

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

FACULTAD DE INGENIERÍA MINAS, GEOLOGÍA Y CIVIL

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**"Modelo machine learning para detectar las placas de vehículos hurtados
Ayacucho, 2023"**

Tesis para optar el título profesional de:

Ingeniera de sistemas

Presentado por:

Bach. Lizbeth Roxana Infante Leva

Asesor:

Dr. Manuel Avelino Lagos Barzola

Ayacucho - Perú

2024

Dedicatoria

Con gratitud infinita a mis padres y hermanos, por su apoyo incondicional ha sido mi mayor inspiración en este camino académico.

A mis amados padres Samuel y Felicitas, quienes con amor y sacrificio guiaron mi camino. Su apoyo es mi mayor fortaleza en esta travesía académica.

Agradecimiento

Quiero expresar mi sincero agradecimiento a la Universidad San Cristóbal de Huamanga por brindarme una experiencia educativa enriquecedora y desafiante. Agradezco a los profesores comprometidos, cuyo conocimiento ha sido una fuente invaluable de aprendizaje.

Mi más profundo reconocimiento va para mi asesor de tesis, cuya guía experta y apoyo constante fueron esenciales en cada fase de este proceso. La universidad no solo me proporcionó conocimientos académicos, sino también oportunidades de crecimiento personal y profesional. Este logro refleja la inspiración y educación de alta calidad que recibí aquí. Llevaré conmigo las valiosas lecciones aprendidas y el afecto por esta institución, que ha dejado una huella imborrable en mi vida. Agradezco a todos los que contribuyeron a mi formación .

Resumen

El trabajo de investigación “Modelo machine learning para detectar las placas de vehículos hurtados Ayacucho, 2023” surge a raíz del aumento de hurtos de vehículos en la región de Ayacucho que va cada vez en ascenso desde año 2015 al 2021 según datos del INEI, con esta investigación se desarrolló un modelo de algoritmo machine learning juntamente con la aplicación web para la detección de vehículos hurtados que permiten proporcionar a las autoridades de Ayacucho un sistema capaz de detectar, validar placas de vehículos y verificar su autenticidad en una base local.

El objetivo de este trabajo de investigación es determinar las características del modelo de machine learning para detectar las placas de vehículos hurtados mediante el tipo de modelo, arquitectura del modelo, eficacia y validez de modelo con la finalidad de detectar las Placas de vehículos hurtados Ayacucho,2023. El tipo de investigación es observacional, transversal y descriptiva.

Los resultados obtenidos evidenciaron una alta precisión en la detección de placas vehiculares y una efectiva verificación del estado de los vehículos en la base de datos. Asimismo, se apreció una notable mejora en la capacidad de reconocimiento del modelo.

El modelo de machine learning desarrollado para la detección de vehículos hurtados junto con tecnologías complementarias sería una herramienta poderosa para abordar el hurto de vehículos . Su capacidad para identificar placas y verificar su estado en tiempo real proporciona a las autoridades locales un recurso eficaz para fortalecer la seguridad vial y reducir la incidencia del crimen relacionado con el robo de vehículos.

Palabras claves: Inteligencia artificial, Machine learning, Redes neuronales, Placas, Vehículos, Tratamiento de imágenes, Recolección de imágenes.

Abstract

The research work 'Machine Learning Model for Detecting Stolen Vehicle Plates in Ayacucho, 2023' arises from the increasing trend of vehicle thefts in the Ayacucho region, which has been steadily rising from 2015 to 2021 according to data from the INEI. With this research, a machine learning algorithm model was developed along with a web application for detecting stolen vehicles, aimed at providing the authorities in Ayacucho with a system capable of detecting, validating vehicle plates, and verifying their authenticity in a local database.

The objective of this research work is to determine the characteristics of the machine learning model for detecting stolen vehicle plates through the type of model, model architecture, model effectiveness, and validity, with the purpose of detecting Stolen Vehicle Plates in Ayacucho, 2023. The type of research is Observational, cross-sectional, and descriptive.

The results obtained showed a high precision in the detection of vehicle plates and an effective verification of the vehicles' status in the database. Likewise, a notable improvement in the model's recognition capacity was observed.

The machine learning model developed for detecting stolen vehicles, along with complementary technologies, would be a powerful tool for addressing vehicle theft. Its ability to identify plates and verify their status in real-time provides local authorities with an effective resource to strengthen road safety and reduce the incidence of crime related to vehicle theft.

Keywords: Artificial Intelligence, Machine Learning, Neural Networks, Plates, Vehicles, Image Processing, Image Collection.

INDICE

Resumen.....	2
Abstract.....	3
Lista de tablas.....	6
Lista de figuras.....	7
INTRODUCCIÓN	9
CAPÍTULO I.....	10
PLANTEAMIENTO DEL PROBLEMA	10
1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA	10
1.2.1. Problema general	13
1.2.2. Problemas específicos	13
1.3.1. Objetivo general.....	13
1.3.2. Objetivo específico	13
1.4. HIPÓTESIS	14
1.5. JUSTIFICACIÓN DE LA INVESTIGACIÓN	14
1.5.1. Práctica	14
1.5.2. Socioeconómica	14
1.5.3. Tecnológica	15
1.6. LIMITACIONES DE LA INVESTIGACIÓN	15
CAPÍTULO II	16
MARCO TEÓRICO	16
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	16
2.1.1. Antecedentes nacionales	16
2.1.2. Antecedentes internacionales.....	16
2.2. MARCO TEORICO	17
2.2.1. Inteligencia artificial.....	17
2.2.2. Machine learning	19
2.2.3. Visión por computador.....	28
2.2.4. Framework y herramientas de desarrollo	29
CAPÍTULO III.....	31
MATERIAL Y MÉTODOS	31
3.1. TIPO Y NIVEL DE INVESTIGACIÓN	31
3.1.1. Tipo de investigación	31
3.1.2. Nivel de investigación	31
3.2. DISEÑO DE LA INVESTIGACIÓN	31

3.3.	VARIABLES	32
3.3.1.	Definición conceptual de variables	32
3.4.	POBLACIÓN Y MUESTRA.....	33
3.4.1.	Población.....	33
3.4.2.	Muestra	33
3.5.	TÉCNICAS E INSTRUMENTOS DE LA INVESTIGACIÓN	34
3.5.1.	Técnicas.....	34
3.5.2.	Instrumentos.....	34
	CAPÍTULO IV	36
	RESULTADOS	36
3.1.	RESULTADOS.....	36
3.1.1.	Recopilación de datos	41
3.1.2.	Preparación de los datos.....	54
3.2.3.	Elegir un modelo	57
3.2.4.	Entrenando el modelo.....	62
3.2.5.	Evaluación del modelo.....	66
3.2.6.	Ajuste de parámetros.....	67
3.2.7.	Hacer predicciones.....	73
	CAPÍTULO V	79
	CONCLUSIONES	79
5.1.	CONCLUSIONES.....	79
	CAPÍTULO VI.....	80
	RECOMENDACIONES	80
	REFERENCIA	81
	Anexo 1. Matriz de consistencia	88
	Anexo 2. Matriz de operacionalización de variables	90
	Anexo 3. Instrumento cámara de celular.....	92
	Anexo 4. Ficha de análisis documental	93

Lista de tablas

Tabla 1. Denuncias por robo de vehículos, según departamento, 2015-2021	11
Tabla 2. Vehículos recuperados por la policía nacional, según departamento, 2015 – 2021	12
Tabla 3. Herramientas tecnológicas	35
Tabla 4. Resultados de las pruebas de eficacia del modelo de detección de placas de vehículos hurtados	38
Tabla 5. Tabla librería_ conductor de la base de datos “bdplacaia”	45
Tabla 6. En la siguiente tabla se muestra los 16 primeros lista de la tabla librería_ conductor	46
Tabla 7. Tabla librería_ persona de la base de datos “bdplacaia”	47
Tabla 8. En la siguiente tabla se muestra los 15 primeros registros de la tabla librería_ persona	48
Tabla 9. Tabla librería_ placas de la base de datos “bdplacaia”	49
Tabla 10. En la siguiente tabla se muestra los ocho primeros registros de la tabla librería_ placas	50
Tabla 11. Tabla librería_ tipodoc de la base de datos “bdplacaia”	50
Tabla 12. En esta tabla se muestran los primeros cinco registros.....	51
Tabla 13. Tabla librería_ vehículo de la base de datos “bdplacaia”	52
Tabla 14. Lista de los 10 primeros registros de la tabla librería_ vehículo	53
Tabla 15. Tabla de base de datos relacional	53
Tabla 16. Código de procesamiento y análisis de información	54
Tabla 17. Resultado obtenido del estado de la placa vehicular "carro4.jpg"	74
Tabla 18. Resultado obtenido del estado de la placa vehicular "carro3.jpg"	76

Lista de figuras

Figura 1. Denuncia de robos de vehículos Ayacucho.....	10
Figura 2. Relación entre AI, ML, NN y aprendizaje profundo.....	18
Figura 3. Regresión lineal.....	22
Figura 4. Redes neuronales.....	24
Figura 5. Arquitectura RNN	26
Figura 6. Arquitectura de CNN	27
Figura 7. Marco de aprendizaje profunda de Pytorch.....	30
Figura 8. Estructura de la arquitectura para detectar placas de vehículos hurtados	37
Figura 9. Prueba de la validez del modelo machine learning para la detección de vehículos hurtados	38
Figura 10. Gráfico de la eficiencia de modelo machine learning para la detección de vehículos hurtados	39
Figura 11. Validez del conjunto de datos de entrenamiento.....	40
Figura 12. Validez del conjunto de datos prueba	40
Figura 13. Pasos para construir un modelo.....	41
Figura 14. Información de imágenes tomados de placas vehiculares.....	42
Figura 15. Script de la extracción de información de la placa de un vehículo	43
Figura 16. Listado de las primeras 30 imágenes mediante el empleo de los códigos de Python	44
Figura 17. Listado de 25 imágenes aleatorias de vehículos para su procesamiento	55
Figura 18. Script de la tabla de librería_ conductor.....	56
Figura 19. Script de la tabla librería_ persona	56
Figura 20. Script de la tabla librería_ placas	56
Figura 21. Script de la tabla librería_ tipodoc	56
Figura 22. Script de la tabla de librería_ vehículo.....	56
Figura 23. Script de la tabla de librería_ vehículo.....	57
Figura 24. Script de la tabla de librería_ placas.....	57
Figura 25. Código fuente del modelo de machine learning para detectar la placa de vehículo hurtados	58
Figura 26. Estructura del Proyecto en Django y Python antes del entrenamiento	63
Figura 27. Código fuente para reconocer los caracteres de una imagen de placa vehicular	63
Figura 28. Reconocimiento de carácter de la placa vehicular	65
Figura 29. Carácter extraído de la imagen de la placa vehicular.....	65
Figura 30. Salida del compilador de nuestro modelo machine learning para la detección de placas de vehículos hurtados.....	66
Figura 31. Evaluación del desempeño utilizando dos imágenes de placas vehiculares.....	67
Figura 32. Script de la aplicación web de reconocimiento de placas vehiculares	67
Figura 33. Clase views.py.....	68
Figura 34. Plantilla de index.html de la carpeta placa.....	71
Figura 35. Selección de la imagen denominada "carro4.jpg" para su posterior importación	73
Figura 36. Reconocimiento de la placa del vehículo "carro4.jpg"	74
Figura 37. Resultado de la búsqueda de la placa de vehículo"carro4.jpg"	75

Figura 38. Selección de la imagen denominada "carro3.jpg" para su posterior importación	75
Figura 39. Reconocimiento de la placa del vehículo "carro3.jpg"	76
Figura 40. Resultado de la búsqueda de la placa de vehículo "carro3.jpg".....	77
Figura 41. Reporte de vehículos hurtados	78
Figura 41. Reporte de vehículos hurtados	78

INTRODUCCIÓN

La investigación " Modelo machine Learning para detectar las placas de vehículos hurtados Ayacucho, 2023" se centró en desarrollar un modelo de algoritmo machine learning juntamente con la aplicación web para la detección de vehículos hurtados en tiempo real y verificar su estado mediante una base de datos local.

El aumento de los incidentes de robo de vehículos en Ayacucho, evidenciado por datos del INEI entre 2015 y 2021, destaca la urgencia de implementar soluciones adecuadas para mejorar la seguridad del transporte en la región. La creación de este modelo de machine learning surge como respuesta a esta situación, con el propósito de dotar a las autoridades locales de una herramienta eficaz para combatir el delito automotriz y salvaguardar a la población.

El objetivo fundamental de esta investigación es proporcionar a las autoridades de Ayacucho un sistema capaz de detectar y validar placas de vehículos robados. Al identificar de manera instantánea placas sustraídas y verificar su autenticidad en una base de datos local, este modelo de machine learning busca fortalecer la capacidad de las autoridades para prevenir el robo vehicular, reducir la tasa delictiva y mejorar la seguridad vial en la región.

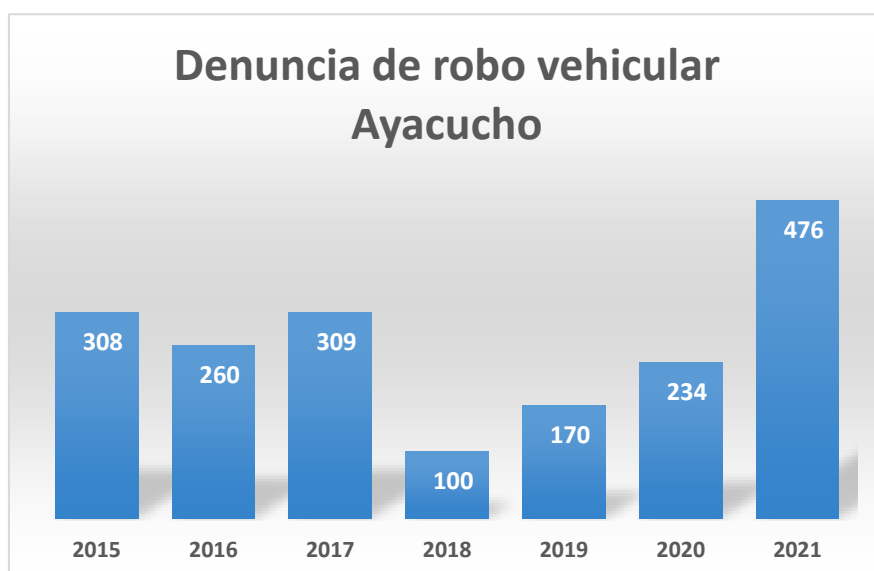
CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA

En la actualidad la región de Ayacucho se ve un incremento en el índice de hurtos de vehículos, viéndose un incremento mayor del año 2020 una cantidad de 234 y en el 2021 una cantidad de 476, y revisando los índices de vehículos recuperados por la policía nacional no llega hacer tan eficiente ya que en el 2020 se recupera 73 y en el año 2021 se recupera 117 vehículos por lo tanto se creará un algoritmo o modelo basado en Machine learning para detectar e identificar aquellos vehículos que fueron hurtados.

Figura 1. Denuncia de robos de vehículos Ayacucho



Nota. Índice de denuncias de vehículos hurtados del año 2015 al 2021,

Fuente (elaboración propia)

En un entorno de cambios continuos, la problemática de la identificación de placas de vehículos hurtados destaca como una necesidad inmediata. Se requiere un modelo o algoritmo que pueda detectar las placas de vehículos hurtados, mejorando así la toma de decisiones. Este modelo debe detectar eficazmente los robos de vehículos, un problema que impacta a toda la comunidad de Ayacucho.

Según el Ministerio del Interior (MININTER), la tabla presenta la cantidad de denuncias de vehículos robados por departamento. En Ayacucho, se observa un incremento en las denuncias de robo de vehículos a lo largo de los años: 304 en 2015, 260 en 2016, 309 en 2017, 100 en 2018, 170 en 2019, 234 en 2020 y 476 en 2021. Este aumento considerable evidencia un alto índice de robos en la región.

Tabla 1. Denuncias por robo de vehículos, según departamento, 2015-2021

Departamento	2015	2016	2017	2018	2019	2020	2021
Total	16 501	17 544	18 106	19 084	20 159	13 984	19 991
Amazonas	44	47	58	153	24	171	396
Áncash	165	191	184	143	157	82	234
Apurímac	16	19	20	75	69	44	89
Arequipa	138	134	141	141	95	62	91
Ayacucho	308	260	309	100	170	234	476
Cajamarca	395	609	509	520	953	685	1 468
Prov. Const. del Callao	313	283	270	1 084	419	267	391
Cusco	93	59	438	130	194	257	188
Huancavelica	5	8	5	20	4	2	2
Huánuco	799	462	642	597	195	306	627
Ica	746	1 466	937	671	410	331	582
Junín	761	780	755	561	614	402	652
La Libertad	1 100	1 270	1 130	1 292	1 664	898	1 545
Lambayeque	662	287	225	1 205	1 114	756	960
Lima Metropolitana 1/ y Lima 2/	6 211	5 649	5 833	6 982	8 637	5 873	6 847
Lima Metropolitana 1/	...	...	...	...	8 031	5 527	6 411
Lima 2/	...	...	...	...	606	346	436
Loreto	1 741	2 062	2 138	2 519	2 605	1 095	1 797
Madre de Dios	730	465	811	544	585	47	436
Moquegua	13	15	23	13	19	16	14
Pasco	4	6	30	39	111	105	125
Piura	350	602	581	785	1058	1 600	2 297
Puno	446	599	539	42	29	52	61
San Martín	194	338	737	1 048	584	440	285
Tacna	76	62	52	20	46	35	58
Tumbes	448	412	343	350	367	203	163
Ucayali	743	1 459	1 396	50	36	21	207

Nota. Esta tabla ilustra la variación en la cantidad de robos de vehículos por departamento.

Fuente (MININTER, 2021)

Según datos del Ministerio del Interior (MININTER), se presenta una tabla que muestra la cantidad de vehículos recuperados por departamento. En Ayacucho, específicamente, se observa que el número de vehículos recuperados fue el siguiente: 112 en 2015, 88 en 2016, 95 en 2017, 111 en 2018, 141 en 2019, 73 en 2020 y 117 en 2021. Estas cifras muestran una variabilidad en la cantidad de vehículos recuperados a lo largo de los años, con algunos períodos de incremento y otros de disminución.

Tabla 2. Vehículos recuperados por la policía nacional, según departamento, 2015 – 2021

Departamento	2015	2016	2017	2018	2019	2020	2021
Total	12 629	12 991	12 676	14 865	13 690	10 309	12 108
Amazonas	125	170	97	153	226	179	322
Áncash	154	134	191	125	147	84	178
Apurímac	18	31	51	40	48	42	54
Arequipa	141	134	135	138	104	49	58
Ayacucho	112	88	95	111	141	73	117
Cajamarca	225	357	512	586	630	537	985
Prov. Const. del Callao	220	241	233	397	252	50	99
Cusco	171	70	257	129	116	94	107
Huancavelica	7	23	18	6	3	1	10
Huánuco	1 006	783	800	276	221	345	449
Ica	291	461	502	319	316	169	197
Junín	215	320	434	477	306	170	349
La Libertad	702	762	641	516	901	790	987
Lambayeque	361	344	117	429	257	210	248
Lima Metropolitana 1/ y Lima 2/	5 943	6 032	5 778	8 190	7 504	5 394	5 432
Lima Metropolitana 1/	...	...	...	...	7 059	4 524	4 272
Lima 2/	...	...	...	...	445	870	1 160
Loreto	1 132	836	756	1 184	956	742	660
Madre de Dios	256	236	251	145	119	66	157
Moquegua	8	14	30	18	21	10	7
Pasco	4	12	31	17	29	57	58
Piura	374	467	472	534	434	460	762
Puno	92	99	110	47	29	46	42
San Martín	330	359	553	625	592	437	538
Tacna	50	45	38	15	37	16	24
Tumbes	188	140	124	258	168	146	118
Ucayali	504	833	450	130	133	142	150

Nota. Esta tabla muestra vehículos recuperados por la policía nacional del año 2015-2021, según departamento.

Fuente (MININTER, 2021)

Surgen preguntas para solucionar este problema de robos vehiculares que afectan día a día a las personas: ¿Cuál es el tipo de modelo de machine learning para detectar las Placas de vehículos hurtados?, ¿Cuál es la arquitectura del modelo de machine learning para

detectar las Placas de vehículos hurtados?, ¿Cuál es la eficacia del modelo de machine learning para detectar las Placas de vehículos hurtados?, ¿Cuál es la validez del modelo de machine learning para detectar las Placas de vehículos hurtados?

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. Problema general

¿Cuáles son las características del modelo de machine learning para detectar las placas de vehículos hurtados Ayacucho, 2023?

1.2.2. Problemas específicos

- a. ¿Cuál es el tipo de modelo de machine learning para detectar las Placas de vehículos hurtados?
- b. ¿Cuál es la arquitectura del modelo de machine learning para detectar las Placas de vehículos hurtados?
- c. ¿Cuál es la eficacia del modelo de machine learning para detectar las Placas de vehículos hurtados?
- d. ¿Cuál es la validez del modelo de machine learning para detectar las Placas de vehículos hurtados?

1.3. OBJETIVOS

1.3.1. Objetivo general

Determinar las características de un modelo machine learning para detectar las placas de vehículos hurtados en la ciudad de Ayacucho, 2023 .

1.3.2. Objetivo específico

- a. Determinar el tipo del modelo machine learning, con la finalidad de detectar las placas de vehículos hurtados.
- b. Crear una arquitectura del modelo machine learning, con la finalidad de detectar las placas de vehículos hurtados.
- c. Medir la eficacia del modelo machine learning que determinará la rapidez del método de algoritmo, con la finalidad de detectar las placas de vehículos hurtados.
- d. Determinar la validez del modelo machine learning, con la finalidad de detectar las placas de vehículos hurtados.

1.4. HIPÓTESIS

Conforme a lo expresado por Hernandez (2014), una hipótesis sirve como orientación en un estudio o investigación, delineando lo que se intenta demostrar y describiéndose como una explicación provisional de un fenómeno objeto de estudio. Estas hipótesis se derivan de teorías preexistentes y deben expresarse como proposiciones; esencialmente, representan respuestas preliminares a las preguntas de investigación. Es crucial reconocer que en nuestra cotidianidad a menudo formulamos suposiciones sobre diversos temas y posteriormente las verificamos para confirmar su veracidad. (p. 104).

Según Bernal (2010), menciona que en una investigación descriptiva, no siempre es necesario formular una hipótesis, se narran hechos, características o rasgos del objeto de estudio y se diseñan productos como modelos o prototipos. No se ofrecen explicaciones, sino que se apoyan en técnicas como encuestas, entrevistas, observación y resúmenes documentales para responder las preguntas de investigación.

En la presente investigación **no se realizó el planteamiento de hipótesis**, considerando que esta investigación es de nivel descriptivo, además no pretendemos pronosticar una cifra o un hecho; sin embargo, se logró los objetivos específicos planteados.

1.5. JUSTIFICACIÓN DE LA INVESTIGACIÓN

1.5.1. Práctica

El desarrollo de un modelo de machine learning junto con una aplicación web para la detección de vehículos hurtados ofrece una herramienta práctica para las autoridades de Ayacucho. Esta investigación busca proporcionar un sistema capaz de identificar y validar placas de vehículos, verificando su autenticidad en una base local. La justificación práctica radica en la necesidad de contar con un recurso eficaz para fortalecer la seguridad vial y reducir la incidencia del crimen relacionado con el robo de vehículos en la región.

1.5.2. Socioeconómica

La implementación de un modelo de machine learning junto con una aplicación web busca proporcionar una solución efectiva que contribuya a mejorar la seguridad pública, proteger el patrimonio de los ciudadanos y promover un entorno favorable para el crecimiento socioeconómico en Ayacucho.

1.5.3. Tecnológica

El desarrollo de un modelo de machine learning junto con una aplicación web para la detección de vehículos hurtados aprovecha los avances en tecnología de inteligencia artificial y visión por computadora. Estas herramientas permiten automatizar el proceso de identificación y validación de placas de vehículos. Además, la aplicación de estas tecnologías permite una mayor precisión y rapidez en la detección de vehículos hurtados.

1.6. LIMITACIONES DE LA INVESTIGACIÓN

Las limitaciones de la investigación abarcan la posible falta de disponibilidad de datos precisos sobre el robo de vehículos en la región de Ayacucho, el modelo podría ser sensible a cambios ambientales, como alteraciones climáticas o de iluminación, durante la captura de imágenes, afectando su precisión al identificar placas robadas, la necesidad de involucrar a la comunidad y la importancia de actualizar el modelo frente a cambios en las estrategias de robo.

CAPÍTULO II

MARCO TEÓRICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1. Antecedentes nacionales

Según Tao et al. (2019), en su artículo de revista “Detección de vehículos basada en el contexto de la escena en imágenes aéreas de alta resolución” de la Universidad Nacional de San Agustín de Arequipa, El autor concluye que el método de detección de vehículos basado en el contexto de la escena es efectivo. Utiliza la clasificación de escenas mediante aprendizaje profundo y detecta vehículos en carreteras y estacionamientos con diferentes detectores. Primero, realizamos la clasificación de escenas usando el método de aprendizaje profundo y, luego, detectamos vehículos en carreteras y estacionamientos por separado a través de diferentes detectores de vehículos. Posteriormente, optimizamos aún más los resultados de detección utilizando diferentes reglas de posprocesamiento según los diferentes tipos de escena.

2.1.2. Antecedentes internacionales

Según Ordóñez et al. (2020), en su en su artículo de revista “Modelo de machine learning para la predicción de las tendencias de hurto en Colombia”, en la Universidad del Cauca, concluye que el modelo de aprendizaje automático se basa en Máquinas de vectores de soporte de regresión (SRV) Adaptado para pronóstico de tendencias robos en Colombia y sus tres ciudades más importantes. El modelo está homologado un conjunto de datos obtenido del Sistema de Información de la fiscalía de Colombia, que incluye 2.662.402 propios colombianos 1960-2019. Los resultados se compararon con los siguientes modelos regresión lineal estándar y modelo SRV no ajustado, resultados obtenidos son prometedores y pueden ser utilizados por las autoridades para la toma de decisiones y apto para antirrobo.

En relacion a este tema Rondón et al. (2021), menciona en su artículo de revista " Detección de placas vehiculares mediante modelo de clasificador en cascada basado en lenguaje Python ", en la universidad Francisco de Paula Santander, concluye que el modelo de clasificador en cascada para la detección de placas vehiculares, utilizando herramienta Python, OpenCV y Cascade Trainer GUI basada en código abierto y aplicaron datos de imágenes digitales de 245 a 1867 para el entrenamiento de detector en

cascada. El modelo de clasificación tardó 17.4 minutos en crearse, y se probó imágenes y videos en ambientes, logrando la detección de placas vehiculares con una efectividad del 90.26%.

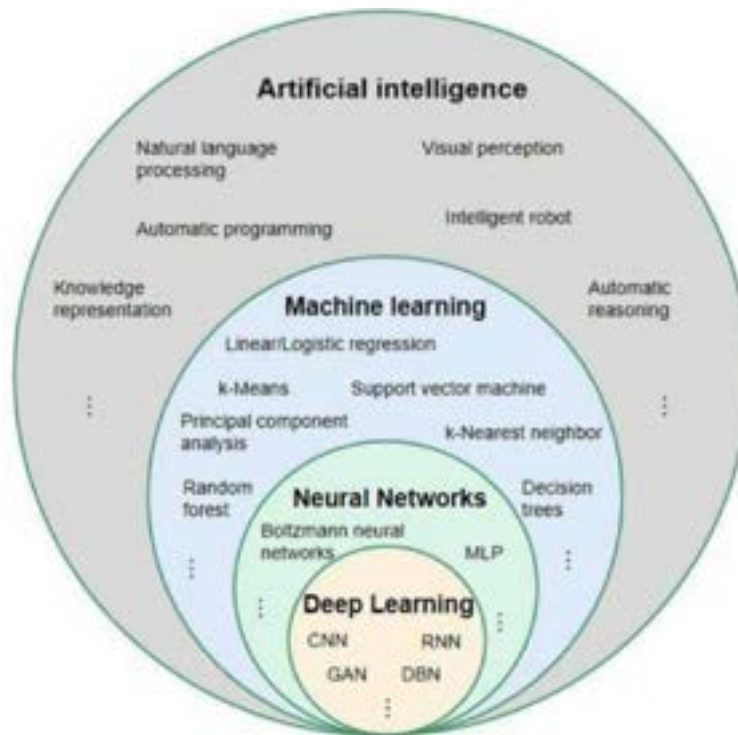
Pinzon (2001), en sus Tesis “desarrollo de nuevos modelos matemáticos para el precálculo básico de daños causados por acción de fenómenos naturales sobre comunidades vulnerables, basado en el uso de técnicas de inteligencia artificial (redes neuronales).” de la Universidad Nacional de Colombia concluye que mediante la aplicación de técnicas de procesamiento de datos por entrenamiento de Redes Neuronales y dinamizar el uso de la información disponible actualmente sobre desastres, con el propósito de que los datos recolectados sean realmente útiles en los procesos de planificación. Las Redes Neuronales no son modelos “mágicos” cuya “inteligencia” resuelve todo. Los resultados del modelo (software) es brindar el apoyo o guía al experto, no reemplazarlo.

2.2. MARCO TEORICO

2.2.1. Inteligencia artificial

Inteligencia artificial (IA), es la capacidad de una computadora digital o un robot controlado por computadora para realizar tareas normalmente asociadas con seres inteligentes. La IA es el resultado de una larga automatización, tras la cual ha sido capaz de repetir ciertas partes del comportamiento inteligente: tareas complejas como leer texto y tomar decisiones o planificar rutas. La inteligencia artificial es la capacidad de las máquinas para reproducir el comportamiento humano (Pacheco, 2023).

Figura 2. Relación entre AI, ML, NN y aprendizaje profundo



Nota. Los diferentes partes de AI, ML, NN y aprendizaje profundo.
Fuente (techringe).

La inteligencia artificial (IA) se refiere a la simulación de la inteligencia humana en máquinas que están programadas para pensar como humanos e imitar su comportamiento. El término también se puede aplicar a cualquier máquina que tenga características asociadas con la mente humana, como el aprendizaje y la resolución de problemas. La inteligencia artificial asocia con el comportamiento de la inteligencia del cerebro humano (Ocaña-Fernández et al, 2019).

La inteligencia artificial se define como la capacidad de las computadoras digitales o robots controlados por computadora para realizar tareas normalmente asociadas con seres inteligentes. La inteligencia artificial también se define como un dispositivo inteligente hecho por humanos capaz de realizar tareas de manera inteligente sin recibir instrucciones claras y capaz de pensar y actuar de manera racional y humana. La inteligencia artificial o IA puede entenderse como una disciplina perteneciente a la informática que ofrece un modelo informático de aprendizaje basado en redes neuronales biológicas humanas (Márquez Díaz, 2020).

La inteligencia artificial (IA) es un conjunto de tecnologías que permiten a las

computadoras realizar una variedad de funciones avanzadas, incluida la capacidad de ver, comprender y traducir lenguaje hablado y escrito, analizar datos, hacer recomendaciones y más. La inteligencia artificial en la actualidad está revolucionando ya que existen algoritmos de tras el entrenamiento actúan de manera autónoma (Ricardo et al, 2021).

2.2.2. Machine learning

Según Barrios (2016), un modelo de aprendizaje automático o machine learning es una expresión para el algoritmo de análisis de grandes cantidades de datos, que busca patrones que predice e investiga. Es un tipo de inteligencia artificial que puede aprender a partir de datos de entrenamiento sin utilizar la programación tradicional. Por lo tanto, cada vez que el algoritmo pasa por el proceso de entrenamiento utilizando una gran cantidad de datos, este produce un modelo más preciso.

Para comprender mejor este concepto, es útil considerar las palabras de Cowley et al. (2021), que los modelos de aprendizaje automático son archivos que están entrenados para reconocer ciertos tipos de patrones. Entrene un modelo en un conjunto de datos, proporciónelo con algoritmos que se puedan usar para inferir y aprender de esos datos. Una vez que haya entrenado un modelo, puede usarlo para considerar y predecir datos que nunca antes había visto. Por ejemplo, supongamos que desea crear un programa que pueda reconocer las emociones de un usuario en función de sus expresiones faciales. Entrenas a un modelo alimentándolo con imágenes de rostros que tienen ciertas emociones, y puedes usar ese modelo en una aplicación que puede reconocer cualquier emoción del usuario.

Según Databricks (2023), un modelo de aprendizaje automático es un programa que encuentra patrones o toma decisiones basadas en conjuntos de datos nunca antes vistos. Por ejemplo, los modelos de aprendizaje automático en el procesamiento del lenguaje natural pueden analizar e identificar la intención de oraciones o combinaciones de palabras que nunca antes se habían escuchado. En el reconocimiento de imágenes, se puede entrenar un modelo de aprendizaje automático para que reconozca objetos como automóviles o perros. Los modelos de aprendizaje automático pueden realizar tales tareas "entrenando" grandes conjuntos de datos. Durante el entrenamiento, los algoritmos de aprendizaje automático se optimizan para encontrar patrones o resultados específicos de

un conjunto de datos en función de una tarea. El resultado de este proceso, generalmente un programa de computadora con reglas especiales y estructuras de datos, se denomina modelo de aprendizaje automático.

Un modelo de aprendizaje automático es un programa que encuentra patrones o toma decisiones basadas en conjuntos de datos nunca antes vistos. Por ejemplo, los modelos de aprendizaje automático en el procesamiento del lenguaje natural pueden analizar e identificar la intención de oraciones o combinaciones de palabras que nunca antes se habían escuchado. En el reconocimiento de imágenes, se puede entrenar un modelo de aprendizaje automático para que reconozca objetos como automóviles o perros. Un modelo de aprendizaje automático es un programa informático que se utiliza para reconocer patrones en los datos o hacer predicciones (Coursera, 2023).

Los modelos de aprendizaje automático pueden realizar tales tareas "entrenando" grandes conjuntos de datos. Durante el entrenamiento, los algoritmos de aprendizaje automático se optimizan para encontrar patrones o resultados específicos de un conjunto de datos en función de una tarea. El resultado de este proceso, generalmente un programa de computadora con reglas especiales y estructuras de datos, se denomina modelo de aprendizaje automático. Los modelos de aprendizaje automático son similares a las funciones matemáticas: toman una solicitud en forma de datos de entrada, predicen esos datos de entrada y luego brindan una respuesta (Paperspace, 2020).

2.2.2.1. Aprendizaje supervisado

El aprendizaje supervisado es una sub categoría del machine learning, es un conjunto de datos etiquetados para entrenar algoritmos que clasifican datos o predice resultados con precisión. El aprendizaje supervisado ayuda a las organizaciones a resolver una variedad de problemas del mundo real a escala, como clasificar el correo no deseado en una carpeta separada de su bandeja de entrada. Un algoritmo de aprendizaje supervisado. Se le dice “supervisado” ya que para su entrenamiento necesita de un conjunto de datos previamente etiquetado y clasificado (Barrionuevo et al, 2020, p.4).

Según JavaTpoint (2021), el aprendizaje supervisado es el modelo de aprendizaje automático más simple para comprender qué datos de entrada se denominan datos de entrenamiento y tienen una etiqueta o resultado conocido como salida. Entonces, funciona

según el principio de los pares de entrada-salida. Requiere crear una función que pueda ser entrenada usando un conjunto de datos de entrenamiento, y luego se aplica a datos desconocidos y hace algún rendimiento predictivo. El aprendizaje supervisado se basa en tareas y se prueba en conjuntos de datos etiquetados.

En el aprendizaje supervisado, la máquina aprende bajo supervisión. Contiene un modelo que puede predecir con la ayuda de un conjunto de datos etiquetado. Un conjunto de datos etiquetado es aquel en el que ya conoce la respuesta objetivo. En el aprendizaje supervisado se busca inducir modelos capaces de predecir el valor de ciertas variables dependientes a partir de variables independientes (González, 2015).

El aprendizaje automático supervisado requiere datos de entrada y salida etiquetados durante la fase de entrenamiento del ciclo de vida del modelo de aprendizaje automático. En el aprendizaje supervisado se busca inducir modelos capaces de predecir el valor de ciertas variables dependientes a partir de variables independientes (González, 2015).

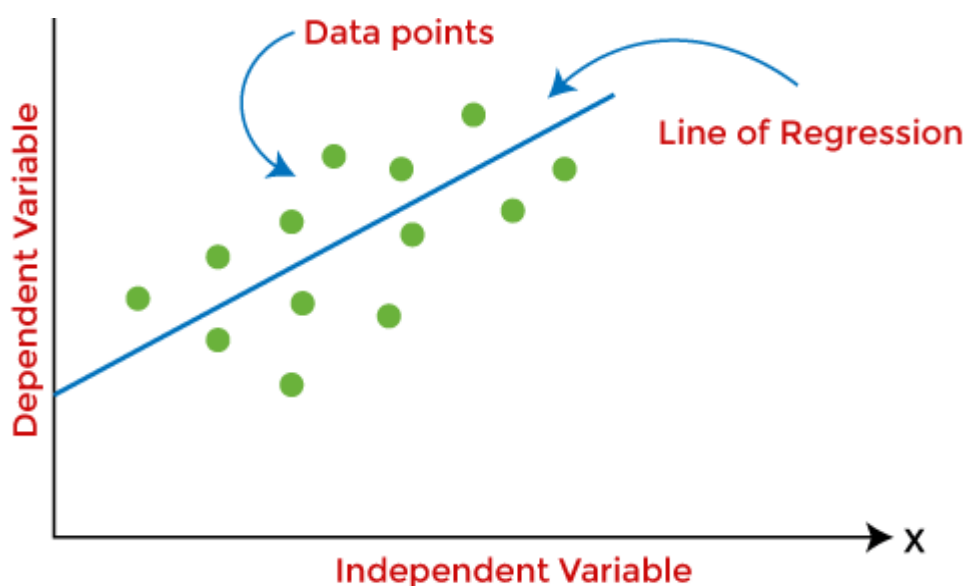
En este método, el algoritmo se entrena con datos etiquetados, donde se conoce la salida deseada. Su propósito es predecir esta salida para nuevos datos. Ejemplos de algoritmos de aprendizaje supervisado son la regresión lineal, árboles de decisión, bosque aleatorio y redes neuronales las cuales detallaremos a continuación.

2.2.2.2. Regresión lineal

La regresión lineal se utiliza para predecir el valor de una variable en función del valor de otra variable. La variable que desea predecir se llama variable dependiente. La variable que estás usando para predecir el valor de la otra variable se llama variable independiente. La regresión es una técnica estadística que consiste en calcular dicha similitud en forma de función matemática. Esta función nos ofrece mucha más información sobre dicha relación (Granados, 2016).

Para entender el concepto básico nos enfocaremos en siguiente gráfico.

Figura 3. Regresión lineal



Nota. La figura muestra la extensión de la regresión lineal a la regresión lineal múltiple (encontrar el plano de mejor ajuste) y la regresión polinomial (encontrar la curva de mejor ajuste), por JavaTpoint, 2021 (<https://www.javatpoint.com/machine-learning-models>).

La regresión lineal intenta modelar la relación entre dos variables ajustando una ecuación lineal a los datos observados. Una variable se considera una variable explicativa y la otra se considera una variable dependiente. A partir de la regresión lineal es posible hacer predicciones sobre la respuesta con base en valores de la variable predictora (Carrasquilla et al, 2016).

Según JavaTpoint (2021), menciona que la regresión lineal es el modelo de aprendizaje automático más simple en el que intentamos predecir una variable de salida utilizando una o más variables de entrada. Una representación de regresión lineal es una ecuación lineal que combina un conjunto de valores de entrada (x) y un conjunto de salidas esperadas (y) para esos valores de entrada. Se representa en forma de línea: $Y = b \times x + c$.

2.2.2.3. Árbol de decisión

Un árbol de decisión es un algoritmo de aprendizaje supervisado no paramétrico, que se utiliza tanto para tareas de clasificación como de regresión. Tiene una estructura de árbol jerárquica, que consta de un nodo raíz, ramas, nodos internos y nodos hoja. Los árboles de decisión intentan dividir un conjunto de puntos en subconjuntos inconexos en función de las respuestas a varias preguntas relacionadas con funciones (Meléndez, 2023, p. 15).

Como menciona JavaTpoint (2021), un árbol de decisiones utiliza una estructura de decisiones similar a un árbol junto con sus posibles consecuencias y resultados. En esto, cada nodo interno se utiliza para representar una prueba sobre un atributo; cada rama se utiliza para representar el resultado de la prueba. Cuantos más nodos tenga un árbol de decisión, más preciso será el resultado.

Un árbol de decisión es una estructura de árbol similar a un diagrama de flujo, donde cada nodo interno denota una prueba en un atributo, cada rama representa un resultado de la prueba y cada nodo de hoja (nodo terminal) tiene una etiqueta de clase. Un árbol de decisión es una forma gráfica y analítica de representar todos los eventos (eventos) como resultado de una decisión tomada en un momento determinado (Fernández et al, 2022, p.3).

2.2.2.4. Bosque aleatorio

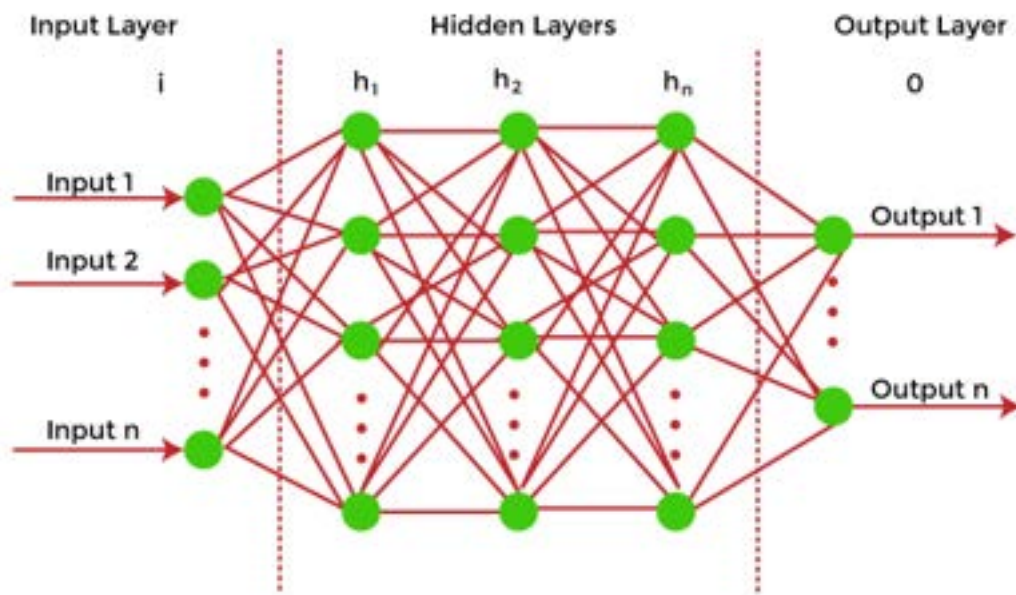
El algoritmo de bosque aleatorio es una extensión del método de embolsado, ya que utiliza tanto el embolsado como la aleatoriedad de características para crear un bosque de árboles de decisión no correlacionado. Los bosques aleatorios generan una gran cantidad de árboles de decisión. Cada observación se ingresa en un árbol de decisión (Panduro, 2022).

Segun JavaTpoint (2021), menciona que el método de aprendizaje por conjuntos, que consiste en una gran cantidad de árboles de decisión. Cada árbol de decisión en un bosque aleatorio predice un resultado, y la predicción con la mayoría de votos se considera como el resultado. Un modelo de bosque aleatorio se puede utilizar tanto para problemas de regresión como de clasificación.

2.2.2.5. Redes neuronales

Para entender el concepto básico nos enfocaremos en siguiente gráfico.

Figura 4. Redes neuronales



Nota. La figura muestra cada neurona artificial está conectada a muchas otras neuronas en una red neuronal, y estos millones de neuronas conectadas forman una estructura cognitiva compleja, por JavaTpoint, 2021 (<https://www.javatpoint.com/machine-learning-models>).

Las redes neuronales son el subconjunto del aprendizaje automático y también se conocen como redes neuronales artificiales. Las redes neuronales están formadas por neuronas artificiales y diseñadas de una manera que se asemeja a la estructura y el funcionamiento del cerebro humano. Cada neurona artificial se conecta con muchas otras neuronas en una red neuronal, y esos millones de neuronas conectadas crean una estructura cognitiva sofisticada. Las redes neuronales convolucionales (CNN) se forman por diversas capas convolucionales, que pueden tener una dimensión única o múltiples dimensiones (Esmaylin, 2023).

Segun JavaTpoint (2021), menciona que las redes neuronales son el subconjunto del aprendizaje automático y también se conocen como redes neuronales artificiales. Las redes neuronales están formadas por neuronas artificiales y diseñadas de una manera que se asemeja a la estructura y el funcionamiento del cerebro humano. Cada neurona artificial se conecta con muchas otras neuronas en una red neuronal, y esos millones de neuronas conectadas crean una estructura cognitiva sofisticada.

Las redes neuronales son un medio para hacer aprendizaje automático, en el que una

computadora aprende a realizar alguna tarea analizando ejemplos de entrenamiento. Las redes neuronales son modelos predictivos, es decir, son capaces de hacer predicciones ante nuevos datos, teniendo en cuenta datos anteriores (Del Barrio et al, 2022).

A continuación, veremos la entrada ponderada:

$$z = \sum_{k=i}^n (w_i \times x_i) + b$$

La entrada ponderada z de una neurona se obtiene sumando el producto de cada entrada x_i por su respectivo peso w_i , para luego sumar el sesgo b .

Después de calcular la entrada ponderada se aplica la función $f()$ para obtener la salida de la neurona:

$$a = f(z)$$

Donde a es la salida de la neurona y la $f()$ es la función de activación que introduce la no linealidad.

En cada neurona de la red, estas ecuaciones se emplean para procesar los datos y predecir si una placa vehicular está robada, basándose en entradas y pesos ajustados durante el entrenamiento del modelo de detección de vehículos hurtados.

2.2.2.5.1.1. Tipo de Redes neuronales

Dependiendo del procesamiento de datos y los objetivos, se explorarán dos tipos de Redes Neuronales Artificiales (RNAs) en la investigación de la presente tesis.

- a) Redes neuronales recurrentes (RNN)
- b) Redes neuronales convolucionales (CNN)

A continuación, describiremos estas variedades de redes neuronales artificiales.

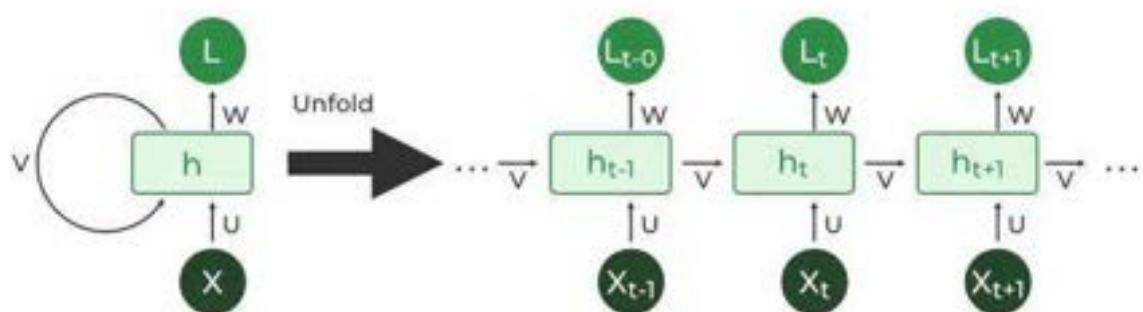
A. Redes neuronales recurrentes (RNN)

La red neuronal recurrente (RNN) es un tipo de red neuronal artificial diseñada para manejar datos secuenciales o series temporales. A menudo se emplean en tareas como traducción, procesamiento de lenguaje natural y reconocimiento de voz. A diferencia de las CNN y las redes de avance, las RNN tienen memoria, lo que les permite considerar información previa en su procesamiento actual. Mientras que las RNN unidireccionales no pueden utilizar eventos futuros en sus predicciones. Estos bucles representan la memoria interna de la red, empleada para comparar las características del dato actual con las de los datos recientes (Arana, 2021).

Según Chilón (2023), La red neuronal recurrente (RNN) es un tipo especial que utiliza la salida previa como entrada actual, con un estado oculto que retiene información sobre la secuencia. Este estado, llamado memoria, recuerda la entrada anterior. Además, utiliza los mismos parámetros en todas las capas ocultas para procesar cada entrada y producir la salida correspondiente.

La red neuronal recurrente (RNN) emplea la salida anterior como entrada actual, con un estado oculto que almacena información de la secuencia. Este estado, denominado memoria, recuerda la entrada previa, utilizando parámetros consistentes para procesar cada entrada y generar la salida correspondiente (Tierrablanca, 2023).

Figura 5. Arquitectura RNN

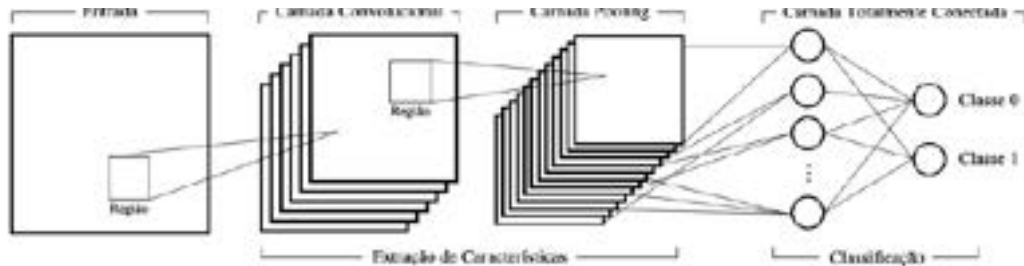


Nota. Esta arquitectura se apoya en el procesamiento de datos organizados en secuencias ordenadas (geeksforgeeks, 2023).

B. *Redes neuronales convolucionales (CNN)*

Para entender el concepto básico nos enfocaremos en siguiente gráfico.

Figura 6. *Arquitectura de CNN*



Nota. Esta arquitectura se basa en reconocimientos de imágenes, a base de múltiples redes neuronales (Nasch, 2015).

Según Artola Moreno (2019), menciona que las redes neuronales son un subconjunto del aprendizaje automático y son la base de los algoritmos de aprendizaje profundo. Consisten en capas de nodos que contienen una capa de entrada, una o más capas ocultas y una capa de salida. Cada nodo está conectado a otro nodo y tiene un peso y un umbral asociados. Si la salida de un nodo supera cierto umbral, ese nodo se activa y envía datos a la siguiente capa de la red. De lo contrario, los datos no se envían a la siguiente capa de red. Una red neuronal tiene que reconocer una gran cantidad de imágenes por sí misma para capturar una imagen, propiedades únicas de cada objeto y así poder generalizarlo.”

Las redes neuronales convolucionales son muy similares a las redes neuronales regulares del capítulo anterior: consisten en neuronas con pesos y sesgos que se pueden aprender. Cada neurona toma una entrada conocida, realiza una producción de puntos y, opcionalmente, la sigue con un procesamiento no lineal. Toda la red sigue representando una única función de puntuación diferenciable: desde píxeles de imagen sin procesar en un extremo hasta puntuaciones de clase en el otro. Es una red neuronal que usa operaciones de convolución para procesar fotografía (Orbe Cisneros & Vargas Vallejo, 2022).

Como menciona Martín de Diego (2022), una red neuronal convolucional es un tipo de red neuronal de realimentación comúnmente utilizada para el análisis de imágenes visuales mediante el procesamiento de datos utilizando una topología similar a una red. También se le llama red convolucional. Las redes neuronales convolucionales se utilizan

para detectar y clasificar objetos en imágenes. Las redes neuronales utilizan una topología en forma de celosía para procesar datos, en lugar de utilizar operaciones de convolución lineal, multiplicar la matriz general por al menos una capa.

La red neuronal convolucional (CNN) es una red neuronal artificial que se utiliza principalmente para el reconocimiento y procesamiento de imágenes debido a su capacidad para reconocer patrones en las imágenes. CNN es una herramienta poderosa, pero el entrenamiento requiere millones de puntos de datos etiquetados. Las CNN deben entrenarse con procesadores de alto rendimiento, como GPU o NPU, para que puedan producir resultados útiles lo suficientemente rápido. Una red neuronal es un modelo que simula la actividad de las neuronas en el cerebro humano. Una red neuronal tiene 3 partes: capa de entrada, capa oculta y capa de salida (Beltrán Morales, 2023).

2.2.3. Visión por computador

La visión por computadora es un campo de la inteligencia artificial dedicado a crear algoritmos y métodos para que las máquinas puedan interpretar y entender información visual extraída de imágenes o videos. Emplea técnicas de procesamiento de imágenes, aprendizaje automático y análisis estadístico para tareas como detección de objetos, seguimiento de movimiento, reconocimiento facial y análisis de escenas en tiempo real. La visión por computadora ha brindado instrumentos que permiten a sistemas con microprocesadores examinar imágenes capturadas por cámaras, procurando obtener datos de relevancia para distintas aplicaciones (Moreo, 2023) .

2.2.3.1. Procesamiento de Imágenes

El procesamiento de imágenes, en la informática e ingeniería, implica modificar y analizar imágenes digitales para mejorar su calidad, obtener información relevante, reconocer patrones y tomar decisiones basadas en contenido visual. Utiliza diversos métodos y algoritmos para trabajar con imágenes digitales. El procesamiento de imágenes se lleva a cabo utilizando imágenes en color, pero estas pueden convertirse a escala de grises, en blanco y negro, para su posterior tratamiento (Cortina, 2023).

2.2.3.2. Extracción de Características

Según Mathworks (2023), menciona que la extracción de características es un paso en el procesamiento de datos y aprendizaje automático que busca identificar las propiedades

más relevantes y distintivas de un conjunto de datos. Estas cualidades transforman los datos originales, resaltan patrones, disminuyen dimensiones y mejoran la eficacia de los algoritmos de análisis y predicción. La extracción de características implica transformar datos crudos en características numéricas, preservando la información del conjunto original para su procesamiento ulterior.

2.2.4. Framework y herramientas de desarrollo

2.2.4.1. Python

Según Jalolov (2023) menciona que Python es un lenguaje de programación interpretado, orientado a objetos, de alto nivel y con semántica dinámica. Sus estructuras de datos integradas y su flexibilidad lo convierten en una elección popular para el desarrollo ágil de aplicaciones y la integración de componentes existentes.

Python es un lenguaje de programación comúnmente empleado en la creación de sitios web, software, automatización de tareas y análisis de datos. Al ser de propósito general, es versátil y aplicable en diversos tipos de programas (coursera, 2024).

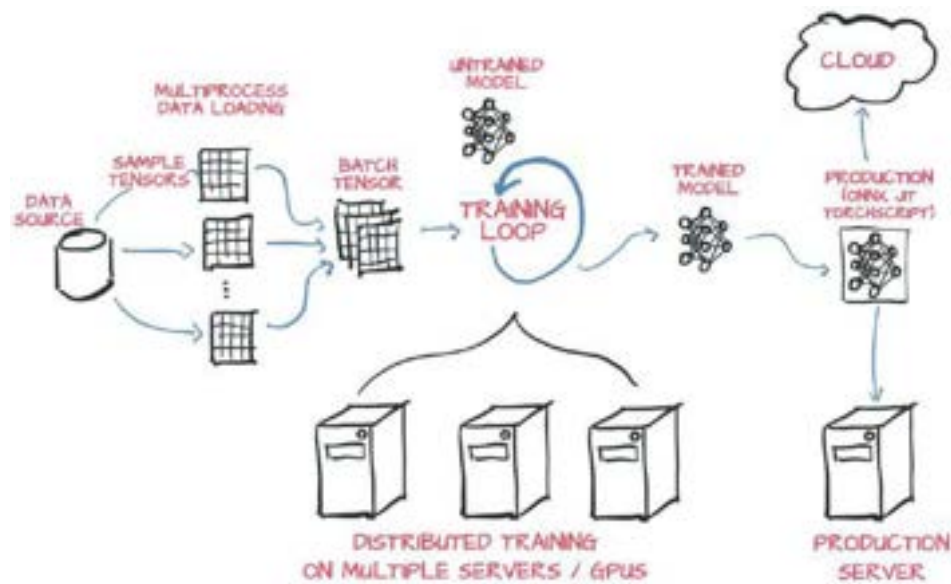
2.2.4.2. Pytorch o tensorflow

PyTorch es una biblioteca de tensores de aprendizaje profundo optimizada basada en Python y pytorch, principalmente para aplicaciones que usan GPU y CPU. Pytorch utiliza gráficos computacionales dinámicos y es completamente Pythonic. Permite a los investigadores, desarrolladores y depuradores de redes neuronales ejecutar y probar fragmentos de código en tiempo real. Por lo tanto, los usuarios no tienen que esperar hasta que se implemente todo el código para verificar si algún código es válido. Pytorch es una biblioteca de Python que facilita las aplicaciones de aprendizaje profundo (Imambi, et al, 2021).

PyTorch, una plataforma integral, simplifica la creación de modelos de aprendizaje profundo, ampliamente utilizados en áreas como el reconocimiento de imágenes y el procesamiento de lenguaje. Basado en Python, ofrece una curva de aprendizaje sencilla y una interacción completa con algoritmos de aprendizaje profundo, siendo una herramienta esencial en el campo del aprendizaje automático. (Orlando et al, 2023).

PyTorch es un pequeño software de computadora basado en la biblioteca Torch. Es un sistema de aprendizaje profundo iniciado por Facebook. PyTorch es una biblioteca de aprendizaje automático para el lenguaje de programación Python que se utiliza para aplicaciones como el procesamiento del lenguaje natural. PyTorch es un marco de aprendizaje profundo combinado con algoritmos de aprendizaje automático (Rozemberczki et al, 2021).

Figura 7. Marco de aprendizaje profunda de Pytorch



Nota. Marco de aprendizaje profunda de Pytorch: Velocidad + usabilidad.

Fuente (sincronizado,2019).

CAPÍTULO III

MATERIAL Y MÉTODOS

3.1. TIPO Y NIVEL DE INVESTIGACIÓN

3.1.1. Tipo de investigación

Según Mousalli y Kayat (2015), en el estudio observacional el investigador se mantiene al margen de los hechos que ocurren y no existe variación intencional de alguna variable para medir su efecto sobre otra si no que se observan los fenómenos tal cual se presentan en su contexto natural (p.30). Por este sustento la investigación será de tipo observacional.

Mousalli y Kayat (2015), en el estudio transversal se recolectan los datos y son medidos en un solo momento, su intención es describirlo o caracterizar el fenómeno a través de la variable incluyendo a la variable de estudio, en un momento dado y estas mediciones tienen el nombre de grupos de muestras independientes (p. 30). Por este sustento la investigación será de tipo transversal.

3.1.2. Nivel de investigación

Según Hernández (2014), los estudios descriptivos se busca especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis. Es decir, únicamente pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren, esto es, su objetivo no es indicar cómo se relacionan éstas y además la meta del investigador consiste en describir fenómenos, situaciones, contextos y sucesos; esto es, detallar cómo son y se manifiestan (p. 92). Por este sustento el nivel de investigación es Descriptiva.

3.2. DISEÑO DE LA INVESTIGACIÓN

Según Hernández (2014), el diseño no experimental se realiza sin la manipulación deliberada de las variables, es decir, se trata de estudios en los que no varían las variables independientes y estas no puedan afectar sobre otras variables. No se genera ninguna situación si no que se observan situaciones ya existentes no son manipuladas por el investigador, las variables independientes ocurren y no es posible manipularlas, no se tiene control directo sobre dichas variables ni se puede influir en ellas, porque ya sucedieron,

al igual que sus efectos (p. 152). Por este sustento, el diseño de investigación es no experimental.

En el diseño transversal se recolectan los datos y son medidos en un solo momento, su intención es describirlo o caracterizar el fenómeno a través de la variable incluyendo a la variable de estudio, en un momento dado y estas mediciones tienen el nombre de grupos de muestras. Este diseño recoge los datos en un solo momento y solo una vez, es como tomar una foto o una radiografía para luego describirlas en la investigación, pueden tener alcances exploratorios, descriptivos y correlaciones (Arias Gonzáles y Covinos Gallardo, 2021, p. 86). Por este sustento la investigación será de tipo transversal.

La investigación se fundamenta en un diseño no experimental, donde las variables no son manipuladas. Emplea un enfoque observacional, analizando fenómenos en su contexto natural. Además, adopta un diseño transversal al medir a los individuos en un único momento.

3.3. VARIABLES

3.3.1. Definición conceptual de variables

3.3.1.1. Variable de interés

A. Modelo machine learning

El modelo de machine learning es una técnica matemática que busca identificar patrones dentro de un conjunto de datos, generalmente derivados de disciplinas como estadística, cálculo y álgebra lineal (Databricks, 2024).

3.3.1.2. Variables descriptivas

B. Tipo de modelo

Un tipo de modelo se define como una categorización o clasificación de un conjunto específico de atributos, estructuras o características que son comunes entre los distintos modelos dentro de un ámbito particular (Rojas, 2023).

C. Arquitectura de modelo

La arquitectura de un modelo describe su disposición y diseño, que abarcan la organización de sus partes internas, las capas de procesamiento, las conexiones entre ellas y los ajustes que regulan su desempeño (Hopsworke, 2024).

D. Eficacia de modelo

La eficacia del modelo se relaciona con su habilidad para alcanzar los objetivos de forma precisa y efectiva, valorando su adecuación a los datos y la obtención de resultados esperados según su propósito y contexto de uso (Castro, 2024).

E. Validez de modelo

La validación del modelo es el proceso de evaluar un modelo entrenado en un conjunto de datos de prueba. Esto asegura la capacidad de generalización del modelo entrenado (Durán, 2022).

3.3.2. Definición operacional de variables

Variable de estudio

X: Modelo machine learning

Variables descriptivas

X1: Tipo de modelo

X2: Arquitectura del modelo

X3: Eficacia del modelo

X4: Validez del modelo

3.4. POBLACIÓN Y MUESTRA

3.4.1. Población

Para Miranda et al. (2016), la población de estudio es un conjunto de casos, definido, limitado y accesible, que formará el referente para la elección de la muestra, y que cumple con una serie de criterios predeterminados. Es necesario aclarar que cuando se habla de población de estudio, el término no se refiere exclusivamente a seres humanos, sino que también puede corresponder a animales, muestras biológicas, expedientes, hospitales, objetos, familias, organizaciones, etc.

La población de estudio está compuesta por 37,000 de placas vehiculares.

3.4.2. Muestra

Según Scharager y Armijo (2001). El muestreo no probabilístico también llamados dirigidas o por conveniencia; La elección de los elementos no depende de la probabilidad

sino de las condiciones que permiten hacer el muestreo, se puede usar este muestreo en poblaciones heterogéneas y cambiantes; también se debe decidir en qué momento se debe recolectar los datos.

Se tomo como muestra 537 imágenes digitales de placas vehiculares para el entrenamiento y procesamiento del modelo machine learning.

3.5. TÉCNICAS E INSTRUMENTOS DE LA INVESTIGACIÓN

3.5.1. Técnicas

- ✓ La fotografía documental se empleó para capturar imágenes de las placas de vehículos utilizando una cámara de celular. Esto permitió recopilar datos visuales de los fenómenos estudiados, facilitando su análisis y utilización posterior en la investigación.

- ✓ El análisis documental se empleó para examinar sistemáticamente documentos relacionados con el modelo de machine learning utilizado en la investigación. Se extrajeron datos esenciales como el tipo de modelo machine learning empleado, la arquitectura específica utilizada, los algoritmos de aprendizaje automático aplicados, los métodos de entrenamiento y validación, y la eficacia y validez del modelo. Además, se revisaron documentos relacionados con investigaciones previas sobre detección de vehículos robados y tendencias de hurto en la región de Ayacucho, lo que proporcionó un marco teórico sólido para el estudio.

3.5.2. Instrumentos

- ✓ Cámara de celular
- ✓ Ficha de análisis documental

3.6. PROCEDIMIENTOS

3.6.1. Técnicas de procesamiento de datos

A. Herramientas tecnológicas

Las herramientas tecnológicas son seleccionadas tomando en cuenta la facilidad y rapidez de su aplicación, se ha seleccionado las tecnologías según la siguiente tabla 3.

Tabla 3. Herramientas tecnológicas

SOFTWARE	FABRICANTE	SERVICIO
Window 10	Microsoft Corporation	Windows ofrece un sistema operativo con interfaz gráfica, administración de archivos y compatibilidad de hardware, simplificando diversas tareas y aplicaciones.
Excel	Microsoft Excel	Para ingresar y gestionar datos inicialmente, además de realizar un análisis estadístico fundamental.
Python (con bibliotecas como Pandas y NumPy)	Pandas	Para la preparación de datos, manipulación avanzada y análisis estadístico.
OpenCV (Open-Source Computer Vision Library)	Intel	Para crear y evaluar modelos de aprendizaje automático.
Scikit-learn (en Python)	Python CC++	Para elaborar modelos de aprendizaje profundo en caso de utilizar redes neuronales.
TensorFlow o PyTorch	Google Brain	Para el desarrollo de modelos de aprendizaje profundo si se utilizan redes neuronales.
Plataformas de Nube (Google Colab, AWS, Azure, etc.)	Amazon AWS y Google Cloud Platform	Para llevar a cabo cálculos intensivos y entrenamiento de modelos en la nube, según sea requerido.
OCR en Python	Google	OCR en Python es la tecnología de reconocimiento óptico de caracteres para convertir imágenes o documentos escaneados en texto editable.

Nota. Tecnologías usadas para procesar la información y los datos.

Fuente (Elaboración propia).

CAPÍTULO IV

RESULTADOS

3.1. RESULTADOS

A. Tipo de modelo machine learning

Tras la preparación de los datos, evaluamos los algoritmos clave de machine learning en visión por computadora para desarrollar un nuevo modelo de detección de placas de vehículos peruanos. Estos algoritmos facilitan la extracción de caracteres de las placas, transformándolos en datos para comparar con nuestra base de datos "bdplacaia", con el fin de mejorar la precisión y eficiencia en la identificación de placas, al integrar la información procesada con los datos almacenados, optimizando así el rendimiento del modelo de detección.

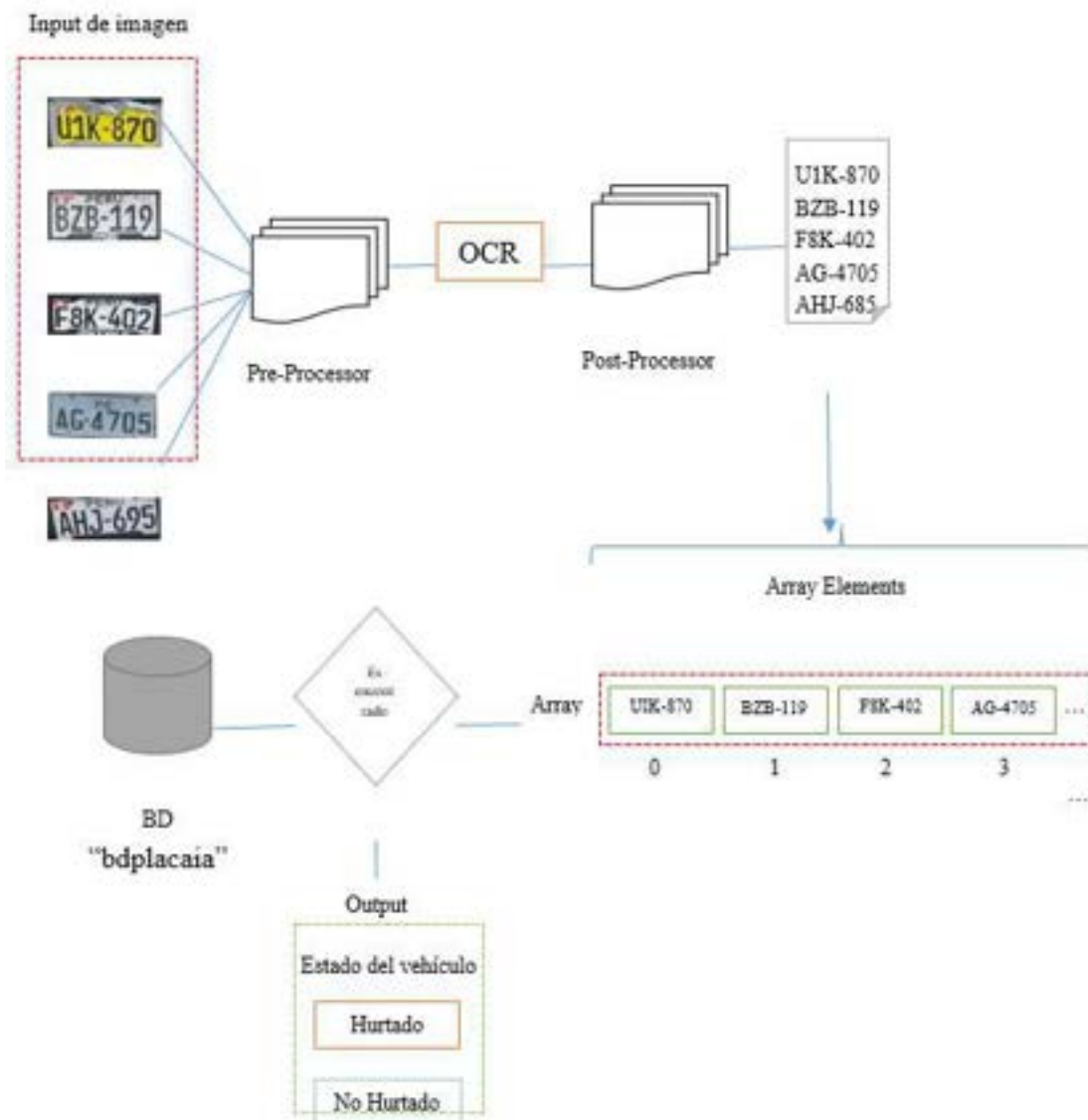
Se inicia el desarrollo de una estructura para detectar placas de vehículos robados en Perú. La primera fase implica crear un modelo específico para identificar estas placas, abordando así el problema de los vehículos robados en el país.

En el proceso, se inicia seleccionando una imagen de una placa vehicular, la cual se convierte en formato binario después de su lectura. Luego, se aplican técnicas de contorno para identificar la placa y se determina el área de los contornos para encontrar el rectángulo que contiene la matrícula. Se discriminan los contornos y se obtiene la proporción de aspecto. Posteriormente, se aplica OCR con Pytesseract para reconocer el texto de la placa. Finalmente, se procede a la visualización de los resultados obtenidos.

B. Arquitectura del modelo machine learning

La figura 8 muestra la estructura de nuestra arquitectura, describiendo las etapas y elementos clave que garantizarán la eficacia en la detección de placas robadas. Este enfoque integral se basa en un modelo diseñado específicamente para abordar el problema de los vehículos robados en Perú.

Figura 8. Estructura de la arquitectura para detectar placas de vehículos hurtados



En la arquitectura propuesta, se observa el enfoque de investigación, con la preparación de datos de imágenes de placas y la información de la base de datos.

En el siguiente paso, ingresaremos las imágenes de las placas vehiculares y las procesaremos utilizando visión computacional (utilizando OpenCV, Pytesseract (OCR) – Python) con algoritmos de aprendizaje automático para extraer caracteres de las imágenes de las placas, como se muestra en la arquitectura, que pueden ser, por ejemplo, U1K-870, BZB-119, F8K-402, AG-4705, AHJ-685, etc. Estos caracteres, una vez extraídos, se colocan en una matriz para luego consultar la base de datos que contiene información sobre el estado del vehículo, el propietario y la licencia del conductor. Al realizar la consulta, devuelve información sobre si el vehículo está registrado como robado o no

(salida). Este es el modelo que se construyó a partir de la arquitectura propuesta para reconocer el estado de robo del vehículo. Además, al finalizar, se muestra una aplicación web desarrollada en esta investigación para detectar el estado de robo del vehículo a partir de una imagen ingresada.

C. Eficacia del modelo machine learning

Para medir la eficacia de nuestro modelo machine learning para la detección de vehículos hurtados se utiliza la siguiente formula:

RA: Resultado alcanzado

RP: Resultado prevista

E: Eficacia

$$E = (RA * 100)/(RP)$$

Figura 9. Prueba de la validez del modelo machine learning para la detección de vehículos hurtados

```

71
72 #print('resultado2: '+numplaca_rec1[1].replace(" ",""))
73 valor1=numplaca_rec1[1].replace(" ","")

PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS

0.6426820755004883
[19/Feb/2024 17:20:19] "POST /addfuncion HTTP/1.1" 200 1498
TIEMPO:
0.5871922969818115
[19/Feb/2024 17:20:38] "POST /addfuncion HTTP/1.1" 200 1498
TIEMPO:
0.7496376037597656
[19/Feb/2024 17:20:56] "POST /addfuncion HTTP/1.1" 200 1498
TIEMPO:
0.7134649753570557
[19/Feb/2024 17:21:07] "POST /addfuncion HTTP/1.1" 200 1498
TIEMPO:
0.6320922374725342
[19/Feb/2024 17:21:24] "POST /addfuncion HTTP/1.1" 200 1498
TIEMPO:
0.6925721168518066
[19/Feb/2024 17:21:44] "POST /addfuncion HTTP/1.1" 200 1498
TIEMPO:
0.676750898361206
[19/Feb/2024 17:21:59] "POST /addfuncion HTTP/1.1" 200 1498
TIEMPO:
0.6899006366729736
[19/Feb/2024 17:22:10] "POST /addfuncion HTTP/1.1" 200 1498

```

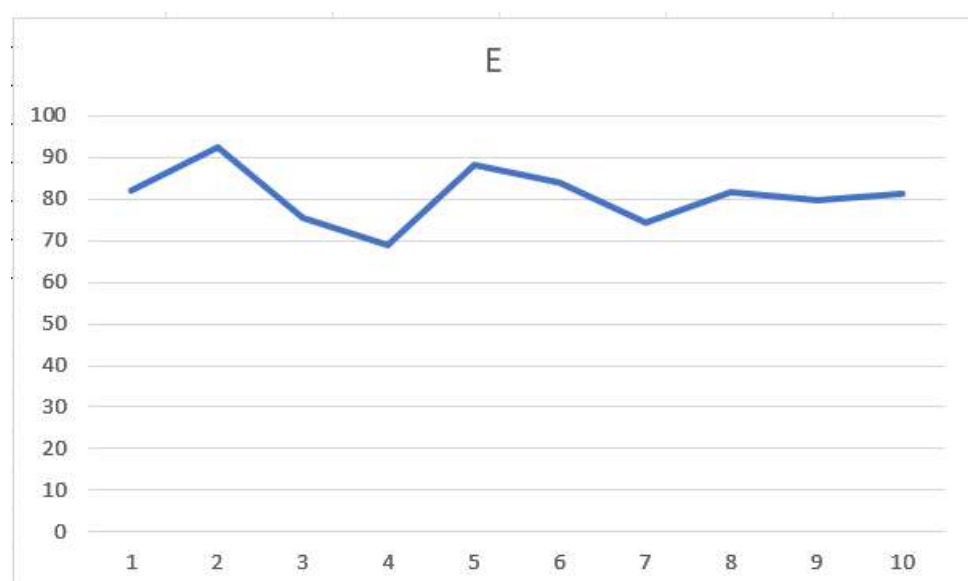
Tabla 4. Resultados de las pruebas de eficacia del modelo de detección de placas de vehículos hurtados

Nº	RA	RP	E
----	----	----	---

PRUEBA N°1	0,697535276	0,85	82,0629737
PRUEBA N°2	0,784631491	0,85	92,3095871
PRUEBA N°3	0,642682076	0,85	75,6096559
PRUEBA N°4	0,587192297	0,85	69,0814467
PRUEBA N°5	0,749637604	0,85	88,1926593
PRUEBA N°7	0,632092237	0,85	74,3637926
PRUEBA N°8	0,692572117	0,85	81,4790726
PRUEBA N°9	0,676750898	0,85	79,6177527
PRUEBA N°10	0,689900637	0,85	81,1647808

En este punto, observamos que nuestra modelo alcanza una eficacia del 92%, lo que representa el rendimiento máximo obtenido. Al graficar estos resultados, se aprecia que, a partir de este punto, la eficacia se mantiene constante como un estándar.

Figura 10. Gráfico de la eficiencia de modelo machine learning para la detección de vehículos hurtados



D. Validez del modelo machine learning

En esta fase, se procede a dividir de forma aleatoria nuestro conjunto de datos en dos partes: un conjunto de entrenamiento y otro de prueba.

El conjunto de entrenamiento consta de 3340 registros de vehículos, actualizados semanalmente a partir de informes de la Policía Nacional. Estos registros incluyen información sobre vehículos tanto hurtados como no hurtados.

Figura 11. Validez del conjunto de datos de entrenamiento

Mostrando filas 0 - 24 (total de 3340, La consulta tardó 0,0005 segundos.)

`SELECT * FROM `libreria_placas``

				id	numero	nombres	num_licencia	clase	categoria	numplaca	imagen	estado	a_fabricac	motor	chasis	chasis2
☐				2572	26313736	RODRIGO DE LA CRUZ ENCINO	Q36413142	A	Des B	RS-1233	na	1	1986	4C31430001	BU700003362	CARSA
☐				2573	70694942	JOHN MARINO ENCO CHAVEZ	342086269	A	Des B	RS-1555	na	1	1991	4C32519403	BU700000608	CARSA
☐				2574	88910395	NELO RODRIGO SIMON NIECA	623651038	A	T095 B	SP-3853	na	1	1984	3L3052179	LH130077689	CARSA
☐				2575	77916395	STALIN YELTISEIN SANCHEZ ARANZA	50389624	A	T095 C	SP-7256	na	1	1992	3L2903001	LH130025803	CARSA

Figura 12. Validez del conjunto de datos prueba

Y tenemos nuestros datos de prueba con un total de 2947 registros de vehículos

Mostrando filas 0 - 24 (total de 2947, La consulta tardó 0,0004 segundos.)

`SELECT * FROM `libreria_placas``

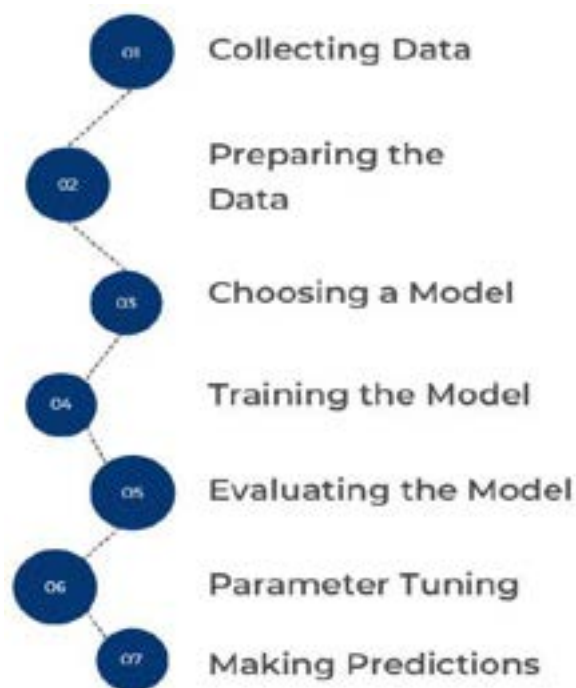
FILTRO														
				numero	nombres	num_licencia	clase	categoria	numplaca	imagen	estado	a_fabricacion	motor	ch
☐				1	70216982	JORGE PEREZ QUEJO			CVL-55718	na	1			
☐				4	25267433	MARCELINO LUIS ANILA MENDOZA			RP-3067	1	1			
☐				2940	70216982	JORGE PEREZ QUEJO	542574471	A	Des B	CVL-55718	imagenes/aut001.jpg	0 1997	3L4258098	3L
☐				2950	25267433	MARCELINO LUIS ANILA MENDOZA	del	A	Des B	RP-3067	1	1 1997	3L4314300	LH
☐				2951	70216982	JORGE PEREZ QUEJO	542574471	A	Des B	RQB-577	na	1 1997	3L4258098	LH
☐				2962	71703771	BREXNER SALTERIO URBANO AGUILAR	541090171	A	Uno	RS-1726	na	1 1997	3L4340632	LH

Esto permite que todos los parámetros sean válidos tras el entrenamiento del modelo, lo que nos proporciona la oportunidad de realizar pruebas reales y evaluar la eficacia del modelo machine learning para la detección de vehículos hurtados. Además, contamos con un total de 537 imágenes de placas vehiculares que se utilizan en el proceso de entrenamiento y pruebas del estado de los vehículos.

En este capítulo, exploraremos estrategias para adquirir, procesar, analizar e interpretar imágenes de matrículas vehiculares peruanas. El propósito es generar datos en forma de variables numéricas y de texto que luego serán procesados por un modelo computacional diseñado para detectar vehículos robados. Concentraremos nuestros esfuerzos en el procesamiento, la percepción y la comprensión de imágenes con el fin de desarrollar un modelo innovador. Este enfoque integral busca mejorar la eficiencia en la identificación de vehículos robados mediante el uso de técnicas avanzadas de procesamiento de imágenes.

Utilizaremos siete pasos descritos por Margüelles (2023) en su artículo de revista como guía para desarrollar nuestro modelo. El artículo proporciona pautas detalladas que seguiremos durante la construcción del modelo.

Figura 13. *Pasos para construir un modelo*



Fuente (elaboración propia).

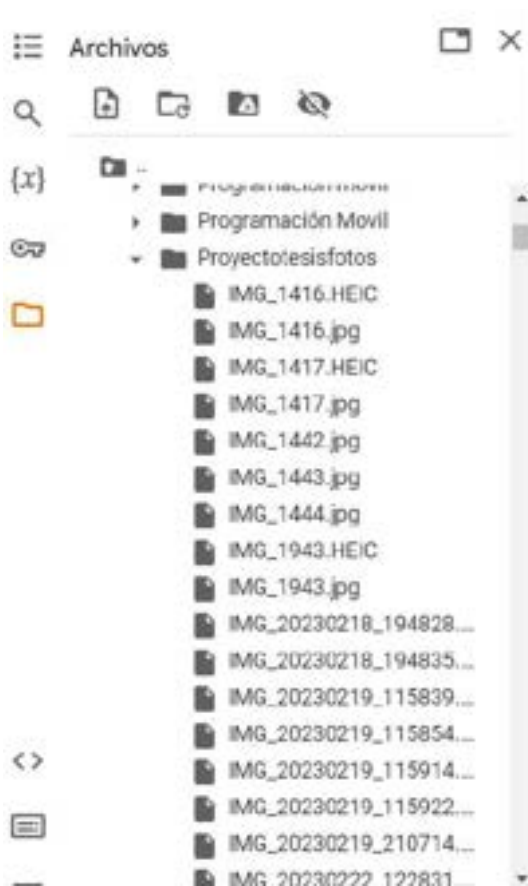
3.1.1. Recopilación de datos

Para recopilar datos, dividiremos en dos grupos: uno de imágenes para el conjunto de datos y otro de datos estructurados para la base de datos. Esta metodología aborda tanto los aspectos visuales como los estructurados, garantizando una recolección completa y equitativa.

a) Dataset de imágenes

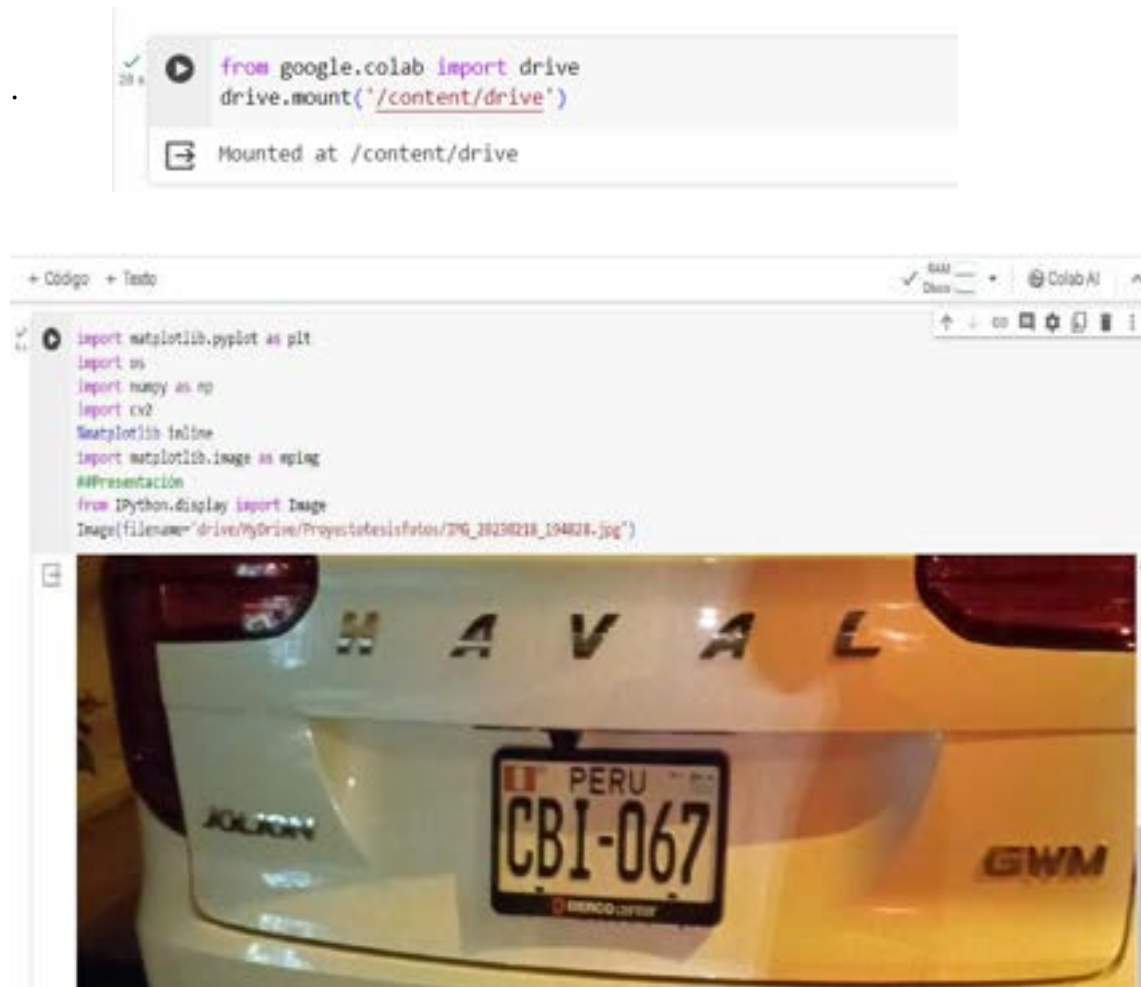
Se obtuvieron datos de placas de vehículos en Ayacucho y Lima, totalizando 537 fotografías. Estas imágenes se almacenaron en Google Drive y se organizaron para su procesamiento. Este proceso incluyó la recopilación de datos visuales de las placas de vehículos en ambas ciudades, seguida de la carga de estas imágenes en el servicio de almacenamiento en la nube. Esto facilitó la preparación y estructuración de la información para su análisis y aplicación en el proyecto.

Figura 14. Información de imágenes tomados de placas vehiculares



Empleamos nuestra herramienta de Colab junto con código Python y algunas bibliotecas para extraer información de nuestras imágenes.

Figura 15. Script de la extracción de información de la placa de un vehículo



Para generar una lista de las primeras 30 imágenes de placas vehiculares, se emplearon códigos Python específicos para el propósito de la investigación

Figura 16. Listado de las primeras 30 imágenes mediante el empleo de los códigos de Python

```
+ Código + Texto
✓ [28] PATH_OF_DATA= 'drive/MyDrive/Proyectotesisfotos'
0s !ls {PATH_OF_DATA}

IMG_1416.HEIC
IMG_1416.jpg
IMG_1417.HEIC
IMG_1417.jpg
IMG_1442.jpg
IMG_1443.jpg
IMG_1444.jpg
IMG_1943.HEIC
IMG_1943.jpg
IMG_20230218_194828.jpg
IMG_20230218_194835.jpg
IMG_20230219_115839.jpg
IMG_20230219_115854.jpg
IMG_20230219_115914.jpg
IMG_20230219_115922.jpg
IMG_20230219_210714.jpg
IMG_20230222_122831.jpg
IMG_20230222_122850.jpg
IMG_20230222_122857.jpg
IMG_20230304_165858.jpg
IMG_20230304_165910.jpg
IMG_20230304_165924.jpg
IMG_20230304_165933.jpg
IMG_20230304_170032.jpg
IMG_20230304_170036.jpg
IMG_20230304_170110.jpg
IMG_20230304_170141.jpg
IMG_20230304_194247.jpg
IMG_20230304_195004.jpg
IMG_20230304_195035.jpg
```

b) Base de datos

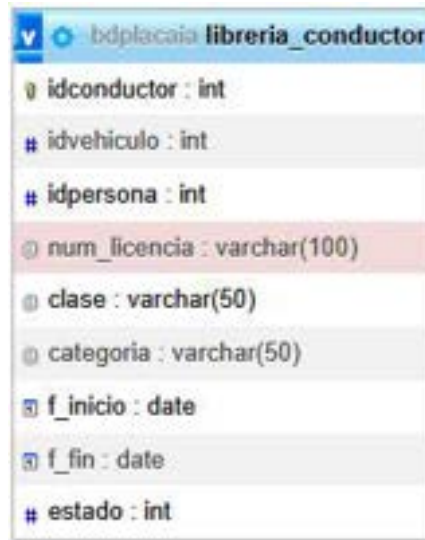
Estableceremos una base de datos enfocada en la identificación de matrículas vehiculares, con la creación de tablas clave como `libreria_conductor`, `libreria_persona`, `libreria_placas`, `libreria_tipodoc` y `libreria_vehiculo`. Estas tablas esenciales facilitan la gestión de datos, permitiendo verificar la presencia de una placa de vehículo en la base de datos. La estructuración de estas tablas proporciona un marco organizado y eficiente para la identificación y administración de información relacionada con conductores, personas, placas, tipos de documentos y vehículos.

• Tabla `librería_conductor`

En esta tabla se documentan datos del conductor, incluyendo variables como `idconductor`, `idvehiculo`, `idpersona`, `num_licencia`, `clase`, `categoría`, `f_inicio`, `f_final` y `estado`. Estas variables encapsulan información vital sobre el conductor, la licencia, el

vehículo y el estado, presentada de manera organizada y estructurada.

Tabla 5. *Tabla librería_conductor de la base de datos “bdplacaia”*



The image shows a screenshot of a database management tool displaying the structure of a table named 'libreria_conductor' within a database named 'bdplacaia'. The table has the following fields:

Field Name	Field Type
idconductor	int
idvehiculo	int
idpersona	int
num_licencia	varchar(100)
clase	varchar(50)
categoria	varchar(50)
f_inicio	date
f_fin	date
estado	int

En este registro, se conservan detalles esenciales sobre los conductores recopilados de la base de datos oficial de la Dirección Regional de Transportes y Comunicaciones de Ayacucho. La información de 2405 conductores se ha extraído y registrado minuciosamente, garantizando una base de datos completa y actualizada. Esto asegura la precisión y la exhaustividad de los detalles relacionados con los conductores al utilizar datos provenientes de la entidad oficial.

Mostrando filas 0 - 24 (total de 2405, La consulta tardó 0,0017 segundos.)

Tabla 6. En la siguiente tabla se muestra los 16 primeros lista de la tabla librería_ conductor

idconductor	idvehiculo	idpersona	num_licencia	clase	categoria	f_inicio	f_fin	estado
2	3	46	dd	A	Dos B	2023-07-15	2023-08-05	1
3	3001	17649	S42574471	A	Dos B	0000-00-00	0000-00-00	1
4	3001	17650	S41990171	A	Uno	0000-00-00	2028-06-18	1
5	3001	15112	S28708047	A	Uno	0000-00-00	0000-00-00	1
6	3001	1504	S80610183	A	Uno	0000-00-00	0000-00-00	1
7	3001	3446	S42030037	A	Dos B	0000-00-00	2025-11-10	1
8	3001	17651	S41355279	A	Dos B	0000-00-00	2026-12-03	1
9	3001	17652	Q10646900	A	Uno	0000-00-00	2031-09-20	1
10	3001	17653	Q70178203	A	Dos B	0000-00-00	2025-03-18	1
11	3001	7921	S07061033	A	TRES C	2023-01-18	2024-01-18	1
12	3001	12183	S80360348	A	TRES C	2023-01-17	2026-01-17	1
13	3001	17654	S47733680	A	Uno	2019-04-12	2029-04-12	1
14	3001	3638	S40648850	A	Dos B	2016-05-25	2025-09-13	1
15	3001	2079	S72663842	A	Dos B	2014-05-07	2024-08-27	1
16	3001	17220	S46809232	A	Dos B	2013-09-13	2024-11-14	1
17	3001	17655	T45989610	A	Uno	2021-05-20	2031-05-20	1

- **Tabla librería_ persona**

En esta tabla, se guardan elementos clave de la persona, como identificación, tipo de documento, número de documento, nombre completo, información de contacto, género y correo electrónico. Estos datos cubren aspectos esenciales de la identidad y el contacto, creando un registro completo y estructurado en la base de datos.

Tabla 7. *Tabla librería_persona de la base de datos “bdplacaia”*



bdplacaia libreria_persona	
🔑	idpersona : int
#	idtipodoc : int
📄	numdoc : varchar(20)
📄	nombrecompleto : varchar(120)
📄	telefono : varchar(15)
📄	direccion : varchar(120)
#	genero : int
📄	email : varchar(50)

Mediante la creación de esta tabla, se guardan los datos obtenidos de la Dirección Regional de Transportes y Comunicaciones de Ayacucho, que abarcan información de 19155 conductores, formando así una base de datos robusta y exhaustiva. Este proceso garantiza la precisión y actualización de los detalles al utilizar información oficial.

Mostrando filas 0 - 24 (total de 19155, La consulta tardó 0,0026 segundos.)

Tabla 8. En la siguiente tabla se muestra los 15 primeros registros de la tabla librería_persona

idpersona	idtipodoc	numdoc	nombrecompleto	telefono	direccion	genero	email
603	1	28267433	MARCELINO LUIS AYALA MENDOZA			0	
602	1	70216952	JORGE PEREZ OZEJO			0	
601	1	71783771	BREKNER BALERIO URBANO AGUILAR			0	
600	1	40819348	PABLO FLORES SULCA			0	
599	1	28269152	ROGER ADRIAN ESPINOZA PARADO			0	
598	1	48429439	ORLANDO ROBERT CURIHUAMAN AYME			0	
597	1	28295024	WILLIAM ROLANDO HEREDIA PALOMINO			0	
596	1	71320979	WALDIR JULIO DE LA CRUZ ALTAMIRANO			0	
595	1	29715418	ROGER LEON RAMIREZ CRUZ			0	
594	1	74096431	RIQUE ROEL HUALLPA NAVARRO			0	
593	1	70151241	BECQUER CANCHO LAURENTE			0	
592	1	41717487	GODOFREDO RODRIGUEZ ATAO			0	
591	1	45365166	FREDY ROMANI JORGE			0	
590	1	74034693	JHONATAN GONZALES SULCA			0	
589	1	48622684	DANY ESPINOZA MONTANO			0	

- **Tabla librería_placas**

En este registro se consignan detalles esenciales, como identificación, licencia, vehículo y sus características, con variables como id, numdoc, nombres, num_licencia, clase, categoría, numplaca, imagen, estado, a_fabricac, motor, chasis, clase_vehic, carrocería, color, placa, modelo_vehic y asiento. Estos datos forman un archivo estructurado y completo para su gestión y análisis efectivos.

Tabla 9. *Tabla librería_placas de la base de datos “bdplacaia”*

bdplacaia libreria_placas	
id	int
numdoc	varchar(200)
nombres	varchar(200)
num_licencia	varchar(20)
clase	varchar(20)
categoria	varchar(20)
numplaca	varchar(100)
imagen	varchar(100)
estado	int
a_fabricac	varchar(20)
motor	varchar(20)
chasis	varchar(20)
clase_vehic	varchar(20)
carroceria	varchar(20)
combustible	varchar(20)
color	varchar(20)
placa_antigua	varchar(20)
marca	varchar(20)
modelo_vehic	varchar(20)
asiento	varchar(20)

En este registro se almacenan datos esenciales de las placas, obtenidos de la base de datos oficial de la Dirección Regional de Transportes y Comunicaciones de Ayacucho. Se incluyen detalles completos de 2947 conductores, asegurando así una base de datos sólida y actualizada con información precisa y detallada sobre las placas y sus propietarios.

Mostrando filas 0 - 24 (total de 2947, La consulta tardó 0,0008 segundos.)

Tabla 10. En la siguiente tabla se muestra los ocho primeros registros de la tabla *librería_placas*

id	asendoc	nombres	num_licencia	clase	categoria	numplaca	imagen	estado	a_fabricac	motor	chasis	cl
1	76216962	JORGE PEREZ OZEJO				CVL65710	null	1				
4	28267433	MARCELINO LUIS AYALA MENDOZA				RP-8957	1	1				
2949	76216962	JORGE PEREZ OZEJO	542574471	A	Dos B	CVL 657 10	imagenes/aut091.jpg	1	1997	3L4298058	3L4298058	CM
2950	28267433	MARCELINO LUIS AYALA MENDOZA	64	A	Dos B	RP-8957	1	1	1997	3L4314306	LH1136153436	CM
2951	76216962	JORGE PEREZ OZEJO	542574471	A	Dos B	RQB-477	null	1	1997	3L4298058	LH1136151340	CM
2952	71183771	BREKNER BALERO URBANO AGUILAR	541956171	A	Uno	RS-1726	null	1	1997	3L4349532	LH1136158130	CM
2953	48819348	PABLO FLORES SULCA	528708647	A	Uno	RE-6849	null	1	1992	3L3119197	LH1136033631	CM
2954	28269152	ROGER ADRIAN ESPINOZA PARADO	580610183	A	Uno	RS-9504	null	1	1993	3L3466115	LH1136046878	CM
2955	48429435	ORLANDO ROBERT	542030637	A	Dos B	RP-9001	null	1	1996	3L4161364	LH1136133229	CM

- **Tabla *librería_tipodoc***

El objetivo de esta tabla es almacenar los diversos tipos de documentos que una persona puede poseer. Incluye variables como *idtipodoc*, *nombre_doc* y *código*, facilitando el registro ordenado y eficaz de información relacionada con distintos documentos disponibles para su asociación con individuos.

Tabla 11. Tabla *librería_tipodoc* de la base de datos “*bdplacaia*”

bdplacaia libreria_tipodoc	
idtipodoc	: int
nombre_doc	: varchar(50)
codigo	: varchar(11)

En la tabla “*librería_tipodoc*” se conserva la información esencial de los tipos de documentos, obtenida de la base de datos oficial de la Dirección Regional de Transportes y Comunicaciones de Ayacucho. Se registran los datos de 5 tipos de documentos en total.

Mostrando filas 0 - 4 (total de 5, La consulta tardó 0,0011 segundos.)

Tabla 12. En esta tabla se muestran los primeros cinco registros

idtipodoc	nombre_doc	codigo
1	DNI	0
2	RUC	0
3	PASAPORTE	0
4	VISA	0
5	CARNET DE EXTRANJERIA	0

- **Tabla librería_vehículo**

La tabla tiene como objetivo principal almacenar datos importantes sobre vehículos, incluyendo variables como idvehículo, idempresa, num_placa, f_inscrip, a_frabricac, motor, chasis, clase_vehic, carrocería, color, placa, modelo_vehic, asiento, ejes, ruedas, peso_netto, peso_seco, carga_util, largo, ancho, alto, estado_vehic y antigüedad_vehic. Estas variables abarcan detalles cruciales sobre la identificación, especificaciones técnicas y estado del vehículo, formando un registro completo y organizado en la base de datos para facilitar el análisis y la gestión eficiente.

Tabla 13. Tabla librería_vehículo de la base de datos “bdplacaia”



Column Name	Data Type
idvehiculo	int
idempresa	int
num_placa	varchar(100)
f_inscrip	date
a_fabricac	varchar(50)
motor	varchar(50)
chasis	varchar(50)
clase_vehic	varchar(50)
carroceria	varchar(20)
combustible	varchar(20)
color	varchar(20)
placa_antigua	varchar(20)
marca	varchar(20)
modelo_vehic	varchar(20)
asiento	varchar(20)
ejes	varchar(20)
ruedas	varchar(20)
peso_neto	varchar(20)
peso_seco	varchar(20)
carga_util	varchar(20)
largo	varchar(20)
ancho	varchar(20)

La tabla librería_vehículo almacena datos obtenidos de la base de datos de la Dirección Regional de Transportes y Comunicaciones de Ayacucho, que incluye 2948 tipos de documentos. Su propósito es facilitar el acceso y uso eficiente de la información vehicular para nuestra aplicación web, asegurando la precisión y completitud de los datos relacionados con los documentos de los vehículos.

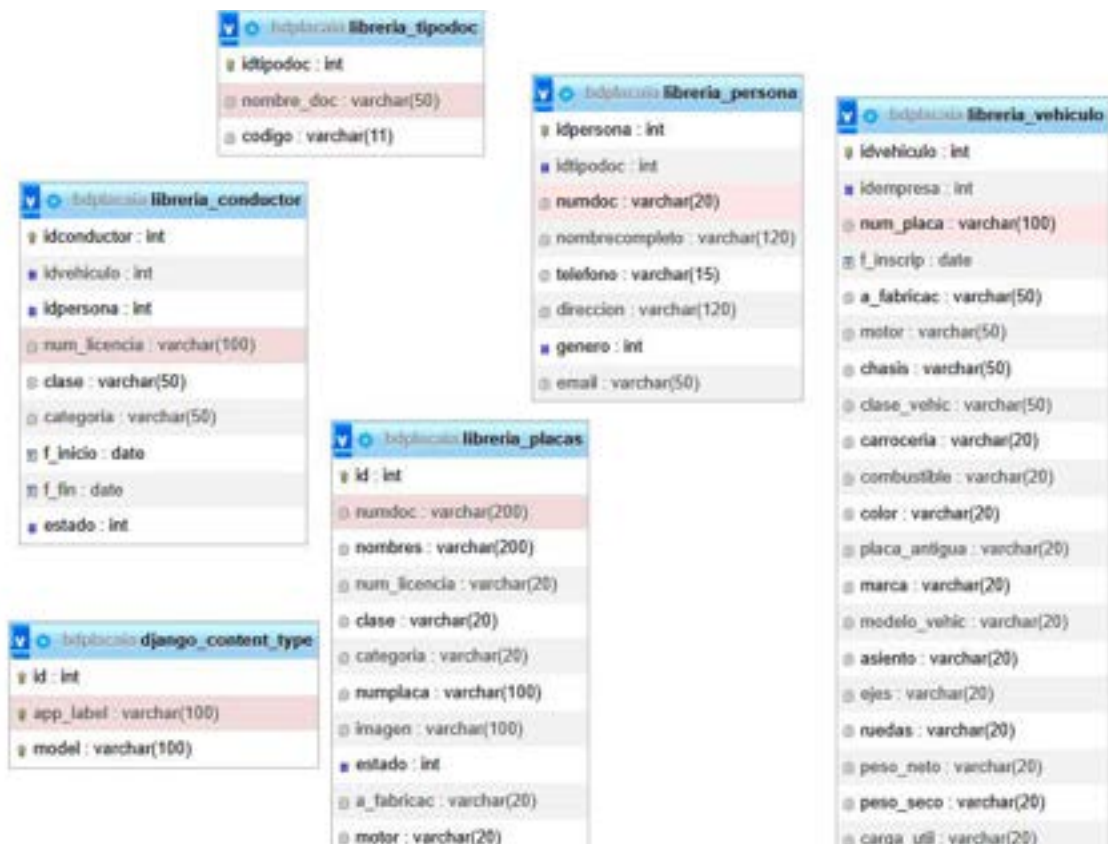
Mostrando filas 0 - 24 (total de 2945, La consulta tardó 0,0020 segundos.)

Tabla 14. Lista de los 10 primeros registros de la tabla *librería_vehículo*

idvehículo	idempresa	num_placa	f_inscrip	a_fabricac	motor	chasis	clase_vehic	carroceria	combustible	color
3	1	RP-8997	0000-00-00	1997	3L4314300	LH1130153436	CMTA.RURAL	METROPOLITANO	PETROLEO	BLANCO CELESTE
4	1	RQB-677	0000-00-00	1997	3L4290095	LH1130151340	CMTA.RURAL	METROPOLITANO	PETROLEO	BLANCO
5	1	RS-1726	0000-00-00	1997	3L4309532	LH1130158130	CMTA.RURAL	METROPOLITANO	PETROLEO	BLANCO
6	1	RE-0549	0000-00-00	1992	3L3119197	LH1130013031	CMTA.RURAL	METROPOLITANO	PETROLEO	BLANCO VERDE
7	1	RS-1684	0000-00-00	1993	3L3456115	LH1130045578	CMTA.RURAL	METROPOLITANO	PETROLEO	BLANCO
8	1	RP-5081	0000-00-00	1996	3L4101304	LH1130133229	CMTA.RURAL	METROPOLITANO	PETROLEO	BLANCO
9	1	CVL-657-18	0000-00-00	1986	4D31498956	FE214E566121	OMNIBUS	METROPOLITANO	PETROLEO	BLANCO VERDE TRES TO
10	1	UD-2540	0000-00-00	1994	4D31A57727	FE440FAE2127	OMNIBUS	METROPOLITANO	PETROLEO	AZUL MULTICOLOR
13	1	RQK-026	0000-00-00	1993	2L3309532	LH1140011510	CMTA.RURAL	METROPOLITANO	PETROLEO	BLANCO
14	1	RS-1441	1000-01-01	1996						

La tabla organiza datos para consultas y análisis precisos de placas, propietarios y estados vehiculares.

Tabla 15. Tabla de base de datos relacional



3.1.2. Preparación de los datos

Durante este proceso, nuestro objetivo es recopilar y limpiar todos los datos, eliminando cualquier información no deseada. Al aplicar el código en nuestra herramienta Google Colab, esperamos obtener los resultados necesarios. Este enfoque asegura la limpieza y organización de los datos, haciendo un uso eficiente de Google Colab para el procesamiento y análisis de la información recolectada.

Tabla 16. Código de procesamiento y análisis de información

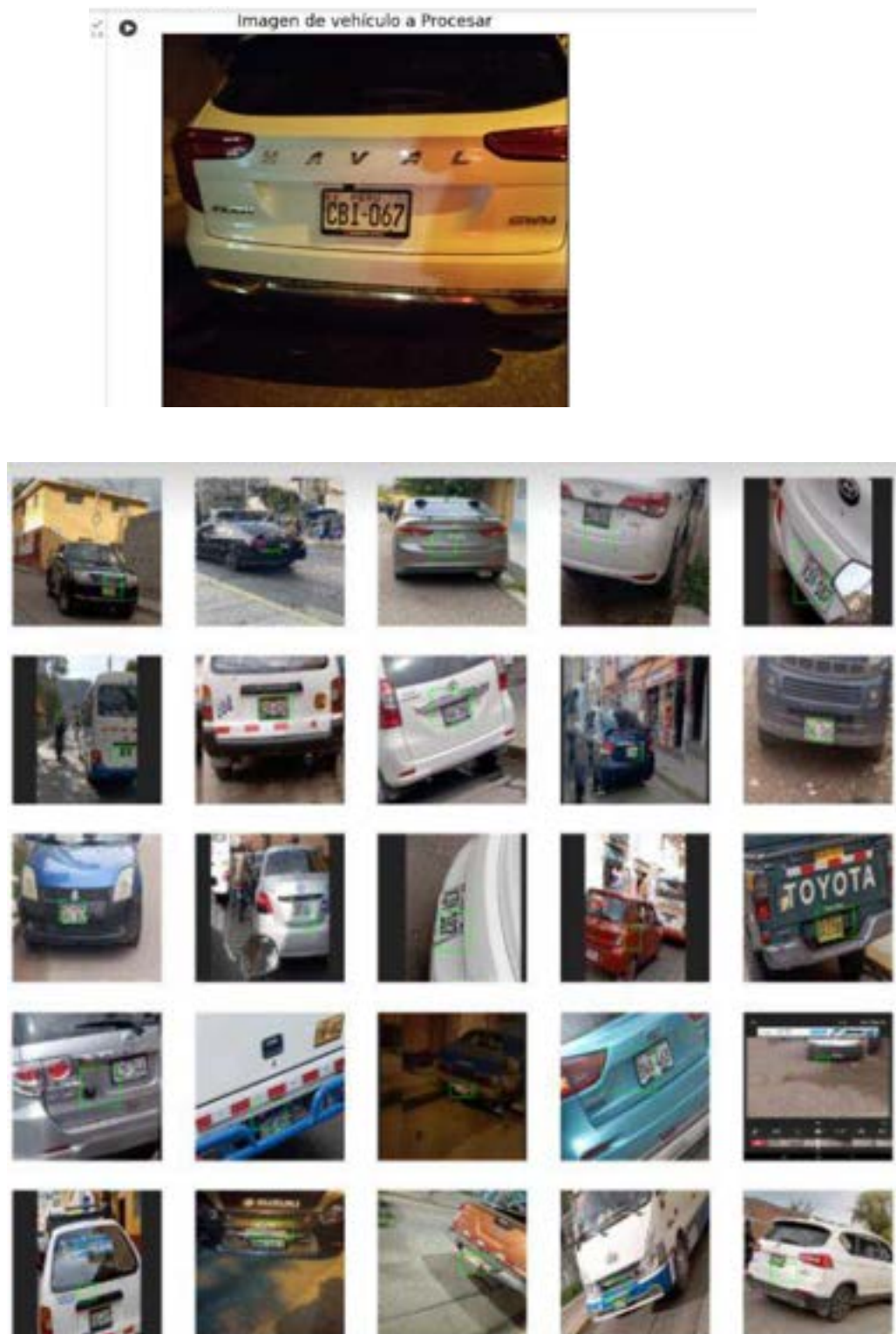
```
+ Código + Texto
import torch.nn.functional as F
from torch.utils.data import DataLoader
from torch.utils.data import sampler
import torchvision.datasets as datasets
import torchvision.transforms as T

def mostrar(imagen, ucm):
    #Hay una diferencia en el ordenamiento de píxeles en OpenCV y Matplotlib.
    #OpenCV sigue el orden BGR, mientras que matplotlib sigue el orden RGB.
    if len(imagen.shape)==3 :
        b,g,r = cv2.split(imagen)      # obtener b,g,r
        imagen = cv2.merge([r,g,b])    # cambiar a r,g,b
    imgplot=plt.imshow(imagen, ucm)
    imgplot.axes.get_xaxis().set_visible(False)
    imgplot.axes.get_yaxis().set_visible(False)

plt.rcParams['figure.figsize'] = 7, 7
# Funcion Principal
plt.title('Imagen de vehiculo a Procesar')
imagen_carro="drive/MyDrive/Proyectotesisfotos/IMG_20230218_194828.jpg"
ejemplocarro=cv2.imread(imagen_carro)
#mostrar(ejemplocarro,None)
#imgplot = plt.imshow(ejemplocarro)
mostrar(ejemplocarro, None)
```

Durante la etapa de preparación de datos de placas de vehículos, se procede a eliminar los archivos que presentan una alta interferencia de píxeles, es decir, las imágenes borrosas. La imagen que resulta tras ejecutar el código previo muestra una calidad óptima. Este proceso es esencial para asegurar la claridad y calidad de las imágenes, lo cual es fundamental para el éxito de futuros análisis y modelos de detección de placas vehiculares. Lo que haremos es mostrar aleatoriamente nuestros 25 primeros datos (25 primeros imagen).

Figura 17. Listado de 25 imágenes aleatorias de vehículos para su procesamiento



Además, en nuestra base de datos, es esencial planificar la inserción de datos en tablas clave como librería_ conductor, librería_ persona, librería_ placas, librería_ tipodoc y librería_ vehículo. Este proceso garantiza la coherencia y organización de los datos en

nuestras diversas tablas.

Tenemos los siguientes scripts de inserciones de cada tabla:

Figura 18. Script de la tabla de librería_ conductor

```
INSERT INTO `libreria_conductor`(`idconductor`, `idvehiculo`, `idpersona`,  
`num_licencia`, `clase`, `categoria`, `f_inicio`, `f_fin`, `estado`) VALUES ([value-  
1],[value-2],[value-3],[value-4],[value-5],[value-6],[value-7],[value-8],[value-  
9])
```

Figura 19. Script de la tabla librería_ persona

```
INSERT INTO `libreria_persona`(`idpersona`, `idtipodoc`, `numdoc`,  
`nombrecompleto`, `telefono`, `direccion`, `genero`, `email`) VALUES ([value-  
1],[value-2],[value-3],[value-4],[value-5],[value-6],[value-7],[value-8])
```

Figura 20. Script de la tabla librería_ placas

```
INSERT INTO `libreria_placas`(`id`, `numdoc`, `nombres`, `num_licencia`, `clase`,  
`categoria`, `numplaca`, `imagen`, `estado`, `a_fabricac`, `motor`, `chasis`,  
`clase_vehic`, `carroceria`, `combustible`, `color`, `placa_antigua`, `marca`,  
`modelo_vehic`, `asiento`) VALUES ([value-1],[value-2],[value-3],[value-  
4],[value-5],[value-6],[value-7],[value-8],[value-9],[value-10],[value-11],[value-  
12],[value-13],[value-14],[value-15],[value-16],[value-17],[value-18],[value-  
19],[value-20])
```

Figura 21. Script de la tabla librería_ tipodoc

```
INSERT INTO `libreria_tipodoc`(`idtipodoc`, `nombre_doc`, `codigo`) VALUES  
([value-1],[value-2],[value-3])
```

Figura 22. Script de la tabla de librería_ vehículo

```
INSERT INTO `libreria_vehiculo`(`idvehiculo`, `idempresa`, `num_placa`, `f_inscrip`,  
`a_fabricac`, `motor`, `chasis`, `clase_vehic`, `carroceria`, `combustible`, `color`,  
`placa_antigua`, `marca`, `modelo_vehic`, `asiento`, `ejes`, `ruedas`, `peso_neto`,
```

```
`peso_seco`, `carga_util`, `largo`, `ancho`, `alto`, `estado_vehic`, `antiguedad_vehic`)
VALUES ([value-1],[value-2],[value-3],[value-4],[value-5],[value-6],[value-
7],[value-8],[value-9],[value-10],[value-11],[value-12],[value-13],[value-
14],[value-15],[value-16],[value-17],[value-18],[value-19],[value-20],[value-
21],[value-22],[value-23],[value-24],[value-25])
```

Luego de introducir los datos esenciales en nuestra base de datos "bdplacaia", es vital realizar la limpieza de campos nulos o vacíos en la columna "numero_placa". Esta operación es esencial, ya que cada vehículo debe tener un número de placa. Para ejecutar esta tarea, empleamos scripts en SQL con los códigos correspondientes, garantizando así la coherencia e integridad de la información almacenada en nuestra base de datos.

Figura 23. Script de la tabla de librería_vehículo

```
DELETE FROM libreria_vehiculo WHERE `num_placa` = NULL
DELETE FROM libreria_vehiculo WHERE `num_placa` = "
```

Figura 24. Script de la tabla de librería_placas

```
DELETE FROM `libreria_placas` WHERE `numplaca` = null
DELETE FROM `libreria_placas` WHERE `numplaca` = "
```

3.2.3. Elegir un modelo

Tras la preparación de los datos, evaluamos algoritmos clave de machine learning en visión por computadora para desarrollar un nuevo modelo de detección de placas de vehículos peruanos. Estos algoritmos facilitan la extracción de caracteres de las placas, transformándolos en datos para comparar con nuestra base de datos "bdplacaia", con el fin de mejorar la precisión y eficiencia en la identificación de placas, al integrar la información procesada con los datos almacenados, optimizando así el rendimiento del modelo de detección.

El modelo fue creado utilizando Python mediante Visual Studio Code. La implementación completa se presenta en el siguiente código, revelando el desarrollo y la estructura detallada del modelo.

Figura 25. Código fuente del modelo de machine learning para detectar la placa de vehículo hurtados

```
from typing import Any
from django.db import models
#plugin de reconocimiento
import cv2
import pytesseract
from pathlib import Path
pytesseract.pytesseract.tesseract_cmd = r'C:/Program Files/Tesseract-OCR/tesseract'

class Placas(models.Model):
    id=models.AutoField(primary_key=True)
    numdoc=models.CharField(max_length=200,verbose_name='N° DNI')
    nombres=models.CharField(max_length=200,verbose_name='Nombres')
    num_licencia = models.CharField(max_length=20)
    clase = models.CharField(max_length=20)
    categoria = models.CharField(max_length=20)
    numplaca=models.CharField(max_length=100,verbose_name='N° placa')
    imagen=models.ImageField(upload_to='imagenes/',verbose_name='Fotos',null=True)
    estado=models.IntegerField(verbose_name='Estado',null=True)
    a_fabricac = models.CharField(max_length=20)
    motor = models.CharField(max_length=20)
    chasis = models.CharField(max_length=20)
    clase_vehic = models.CharField(max_length=20)
    carroceria = models.CharField(max_length=20)
    combustible = models.CharField(max_length=20)
    color = models.CharField(max_length=20)
    placa_antigua = models.CharField(max_length=20)
    marca = models.CharField(max_length=20)
    modelo_vehic = models.CharField(max_length=20)
    asiento = models.CharField(max_length=20)
    def __str__(self):
        fila="N° DNI: "+self.numdoc+" - "+"Nombre: "+self.nombres
```

```

        return fila

    def delete(self, using=None, keep_parents=False):
        self.imagen.storage.delete(self.imagen.name)
        super().delete()

    def reconocer(self):
        placa = []
        image =
cv2.imread('D:/PROYECTO/liz/sysplaca/libreria/templates/placas/auto004.jpg')
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        gray = cv2.blur(gray,(3,3))
        canny = cv2.Canny(gray,150,200)
        canny = cv2.dilate(canny,None,iterations=1)
        cnts,_ =
cv2.findContours(canny,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
        for c in cnts:
            area = cv2.contourArea(c)
            x,y,w,h = cv2.boundingRect(c)
            epsilon = 0.09*cv2.arcLength(c,True)
            approx = cv2.approxPolyDP(c,epsilon,True)

            if len(approx)==4 and area>9000:
                aspect_ratio = float(w)/h
                if aspect_ratio>2.4:
                    placa = gray[y:y+h,x:x+w]
                    text = pytesseract.image_to_string(placa,config='--psm 11')

                    numplaca=text
                    lista = [numplaca,text]
        return lista

class Tipodoc(models.Model):
    idtipodoc = models.AutoField(primary_key=True)
    nombre_doc = models.CharField(max_length=50)
    codigo = models.CharField(max_length=11)

```

```

class Persona(models.Model):
    idpersona = models.AutoField(primary_key=True)
    idtipodoc = models.IntegerField()
    numdoc = models.CharField(max_length=20)
    nombrecompleto = models.CharField(max_length=120)
    telefono = models.CharField(max_length=15)
    direccion = models.CharField(max_length=120)
    genero = models.IntegerField()
    email = models.CharField(max_length=50)

    def __str__(self):
        fila="Nº DNI: "+self.numdoc+" - "+" Nombre: "+self.nombrecompleto
        return fila

class Vehiculo(models.Model):
    idvehiculo = models.AutoField(primary_key=True)
    idempresa = models.IntegerField(blank=True, null=True)
    num_placa = models.CharField(max_length=100)
    f_inscrip = models.DateField()
    a_fabricac = models.CharField(max_length=50)
    motor = models.CharField(max_length=50)
    chasis = models.CharField(max_length=50)
    clase_vehic = models.CharField(max_length=50)
    carroceria = models.CharField(max_length=20)
    combustible = models.CharField(max_length=20)
    color = models.CharField(max_length=20)
    placa_antigua = models.CharField(max_length=20)
    marca = models.CharField(max_length=20)
    modelo_vehic = models.CharField(max_length=20)
    asiento = models.CharField(max_length=20)
    ejes = models.CharField(max_length=20)
    ruedas = models.CharField(max_length=20)
    peso_netto = models.CharField(max_length=20)

```

```

peso_seco = models.CharField(max_length=20)
carga_util = models.CharField(max_length=20)
largo = models.CharField(max_length=20)
ancho = models.CharField(max_length=20)
alto = models.CharField(max_length=20)
estado_vehic = models.CharField(max_length=20)
antiguedad_vehic = models.CharField(max_length=20)

def __str__(self):
    fila="N° DNI: "+self.marca+" - "+" asiento: "+self.asiento
    return fila

def reconocer2(self):
    placa = []
    image =
cv2.imread('D:/PROYECTO/liz/sysplaca/libreria/templates/placas/auto001.jpg')

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    gray = cv2.blur(gray,(3,3))
    canny = cv2.Canny(gray,150,200)
    canny = cv2.dilate(canny,None,iterations=1)
    cnts,_ =
cv2.findContours(canny,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
    for c in cnts:
        area = cv2.contourArea(c)
        x,y,w,h = cv2.boundingRect(c)
        epsilon = 0.09*cv2.arcLength(c,True)
        approx = cv2.approxPolyDP(c,epsilon,True)

        if len(approx)==4 and area>9000:
            aspect_ratio = float(w)/h
            if aspect_ratio>2.4:
                placa = gray[y:y+h,x:x+w]
                text = pytesseract.image_to_string(placa,config='--psm 11')

```

```

    lista = [text]
    return text

class Conductor(models.Model):
    idconductor = models.AutoField(primary_key=True)
    idvehiculo = models.IntegerField()
    idpersona = models.IntegerField()
    num_licencia = models.CharField(max_length=100)
    clase = models.CharField(max_length=50)
    categoria = models.CharField(max_length=50)
    f_inicio = models.DateField()
    f_fin = models.DateField()
    estado = models.IntegerField()

```

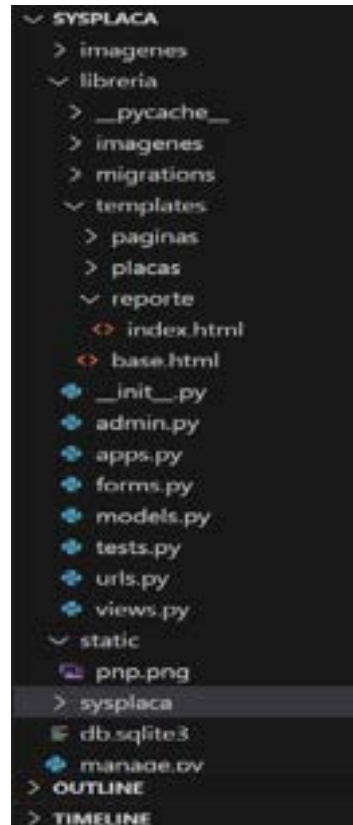
El código fuente de nuestro modelo muestra que su objetivo es identificar placas, extraer caracteres usando el algoritmo OCR y luego compararlos con la base de datos para determinar el estado de hurto del vehículo. El OCR es vital para extraer los alfanuméricos de las imágenes de placas, lo que facilita las operaciones mencionadas con precisión y eficacia.

3.2.4. Entrenando el modelo

En esta fase, nos adentraremos en el entrenamiento de nuestro modelo, creado según la arquitectura propuesta. A través del código en Python previamente presentado, el modelo examina patrones. Específicamente, se ocupa de extraer caracteres de imágenes de placas vehiculares peruanas y busca coincidencias en nuestra base de datos. Este proceso de entrenamiento permite al modelo perfeccionar su capacidad de reconocimiento, mejorando la precisión y eficacia en la identificación de placas y su correspondencia con la información almacenada.

Antes de iniciar el entrenamiento de nuestro código fuente, primero presentaremos el proyecto desarrollado en Django y Python.

Figura 26. Estructura del Proyecto en Django y Python antes del entrenamiento



Con el fin de llevar a cabo el entrenamiento, hemos creado un archivo llamado test.py dentro de nuestro proyecto sysplaca. En este archivo, se aplicó nuestro modelo de OCR para reconocer los caracteres de una imagen de una placa vehicular. A continuación, se presenta el código fuente que utiliza OpenCV, Pytesseract (OCR) y Python, desempeñando un papel fundamental en el desarrollo y mejora de nuestro modelo de detección de placas.

Figura 27. Código fuente para reconocer los caracteres de una imagen de placa vehicular

```
import cv2
import pytesseract
from pathlib import Path

pytesseract.pytesseract.tesseract_cmd = r'C:/Program Files/Tesseract-OCR/tesseract'
placa = []
```

```

image =
cv2.imread('D:/PROYECTO/liz/sysplaca/libreria/templates/placas/auto004.jpg')

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray = cv2.blur(gray,(3,3))
canny = cv2.Canny(gray,150,200)
canny = cv2.dilate(canny, None, iterations=1)
cnts, _ = cv2.findContours(canny, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)

for c in cnts:
    area = cv2.contourArea(c)
    x,y,w,h = cv2.boundingRect(c)
    epsilon = 0.09*cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, epsilon, True)
    if len(approx)==4 and area>9000:
        print('area=', area)
        aspect_ratio = float(w)/h
        if aspect_ratio>2.4:
            placa = gray[y:y+h, x:x+w]
            text = pytesseract.image_to_string(placa, config='--psm 11')
            cv2.imshow('PLACA', placa)
            cv2.moveWindow('PLACA', 780, 10)
            cv2.rectangle(image, (x,y), (x+w,y+h), (0,255,0), 3)
            cv2.putText(image, text, (x-20,y-10), 1, 2.2, (0,255,0), 3)

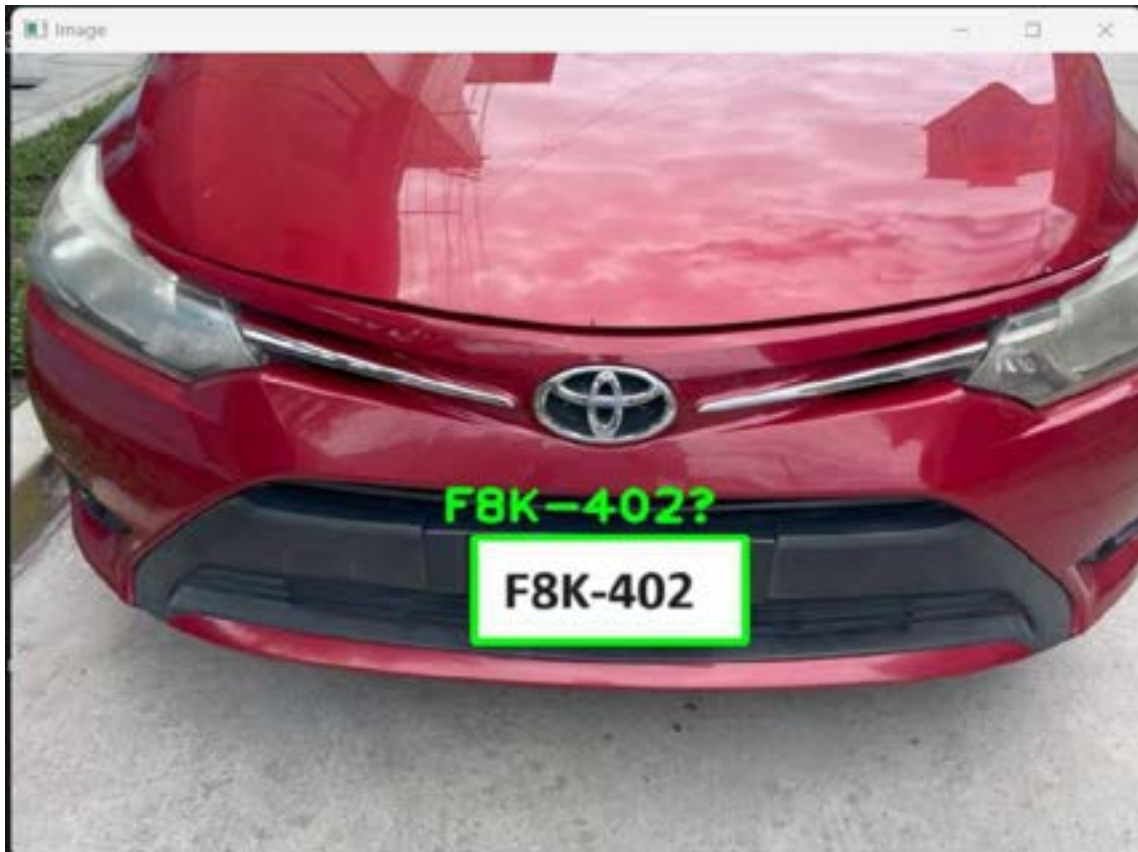
numplaca=text
print(numplaca)
cv2.imshow('Image', image)
cv2.moveWindow('Image', 45, 10)
cv2.waitKey(0)

```

Como se puede apreciar en la imagen que sigue, el reconocimiento óptico de caracteres (OCR), junto con la visión por computadora, identifica con precisión la placa del

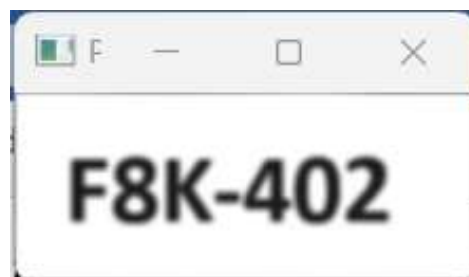
vehículo, resaltándola en verde y luego extrayendo los caracteres alfanuméricos. Esta secuencia completa demuestra la capacidad del OCR para identificar y procesar los detalles específicos de las placas vehiculares, lo que mejora la efectividad en la adquisición de información relevante.

Figura 28. Reconocimiento de carácter de la placa vehicular



En la siguiente imagen lo que se muestra es solamente el carácter extraído de la imagen.

Figura 29. Carácter extraído de la imagen de la placa vehicular



Además, en la imagen que sigue, se muestra la salida del compilador de nuestro modelo,

que extrae exclusivamente los caracteres de la placa, como "F8K-402". Este resultado es esencial, ya que facilita la inclusión de estos caracteres en un conjunto de datos, lo que simplifica la aplicación de filtros en nuestra base de datos. Este enfoque permite verificar el estado del vehículo asociado a la placa identificada, mejorando la eficacia del sistema de detección de placas.

Figura 30. Salida del compilador de nuestro modelo machine learning para la detección de placas de vehículos hurtados



```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS
Python Debug Console

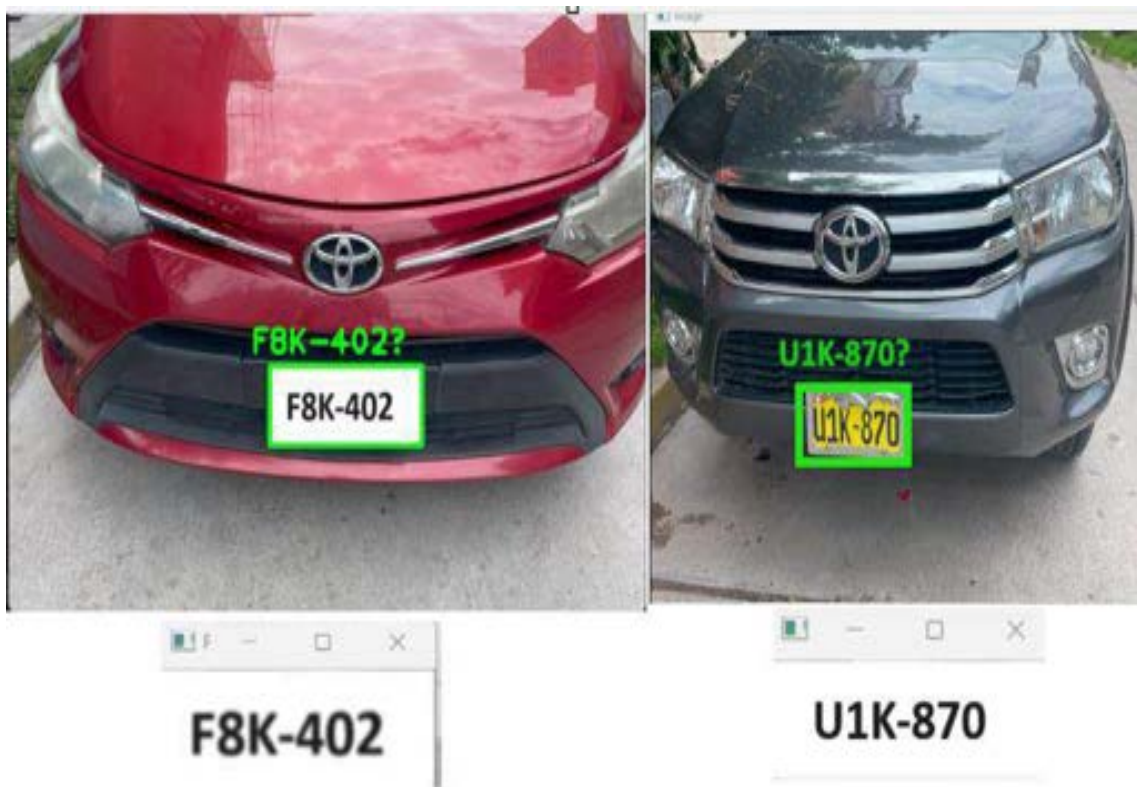
Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS D:\PROYECTO\lir\sysplaca> & 'C:\Users\kiser\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\kiser\.vscode\extensions\ms-python.python-2023.12.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '54968' '--' 'D:\PROYECTO\lir\sysplaca\libreria\templates\placas\test.py'
area= 11841.8
F8K-402
```

3.2.5. Evaluación del modelo

Con nuestro modelo debidamente entrenado, procederemos a evaluar su desempeño utilizando otras imágenes para comprobar su efectividad en la identificación de placas. Se han seleccionado dos imágenes particulares cuyos resultados se detallan a continuación. Esta evaluación nos permitirá examinar la precisión y la capacidad del modelo en diferentes escenarios, proporcionando información relevante sobre su rendimiento y su habilidad general para identificar placas vehiculares.

Figura 31. Evaluación del desempeño utilizando dos imágenes de placas vehiculares



Como se ilustra en las imágenes, se logró detectar satisfactoriamente las placas de dos vehículos, identificadas como F8K-402 y U1K-870.

3.2.6. Ajuste de parámetros

En este paso, incorporamos las clases y vistas necesarias para simplificar la interacción con la aplicación web de reconocimiento de placas vehiculares. Como se mencionó previamente, la aplicación web se ha desarrollado utilizando Django. Este proceso involucra la conexión con la base de datos "sysplacaia", asegurando una integración efectiva entre la interfaz web y la información almacenada.

Figura 32. Script de la aplicación web de reconocimiento de placas vehiculares

```
# Database
# https://docs.djangoproject.com/en/5.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
```

```

    'NAME': 'bdplacaia',
    'USER': 'root',
    'PASSWORD': '',
    'HOST': 'localhost',
    'PORT': '3306',
    "OPTIONS": {
        'init_command': "SET sql_mode='STRICT_TRANS_TABLES',
innodb_strict_mode=1",
        'charset': 'utf8mb4',
        "autocommit": True,
    }
}
}

```

En el ámbito de la investigación mencionada, la clase `views.py` es un componente vital de los proyectos web desarrollados en Django. Encargada de gestionar solicitudes, procesar datos y mostrar resultados, esta clase asegura una interacción eficaz entre el usuario y la aplicación, ofreciendo una experiencia fluida y funcional.

Figura 33. Clase *views.py*

```

from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect
from .models import Placas
from .models import Persona
from .models import Tipodoc
from .models import Vehiculo
from .models import Conductor
from .forms import PlacasForm
from django import template

import unicodecode

# Create your views here.

```

```

from django.views.decorators.csrf import csrf_exempt

def inicio(request):
    return HttpResponse("<h1>bienvenido LIZBETH</h1>")

def nosotros(request):
    return render(request, 'paginas/nosotros.html')

def placas(request, *args, **kwargs):
    placas=Placas.objects.all()
    return render(request, 'placas/index.html',
        {'placas':placas
        }
    )

def reporte(request):
    placas=Placas.objects.filter(estado=0)

    return render(request, 'reporte/index.html',
        {'placas':placas}
    )

@csrf_exempt
def addlibreria_placas(request, *args, **kwargs):
    numdoc=request.POST.get("numdoc")
    nombres=request.POST.get("nombres")
    numplaca=request.POST.get("numplaca")
    #imagen=request.POST.get("imagen")
    estado=request.POST.get("estado")

    formulario=PlacasForm(request.POST or None,request.FILES or None)
    if formulario.is_valid():
        #formulario.save()
        numdoc = formulario.cleaned_data.get('numdoc')
        nombres = formulario.cleaned_data.get('nombres')

```

```

    numplaca = formulario.cleaned_data.get('numplaca')
    imagen = formulario.cleaned_data.get('imagen')
    numdoc = formulario.cleaned_data.get('numdoc')
    estado = formulario.cleaned_data.get('estado')

    #Placas.objects.create(numdoc=numdoc,nombres=nombres,numplaca=numplaca,i
magen=imagen,estado=estado)
    numplacax=numplaca
    #inicio enviar a la funcion reconomiento de placa

    objplacas=Placas()
    numplaca_rec1=objplacas.reconocer()

    objvehiculo=Vehiculo()
    numplaca_rec2=objvehiculo.reconocer2()
    #consultabd=Placas.objects.filter(numplaca=numplaca_rec[1])
    numpla=[numplaca_rec2,'CVL 657 18']

    print('resultado1: '+numpla[0])
    datavehiculobd=Vehiculo.objects.filter(num_placa="+numpla[0])

    print('resultado2: '+numplaca_rec1[1].replace(" ",""))
    valor1=numplaca_rec1[1].replace(" ","")
    consultabd=Placas.objects.filter(numplaca=numplaca_rec1[1])

    #Placas.objects.create(numdoc=numdoc,nombres=nombres,numplaca=numplaca_rec
1[1],estado=estado)

    #fin de la funcion reconocimiento
    return render(request,'placas/divform.html',{
        'consultabd':consultabd

    })
def crear_placa(request):
    formulario=PlacasForm(request.POST or None)

```

```

print(render(request, 'placas/crear.html',{'formulario':formulario}))

def editar_placa(request):
    return render(request, 'placas/editar.html')
# Create your views here.

```

Después de extraer los caracteres de la placa del modelo, los guardamos en una matriz, por ejemplo, `valor = [numplaca1, numplaca2, numplaca3]`. Esta matriz se envía al controlador desde la vista, donde se llevan a cabo las siguientes operaciones: si la placa no está presente en la base de datos, se guarda; si está presente, se verifica el estado del vehículo, devolviendo información detallada como el estado (si está robado o no), número de placa, motor, modelo, color, nombre del propietario, número de DNI, número de licencia, clase y tipo de vehículo. Esta clase también maneja todas las vistas en nuestra aplicación web.

Figura 34. Plantilla de index.html de la carpeta placa

```

{% extends "base.html" %}
{% block titulo %} SISTRECPLACA {% endblock %}
{% block contenido %}
<h1>REPORTE DE VEHÍCULOS HURTADOS</h1>
<div class="card">
    <div class="card-header">Listar placas Reconocidos</div>
    <div class="card-body">
        <h4 class="card-title">placas</h4>

        <div class="table-responsive">
            <table class="table table-primary">
                <thead>
                    <tr>
                        <th scope="col">Nº</th>
                        <th scope="col">Nº DE PLACA</th>
                        <th scope="col">DNI - NOMBRE COMPLETO</th>
                        <th scope="col">ESTADO</th>

```

```

        <th scope="col">PLACA IMG</th>
    </tr>
</thead>
<tbody>
    {% for placas in placas %}
    <tr>
        <td scope="row">{{ forloop.counter }}</td>
        <td>{{ placas.numplaca }}</td>
        <td>{{ placas.numdoc }} - {{ placas.nombres }}</td>
        <td>
            {% if placas.estado == 0 %}
                <span class="badge text-bg-danger">HURTADO</span>
            {% elif placas.estado == 1 %}
                <span class="badge text-bg-primary">NO HURTADO</span>
            {% else %}
                <span class="badge text-bg-light">OTRO</span>
            {% endif %}
        </td>
        <td>
            
        </td>
    </tr>
    {% endfor %}
</tbody>
</table>
</div>
<div>
    <div class="card-footer text-muted">Footer</div>
</div>
{% endblock %}

```

La plantilla de "index.html" en la carpeta "placa" es un archivo fundamental dentro de la estructura de un proyecto web. Este archivo define la estructura y el contenido inicial de la página principal de un sitio web relacionado con el reconocimiento de placas vehiculares. En este contexto, la plantilla index.html en la carpeta placa incluye elementos

Figura 36. Reconocimiento de la placa del vehículo "carro4.jpg"



La búsqueda en la base de datos proporciona un análisis exhaustivo del estado del vehículo vinculado a la placa identificada, ofreciendo una amplia gama de detalles, como el número de placa, motor, chasis, modelo, marca, color, número de DNI del propietario, nombre completo del dueño del vehículo, número de licencia, clase y categoría. Un ejemplo ilustrativo del resultado obtenido de una imagen demuestra la claridad y la integridad de la información del vehículo extraída de la base de datos.

Tabla 17. Resultado obtenido del estado de la placa vehicular "carro4.jpg"

DESCRIPCIÓN	RESULTADO
ESTADO:	NO HURTADO
N° DE PLACA:	F8K-402
MOTOR – CHASIS:	M9TA876C032001 – 93YMEH4KEHJ453080
MODELO – MARCA:	HIACE – TOYOTA
COLOR:	ROJO
N° DNI – DUEÑO DEL VEHÍCULO:	48100939 – LOAIZA HUAITA NORMAM ALAYING
N° LIC. – CLASE Y CATEGORIA:	T48989610 – A Uno

Figura 37. Resultado de la búsqueda de la placa de vehículo "carro4.jpg"

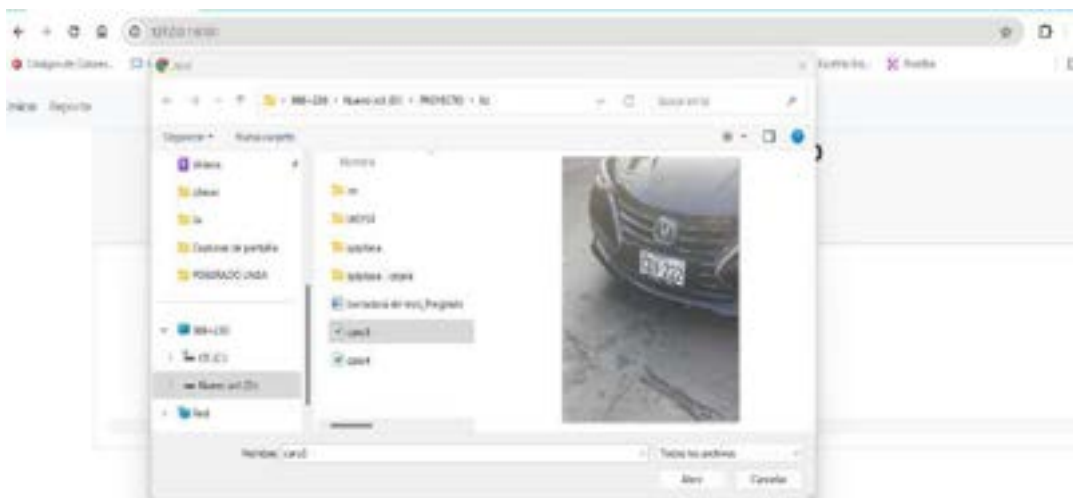
The screenshot shows a web application interface for vehicle license plate recognition. At the top, it says 'RECONOCIMIENTO DE PLACAS DE VEHÍCULO' with a police emblem. Below this is a 'Seleccionar imagen' section where 'Seleccionar archivo' is set to 'carro4.jpg' and a 'Consultar' button is present. The main section, 'RESULTADO DE LA BUSQUEDA', contains a table with the following data:

DESCRIPCIÓN	RESULTADO
USUARIO	NO HURTADO
Nº DE PLACA	YBK-402
MOTOR - CHASSI	MITSUBISHI 2001 - SIMMONDS HOLLAND
MODELO - MARCA	HACE - TOYOTA
COLOR	ROJO
Nº DNI - DUEÑO DEL VEHÍCULO	4838203 - LIDIA HUITA NORMAN ALARIN
Nº DIC - CLASE Y CATEGORÍA	24588883 - A. Uno

Según los datos obtenidos de nuestra base de datos, la imagen indica que el vehículo no está registrado como hurtado, ofreciendo una conclusión definida sobre su estado actual.

Vamos a realizar la próxima evaluación utilizando otra imagen, "carro3.jpeg". Para ello, comenzaremos seleccionando la imagen desde nuestro dispositivo local.

Figura 38. Selección de la imagen denominada "carro3.jpg" para su posterior importación



Después de importar la imagen "carro3.jpeg", como se muestra en la siguiente ilustración,

procedemos haciendo clic en el botón de consulta para visualizar el estado del vehículo.

Figura 39. Reconocimiento de la placa del vehículo "carro3.jpg"



Al realizar la consulta en la base de datos, se obtiene información exhaustiva sobre el estado del vehículo, que incluye detalles como el número de placa, motor, chasis, modelo, marca, color, número de DNI del propietario, nombre completo del dueño del vehículo, número de licencia, clase y categoría. Analicemos un ejemplo representativo del resultado obtenido a partir de una imagen, proporcionando una visión detallada y completa de la información del vehículo recuperada, presentada de manera clara y comprensible para el usuario.

Tabla 18. Resultado obtenido del estado de la placa vehicular "carro3.jpg"

DESCRIPCIÓN	RESULTADO
ESTADO:	HURTADO
N° DE PLACA:	CDV-222
MOTOR – CHASIS:	1KD2849223 - JTFST22P4K0039450
MODELO – MARCA:	DIESEL – DIESEL
COLOR:	NEGRO
N° DNI – DUEÑO DEL VEHÍCULO:	72093232 - DANNY BERNAOLA CURO
N° LIC. – CLASE Y CATEGORIA:	S42574471 - A Uno

De acuerdo con la imagen, se indica que el vehículo fue reportado como robado según la información obtenida de nuestra base de datos.

Figura 40. Resultado de la búsqueda de la placa de vehículo "carro3.jpg

InicioReporte

RECONOCIMIENTO DE PLACAS DE VEHÍCULO



Seleccionar Imagen

Seleccionar archivo: carro3.jpg

Consultar

RESULTADO DE LA BUSQUEDA

DESCRIPCIÓN	RESULTADO
ESTADO:	ROBADO
N° DE PLACA:	CDV-220
MOTOR - CHASIS:	14ED2H42C10 - 875T224H4003400
MODELO - MARCA:	CRUEL - CRUEL
COLOR:	NEGRO
N° DNI - DUEÑO DEL VEHICULO:	72863232 - CHAMAN BERNALDI - DURO
N° LIC - CLASE Y CATEGORIA:	SALES/NETT - A Uno

Además, contamos con una función de generación de informes que proporciona un análisis exhaustivo de todos los vehículos catalogados como robados. Este informe incluye detalles como el número de placa, el número de identificación del propietario, el nombre completo del propietario, el estado del vehículo y una imagen de la placa capturada. Esta funcionalidad ofrece una visión completa y organizada de los vehículos identificados como robados, facilitando la gestión y seguimiento de esta información vital.

Figura 41. Reporte de vehículos hurtados

Inicio Reporte

REPORTE DE VEHÍCULOS HURTADOS

Listar placas reconocidas

placas

N°	N° DE PLACA	DNI - NOMBRE COMPLETO	ESTADO	PLACA IMG
1	CVL-15718	78219152 - JORGE PEREZ OLIVERO	HURTADO	
2	CDV-222	72612152 - DANNY BERNALDA OLIVERO	HURTADO	

Página 1

CAPÍTULO V

CONCLUSIONES

5.1. CONCLUSIONES

- A. El tipo de modelo que se está usando para la detección de vehículos hurtados utiliza OCR en Python a través de Visual Studio Code para extraer caracteres de las imágenes. Posteriormente, estos resultados se encapsulan en un arreglo para su comparación con la base de datos local.

- B. Se desarrolló una arquitectura para la detección de vehículos hurtados, como se evidencia en la Figura 8 de la página 37. Esta arquitectura extrae las características de las placas, una vez extraídos se colocan en una matriz para luego consultar la base de datos que contiene información sobre el estado del vehículo, el propietario y la licencia del conductor. Al realizar la consulta, devuelve información sobre si el vehículo está registrado como robado o no (salida).

- C. La eficacia del modelo machine learning para la detección de placas de vehículos hurtados se logra una eficacia del 92%, marcando el máximo rendimiento alcanzado. Se observa que en la figura 10 de la página 39, la eficacia se mantiene constante como un estándar.

- D. Se realizó la validación del modelo de machine learning para detectar placas de vehículos hurtados mediante un conjunto de datos de entrenamiento que consta de 3340 registros y un conjunto de datos de prueba de 2947, cuyo entrenamiento tardó 0,0004 segundos lo cual es óptimo. Esta validación garantiza que todos los parámetros sean válidos después del entrenamiento del modelo, lo que nos permite realizar pruebas reales y evaluar su eficacia del modelo.

CAPÍTULO VI

RECOMENDACIONES

6.1. RECOMENDACIONES

- A. Para fortalecer la robustez del modelo, se aconseja diversificar y ampliar el conjunto de datos de entrenamiento, incorporando imágenes con diversas condiciones de iluminación, ángulos y contextos para potenciar la capacidad de generalización del modelo.
- B. Realizar pruebas y evaluaciones adicionales del modelo en entornos del mundo real para verificar su rendimiento en condiciones variables y asegurar su eficacia en situaciones operativas reales.
- C. Ampliar el alcance geográfico de la investigación para incluir otras regiones afectadas por el robo de vehículos, lo que podría proporcionar una visión más completa de los patrones de hurto y mejorar la generalización del modelo.
- D. Realizar estudios de validación adicionales para confirmar la eficacia del modelo en entornos del mundo real y evaluar su capacidad para adaptarse a diferentes condiciones y escenarios de detección de vehículos hurtados.
- E. Continuar investigando y desarrollando nuevas técnicas y metodologías en el campo del machine learning aplicado a la seguridad pública, con el objetivo de mejorar constantemente las herramientas y estrategias disponibles para combatir el robo de vehículos y otros delitos relacionados.

REFERENCIA

- A.gov. (2023). Obtenido de <https://www.dmv.ca.gov/portal/es/vehicle-registration/license-plates-decals-and-placards/california-license-plates/>
- Aguilar, J. (2023). Contaminación visual en Lima: Una revisión de literatura. *Revista de Ciencias Sociales*. 123-135.
- Aguirre Ascona, Y. D. (2019). Métodos de aprendizaje supervisado para la predicción de diabetes: una revisión sistemática de la literatura.
- Alimentación, O. d. (2006). *Evapotranspiración del cultivo*. Roma: fao.
- Anasi Nasimba, R. A., & Martínez Arellano, G. A. (2023). Sistema de control para distribución de combustibles mediante la identificación de placas en vehículos utilizando visión artificial (Bachelor's thesis).
- Aracena, C. V. (2022). Aplicaciones de aprendizaje automático en salud. . *Revista Médica Clínica Las Condes*, 568-575. Obtenido de <https://doi.org/10.1016/j.rmclc.2022.10.001>
- Arana, C. (2021). Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales (No. 797. *Serie Documentos de Trabajo*.
- Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python. Obtenido de <https://hdl.handle.net/11441/89506>
- Bandyopadhyay, H. (julio de 2021). V7. Obtenido de <https://www.v7labs.com/blog/image-classification-guide>
- Barnes, D., & Kölling, M. (2007). *Programación orientada a objetos con Java* (Tercera ed.). (B. I. Brenta, Trad.) Madrid: Pearson Educación, S.A.
- BARNES, D., & KÖLLING, M. (2007). *Programación orientada a objetos con Java* (Tercera ed.). (B. I. Brenta, Trad.) Madrid: PEARSON EDUCACIÓN, S.A.
- Beltrán Morales, J. (24 de Enero de 2023). Red neuronal convolucional para separar basura. Universidad de los Andes. Obtenido de <http://hdl.handle.net/1992/64238>
- Bernal, C. A. (2010). En *Metodología de investigacion. Tercera edicion* (pág. 320). Pearson Educacion, Colombia, 2010. doi:978-958-699-128-5
- Booch, G. (2001). *Análisis y Diseño Orientado a Objetos*. Mexico: S.A. ALHAMBRA.
- Booch, G. (2001). *Análisis y Diseño Orientado a Objetos*. Mexico: S.A. Almahambra.
- Brownlee, J. (17 de abril de 2020). *machine learning mastery*. Obtenido de <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>
- Camargo, M. d. (2013). *Sistema de Control de riego automático mediante el monitoreo de humedad del suelo vía internet*. Universidad Autónoma de Querétaro. Mexico: Centro Universitario Querétaro.
- Camejo E., Lorenzo; Duarate S., Leonel; Companioni L., José. (2010). Tecnología de riego y fertirrigación en ambientes controlados. *Ciencias Técnicas Agropecuarias*, 19, 95-97.
- Camejo E., Lorenzo; DUARTE S., Leonel; COMPANIONI L., José. (2010). Tecnología de riego y fertirrigación en ambientes controlados. *Ciencias Técnicas Agropecuarias*, 19, 95-97.
- Casanova Rivera, H. F. (2022). Diseño de un mecanismo de generación de

- actualizaciones dinámico utilizando aprendizaje reforzado en comunicaciones IOT celular para optimizar la frescura de la información. Obtenido de <http://dspace.unach.edu.ec/handle/51000/9942>
- Castro, U. (2024). Pronóstico de ventas a corto plazo en el sector de consumo masivo a través de modelos de machine learning.
- Castruita, M. A. (2017). *Modelación de la evapotranspiración potencial*. Mexico: Torreón.
- Ccama, D. A. (28 de abril de 2022). *rpubs*. Obtenido de https://rpubs.com/dsotilccama/CNN_Markdown
- Chilón, A. L. (2023). Redes neuronales recurrentes y modelos arima para el pronóstico de la inflación en el Perú.
- Choudhary, J. R. (15 de julio de 2020). *Medium*. Obtenido de <https://medium.com/analytics-vidhya/what-is-model-validation-257686d0253e>
- Chunqu Xiao, H. W. (Enero de 2023). Dense is not green: How visual density influences greenness evaluation on environmentally friendly products. *Didier Soopramanien Xi'an Jiaotong-Liverpool University, China*. Obtenido de <https://doi.org/10.3389/fpsyg.2022.1035021>
- Coleman, J. (23 de setiembre de 2020). *ODI open data institute*. Obtenido de <https://www.theodi.org/article/what-is-a-computer-model/>
- Collins. (2023). Obtenido de <https://www.collinsdictionary.com/dictionary/english/license-plate>
- Connor, R. (2021). *Valor del agua*. Italia: Colombella.
- Cortina, R. E. (2023). Sistema de orientación utilizando procesamiento de imágenes para personas con discapacidad visual que practican actividad física (Master's thesis, Tesis (MC). *Centro de Investigación y de Estudios Avanzados del IPN Departamento de Ingeniería Eléctrica. Sección de Bioelectrónica*). Obtenido de <chrome-extension://efaidnbmninnibpcajpcglclefindmkaj/https://repositorio.cinvestav.mx/bitstream/handle/cinvestav/4237/SSIT0019258.pdf?sequence=1>
- Coursera. (09 de 01 de 2023). *Coursera*. Obtenido de <https://www.coursera.org/articles/machine-learning-models>
- coursera. (2024). *coursera*. Obtenido de [coursera: https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python](https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python)
- Databricks. (2024). *Databricks*. Obtenido de <https://www.databricks.com/glossary/machine-learning-models>
- Davis, J. (2023). *itstillruns*. Obtenido de <https://itstillruns.com/registration-number-car-registration-6319449.html>
- Deitel, Paul; Deitel, Harvey;. (2012). *How to Program Java* (Novena ed.). Mexico: Pearson Educación.
- Delrieux, C., Ramoscelli, G., & Chia, D. (2001). Visualización y Procesamiento de Imágenes Satelitales. *III Workshop de Investigadores en Ciencias de la Computación*. Argentina.
- Devlin, J. C.-W. (2018). Bert: Pre-training of deep bidirectional transformers for language

- understanding. *Nature*.
- Díaz Labrador, M., & Collazo Garcia, A. (2013). *La Programación Extrema*.
- Durán, J. Á. (2022). VALIDEZ PREDICTIVA DE UN MODELO DE MACHINE LEARNING PARA LA DETECCIÓN INTRA-OPERATIVA DE TUMOR CEREBRAL.
- Enddrowning. (2015). *Enddrowning*. Obtenido de <https://enddrowning.org/steps/step-4/determine-resource-requirements>
- Esmaylin, C. F. (2023). Modelado de las etapas de amplificación y distorsión de guitarra eléctrica mediante redes neuronales recurrentes y redes neuronales convolucionales. Obtenido de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/<https://repositorio.espe.edu.ec/bitstream/21000/36731/1/T-ESPE-052862.pdf>
- Faci Gonzales, J. m., & Martinez Cob, A. (1991). *Cálculo de la evapotranspiración de referencia en Aragon*. Zaragoza.
- Fao. (22 de Julio de 2013). *Fao.com*. Obtenido de http://www.fao.org/nr/water/cropinfo_alfalfa.html
- Fernandez, J. C. (2022). Sistema de reconocimiento de patrones con inteligencia artificial para detectar placas de vehículos requisitoriados para la Policía Nacional del Perú.
- García Petillo, M., Puppo, L., Hayashi, R., & Morales, P. (2008). *Metodología para determinar los parametros hidricos de un suleo de campo*. Mexico.
- García, J. (2023). Memoria de tesis doctoral.
- geeksforgeeks. (2023). *geeksforgeeks*. Obtenido de <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>
- Girón, M. A. (2023). Consecuencias neuropsicológicas por contaminación visual en la población de Guatemala. *Académica CUNZAC*, 69-75.
- González, F. A. (2015). Machine learning models in rheumatology. *Colombiana de Reumatología*, 77-78. Obtenido de <http://www.scielo.org.co/pdf/rcr/v22n2/v22n2a01.pdf>
- Granados, R. M. (2016). Modelos de regresión lineal múltiple. Obtenido de https://www.ugr.es/~montero/matematicas/regresion_lineal.pdf
- Hernandez. (2014). Metodología de la investigacion. doi:978-1-4562-2396-0
- Herrera Acosta, R., Palomino Pacheco, K., Reyes Jiménez, F., & Valencia Ochoa, G. (2018). *Análisis Estadístico Descriptivo e Inferencial de la Velocidad y Dirección del viento en la Costa Colombiana*. Colombia.
- Hopsworks. (2024). *Hopsworks*. Obtenido de <https://www.hopsworks.ai/dictionary/model-architecture#:~:text=Model%20architecture%20consists%20of%20layers,%2C%20feature%20extraction%2C%20or%20prediction.>
- Huamán Gutierrez, H. A. (2015). *Estimacion espacial de la evapotranspiracion real usando imágenes satelitales mediante algoritmo sebal caso_ irrigacion majes I*. Arequipa.
- Infoagro. (18 de Julio de 2015). <http://www.infoagro.com/>. Obtenido de <http://www.infoagro.com/herbaceos/forrajes/alfalfa.htm>
- Infoclima, a. (22 de Julio de 2015). <http://agro.infoclima.com/>. Obtenido de

- http://agro.infoclima.com/?page_id=506
- Jalolov, T. (2023). Resolver problemas complejos en Python. *Revista Estadounidense de Lenguaje, Alfabetización y Aprendizaje en Educación STEM*.
- Joaquín Guitierrez Jagüey, Miquel Ángel Porta Gándara, Eduardo Romero Vivas, José Francisco Villa Medina. (2012). *Sistema de riego Automatizado*. Playa Palo de Santa Rita Sur, Baja California Sur, Mexico: Centro de Investigaciones Biológicas del Noreste.
- Juhi, R. (5 de agosto de 2020). *Medium*. Obtenido de <https://towardsdatascience.com/why-is-model-validation-so-darn-important-and-how-is-it-different-from-model-monitoring-61dc7565b0c>
- keepcoding. (19 de diciembre de 2022). *keepcoding*. Obtenido de <https://keepcoding.io/blog/capas-pooling-red-neuronal-convolucional/>
- Koncagül, E., Michael, T., & Connor, R. (2020). *Agua y el Cambio Climático*. Perugia, Italia: Colombella.
- Kuhne, T. (2005). Obtenido de <http://drops.dagstuhl.de/opus/volltexte/2005/23>
- La Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. (2021). *Informe Mundial de las Naciones Unidas sobre el Desarrollo de los Recursos Hídricos 2021*. Obtenido de https://unesdoc.unesco.org/ark:/48223/pf0000375750_spa
- La Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. (s.f.). <https://es.unesco.org/themes/garantizar-suministro-agua/hidrologia/escasez-calidad>.
- Leiva, I. G. (2019). Técnicas y usos en la clasificación automática de imágenes. Obtenido de <https://osf.io/d6ax4/download>
- Lincol Zotarelli, Michael D. Dukes; T. Morgan, Kelly;. (2013). Interpretación del contenido de la humedad del suelo para determinar capacidad de campo y evitar riego excesivo en suelos arenosos utilizando sensores de Humedad. *EDIS*.
- Lincol Zotarelli, Michael D. Dukes; T. Morgan, Kelly;. (2013). Interpretación del contenido de la humedad del suelo para determinar capacidad de campo y evitar riego excesivo en suelos arenosos utilizando sensores de Humedad. *Edis*.
- López Acevedo, M. (2014). *Edafología uso y proteccion de los Suelos*. Cataluña: Mundi-Prensa.
- Mamani, U. Q. (2022). CONTAMINACIÓN VISUAL DEL PAISAJE URBANÍSTICO DE LA CIUDAD DE JULIACA. . *Ñawparisun-Revista de Investigación Científica*.
- Margüelles, A. (2023). Detección en tiempo real de comida: Un enfoque innovador para la clasificación automatizada de alimentos con CNN. Obtenido de chrome-extension://efaidnbmnnnibpcajpglclefndmkaj/https://oa.upm.es/75743/1/TFG_DAVID_MARGUELLES_ANSORENA.pdf
- Márquez Díaz, J. (2020). Inteligencia artificial y Big Data como soluciones frente a la COVID-19. *Revista de bioética y derecho*, 50, 315-331. Obtenido de https://scielo.isciii.es/scielo.php?pid=S1886-58872020000300019&script=sci_arttext&tlng=pt
- Martín de Diego, E. (2022). Redes neuronales convolucionales. Obtenido de

- <https://uvadoc.uva.es/handle/10324/582257>
- Mathworks. (2023). *mathworks*. Obtenido de <https://www.mathworks.com/discovery/feature-extraction.html#:~:text=Feature%20extraction%20refers%20to%20the,in%20the%20original%20data%20set>.
- Meteorologiaenred. (07 de 10 de 2021). *meteorologia en red*. Obtenido de <https://www.meteorologiaenred.com/>
- Miranda Novales, M. G., Arias Gómez, J., & Villasís Kever, M. A. (2016). El protocolo de investigación III. *Alergia México*, 63(2), 201-206. Obtenido de <https://www.redalyc.org/articulo.oa?id=486755023011>
- Moreno. (2005). Metodología de la Investigación. Obtenido de <http://biblioteca.udgvirtual.udg.mx/jspui/handle/123456789/3830>
- Moreno, J. (2022). *HubSpot*. Obtenido de <https://blog.hubspot.es/marketing/espectaculares-carteles-publicitarios-interactivos#:~:text=%C2%BFQu%C3%A9%20es%20un%20cartel%20publicitario,%2C%20servicio%2C%20suceso%20o%20evento>.
- Moreo, F. J. (2023). Diseño e implementación en hardware reconfigurable de un sistema de reconocimiento de gestos de la mano basado en visión por computador (Doctoral dissertation, Universidad Politécnica de Cartagena). Obtenido de <https://dialnet.unirioja.es/servlet/dctes?codigo=312756>
- Nasch, O. e. (2015). Reconocimientos de imágenes, a base de múltiples redes neuronales.
- Oliveira, A. B. (2013). Impacto Psicosocial. En: Gellman, MD, Turner, JR (eds). *Enciclopedia de medicina conductual*.
- Openweathermap. (07 de 10 de 2021). *open weather map*. Obtenido de <https://openweathermap.org/technology>
- Orbe Cisneros, E. A., & Vargas Vallejo, V. C. (2022). Diseño e implementación de un sistema de detección de caídas en el adulto mayor mediante visión artificial utilizando redes neuronales convolucionales (Doctoral dissertation, Tesis de Ingeniería, Universidad De Las Fuerzas Armadas]. Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/31742/1/TESPE-052408.pdf>.
- Ordoñez-Matute, R. (2022). Dispositivo Escénico Relacional: nuevos formatos de activación, traducción, desciframiento e interferencia para las artes escénicas contemporáneas. *telondefondo. Revista de Teoría y Crítica Teatral*.
- Orozco Medina, I., Alvarado Barrientos, M. S., López de la Cruz, J., & Ramírez Orozco, A. I. (2019). *Evaluación del efecto de distintas aproximaciones de la radiación en la estimación de la evapotranspiración usando el modelo FAO Penman-Monteith*. Mexico.
- Orozco Medina, I., Alvarado Barrientos, S., López de la Cruz, J., & Ramírez Orozco, A. (2019). *Evaluación del efecto de distintas aproximaciones de la radiación*. Mexico.
- Oziel Lugo Espinoza, A. Q. (2011). Prototipo para Automatizar un Sistema de riego Multicultivo. *Revista Mexicana de Ciencias Agrícolas*, 659-672.
- Pacheco, P. H. (2023). El uso de la inteligencia artificial como herramienta del juzgador . Obtenido de <https://repositorio.unitec.edu/xmlui/handle/123456789/10793>

- Palomino Dongo, E. (2019). *Producción de forraje en líneas a diferente distanciamiento en cuatro variedades de alfalfa (Medicago sativa), a 2750 msnm. Ayacucho. Ayacucho.*
- Palomino Palomino, E. (2013). *Análisis de los sistemas productivos y la comercialización de palto en el valle de Iuricocha.* Haancavelica.
- Palomino, D. (2019). *Producción de forraje en líneas a diferente distanciamiento en cuatro variedades de alfalfa.* Ayacucho.
- Panduro, B. C. (2022). Aplicación del algoritmo del bosque aleatorio a un modelo de clasificación de la anemia en niños peruanos. *MediCiego*, 28(1). Obtenido de <https://revmediciego.sld.cu/index.php/mediciego/article/view/3471/0>
- Paperspace. (2020). *paperspace.* Obtenido de <https://machine-learning.paperspace.com/wiki/machine-learning-models-explained>
- Pérez, M. E. (2006). Fluctuaciones Climáticas y Variabilidad Temporal del Clima en el Norte Argentino – 1931/2005. *XI Encuentro de Profesores en Geografía del Nordeste.* Argentina.
- Pérez, M. E. (2006). FLUCTUACIONES CLIMÁTICAS Y VARIABILIDAD TEMPORAL DEL CLIMA EN EL NORTE ARGENTINO – 1931/2005. *XI ENCUENTRO DE PROFESORES EN GEOGRAFIA DEL NORDESTE.* Argentina.
- Perfumo, M. S., & Ares, M. V. (2023). Indicadores para la evaluación de la evolución de la matrícula en instituciones de educación superior. . *Digital Publisher CEIT.*
- Porras, E. (2010). *Comparación de los procesos de desarrollo de software usando los métodos ICONIX y XP, caso: Comercialización de la tara en la región de Ayacucho.* Lima, Peru.
- Quevedo Nolasco, A. (2008). Sistema de Riego Automatizado En Tiempo Real Con Balance Hídrico, Medición De Humedad Del Suelo Y Lisímetro. *Insitut for Agrartechnik*, 34, 459-470.
- Retamozo Godoy, W. J. (2015). *Estimación de la evapotranspiración potencial a partir de imágenes satelitales Modis aplicadas a la zona agrícola de canaán (2756 msnm) - ayacucho 2015.* Ayacucho.
- Rojas, J. D. (2023). Desarrollo de un modelo de Machine Learning para la clasificación de tipos de dengue de acuerdo a su nivel de severidad: Un estudio de caso de Bucaramanga, Colombia. Obtenido de Rojas Sánchez, J. D. (2023).
- Ruiz Lopez, C. R. (2001). *Diseño de un Sistema de Riego Movil por Goteo.* Escuela Superior Politecnica del Litoral. Guayaquil: Centro Universitario.
- Salgado Ortiz, S. (2017). *Diseño y Construcción de una Estación Portátil de Medición de Presión Atmosférica, Humedad Relativa y Temperatura.* Bogotá.
- Salgado Ortiz, S. (2017). *DISEÑO Y CONSTRUCCIÓN DE UNA ESTACIÓN PORTÁTIL DE MEDICIÓN DE PRESIÓN ATMOSFÉRICA, HUMEDAD RELATIVA Y TEMPERATURA.* Bogotá.
- Sálmon Cuspinera, Y., Rodríguez Gross, R., & Rosales Montoya, A. (2016). *Efecto de las Variables Climáticas sobre el Rendimiento Industrial.* Cuba.
- Sálmon Cuspinera, Y., Rodríguez Gross, R., & Rosales Montoya, A. (2016). *EFEECTO DE LAS VARIABLES CLIMÁTICAS SOBRE EL RENDIMIENTO INDUSTRIAL.*

Cuba.

- Saturncloud. (2023). *saturncloud*. Obtenido de <https://saturncloud.io/blog/calculating-input-and-output-size-for-conv2d-in-pytorch-for-image-classification/>
- Segovia Granda, H. P. (2022). Revisión sistemática y análisis de metodologías que utilizan técnicas de minería de datos y aprendizaje automático para detección del trolling en las redes sociales. Obtenido de <http://repositorio.uisek.edu.ec/handle/123456789/4864>
- Smith. (2023). Clasificación automática de imágenes. P. 123.
- Tierrablanca, P. H. (2023). *Predicción de precios de productos agrícolas mediante redes neuronales recurrentes*. Obtenido de <https://ri-ng.uaq.mx/handle/123456789/8369>
- Turley, Paul; Wood, Dan;. (2009). *Beginning T.SQL*. Canada: Wiley Publishing, Inc.
- Valdiviezo Camacho, F. I. (2020). *Estimación de la Evapotranspiración de los cultivos mediante MOD16*. Quito.
- Vera, G. E. (2014). *Implementación de Sistema de control de riego Mediante Red inalámbrica de Sensores de humedad*. Universidad Técnica Estatal de Quevedo. Ecuador: Quevedo.
- zenvia. (07 de 10 de 2021). *zenvia*. Obtenido de <https://www.zenvia.com/es/blog/que-es-api/>
- Zvornicanin, E. (05 de mayo de 2023). *baeldung*. Obtenido de <https://www.baeldung.com/cs/cnn-depth>

Anexos

Anexo 1. Matriz de consistencia

Título: **Modelo machine learning** para **detectar** las **placas de los vehículos hurtados** Ayacucho, 2023.

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	MÉTODO DE INVESTIGACIÓN
PROBLEMA GENERAL ¿Cuáles son las características del modelo de machine learning para detectar las placas de vehículos hurtados Ayacucho,2023?	OBJETIVO GENERAL Determinar las características de un modelo machine learning para detectar las placas de vehículos hurtados en la ciudad de Ayacucho,2023 .	HIPÓTESIS GENERAL Según Hernández (2014), Una hipótesis sirve como orientación en un estudio o investigación, delineando lo que se intenta demostrar y describiéndose como una explicación provisional de un fenómeno objeto de estudio. Estas hipótesis se derivan de teorías preexistentes y deben expresarse como proposiciones; esencialmente, representan respuestas preliminares a las preguntas de investigación. Es crucial reconocer que en nuestra cotidianidad a menudo formulamos suposiciones sobre diversos temas y posteriormente las verificamos para confirmar su veracidad. (p. 104)”.	VARIABLE DE ESTUDIO X: Modelo machine learning VARIABLES DESCRIPTIVAS X1: Tipo de modelo X2: Arquitectura del modelo X3: Eficacia del modelo X4: Validez del modelo	TIPO DE INVESTIGACIÓN La investigación es de tipo observacional, transversal y descriptiva. NIVEL DE INVESTIGACIÓN Descriptivo DISEÑO No experimental, observacional. POBLACIÓN Está compuesto por más de 37,000 vehículos registrados en la región de Ayacucho. MUESTRA Se estudiarán 537 imágenes de placas vehiculares. TÉCNICA Fotografía documental
PROBLEMAS ESPECÍFICOS a. ¿Cuál es el tipo de modelo de machine learning para detectar las Placas de vehículos hurtados ? b. ¿Cuál es la arquitectura del modelo de machine learning para detectar las Placas de vehículos hurtados ? c. ¿Cuál es la eficacia del modelo de machine learning para detectar las	OBJETIVOS ESPECÍFICOS a. Determinar el tipo del modelo machine learning , con la finalidad de detectar las placas de vehículos hurtados . b. Crear una arquitectura del modelo machine learning , con la finalidad de detectar las placas de vehículos hurtados . c. Medir la eficacia del modelo machine learning que	En la presente investigación no se realizará el planteamiento de hipótesis, considerando que esta		

Placas de vehículos hurtados?	determinará la rapidez del método de algoritmo, con la finalidad de detectar las placas de vehículos hurtados . investigación es de nivel descriptivo, además no pretendemos pronosticar una cifra o un hecho; sin embargo, lograremos los objetivos específicos planteados”.	Análisis documental
d. ¿Cuál es la validez del modelo de machine learning para detectar las Placas de vehículos hurtados ?	d. Determinar la validez del modelo machine learning , con la finalidad de detectar las placas de vehículos hurtados .	INSTRUMENTO Cámara de celular Guía de análisis documental

Anexo 2. Matriz de operacionalización de variables

Variables	Dimensiones	Indicadores	Pregunta
Modelo machine learning	Tipo de modelo	Tipos de algoritmos utilizados	¿Qué algoritmos de machine learning se emplearon en el modelo?
		Datos de entrada	¿Qué tipo de datos de entrada se utilizan para entrenar el modelo de machine learning?
	Arquitectura del modelo	Métodos de entrenamiento y validación	¿Qué métodos se emplearon para entrenar y validar la arquitectura del modelo?
		Precisión	¿Qué precisión ha demostrado el modelo en la predicción de resultados?
	Eficacia del modelo	Sensibilidad y especificidad	¿Cuál es la sensibilidad y especificidad del modelo en la detección de casos positivos y negativos?
		Validez del	Evaluación externa

modelo

cabo alguna
evaluación
externa del
modelo por
parte de
expertos en el
campo?

Fuente: Elaboración propia.

Anexo 3. Instrumento cámara de celular

Este instrumento se utiliza para capturar datos relacionados con la variable de clasificación automática de imágenes a través de la fotografía documental.

CAMARA DE CELULAR

NOMBRE DEL INSTRUMENTO: Cámara de Celular Redmi 10 (2022)

MARCA Y MODELO: Redmi 10

AÑO DE FABRICACIÓN: 2022

DESCRIPCIÓN: Teléfono celular con capacidad fotográfica integrada.

RESOLUCIÓN DE LA CÁMARA PRINCIPAL: 50 megapíxeles.

RESOLUCIÓN DE LA CÁMARA FRONTAL: 16 megapíxeles.

TIPO DE ENFOQUE: Autofoco

ZOOM ÓPTICO: No

APERTURA DEL DIAFRAGMA: f/1.8

ESTABILIZACIÓN DE IMAGEN: Sí (Estabilización óptica de imagen)

FLASH INCORPORADO: Si

FUNCIONES ADICIONALES: Detección de rostros, HDR, modo nocturno

USO EN LA INVESTIGACIÓN: Captura de imágenes para la recolección de datos en el estudio sobre el modelo de machine learning para detectar placas de vehículos hurtados].

PROCEDIMIENTO DE UTILIZACIÓN: Para utilizar el instrumento, encienda el celular y abra la aplicación de la cámara. Ajuste los parámetros según sea necesario (resolución, modo de disparo, etc.) y capture las imágenes de las placas de vehículos

LIMITACIONES Y CONSIDERACIONES: La calidad de la imagen puede disminuir en condiciones de poca luz. Se recomienda utilizar el flash en tales circunstancias. La duración de la batería puede ser limitada con un uso extenso de la cámara.

RESPONSABLE DEL INSTRUMENTO: Lizbeth Roxana Infante Leva

ESTADO DEL INSTRUMENTO: Nuevo

Anexo 4. Ficha de análisis documental

FICHA DOCUMENTAL
TITULO: _____
AUTOR: _____
FECHA: _____
FUENTE: _____
CONTEXTO: _____
OBJETIVOS: _____
METODOLOGIA: _____
RESULTADOS: _____
CONCLUSIONES: _____
RELEVANCIA Y APLICABILIDAD: _____
REFERENCIAS: _____
PALABRA CLAVES: _____
CIUDAD: _____
RESUMEN DEL CONTENIDO

**UNSCH**FACULTAD DE
INGENIERÍA
DE MINAS, GEOLOGÍA Y CIVIL**ACTA No 040-2024-FIMGC: ACTA DE SUSTENTACION DE TESIS****PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERA DE SISTEMAS**

En la Universidad Nacional de San Cristóbal de Huamanga de la ciudad de Ayacucho, en cumplimiento a la Resolución Decanal N° 448-2024-FIMGC-D y al Memorando N° 297, a los nueve días del mes de julio de 2024, siendo las 04:00 p.m, reunidos en el Auditorio de la Escuela Profesional de Ingeniería de Minas, bajo la presidencia del Mg. Ing. Edith Felicitas Guevara Morote y los miembros; Mg. Ing. Eloy Vila Huaman, Mg. Ing. José Antonio Guerrero Hinostroza, Mg. Ing. Manuel Avelino Lagos Barzola actuando como secretario docente el MSc. Kelvis Berrocal Argumedo, para proceder a la sustentación de tesis para optar el Título Profesional de Ingeniera de Sistemas, del bachiller:

Lizbeth Roxana Infante Leva

Quien presentó la tesis denominada:

"MODELO MACHINE LEARNING PARA DETECTAR LAS PLACAS DE VEHÍCULOS HURTADOS AYACUCHO, 2023"

Los señores miembros del jurado luego de expuesto la tesis y absueltas las preguntas, delibera y lo declaran:

Aprobado con nota quince (15)

Siendo las 05:20 p.m. del día 09 de julio de 2024, culmina el acto de sustentación de tesis, y en conformidad a lo actuado los miembros del jurado firmamos al pie del presente.

Mg. Ing. Edith Felicitas Guevara Morote
Presidente

Mg. Ing. Eloy Vila Huaman
Miembro

Mg. Ing. José Antonio Guerrero Hinostroza
Miembro

Mg. Ing. Manuel Avelino Lagos Barzola
Miembro

MSc. kelvis Berrocal Argumedo
Secretario docente de la FIMGC

cc:

Archivo



CONSTANCIA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN

CONSTANCIA N° 006-2024-KPS-FIMGC/UNSCH

El que suscribe; responsable verificador de originalidad de trabajos de tesis de pregrado con el software Turnitin, en segunda instancia para las **Escuelas Profesionales** de la **Facultad de Ingeniería de Minas, Geología y Civil**; en cumplimiento a la **Resolución de Consejo Universitario N° 039-2021-UNSCH-CU**, Reglamento de Originalidad de Trabajos de Investigación de la Universidad Nacional San Cristóbal de Huamanga y **Resolución Decanal N° 473-2023-FIMGC-D**, deja constancia de originalidad de trabajo de investigación, que el/la Sr./Srta.

Nombres y Apellidos : LIZBETH ROXANA INFANTE LEVA
Escuela Profesional : INGENIERÍA DE SISTEMAS
Título de la Tesis : "MODELO MACHINE LEARNING PARA DETECTAR LAS PLACAS DE VEHÍCULOS HURTADOS AYACUCHO, 2023"
Evaluación de la Originalidad : 8 % Índice de Similitud
Identificador de la entrega : 2443148070

Por tanto, según los Artículos 12, 13 y 17 del Reglamento de Originalidad de Trabajos de Investigación, es **PROCEDENTE** otorgar la **Constancia de Originalidad** para los fines que crea conveniente.

En señal de conformidad y verificación se firma la presente constancia

Ayacucho, 04 de setiembre de 2024



Firmado digitalmente por:
PERALTA SOTOMAYOR Karel
FAU 20143680754 soft
Motivo: Soy el autor del documento
Fecha: 04/09/2024 17:13:16-0500

“MODELO MACHINE LEARNING PARA DETECTAR LAS PLACAS DE VEHÍCULOS HURTADOS AYACUCHO, 2023”

por Lizbeth Roxana Infante Leva

Fecha de entrega: 02-sep-2024 08:24a.m. (UTC-0500)

Identificador de la entrega: 2443148070

Nombre del archivo:

MODELO_MACHINE_LEARNING_PARA_DETECTAR_LAS_PLACAS_DE_VEHÍCULOS_HURTADOS_AYACUCHO_2023.pdf
(1.39M)

Total de palabras: 18420

Total de caracteres: 114554

"MODELO MACHINE LEARNING PARA DETECTAR LAS PLACAS DE VEHÍCULOS HURTADOS AYACUCHO, 2023"

INFORME DE ORIGINALIDAD

8%

INDICE DE SIMILITUD

8%

FUENTES DE INTERNET

2%

PUBLICACIONES

3%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

1

cdn.www.gob.pe

Fuente de Internet

4%

2

omes-va.com

Fuente de Internet

2%

3

Submitted to Universidad Nacional de San
Cristóbal de Huamanga

Trabajo del estudiante

1%

4

hdl.handle.net

Fuente de Internet

<1%

5

Submitted to Stefan cel Mare University of
Suceava

Trabajo del estudiante

<1%

6

Submitted to City University

Trabajo del estudiante

<1%

7

forum.djangoproject.com

Fuente de Internet

<1%

8

repositorio.unsch.edu.pe

Fuente de Internet

<1%

9

Submitted to Oklahoma State University

Trabajo del estudiante

<1 %

10

repositorio.unheval.edu.pe

Fuente de Internet

<1 %

Excluir citas

Activo

Excluir coincidencias < 30 words

Excluir bibliografía

Activo