ciclo de diseño de un clasificador. Fuente: (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

1.3.11.3. Algoritmos de clasificación

En la aplicación del reconocimiento de modelos o patrones se afrontan con dificultades existentes, se han perfeccionado y se prolongan desarrollándose, con una abundancia de algoritmos de amaestramiento, con sus concernientes fortalezas y restricciones, como redes neuronales, los árboles de decisión, aparatos de vectores soporte, las redes bayesianas, entre otros. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

1.3.11.3.1. El algoritmo K-means

Tenemos al K-means nos dice que es un procedimiento de clasificación pero no controlada, muy pública para el estudio de clusters o agrupamientos, la expresión fue usada por originaria vez por MacQueen (1967) pero el algoritmo fue presentado por Lloyd (1957), la meta del

algoritmo se debe establecer para todos y cada uno de los datos para el clusters o grupos, de tal modo que los segmentos de cada grupo o clúster, estén siempre análogos entre sí, en todo lo que a que sus particularidades, que deben estar más diferentes y próximas a los de otro clúster o grupos (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

La forma radica en dividir a N reflexiones en k clusters o grupos, de forma que cada reflexión corresponde al clusters o grupo más colindante, esto nos da lugar a un fraccionamiento del área de datos en celdas de Voronoi. (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

Consideremos para el algoritmo, es el beneficiario el que resuelve a priori la cantidad de clusters o grupos (k), en los que se desee congregarse todos los datos, tenemos a continuación se explican las gestiones a seguir con un ejemplo preciso (la figura 24 incluye una síntesis del algoritmo): (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

Paso 1: Aleatoriamente se inicializan los k ejes, en este caso, $k = 3$, están incorporados en la figura 9.10 en color azul, rojo y verde.

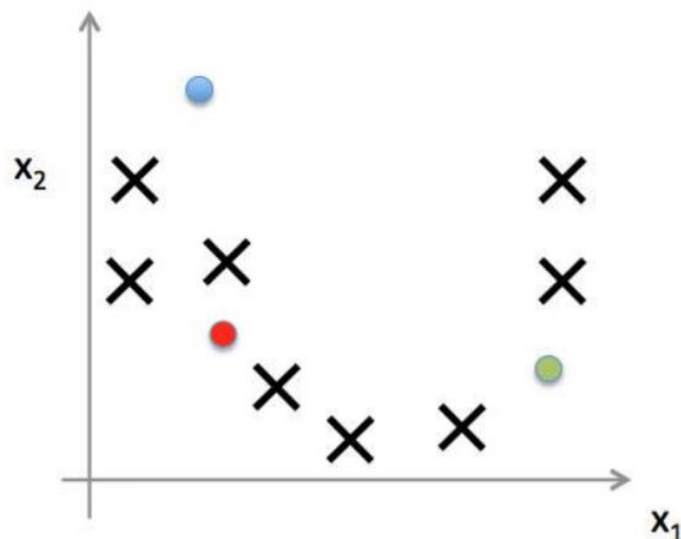


Figura 24: Paso 1: La aleatoria inicialización de los centros. Fuente: (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

- Paso 2: Se determinan los datos a esos ejes en función de los recorridos, figura 25. El color de los fichas simboliza el centro al cual ha sido establecido en función de su cercanía.

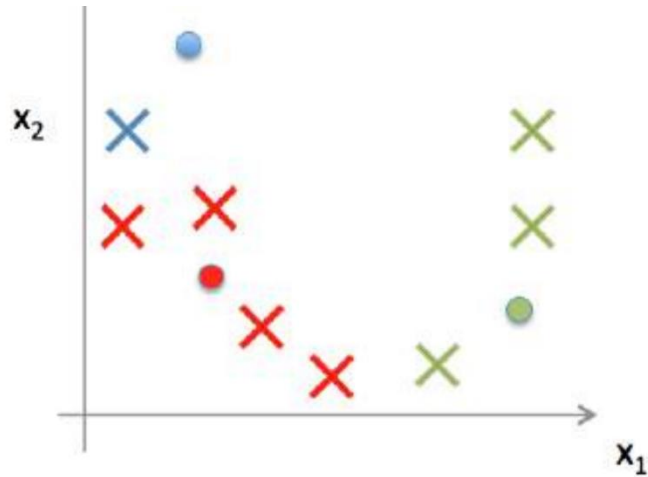


Figura 25: Paso 2: En la asignación de puntos a ejes en función de los recorridos.

Fuente: (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

- Paso 3: Se traslada el centro asumiendo que los puntos que se le han determinado, computarizándolo como la media de todos los elementos establecidos a su grupo, figura 26.

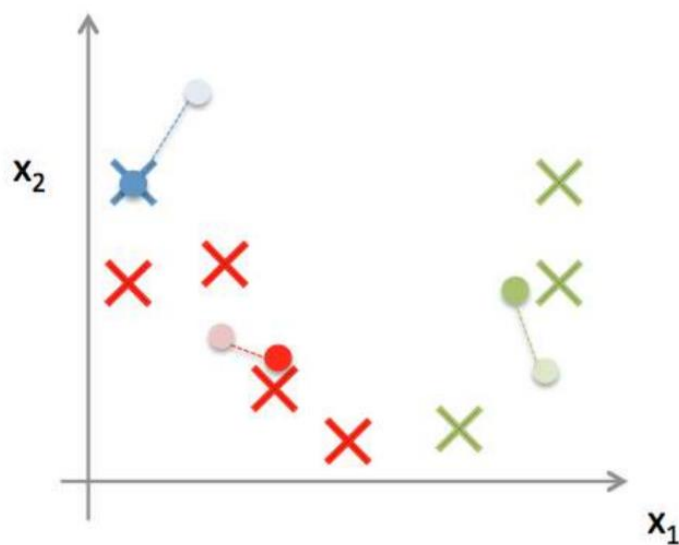


Figura 26 Paso 3: El centro se traslada atravesando a ser el centroide de todos los puntos que había determinados. Fuente: (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

- Paso 4: Se repite la asignación de puntos y la reubicación de los ejes, es decir, los intentos 2 y 3 hasta lograr que los ejes no cambien de lugar en dos insistencias consecutivas.

Entrada: k , conjunto de N datos sin etiquetar: $\{x^1, x^2, \dots, x^N\}$

1. Inicializar aleatoriamente los k centros dentro del espacio representado por los objetos.
2. Asignar cada muestra al centro más cercano.
3. Cuando todas las muestras hayan sido asignadas al centro correspondiente, recalculan las posiciones de los centros de manera que se conviertan en el centroide de las muestras que tienen asignadas dichos centros.
4. Repetir los pasos 2 y 3 hasta que los centros ya no se muevan. Esto produce una separación de los datos en k grupos de manera que muestras más parecidas entre sí pertenezcan al mismo grupo.

Figura 27: Algoritmo K-means. Fuente: (GARCÍA ORDÁS, ALAIZ RODRÍGUEZ, & ALEGRE, 2016).

1.4. **Formulación del Problema.**

¿De qué manera podremos reconocer placas vehiculares para mejorar el acceso aun parqueadero?

1.5. **Justificación e Importancia del estudio.**

En el aspecto tecnológico fue orientado a generar un producto de tal manera que podamos ahorrar tiempo en el control de acceso de vehículos a un parqueadero. En el aspecto económico permitió ahorrar costo en horas hombre (la automatización permitió que la aplicación se encargue íntegramente de reconocimiento). En el aspecto académico permitió contar con un análisis de rendimiento del algoritmo KNN, conocer su complejidad temporal Precisión, exactitud, Recall, . En cuanto a la viabilidad de la presente investigación no afectó en lo económico puesto que aún no se va a realizar la implantación del sistema y los cotos de obtención de las imágenes del dataset serán

imágenes fotográficas reales y también imágenes sintéticas obtenidas de un repositorio de imágenes de placas vehiculares.

1.5.1. Limitaciones de la Investigación

Una de las limitaciones más sensibles para detectar el vehículo entre el día y la noche es la iluminación del entorno; obteniendo imágenes con muy bajo contraste, reflexiones, sombras y detalles que impiden la correcta visualización de las placas; otra limitación considerable es la complejidad del contenido de la placa peruana puesto que contiene muchos elementos distractores y que incrementan la dificultad de obtener los caracteres correctos de la placa.

1.5.2. Delimitación de la Investigación.

El alcance de este proyecto incluirá el desarrollo de un algoritmo de detección de placas vehiculares peruanas en ambiente controlado, para el reconocimiento de placas vehiculares para mejorar el acceso a un parqueadero. en el desarrollo de la investigación se consideró al estudiante como investigador y desarrollador del sistema; todo esto durante el año 2021.

1.6. Hipótesis.

1.6.1. Hipótesis general.

Si se implementa un sistema haciendo uso de visión computacional entonces se podrá reconocer placas vehiculares para mejorar el acceso a un parqueadero.

1.7. Objetivos.

1.7.1. Objetivo general.

Desarrollar un sistema de reconocimiento de placas vehiculares mediante visión computacional para mejorar el acceso a un parqueadero.

1.7.2. Objetivos específicos.

- 1) Realizar el análisis de requerimientos funcionales y no funcionales
- 2) Diseñar la arquitectura del sistema
- 3) Implementar los algoritmos seleccionados
- 4) Evaluar el rendimiento del sistema

II. MATERIAL Y MÉTODO

De acuerdo con el procedimiento técnico desarrollado para realizar esta investigación fue obligatorio exponer la tipología y diseño que permitieron la adecuada estructuración y recogida de datos favorecedores para los resultados del mismo. Esta se presenta como aplicada y tecnológica según su propósito mediante el procesamiento de imágenes digitales.

2. 1. Tipo y Diseño de Investigación.

2.1.1. Tipo de estudio.

Mi trabajo de investigación corresponde al enfoque cuantitativa, de acuerdo a su naturaleza, el estudio corresponde a una tipología aplicada, y según sus objetivos tecnológica, ya que por la naturaleza que se deriva del conocimiento científico apoyando por la ciencia computacional. Asimismo, es importante mencionar que los métodos estudiados anteriormente y llevados a la práctica, permitieron medir las cualidades con las que se trabaja con valores como exactitud, precisión, recall.

2.1.2. Diseño de estudio.

En la presente investigación realizada se utilizó el experimento como exploración para recabar datos y verificar la hipótesis basada en la estadística, la matemática, la lógica es por ello que la presente investigación corresponde al diseño experimental, del sub tipo cuasi experimental.

2. 1. Población y Muestra

2.2.1. Población.

Se considero la población que está constituida por 510 imágenes de placas vehiculares que de acuerdo a las investigaciones realizadas por los autores. (Satsangi, Yadav, & Sewak, 2019, pág. 5)

2.2.2. Muestra.

La muestra es el mismo el tamaño de mi población y está constituido por 510 imágenes de placas vehiculares, según los autores esta es la cantidad de imágenes adecuadas para la investigación. (Satsangi, Yadav, & Sewak, 2019, pág. 5)

2.3. Variables, Operacionalización.

2.3.1. Variables

2.3.1.1. Variable independiente.

Sistema de visión computacional

2.3.1.2. Variable dependiente.

Reconocimiento de placas vehiculares.

Matriz de operacionalización de las variables de la investigación

Tabla 1: Matriz de operacionalización de las variables de la investigación.

Variable	Dimensión	Indicador	Formula	Técnicas e instrumentos de recolección de datos
Variable independiente : Sistema de Visión Computacional	Tiempo	Tiempo de respuesta	$Tr = \sum_j^n \frac{tf_j - tf_i}{n}$	Técnica de Observación Ficha de Observación
	Consumo	Consumo de memoria RAM	$C_{RAM} = \frac{RAM_T \times 100}{RAM_u}$	
		Consumo de CPU	$C_{MCPU} = \frac{MCPU_T \times 100}{GPU_U}$	
Variable dependiente: Reconocimiento de placas vehiculares	Reconocimiento	Exactitud	$E = \frac{V}{V + F}$	Técnica de Observación Ficha de Observación
Reconocimiento de placas		Recall o sensibilidad	$\text{Recall} = \frac{VP}{FN + VP}$	
vehiculares		Precisión	$P = \frac{VP}{VP + FP}$	

Fuente: Elaboración propia

2.4. Técnicas e Instrumentos de Recolección de Datos, Validez y confiabilidad

En mi investigación del presente estudio se utilizaron los siguientes métodos: El análisis, la Observación, el Resumen, experimental.

Análisis: en este proceso realizamos los procedimientos para someter la información recabada para alcanzar los objetivos de estudio.

Observación: Como ciencia nos permitirán describir y explicar el comportamiento para obtener datos fiables en el presente estudio. estas particularidades en la presente investigación tanto como investigador, asesores y jurado calificador.

Resumen: Una vez sintetizado el problema propuesto y hemos seleccionado los métodos de visión computacional a implementar, se realizará el desarrollo de la solución con los métodos elegidos.

Experimental: Este estudio se realizó sobre una situación real, de un problema social y de ingeniería, orientado a implementar los métodos y técnicas de visión por computador, teniendo como base primordial la elaboración y contrastación de la hipótesis.

2.4.1. Técnicas:

a) Observación: Esta técnica se define en forma más normalizada como: La razón para el registro verificable y visual de lo que se intenta conocer; es decir, es atraer de la modo más objetivo posible, lo que sucede en el mundo existente, ya sea para analizarlo o explicarlo, describirlo, desde un aspecto científico; a opuesto de lo que ocurre en el mundo empírico, en el cual la persona en común utiliza el dato o la información observada de manera práctica para satisfacer o resolver problemas de sus necesidades (Campos y Emma, 2013,)

b) Documentación: se realizará análisis bibliográficos de distintas bases de datos con material bibliográfico (libros, artículos científicos, revistas, etc)

2.4.2. Instrumento de recolección de datos

a) Ficha de observación

Mediante la técnica y su instrumento de investigación podremos recolectar y evaluar los datos, con respecto a un objetivo específico, para poder determinar las variables específicas y se usan para registrar los datos a fin de obtener resultados para la mejora o análisis correspondiente.

2.5. Procedimiento de análisis de datos

a) **Análisis documental:** en el presente estudio usamos el análisis fundamentado para la recolección de los datos el cual se realizó en los siguientes pasos.

a) Adquisición de imágenes formada por 510 según el tamaño de muestras, tomadas en el año 2021. En esta base de datos encontramos diversas posiciones de los vehículos tomadas de forma frontal y posterior. Por último, cabe mencionar que todas las fotos fueron tomadas bajo condiciones de iluminación y parámetros para una obtención nítida de las placas en las imágenes.

b) Realizar revisiones bibliográficas de investigaciones anteriores para determinar las técnicas de visión computacional utilizadas en el reconocimiento de placas vehiculares.

c) Investigar sobre métodos de visión computacional para poder cumplir con todos objetivos propuestos en la investigación del presente estudio.

- d) Observación al haberse implementado las técnicas de reconocimiento de placas vehiculares, el siguiente paso será la valorar el rendimiento de las técnicas implementadas.

2.5.1. Análisis Estadístico e Interpretación de los datos

Se procesará los datos estadísticos utilizando Microsoft Excel 2016, para su análisis e interpretación de los resultados precisados en los indicadores según la matriz de operacionalización de las variables de la investigación. En los que se podrá detallar los datos obtenidos de los resultados en promedio de la suma total, la varianza y cuadros estadísticos.

b) Indicadores para la variable dependiente

Tiempo de respuesta: Tiempo que se demora en ejecutar el algoritmo.

Fórmula:

$$TR = TI - TF$$

Donde:

TI = tiempo inicio

TF = tiempo final

Grado de consumo de memoria RAM: al ejecutar el algoritmo.

Fórmula:

$$\text{Utilización de Memoria} = C_{RAM} = \frac{RAM_T \times 100}{RAM_u}$$

Donde:

C_{RAM} es Consumo de RAM

RAM_u es RAM utilizada

RAM_T es RAM total

Grado de consumo de CPU: Es el tiempo al ejecutar un proceso.

Fórmula:

$$C_{MCPU} = \frac{MCPU_T \times 100}{GPU_U}$$

Donde:

C_{MCPU} es Consumo de memoria de CPU

$MCPU_U$ es memoria de CPU utilizada

$MCPU_T$ es memoria de CPU total

c) Indicadores para la variable independiente

Exactitud: Indica la aproximación de los resultados del cálculo con relación al valor verdadero.

Formula:

$$E = \frac{V}{V + F}$$

Donde:

V = Total de Positivos Verdaderos

V = Total de Positivos Falsos

F=Total de Negativos Falsos

Precisión: Mide el grado de confiabilidad, es decir dar el mismo resultado en cálculos desiguales con similar situación.

Formula:

$$\text{Precisión} = \frac{VP}{VP + FP}$$

Donde:

VP = Total de Verdaderos Positivos

VFP = Total de Falsos Positivos

RECALL: Indica la cantidad de positivos verdaderos que identificaron correctamente. También conocido como sensibilidad.

Formula:

$$\text{Recall} = \frac{VP}{FN + VP}$$

Donde:

VP = Total de Verdaderos Positivos

FN = Total de Falsos Negativos

2.6. Criterios Éticos

Para el progreso de esta investigación se respetó la identidad de los dueños de los vehículos que formaron parte del mismo, así como su identificación y detalles inherentes a su procedencia. De igual manera la información recaba es fidedigna, recopilada mediante técnicas de investigación específicas que permiten la autenticada de la misma y el respeto a la propiedad de autor en el caso de las fuentes documentales usadas.

2.7. Criterios de Rigor Científico

a) Fiabilidad:

Del proyecto de investigación cumplió con las perspectivas mencionadas en todo su contenido y su ejecución con el modelo y políticas para su buen desarrollo.

b) Validez:

Los antecedentes conseguidos por la tesis o proyecto fueron analizados y evaluados, por los profesionales especialistas en el tema para determinar su autenticidad.

c) Consistencia:

el trabajo de la investigación representó material sólido y certificado por la comunidad científica.

III. RESULTADOS

En el capítulo 3 del presente estudio se explica los resultados del sistema propuesto, así como la interpretación de los mismos, Se utilizaron un total de 198 imágenes de automóviles organizadas en dataset las mismas que han sido tomadas en diferentes horas del día y de la noche, y en condiciones climáticas distintas las cuales nos permitirán medir los indicadores.

A continuación, se presentan los resultados logrados por la visión computacional en sus dos etapas de reconocimiento y pruebas. Se requieren medir diversos indicadores, como son el consumo del CPU y consumo de la memoria RAM que se calculan por medio de la herramienta administrador de procesos de Windows; con respecto a la medición del tiempo de respuesta, Precisión, Recall, Exactitud se usó la librería Python Sklearn. Se entreno el modelo mediante el algoritmo KNN con imágenes representativas que contienen vehículos con sus correspondientes placas.

3.1. Resultados en tablas y figuras.

3.1.1. Tiempo de respuesta

El tiempo medio del procesamiento de la aplicación para el reconocimiento de un vehículo autorizado teniendo en cuenta su placa vehicular es de 0.48 s, es decir se obtiene un rendimiento de hasta 29 imágenes procesadas en 14 segundos.

3.1.2. Consumo de CPU

las medidas de consumo de CPU permitieron conocer cuál es el consumo en porcentaje que utiliza el algoritmo; este indicador es importante porque se desea aplicar la investigación en un ambiente real donde se requerirá un buen rendimiento del sistema. Para esto se realizó tres mediciones de consumo de CPU por la ejecución del algoritmo como se observa en la tabla.

Tabla 2.

Porcentaje de consumo de CPU

Sistema de Reconocimiento	Porcentaje de consumo de CPU
Medición 1	20%
Medición 2	21%
Medición 3	20%

Fuente: Elaboración propia

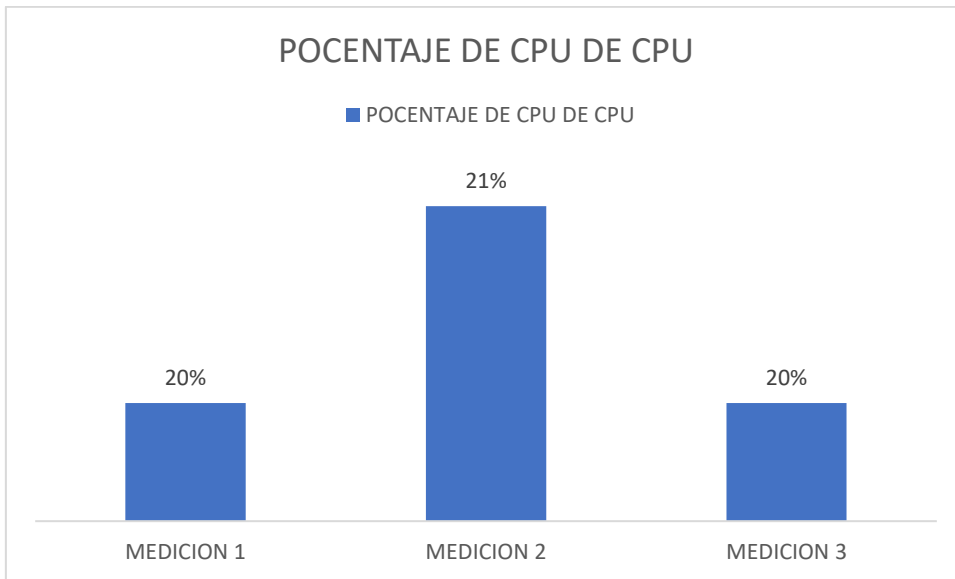


Figura 28: Porcentaje de consumo de CPU. Fuente: Elaboración propia

En la figura 28 se aprecia los resultados obtenidos en cada medición de Porcentaje de CPU. En donde se aprecia un consumo mínimo del 20% y un promedio del 20.33%.

3.1.3. Consumo de Memoria RAM

las medidas de consumo de Memoria RAM permitieron conocer cuál es el consumo en porcentaje que utiliza el algoritmo. Para esto se realizó tres mediciones de consumo de Memoria RAM por la ejecución del algoritmo como se observa en la tabla.

Tabla 3.

Porcentaje de consumo de memoria RAM.

Sistema de Reconocimiento	Porcentaje de Consumo de memoria RAM
Medición 1	20%
Medición 2	21%
Medición 3	20%

Fuente: Elaboración propia

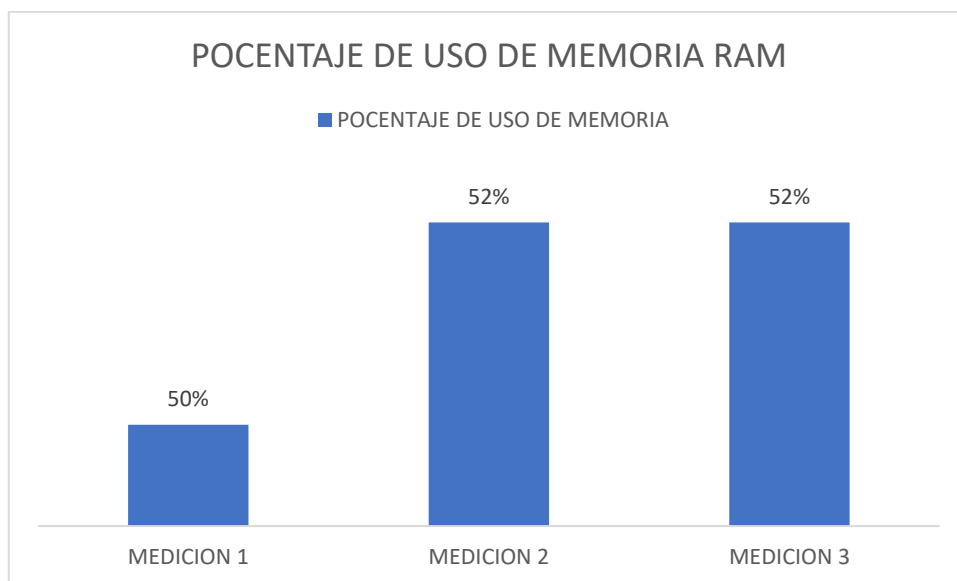


Figura 29: Porcentaje de consumo de memoria RAM.

En la figura 29 se aprecia los resultados obtenidos en cada medición de Porcentaje de consumo de memoria RAM. En donde se aprecia un consumo mínimo del 50% y un promedio del 51.33%. Fuente: Elaboración propia

3.1.4. Exactitud

El uso de un sistema de reconocimiento de caracteres de una placa requiere de calcular la exactitud porque ésta permite medir el porcentaje de positivos que el modelo de esta investigación ha acertado en el reconocimiento total de una placa vehicular.

Para la medición de la exactitud se realizó una única medición como se puede apreciar en la siguiente tabla.

Tabla 4.

Exactitud del algoritmo kNN.

Sistema de Reconocimiento	Exactitud
Algoritmo kNN	36.4%

Fuente: Elaboración propia

En la figura 30 podemos apreciar que en la medición realizada al sistema obtenemos un 36.4% de exactitud debido a que se usó una muestra aleatoria de imágenes que no contenían elementos favorables para que los caracteres fueran reconocidos acertadamente en su totalidad.

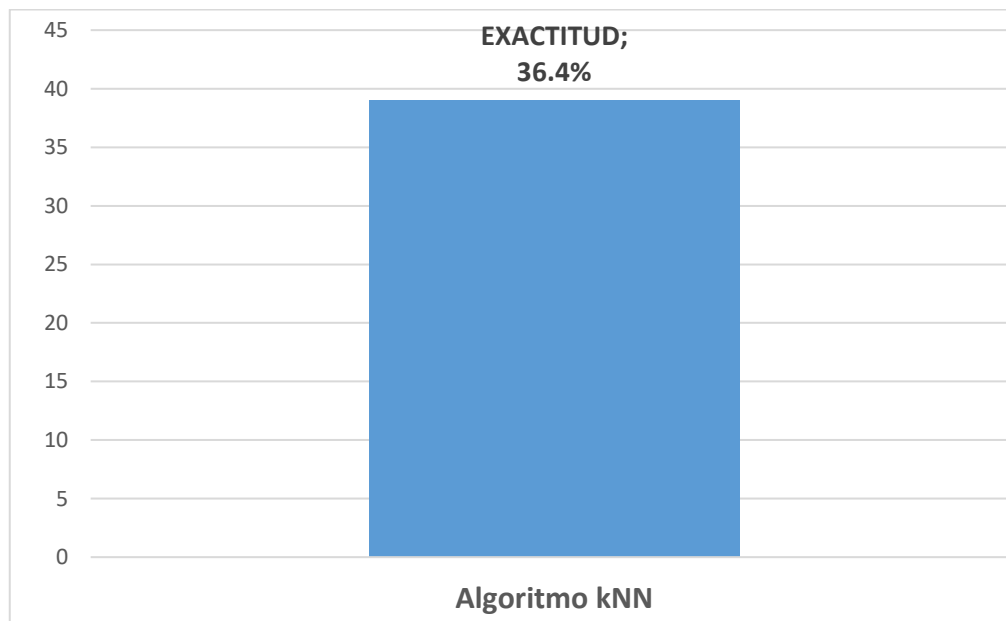


Figura 30: Exactitud del algoritmo kNN. Fuente: Elaboración propia

3.1.5. Precisión

En el grafico 4 podemos apreciar que en la medición realizada al sistema obtenemos un 89.9% de precisión en reconocer números y 80.5% en reconocer letras debido a que se obtuvieron buenos resultados por el algoritmo KNN y el preprocesamiento de imágenes adecuado; por tal razón el algoritmo fallará en aquellos casos en donde los caracteres de la placa vehicular estén en mal estado como caracteres mal pintados, en condiciones deterioradas o posición inadecuada.

Tabla 5.

Precisión del algoritmo kNN.

Sistema de Reconocimiento	Precisión en caracteres numéricos	Precisión caracteres del alfabeto
Algoritmo kNN	89.9	80.5

Fuente: Elaboración propia

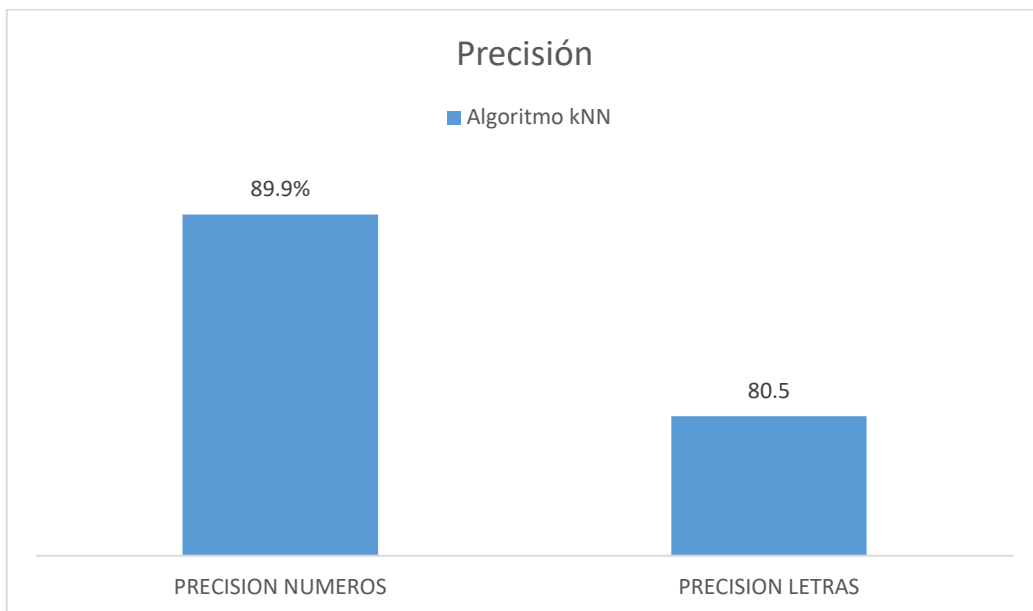


Figura 31: Precisión del algoritmo kNN. Fuente: Elaboración propia

3.1.6. Recall

En el grafico 5 podemos apreciar que en la medición realizada al sistema obtenemos un 46.9% de recall, que no brindo información sobre la cantidad de aciertos que el modelo fue capaz de identificar. Este indicador nos permite una mejor valoración en la exactitud.

Tabla 6.

Recall del algoritmo kNN

Sistema de Reconocimiento	Recall
Algoritmo kNN	46.9

Fuente: Elaboración propia

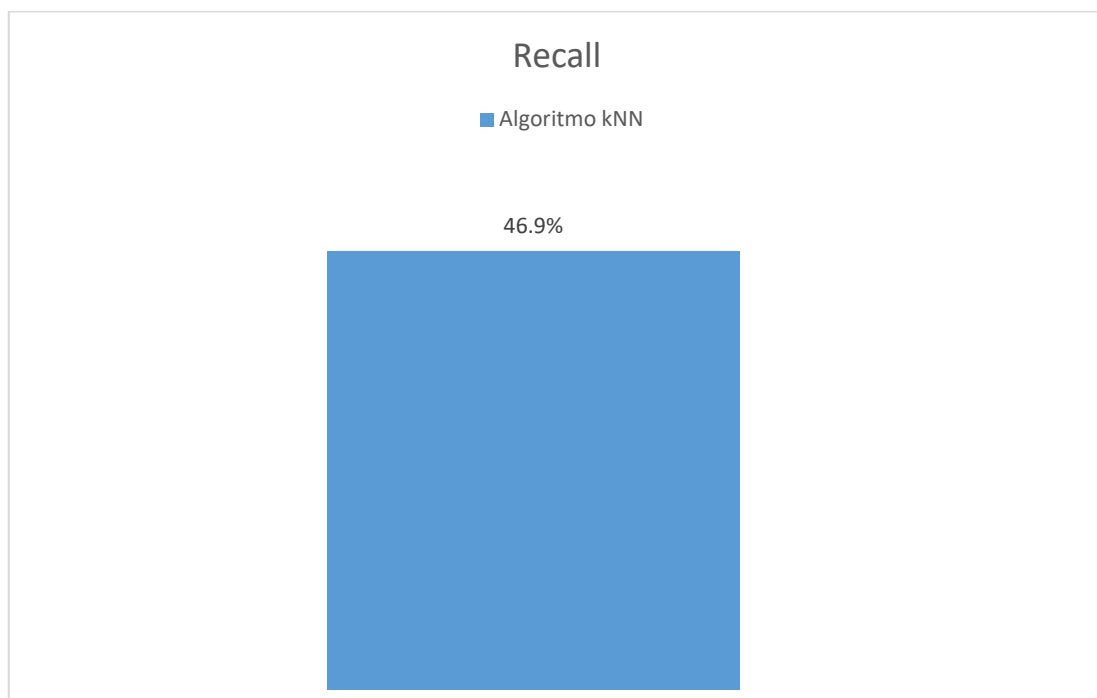


Figura 32: Recall del algoritmo kNN. Fuente: Elaboración propia

3.2. Discusión de resultados.

La presente investigación se partió de la revisión bibliográfica de diferentes estudios, internacionales y nacionales a fin de ejecutar los objetivos que se plantearon.

3.2.1. Discusión numero 1:

(Quiros, y otros, 2017), En su investigación: Un enfoque basado en kNN para la visión artificial del reconocimiento de caracteres de los números de matrícula, en donde proponen la automatización de los procesos del reconocimiento utilizando una cámara IP para la captura de la imagen. Los contornos de los caracteres en una imagen de placa determinada se detectan, violan y aíslan de la imagen principal. Esto resulta en caracteres segmentados. Para la identificación de los caracteres se utilizó un algoritmo de vecinos más cercanos (kNN). El algoritmo kNN fue entrenado utilizando conjuntos de datos que contenían 36 caracteres cada uno. Las pruebas al algoritmo se realizaron con caracteres previamente segmentados. Se logró una precisión del 87.43%. El tiempo medio de procesamiento fue de 0.034 s. En la presente investigación con el algoritmo kNN implementado se obtuvo una precisión de 89.9% para caracteres numéricos y un 80.5% para los caracteres del alfabeto.

3.2.2. Discusión numero 2:

(Zhang, Yu, Su, & Li, 2019), en su estudio Diseño de sistema de reconocimiento de matrículas basado en aprendizaje automático, se desarrolló en la plataforma que consta de Windows 10, Visual C++ y la librería OpenCV3, la interfaz fue diseñada en QT, el entrenamiento y reconocimiento se basó en Tensorflow y Python3.6, para evaluar la efectividad del sistema se recopiló imágenes reales de placas vehiculares las mismas que fueron normalizadas a un tamaño de 32*32 para realizar pruebas. El experimento de desarrollo de la siguiente manera.

1. Se experimentó el reconocimiento de una sola placa de vehículo en condiciones normales con el objetivo de probar la precisión del

reconocimiento del sistema se probaron 172 imágenes de placas vehiculares con una precisión de 100%. Las principales razones fueron: las imágenes de placas vehiculares tienen alta resolución, y el ángulo, la distancia y la iluminación eran ideales.

2. Se experimento el reconocimiento de una sola placa de vehículo en condiciones difíciles con la meta de probar el beneficio del sistema. Las condiciones probadas en este documento incluyen principalmente: imágenes de gran ángulo, día lluvioso niebla, etc. Se probaron 21 imágenes en condiciones no favorable encontrando 2 errores de reconocimiento que conformaría una tasa de 90.5%.

Para el presente estudio se diseñó el sistema de reconocimiento de matrículas basado en aprendizaje automático, se desarrolló en la plataforma que consta de Windows 10, Python 3.9 y la librería OpenCV, la interfaz fue diseñada usando la librería Tkinter, el entrenamiento y reconocimiento se basó en el desarrollo de un algoritmo kNN, para evaluar la efectividad del sistema se recopiló imágenes reales de placas vehiculares las mismas que fueron normalizadas, En la presente investigación con el algoritmo kNN implementado se probaron 198 imágenes, obteniendo una precisión de 89.9% para caracteres numéricos y un 80.5% para los caracteres del alfabeto.

3.3. Aporte Práctico

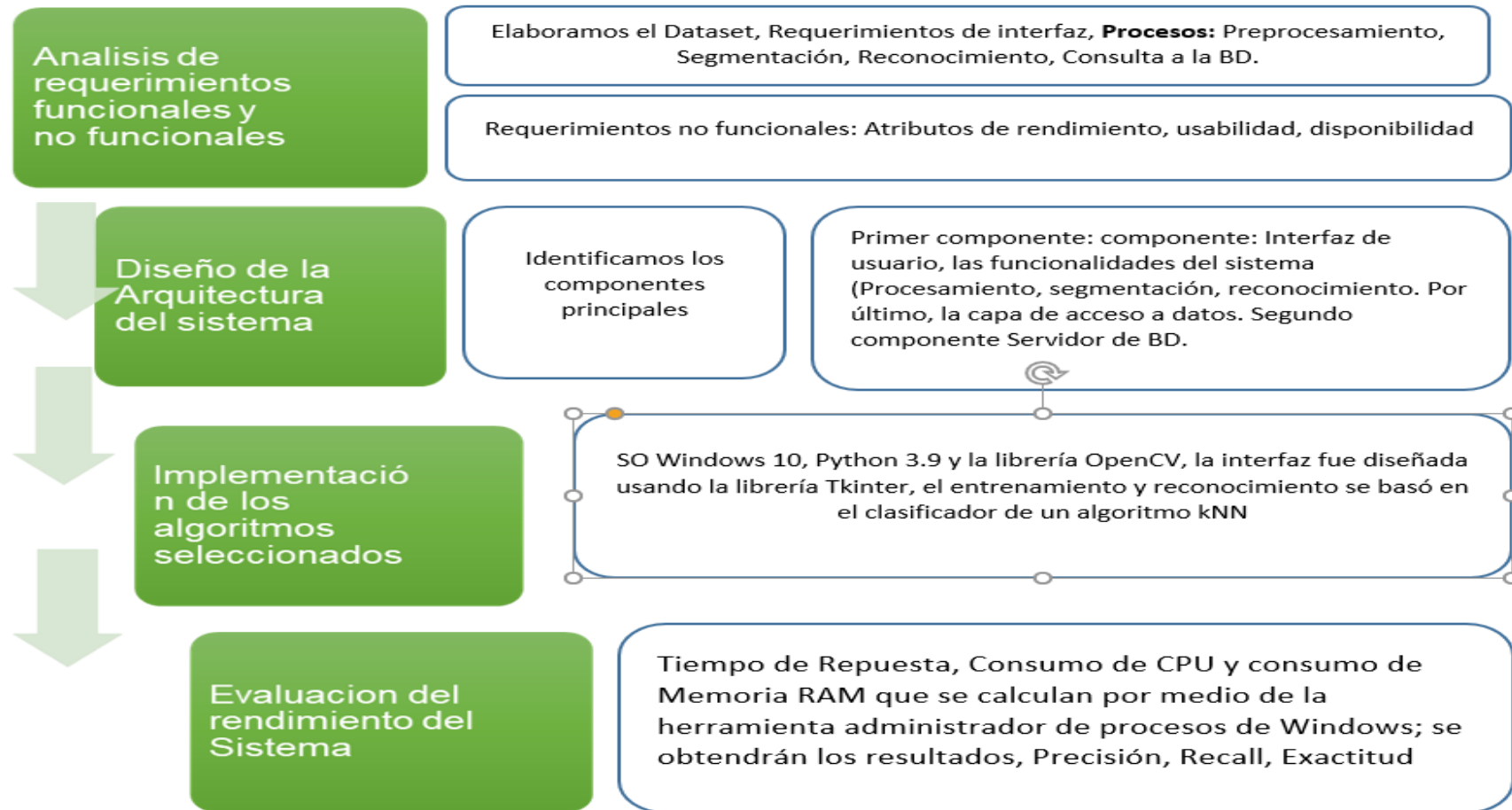


Figura 32: Diagrama de objetivos. Fuente: Elaboración propia



Figura 33: En la presente figura se describir cómo los elementos planteados interactúan y apoyan el algoritmo de visión artificial implementado, la entrada es una imagen capturada, la primera fase es el preprocesamiento donde se condiciona la imagen para el buen funcionamiento del proceso, luego sigue la fase de segmentación donde los caracteres de la matricula son separados, se prosigue a la fase de Reconocimiento Óptico de Caracteres con la finalidad de reconocer y convertir a código alfanumérico los caracteres de la matricula, los mismos que serán contrastados con una BD SQL y determinar si el vehículo tiene acceso permitido al parqueadero.

3.3.1. Requerimientos funcionales y no funcionales

En este apartado se presenta el documento cuya intención es definir los requerimientos funcionales y no funcionales del sistema de reconocimiento de placas vehiculares, para la implementación del sistema informático el cual será un sistema que funcionará en un ambiente de escritorio que permitirá controlar el acceso a un parqueadero.

3.3.1.1. Personal involucrado

Tabla 7.

Personal involucrado en el desarrollo del sistema

Nombre	David Enrique Bances Saavedra
Rol	Asesor
Categoría	Ingeniero Magister
Profesional	
Responsabilidades	Brindar asesoramiento en el desarrollo del proyecto

Fuente: elaboración propia

Tabla 8.

Personal involucrado en el desarrollo del sistema

Nombre	Edwin Wildor Pérez Silva
Rol	Desarrollador
Categoría	Estudiante
Profesional	
Responsabilidades	Analizar, e implementar y evaluar el Sistema

Fuente: elaboración propia

3.3.1.2. Requisitos de la interfaz

- Interfaz de Usuario:** La interfaz con la que el usuario interactuará contiene botones y etiquetas que permita el uso adecuado, que fue construida únicamente para el sistema propuesto.

b) **Interfaz de Hardware:** Para el adecuado funcionamiento del software se necesitan equipos que cumplan con las siguientes características:

- Procesador de 3.90 GHz o superior
- Memoria RAM de 8 GB
- Mouse
- Teclado

c) **Interfaz de Software:** El software será ejecutado en sistema operativo Windows.

3.3.1.3. *Requerimientos específicos de software*

Tabla 9.

Captura de imagen

Numero de requisito	RF-001
Nombre de requisito	Captura
Prioridad del requisito	Alta
Descripción	El sistema permitirá capturar imagen desde una cámara

Fuente: elaboración propia

Tabla 10:

Preprocesamiento de la imagen

Numero de requisito	RF-002
Nombre de requisito	Preprocesamiento de la imagen
Prioridad del requisito	Alta
Descripción	El sistema mediante algoritmos de visión computacional permitirá preparar la imagen para su posterior análisis

Fuente: elaboración propia

Tabla 11.

Localización de la placa del vehículo

Numero de requisito	RF-003
Nombre de requisito	Localización de la placa del vehículo
Prioridad del requisito	Alta

Descripción	El sistema mediante los algoritmos implementados deberá localizar la placa dentro de la imagen previamente capturada
--------------------	--

Fuente: elaboración propia

Tabla 12.

Segmentación de caracteres

Numero de requisito	RF-004
Nombre de requisito	Segmentación de caracteres.
Prioridad del requisito	Alta
Descripción	El sistema mediante los algoritmos implementados deberá segmentar los caracteres que se encuentre dentro de la placa del vehicular.

Fuente: elaboración propia

Tabla 13.

Reconocimiento de caracteres de la placa vehicular

Numero de requisito	RF-005
Nombre de requisito	Segmentación de caracteres.
Prioridad del requisito	Alta
Descripción	El sistema deberá mostrar los caracteres detectados en la placa vehicular

Fuente: elaboración propia

3.3.1.4. Requerimientos no funcionales

a) RNF – 001 Requerimiento de rendimiento del sistema.

La capacidad de respuesta a peticiones será de manera eficiente

b) RNF – 002 Usabilidad

- El sistema contará con una interfaz gráfica de usuario fácil de manejar.
- Podrá ser usado desde un computador.

c) RNF – 003 Disponibilidad

La disponibilidad del sistema será las 24 horas del día

3.3.2. Diseño de la Arquitectura del Sistema

Considerando los requerimientos funcionales y no funcionales identificados y detallados en el punto anterior se ha diseñado la arquitectura del sistema de software que se ha desarrollado en la presenta investigación. En la figura 10 se muestra la vista lógica de la arquitectura y en la tabla 2 se explica cada componente que la conforma, su responsabilidad y propiedades.

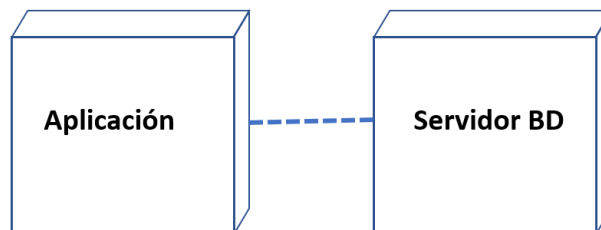


Figura 34:

Vista lógica de los compontes de la arquitectura del sistema desarrollado.

Fuente: elaboración propia

Tabla 14.
Componentes de la arquitectura del sistema.

Elemento	Responsabilidad	Propiedades
Aplicación	Engloba los componentes de coordinación que se encargan de llevar a cabo la ejecución del sistema para la autorización de vehículos mediante la identificación de su placa.	- Lenguaje de Programación: Python - Librería: OpenCv
Servidor de Base de datos	Sistema de gestión de base de datos para almacenar los datos de los vehículos en una base de datos Relacional.	- Sistema de Gestión de Base de Datos: MySQL

Fuente: Elaboración propia

En la figura 11 se muestra la vista física de la arquitectura y en la tabla 3 se explica cada componente que la conforma, su responsabilidad y propiedades.

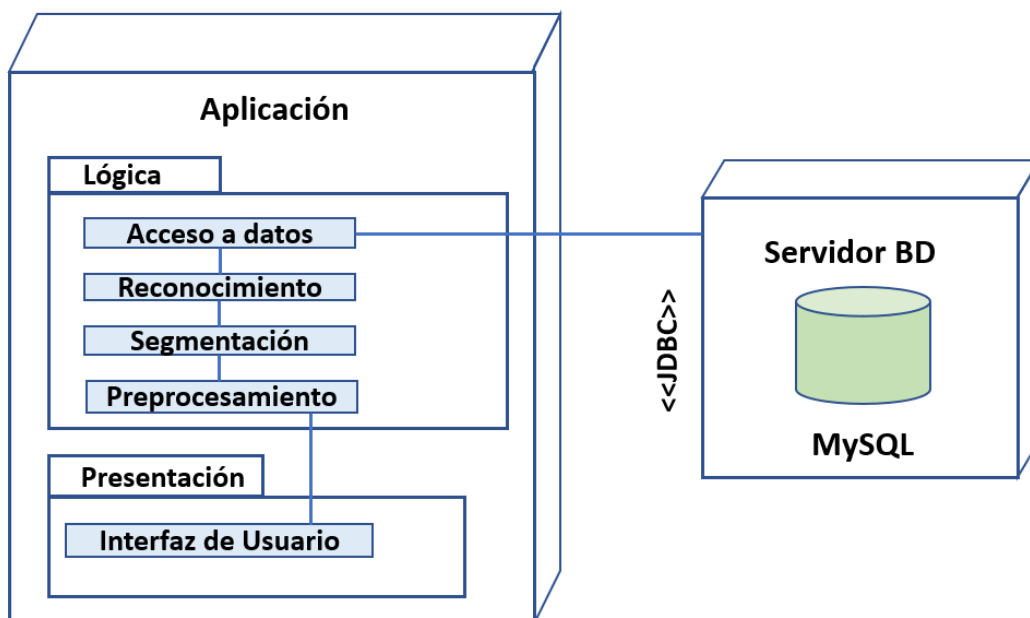


Figura 35. Vista física de los componentes de la arquitectura del sistema desarrollado. Fuente: elaboración propia

Tabla 15.

Elementos que componen arquitectura del sistema

Elemento	Responsabilidad	Propiedades
Interfaz de usuario	Formulario que presenta la opción de poder seleccionar la imagen del vehículo a procesar, en este mismo formulario se muestra el resultado de reconocimiento.	- Lenguaje de Programación: Python - Librería: Tcl/Tk
Preprocesamiento	El algoritmo desarrollado basado en OpenCv realiza primero la conversión de la imagen de BGR a HSV, después se maximiza el contraste y por último se realiza la umbralización, esta imagen resultante será utilizada en la segmentación	- Lenguaje de Programación: Python - Librería: OpenCv
Segmentación	El algoritmo permite la segmentación de la región de interés en este caso de la placa identificada en la imagen del vehículo, esta imagen segmentada será usada en la etapa de reconocimiento.	- Lenguaje de Programación: Python - Librería: OpenCv
Reconocimiento	El algoritmo identifica los caracteres que conforman la placa y compare ese valor con los registros de vehículos autorizados, de esta manera se determina si el vehículo es autorizado o no.	- Lenguaje de Programación: Python - Librería: OpenCv

Acceso a datos	El algoritmo permite la conexión a la base de datos MySQL mediante funciones de consulta utilizando la librería JDBC.	- Lenguaje de Programación: Python - Librería: JDBC
----------------	---	---

Base de datos	La base de datos donde se almacena los registros de vehículos esta implementada en MySQL	- Sistema de Gestión de Base de Datos: MySQL
---------------	--	---

Fuente: Elaboración propia

3.3.3. Implementación de los algoritmos seleccionados en lenguaje de programación.

El desarrollo de esta investigación se llevó a cabo mediante las etapas comprendidas en visión artificial utilizada para poder realizar el reconocimiento de placas vehiculares. Se realizó la recopilación de la información, para ello fue necesario conocer como están estructuradas y que organismos rigen la elaboración de las placas vehiculares peruanas. El Ministerio de Transportes y comunicación (MTC) presenta un documento, Reglamento del Sistema de placa única nacional de rodaje [17], la placa única de rodaje debe cumplir con la norma ISO 7591, DIN 74069 y/o ASTM E-810.



Fuente: Reglamento del Sistema de placa única nacional de rodaje: Pág. 18

Figura 36: Distribución del contenido de la placa de rodaje,

3.3.3.1. Adquisición De Imágenes

Luego del análisis de la estructura de las placas peruanas, se procedió a construir un dataset para poder medir el rendimiento del sistema. Para ello se han obtenido imágenes de placas de vehiculares, de las cuales se conformó dos grupos. El primer grupo son imágenes reales, las mismas han sido recolectadas en diferentes horas del día y bajo condiciones climáticas distintas, tenemos a continuación que se detallara el proceso para la obtención de imágenes, las cuales fueron tomadas desde un celular inteligente con resolución de cámara de 13 mega pixeles. El procedimiento de

captura para la recolección de las imágenes se realizó de la siguiente manera: Se selecciona un vehículo y enfocar su parte frontal o posterior; Ubicarse a una distancia entre 1m o hasta 2m, y enfocar la placa del vehículo en un ángulo descendente no mayor de 45 grados.

El segundo grupo de imágenes de placas sintéticas se han conseguido mediante las herramientas gratuitas de la nube en los sitios Web *platesmania.com*. los dos grupos de imágenes han sido almacenadas en un directorio de nuestro ordenador.

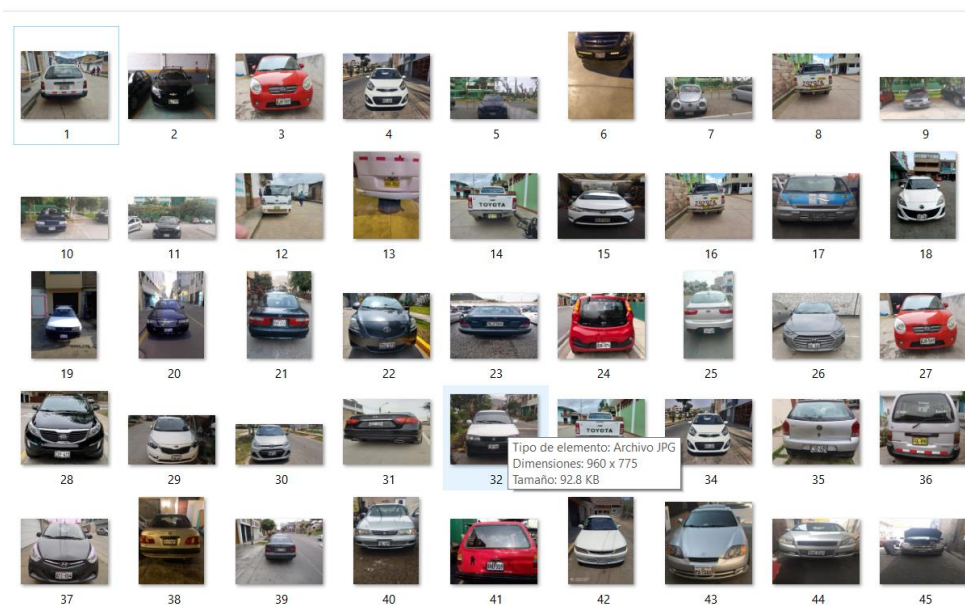


Figura 37: Dataset implementado con imágenes de placas vehiculares.

Fuente: elaboración propia

3.3.3.2. Preprocesamiento

Conversión de imagen de BGR a HSV

Una vez adquirida la imagen realizamos la conversión de formato en la imagen original para aprovechar la información que nos provee el canal de HSV (MATIZ, SATURACION Y VALOR) nos permitirá contar con más información de la imagen como para tener como resultado una mejor segmentación y contorno.

```
def extraerValores(imgOriginal):
```

```
    alto, ancho, numChannels = imgOriginal.shape
```

```
    #obtener altura, ancho y numero de canales de la imagen
```

```

imgHSV = np.zeros((alto, ancho, 3), np.uint8)
#vector de ceros
imgHSV = cv2.cvtColor(imgOriginal, cv2.COLOR_BGR2HSV)
#obtenemos imagen en canal HSV
imgHue, imgSaturation, imgValue = cv2.split(imgHSV)
# campos del canal HSV
return imgValue

```



Figura 38: Conversión de imagen a formato HSV. Fuente: elaboración propia

Maximizar contraste

Se trabajó con la imagen necesariamente a escala de grises y se realizó una maximización de contraste para poder resaltar tonos y contornos en la imagen, para ello llamamos los métodos de OpenCV: `cv2.getStructuringElement`, `cv2.morphologyEx`, `cv2.subtract`. Estas funciones se basan en transformaciones morfológicas, normalmente son aplicadas en imágenes binarizadas.

```
def maximizarContraste(imgGrayscale):
```

```
#maximizamos contraste para resaltar y destacar los tonos de la imagen en escala de gris
```

```
    alto, ancho = imgGrayscale.shape
```

```
    imgTopHat = np.zeros((alto, ancho, 1), np.uint8)
```

```

imgBlackHat = np.zeros((alto, ancho, 1), np.uint8)

structuringElement = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))

imgTopHat = cv2.morphologyEx(imgGrayscale, cv2.MORPH_TOPHAT,
structuringElement)

imgBlackHat = cv2.morphologyEx(imgGrayscale, cv2.MORPH_BLACKHAT,
structuringElement)

imgGrayscalePlusTopHat = cv2.add(imgGrayscale, imgTopHat)

imgGrayscalePlusTopHatMinusBlackHat = cv2.subtract(imgGrayscalePlusTopHat,
imgBlackHat)

return imgGrayscalePlusTopHatMinusBlackHat

```



Figura 39: Maximizar contraste. Fuente: elaboración propia

Desenfocado Gaussiano

Se aplico la técnica desenfocada Gaussiano para ayudar a los tonos de los pixeles estén cercanos uno al otro, lo que provoque que la imagen resalte bordes y detalles, para ello se utilizó el método de OpenCV: cv2.GaussianBlur. El cual nos ayudará a reducir el ruido de la imagen

```

imgBlurred = np.zeros((alto, ancho, 1), np.uint8)

imgBlurred = cv2.GaussianBlur(imgMaxContrastGrayscale,
TAMAÑO_FILTRO_SUAVE_GAUSIANO, 0)

#filtro gaussiano

#convertir a blanco y negro por umbralización especial

```



Figura 40: Desenfocado Gaussiano. Fuente: elaboración propia

Umbralización

Es un tipo de Binarización, en donde la imagen está en escala de grises obtenida hasta ahora se convierte en una imagen en blanco y negro, con la finalidad de poder hacer más fácil y eficiente el procesamiento de los píxeles, para ello llamamos al método de OpenCv: `cv2.adaptiveThreshold`. que es un umbral adaptativo que funciona bien para condiciones variables como iluminación no uniforme, la meta es convertir una imagen en escala de grises a una nueva, con sólo 02 niveles, de modo que los objetos permanezcan apartados del fondo. De lo contrario, la tarea de segmentación fallará.

```
imgThresh = cv2.adaptiveThreshold(imgBlurred, 255.0, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,  
cv2.THRESH_BINARY_INV,
```

```
ADAPTIVE_THRESH_TAMAÑO_BLOQUE, ADAPTIVE_THRESH_PESO)
```

```
return imgGrayscale, imgThresh
```



Figura 41: Imagen binarizada por umbralización. Fuente: elaboración propia

3.3.3.3. Segmentación

Terminada la etapa de preprocesamiento, se continuó con la etapa de segmentación, la cual tiene como objetivo separar la imagen principal en regiones significativas, donde se realizaron los procesos de segmentación de placa y segmentación de caracteres.

Segmentación de placa de la imagen

Este proceso es de vital importancia. Este paso tiene como objetivo identificando la matrícula y separarla de la imagen.

El proceso es el siguiente: Se buscan todas las áreas de interés que codician con la forma valida de una placa, de las cuales se selecciona aquella que cumpla lo más cercano a los requerimientos de la placa, (forma, caracteres, dimensión). Se usó la función “detectarPlacasenEescena” para que devolviera

una lista de todas las posibles regiones de interés, así como sus propiedades como área, cuadro delimitador, etc. Se usó la función "listOfPossibleCharsInScene" esta función primero encuentra todos los contornos, luego solo incluye los contornos que podrían ser caracteres.

```
def detectarPlacasEnEscena(imgOriginalScene):
    listaDePosiblesPlacas = []
    height, width, numChannels = imgOriginalScene.shape
    imgGrayscaleScene = np.zeros((height, width, 1), np.uint8)
    imgThreshScene = np.zeros((height, width, 1), np.uint8)
    imgContornos = np.zeros((height, width, 3), np.uint8)
    imgGrayscaleScene, imgThreshScene = Preprocesamiento.preproceso(imgOriginalScene)
    # preproceso para obtener imágenes en escala de grises y umbral
    listOfPossibleCharsInScene = VerificarPosiblesCaracteresEnEscena(imgThreshScene)
    listOfListsOfMatchingCharsInScene =
    DeteccionCaracteres.findListOfListsOfMatchingChars(listOfPossibleCharsInScene)
    for ListaDeMatchingChars in listOfListsOfMatchingCharsInScene:
        # para cada grupo de caracteres coincidentes
        possiblePlate = ExtraerPlaca(imgOriginalScene, ListaDeMatchingChars)
        # intento de extraer la placa
        if possiblePlate.imgPlate is not None:
            # si se encontró la placa
            listaDePosiblesPlacas.append(possiblePlate)
        # agregar a la lista de posibles placas
    print("\n" + str(len(listaDePosiblesPlacas)) + " posibles placas encontradas")
    return listaDePosiblesPlacas
```



Figura 42: Segmentación de la placa de la imagen. Fuente: elaboración propia

Segmentación de caracteres

El objetivo separar los personajes en la placa, lo cual es importante para reconocer a los personajes correctamente. Se obtuvo las regiones de interés de cada carácter teniendo en cuenta principalmente que sean caracteres válidos (se contrastaron con aquellos caracteres que han sido entrenados con el algoritmo KNN y que se encuentran ordenados de izquierda a derecha) y que estén a una distancia dentro del rango adecuado (que tenga un mínimo de 8 píxeles de ancho por alto). Se usó la función de OpenCv: "listOfListsOfMatchingCharsInPlate" para reconocer todos los personajes de la placa del vehículo.

```
def detectarCaracteresEnPlacas(listaDePosiblesPlacas):  
    intPlateCounter = 0  
    imgContours = None  
    contornosLista = []  
    if len(listaDePosiblesPlacas) == 0:  
        # si la lista de posibles placas está vacía  
        return listaDePosiblesPlacas  
    for posiblePlacai in listaDePosiblesPlacas:  
        # para cada placa posible  
        posiblePlacai.imgGrayscale, posiblePlacai.imgThresh =  
        Preprocesamiento.preproceso(posiblePlacai.imgPlate)  
        # preprocesar para obtener imágenes en escala de grises y umbral  
        posiblePlacai.imgThresh = cv2.resize(posiblePlacai.imgThresh, (0, 0), fx = 1.6, fy = 1.6)
```

```

thresholdValue, posiblePlacai.imgThresh = cv2.threshold(posiblePlacai.imgThresh, 0.0,
255.0, cv2.THRESH_BINARY | cv2.THRESH_OTSU)

listOfPossibleCharsInPlate =
encontrarPosiblesCaracteresEnPlaca(posiblePlacai.imgGrayscale,
posiblePlacai.imgThresh)

listOfListsOfMatchingCharsInPlate =
findListOfListsOfMatchingChars(listOfPossibleCharsInPlate)

    for i in range(0, len(listOfListsOfMatchingCharsInPlate)):

        # recorremos cada lista de caracteres coincidentes

        listOfListsOfMatchingCharsInPlate[i].sort(key = lambda matchingChar:
matchingChar.intCenterX)

        # ordenar caracteres de izquierda a derecha

        listOfListsOfMatchingCharsInPlate[i] =
eliminarCaracteresSuperpuestosInternos(listOfListsOfMatchingCharsInPlate[i])
        # y eliminar los caracteres superpuestos internos

        # dentro de cada placa posible, suponga que la lista más larga de caracteres
coincidentes es la lista real de caracteres

        intLenOfLongestListOfChars = 0

        intIndexOfLongestListOfChars = 0

        for i in range(0, len(listOfListsOfMatchingCharsInPlate)):

            if len(listOfListsOfMatchingCharsInPlate[i]) > intLenOfLongestListOfChars:

                intLenOfLongestListOfChars = len(listOfListsOfMatchingCharsInPlate[i])

                intIndexOfLongestListOfChars = i

            longestListOfMatchingCharsInPlate =
listOfListsOfMatchingCharsInPlate[intIndexOfLongestListOfChars]

            posiblePlacai.strChars = reconocerCaracteresEnPlaca(posiblePlacai.imgThresh,
longestListOfMatchingCharsInPlate)

return listaDePosiblesPlacas

```



Figura 43: Segmentación de los caracteres de la placa vehicular.

Fuente: elaboración propia

3.3.3.4. Reconocimiento e interpretación

Para el proceso de reconocimiento de caracteres realiza la detección de los caracteres individuales dentro de la placa previamente detectada, los ordena y clasifica en un carácter alfanumérico. En consecuencia, el resultado del proceso es una cadena de caracteres.

El proceso inicia con la segmentación de la imagen de la placa en caracteres individuales. La siguiente imagen muestra los caracteres individuales detectados y separados por cuadros delimitadores.

Cada personaje fue sometido a un algoritmo KNN que clasifica al personaje. La clasificación por KNN se tuvieron un conjunto de datos con imágenes de caracteres de la A-Z (5 tipos diferentes con 26 caracteres) y del 0-9 (5 tipos diferentes con 10 caracteres) con dimensiones 20 x 30 píxeles. A continuación, se muestra las plantillas de fuentes. Se generó un archivo en imagen y un archivo de clasificación "clasificador" que contiene la matriz de píxeles de cada carácter en la plantilla de fuente y el código ASCII que corresponde a los caracteres asignados después del entrenamiento, a los cuales se accedió durante la parte de las pruebas.

0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Figura 44. Plantilla de fuente de muestra. Fuente: elaboración propia



Figura 45: Reconocimiento e interpretación de caracteres con kNN. Fuente: elaboración propia

Finalizada cada una de las etapas correspondiente al procesamiento de imágenes, y con el resultado (caracteres de la placa vehicular) obtenidos específicamente en la última etapa, se procedió a comparar dichos resultados contra una base de datos, que está directamente conectada con la aplicación, la cual nos permitirá verificar si dicha placa es aceptada y considerada segura o es rechazada y negado su acceso.

3.3.3.5. Implementación de la interfaz de usuario

Culminadas las etapas correspondientes al procesamiento digital de imágenes, se procedió a realizar una interfaz de usuario, que permitió la interacción entre el modelo de reconocimiento y el usuario final.

3.3.4. Evaluar el rendimiento del sistema

Para evaluar el rendimiento del sistema se han realizado pruebas en un equipo con las siguientes características:

Procesador: Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz 2.59 GHz

Memoria RAM: 8.00 GB (7.83 GB utilizable)

Sistema Operativo: Sistema operativo de 64 bits, procesador x64

El desarrollo de este objetivo nos permite evaluar el sistema, para ello se ha elaborado un dataset conformado por una muestra de 198 imágenes de placas vehiculares previamente registradas en la base de datos con sus placas reales. A continuación, explicaremos el procesamiento para calcular las métricas más importantes de la matriz de confusión, es decir Exactitud, Recall y Precisión.

Para calcular la exactitud se analizan aquellas placas que el software nos indica los mismos caracteres y números sin errores que su correspondiente valor de placa en la base de datos, se tiene en cuenta la muestra de 198 imágenes. Su valor de exactitud se calcula después de recorrer la muestra antes mencionada mediante una división tal cual como se indicó en su fórmula previamente.

Para calcular la precisión, en primer lugar, se hace una revisión de distancias de la placa obtenida por el software contra la lista total de placas que están registradas en la base de datos devolviendo aquella que se encuentre más cercana a su distancia. A continuación, se procede a recorrer un a uno la placa similar encontrada contra la placa calculada por el software buscando los verdaderos positivos en números y en letras, así como también los falsos positivos en números y en letras. Finalmente, el valor de precisión se calcula después de recorrer la

muestra aplicando su fórmula correspondiente tanto para la clase números y la clase letras

Para calcular el Recall tomamos en cuenta todas las placas sin errores en caracteres o números teniendo en cuenta los valores en placa en base de datos. El valor de recall se calcula posterior a la exactitud tal como se aprecia en la fórmula descrita anteriormente

3.3.4.1. Exactitud:

La exactitud nos da una visión general de los aciertos que ha tenido el sistema, es decir en donde los caracteres de la placa determinada por el sistema son exactamente igual a los caracteres de la placa real.

$$E = \frac{77}{198} = 36.4.0\%$$

3.3.4.2. Precisión:

Nos permite conocer la exactitud en cada clase que determina el sistema (caracteres numéricos o caracteres del alfabeto).

Precisión en caracteres numéricos

$$Precisión = \frac{608}{608 + 87} = 89.9\%$$

Precisión en caracteres del alfabeto

$$Precisión = \frac{359}{359 + 87} = 80.5\%$$

3.3.4.3. Recall:

Nos permite medir la sensibilidad de los resultados del sistema, de esta manera tener una visión más clara de la exactitud.

$$\text{Recall} = \frac{77}{77 + 87} = 46.9\%$$

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones:

- a) Para el diseño del sistema de reconocimiento de placas vehiculares se ha considerado como requerimientos funcionales los procesos de captura de la imagen de los vehículos, preprocesamiento de la imagen, localización de la placa, segmentación y reconocimiento de los caracteres que conforman la placa del vehículo para determinar mediante una consulta a la base de datos de vehículos autorizados si este tiene acceso autorizado al parqueadero. Los requerimientos no funcionales para el diseño del sistema de reconocimientos de placas vehiculares fueron agrupados en los atributos de rendimiento, usabilidad y disponibilidad.
- b) Considerando los requerimientos funcionales y no funcionales identificados se ha diseñado la arquitectura de software conformado por dos componentes principales, el primer componente comprende la capa de interfaz de usuario, la capa lógica de la aplicación que engloba las funcionalidades de preprocesamiento, segmentación y reconocimiento de las placas vehiculares y por último la capa de acceso a datos que permite la comunicación con el servidor de base de datos; el segundo componente de la arquitectura es el servidor de base de base de datos donde se almacenan todos los registros de los vehículos autorizados.
- c) Para obtener mayor información de la imagen vehicular se ha realizado un preprocesamiento, inicialmente se ha convertido la imagen del modelo RGB a HSV, luego se ha maximizado el contraste para resaltar tonos y contornos, a la imagen resultante se la ha suavizado utilizando el desenfoque gaussiano, y por último se ha convertido la imagen resultante a escala de grises para poder optimizar los cálculos posteriores. En la etapa de segmentación de la imagen se ha implementado la funcionalidad para para identificar la placa vehicular en la imagen ingresada, realizando además la segmentación de cada carácter que conforma la placa. En la etapa de reconocimiento los caracteres segmentados son contrastados con la matriz

de caracteres aplicando el algoritmo KNN para la clasificación de cada carácter segmentado.

- d) En las pruebas realizadas en un entorno supervisado el sistema obtiene una precisión de 89.9% para el reconocimiento de caracteres numéricos y 80.5% para el reconocimiento de caracteres alfabéticos, determinando que los algoritmos empleados en el sistema de reconocimiento alcanzan un nivel aceptable.
- e) El tiempo medio del procesamiento de la aplicación para el reconocimiento de un vehículo autorizado teniendo en cuenta su placa vehicular es de 0.034 s, es decir se obtiene un rendimiento de hasta 29 imágenes procesadas por segundo. Las funcionalidades del sistema de reconocimiento de placas vehiculares se han desarrollado en el lenguaje de programación Python utilizando la librería de visión por computador OpenCV, se ha seleccionado estas tecnologías para permitir que el sistema de reconocimiento de placas vehiculares funcione en los sistemas operativos Windows, Linux y Mac OS.

5.1. Recomendaciones.

- a) Se recomienda trabajar con imágenes de alta calidad para mejorar el porcentaje de aciertos en el reconocimiento de caracteres. Se recomienda mejoras en el desarrollo de entrenamiento del algoritmo para lograr un porcentaje del 100% de precisión.
- b) Es importante utilizar otros algoritmos y de esta manera lograr mejoras en las investigaciones futuras de sistemas de visión computacional porque nos permiten atomizar los procesos de visión, las mismas que son muy beneficioso en el control de acceso de los vehículos.
- c) Si se necesita obtener un mejor tiempo de respuesta para el reconocimiento de placas vehiculares se recomienda utilizar un equipo de cómputo con mejores características que las empleadas en esta investigación.
- d) Para la recolección de las imágenes es importante tener en cuenta el ángulo, la iluminación y la distancia ya que son factores de mucha importancia para el análisis de cada placa y la precisión final del algoritmo de reconocimiento.
- e) Dado que el sistema de tráfico inteligente (ITS) es un medio importante para viajes más seguros es recomendable que en trabajos futuros implementar sistemas que permitan generar la información necesaria para identificar vehículos que cuenten con restricciones de circulación en las vías.

Referencias

- Agbeyangi, b. O., Alashiri, O. A., & Otunuga, A. E. (2020). Automatic Identification of Vehicle Plate Number. *2020 International Conference in Mathematics, Computer Engineering and Computer Science*. Nigeria.
- Aldás, D., Collantes, S., Reyes, J. (2016). Procesamiento de imágenes con visión artificial para el reconocimiento de de placas vehiculares. *SATHIRI* (11), .133 - 142. . doi: ISSN 1390-6925
- Aranda, J. F. (2016). *Operaciones sobre el histograma y filtrado de la imagen*. En E. Alegre, G. Pajares, & A. D. Escalera (Edits.), *Conceptos y Métodos en visión por computador* (págs. 31-46). España : Grupo de Visión del Comité Español de Automática.
- Arredondo, M., Carmona, J., Padilla, A., Prado, J., Barranco, I. (2016). Implementación de sistema de reconocimiento de automóviles en imágenes basado en patrones locales binarios y clasificador tipo Adaboost. *CONIITEC*, 1-6. Obtenido de <http://coniitec.utrng.edu.mx/>
- Atikuzzaman, M., Asaduzzaman, M., & Zahidul Islam, M. (2019). Vehicle Number Plate Detection and Categorization Using CNNs. *2019 International Conference on Sustainable Technologies for*. Dhaka.
- Atikuzzaman, M., Asaduzzaman, M., & Zahidul Islam, M. (2019). Vehicle Number Plate Detection and Categorization Using CNNs. *2019 International Conference on Sustainable Technologies for*. Dhaka.
- Bit, A. (2018). *Design and development of affordable healthcare technologies [Diseño y desarrollo de tecnologías médicas asequibles]*. Estados Unidos de América: IGI Global.
- Cancelas, J. G. (2016). *Procesamiento morfológico*. En E. Alegre, G. Pajares, & A. D. Escalera (Edits.), *Conceptos y métodos en visión por computador* (págs. 77-96). España: Grupo de Visión del Comité Español de Automática.
- Catalán, A. (2019). Técnicas de procesamiento digital de imágenes. *Revista de Marina*, 68-71. doi:ISSN 0034-8511
- CEPLAN. (2021). *Plan Estratégico de Desarrollo Nacional Actualizado* .
- Chacón, M. (2016). *Percepción visual aplicada a la robótica*. México : Alfaomega Grupo Editor.
- Chen, J. (2017). Chinese License Plate Identification Based On Android Platform. *IEEE International Conference on "Computational Intelligence and Communication Technology" (IEEE-CICT 2017)*. Shanghai.
- Dhar, P., Zainal Abedin, M., Karim, R., Tuj Johora, F., & Shahadat Hossain, M. (2019). Bangladeshi License Plate Recognition Using Adaboost Classifier. *Joint 2019 8th International Conference on Informatics, Electronics & Vision*

- (ICIEV) & 3rd International Conference on Imaging, Vision & Pattern Recognition (IVPR) . Bangladesh.
- Dhar, P., Zainal Abedin, M., Karim, R., Tuj Johora², F., & Shahadat Hossain, M. (2019). Bangladeshi License Plate Recognition Using. *Joint 2019 8th International Conference on Informatics, Electronics & Vision (ICIEV) & Bangladesh: IEEE.*
- Espinosa, G. (2014). *Sistema de reconocimiento de patrones en placas vehiculares para el acceso automático de visitas a un edificio.*
- Espinoza, D., y Salinas, C. (2015). *Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares.* Obtenido de <http://dSPACE.uazuay.edu.ec/bitstream/datos/5047/1/11486.PDF>
- Esqueda, J. y. (2005). *Fundamentos para el procesamiento de imágenes.* Mexico: Universidad Autónoma de Baja California.
- F. Quiros, A. R., Anjerome Bedruz, R., Uy, A. C., Abad, A., Bandala, A., & Dadios, E. P. (2017). Un enfoque basado en kNN para la visión artificial del reconocimiento de caracteres de los números de matrícula. *Proc. de la Conferencia IEEE Región 10 de 2017 (TENCON), Malasia, 5-8 de noviembre de 2017.* Salle Manila.
- GARCÍA ORDÁS, M. T., ALAIZ RODRÍGUEZ, R., & ALEGRE, E. (2016). *Conceptos y Métodos en Visión por Computador.* España.
- García, M. A. (2016). *Clasificación y reconocimiento de patrones.* En E. Alegre, G. Pajares, & A. D. Escalera (Edits.), *Conceptos y métodos en visión por computador (págs. 159-180).* España: Grupo de Visión del Comité Español de Automática.
- García, M. y. (2016). *Sistema de reconocimiento automático de placa automotriz paraguaya.* *FPUNE Scientific(10), 43-47.* Obtenido de <http://www.une.edu.py:83/fpunescientific/index.php/fpunescientific/article/view/139/137>
- Gavilanes, J., Jácome, J., y Pazmiño, A. (2018). Diseño de sistemas embebidos para controlar la entrada y salida de vehículos en línea, en el acceso principal de la ESPOCH. *Revista Ciencia E Investigación., 3, 113-120.* doi:ISSN 2528-8083
- Górriz, H. Á. (2018). *Procesamiento digital de imagen para caracterización morfométrica de la de la especie Frankliniella Occidentalis.* España: Libros Electrónico Nro. 86, Universidad Almería.
- Guerrero Ibañez, J. A., Acosta Mendoza, J., & Contreras Castillo, J. J. (2015). Dándole Inteligencia a la Gestión de Estacionamientos en Zonas Públicas a través. *SISTEMAS, CIBERNÉTICA E INFORMÁTICA, 54.*

- Hernández, Fernández y Baptista. (2014). *Metodología de la Investigación*. México: Trillas.
- Hoffstaetter, S. (2018). *Pytesseract* . *PyPI*. Obtenido de <https://pypi.org/project/pytesseract/>
- Huamaní, F. y Rojas, J. (2015). Procesamiento digital de imágenes para estimación de velocidad y medición de flujo vehicular. *ELECTRÓNICA*, 67-74. Obtenido de http://revistas.urp.edu.pe/index.php/Perfiles_Ingenieria/article/view/406
- Hung Lin, C., & Sin Sie, Y. (2019). Two-Stage License Plate Recognition System Using Deep learning. *2019 8th International conference on Innovation, Communication and Engineering (ICICE)*. Taiwan.
- Hung Lin, C., Sin Lin, Y., & Chen Liu, W. (2018). An Efficient License Plate Recognition System Using Convolution Neural Networks. *Proceedings of IEEE International Conference on Applied System Innovation 2018* . Taipei.
- Isasi, P. y. (2004). *Redes neuronales artificiales un enfoque práctico*. España: Pearson Education.
- Jiménez, N., Díaz, R., Ambrocio, S., y Ramírez, M. . (2016). El procesamiento digital de imágenes aplicado al reconocimiento morfológico de estructuras de impacto en la superficie terrestre. *Revista Digital Universitaria* , 6(16). doi:ISSN 1607 - 6079
- Krishna Thangallapally , S., Maripeddi, R., & Krishna Banoth , V. (2018). E-Security System for Vehicle Number Tracking at. *2018 IEEE International Students 'Conference on Electrical, Electronics and Computer Science*. Nagpur.
- Morales, R. (2011). *Procesamiento y análisis digital de imágenes*. RA-MA S.A. Editorial y Publicaciones.
- Orellana, J., y Ortega, J. (2019). Tiempo de viaje, sistema de reconocimiento automático de placas vehiculares (ANPR), para detección de infractores en ruta Machala, Santa Rosa. *Polo del Conocimiento*, 5(1), 400-412. doi:ISSN: 2550 - 682X DOI: 10.23857/pc.v5i01.1230
- Oyala, V. (2014). *Sistemas de información geográfica*. España: CreateSpace Independent Publishing Platform (Amazon).
- Perdomo, E., Amaral, O., Monsanto, D., Pérez, J. (2018). Algoritmo para la detección e identificación de matrículas cubanas sobre raspberry pi. *VI Simposio Internacional de Electrónica: Diseño, Aplicaciones, Técnicas Avanzadas y Retos Actuales*, (págs. 1-10). Cuba. Obtenido de <http://www.informaticahabana.cu/sites/default/files/ponencias2018/ELE11.pdf>
- Quiros, A. R., Anjerome Bedruz, R., Christian Uy, A., Abad, A., Bandala, A., & Dadios, E. (2017). A kNN-based Approach for the Machine Vision of.

Conferencia IEEE Región 10 de 2017 (TENCON), Malasia, 5-8 de noviembre de 2017. Salle Manila.

- Quiros, A. R., Anjerome Bedruz, R., Uy, A. C., Abad, A., Bandala, A., & Dadios, E. P. (2017). Un enfoque basado en kNN para la visión artificial del reconocimiento de caracteres de los números de matrícula. *Proc. de la Conferencia IEEE Región 10 de 2017 (TENCON), Malasia, 5-8 de noviembre de 2017. Salle Manila, Filipinas.*
- Ruiz, J., y Fragoso, L. (2017). Técnica de imágenes digitales como herramienta de medición de daño en rompeolas. *INGENIERÍA HIDRÁULICA Y AMBIENTAL*, 38(2), 73-86. doi:ISSN 1815-591X
- Satsangi, M., Yadav, M., & Sewak Sudhish, P. (2018). License Plate Recognition. Agra, India.
- Satsangi, Yadav, & Sewak. (2019). License Plate Recognition., (pág. 5).
- Simon Mandi, M., Shibwabo, B., & Mutua Raphael, K. (2017). An Automatic Number Plate Recognition System for Car . *International Journal of Computer Applications*.
- Sucar, L. E. (2017). *Visión Computacional*. Puebla, México: Instituto Nacional de Astrofísica, Óptica y Electrónica.
- Sucar, L. E., & Gómez, G. (s.f.). *Visión Computacional*. Puebla, México.
- Sucar, L. E., & Gómez, G. (s.f.). *Visión Computacional*. Neuherberg, Alemania.
- Taufiq Musaddid, A., Bejo, A., & Hidayat, R. (2019). Improvement of Character Segmentation for Indonesian License Plate Recognition Algorithm using CNN . *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. Yogyakarta.
- Two-Stage License Plate Recognition System Using Deep learning. (2020). En C. Hung Lin, & Y. Sin Sie, *Departamento de Ingeniería Eléctrica, Universidad Nacional Normal de Taiwán*. Taiwán.
- Zainal Abedin, M., Chandra Nath, A., Dhar, P., Deb, K., & Shahadat Hossain, M. (2017). License Plate Recognition System Based On Contour . *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. Bangladesh.
- Zambrana, O. (2018). Detección de placas vehiculares, mediante vision artificial. *Illuminate*, 10, 17-30. doi:ISSN: 2415-2323
- Zhang, M., Yu, W., Su, J., & Li, W. (2019). Design of License Plate Recognition System Based on Machine Learning. *2019 IEEE 4th International Conference on Image, Vision and Computing*. China.

Anexos

Anexo N° 01

Resolución de aprobación de tema de tesis

 | SENOR DE SIPAN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N°0447-2021/FIAU-USS

Pimentel, 28 de mayo de 2021

VISTO:

El Acta de reunión N°1305-2021 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante oficio N°0227-2021/FIAU-IS-USS de fecha 19 de mayo de 2021, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la Facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

Que, según documentos de Vistos el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS acuerdan aprobar los temas de las Tesis a cargo de los estudiantes del curso de Investigación II que se detallan en el anexo de la presente Resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1°: APROBAR, el tema de la Tesis perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de los estudiantes del Programa de estudios de INGENIERÍA DE SISTEMAS según se detalla en el anexo de la presente Resolución.

ARTÍCULO 2°: ESTABLECER, que la inscripción del Tema de la Tesis se realice a partir de emitida la presente resolución y tendrá una vigencia de dos (02) años.

ARTÍCULO 3°: DEJAR SIN EFECTO, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE


 Dr. Néstor Paredón Barrios Huanc
Decano - Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.


 MSc. María Soledad Rojas Flores
Decana de Estudios / Decana de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N°0447-2021/FIAU-USS

Pimentel, 28 de mayo de 2021

N°	AUTOR (ES)	TÍTULO APROBADO
16	• LUQUE CONDORI BASILIO	EVALUACIÓN DE LA CALIDAD DE APLICACIONES GENERADAS POR BPM PARA LA GESTIÓN DE TRÁMITES EN UNA GERENCIA REGIONAL DE TRANSPORTE PERUANO
17	• PALOMINO PALOMINO CLAUDIO JESUS	ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE INTELIGENCIA DE NEGOCIOS ORIENTADOS A LAS PYMES
18	• PEREZ SILVA EDWIN WILDOR	RECONOCIMIENTO DE PLACAS VEHICULARES MEDIANTE VISION COMPUTACIONAL PARA MEJORAR EL ACCESO A UN PARQUEADERO
19	• ROJAS CAHUATA GUILLERMO	AUTOMATIZACION DE PROCESOS CONSIDERANDO LA METODOLOGIA BPM EN EL CENTRO DE CONTROL EN UNA ENTIDAD BANCARIA.
20	• SANTISTEBAN VALDERA ELVIS ALBERTO • VILLANUEVA CASTRO JOSE SEBASTIAN	DESARROLLO DE UN MODELO DE GESTIÓN DE INCIDENCIAS BASADO EN MARCOS DE REFERENCIA DE BUENAS PRÁCTICAS PARA MEJORAR EL SERVICIO DE TI. CASO DE ESTUDIO MUNICIPALIDAD DISTRITAL DE MÓRROPE - LAMBAYEQUE
21	• VILCHEZ SILVA OMAR JHONATHAN	ANÁLISIS COMPARATIVO DE ALGORITMO DE VITERBI Y BAUM-WELCH DE RECONOCIMIENTO DE VOZ PARA LA IDENTIFICACIÓN DE PERSONAS