

UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENIERIA Y ARQUITECTURA

Escuela Profesional de Ingeniería Civil



Una Institución Adventista

Aplicación de inteligencia y visión artificial para la obtención del aforo vehicular

Tesis para obtener el Título Profesional de Ingeniero Civil

Por:

Fiorella Yaneth Coanqui Apaza

Rudhy Fabian Estofanero Yanapa

Helio Weiss Mamani Condori

Asesor:

Ing. Percy Armando Cota Mayorga

Juliaca, octubre de 2022

DECLARACIÓN JURADA DE AUTORÍA DE TESIS

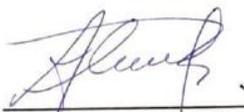
Ing. Percy Armando Cota Mayorga, de la Facultad de Ingeniería y Arquitectura,
Escuela Profesional de Ingeniería Civil, de la Universidad Peruana Unión.

DECLARO:

Que la presente investigación titulada: **“APLICACIÓN DE INTELIGENCIA Y VISIÓN ARTIFICIAL PARA LA OBTENCIÓN DEL AFORO VEHICULAR”** constituye la memoria que presenta los Bachilleres Fiorella Yaneth Coanqui Apaza, Rudhy Fabian Estofanero Yanapa y Helio Weiss Mamani Condori para obtener el título de Profesional de Ingeniero Civil, cuya tesis ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en la ciudad de Juliaca, a los 05 días del mes de octubre del año 2022.



Ing. Percy Armando Cota Mayorga

ACTA DE SUSTENTACIÓN DE TESIS



En Puno, Juliaca, Villa Chullunquiari, a 04 día(s) del mes de octubre del año 2022 a las 9:30 horas, se reunieron los miembros del jurado en la Universidad Peruana Unión Campus Juliaca, bajo la dirección del

(de la) presidente(a):

Ing. Xerxon Quiberly Pari Lusi el (la) secretario(a): Mg. Efraim Velazquez

Mamani y los demás miembros: Ing. Rina Sugmeri
Zampara Ficona y el (la) asesor(a) Ing. Percy Armando Lota

Mayorga con el propósito de administrar el acto académico de sustentación de la tesis titulado:

Aplicación de inteligencia y visión artificial para la obtención del aforo vehicular

del(los) bachiller(es): a) Rudhy Fabian Estofanero Yanapa

b) Fiorella Yaneth Loanqui Spaza

c) Helio Weiss Mamani Londeri

conducente a la obtención del título profesional de:

Ingeniero Civil
(Denominación del Título Profesional)

El Presidente inició el acto académico de sustentación invitando al (a la) / a (los) (las) candidato(a)s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por al (a la) / a (los) (las) candidato(a)s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Bachiller (a): Rudhy Fabian Estofanero Yanapa

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
<u>Aprobado</u>	<u>15</u>	<u>B-</u>	<u>Buena</u>	<u>Muy Bueno</u>

Bachiller (b): Fiorella Yaneth Loanqui Spaza

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
<u>Aprobado</u>	<u>15</u>	<u>B-</u>	<u>Buena</u>	<u>Muy Bueno</u>

Bachiller (c): Helio Weiss Mamani Londeri

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
<u>Aprobado</u>	<u>15</u>	<u>B-</u>	<u>Buena</u>	<u>Muy Bueno</u>

() Ver parte posterior

Finalmente, el Presidente del jurado invitó al (a la) / a (los) (las) candidato(a)s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

[Firma]
Presidente/a

[Firma]
Asesor/a

[Firma]
Bachiller (a)

[Firma]
Miembro

[Firma]
Bachiller (b)

[Firma]
Secretario/a

[Firma]
Miembro

[Firma]
Bachiller (c)

ÍNDICE GENERAL

ÍNDICE DE TABLAS	vi
ÍNDICE DE FIGURAS	vii
RESUMEN:	8
ABSTRACT:	9
1 INTRODUCCIÓN	10
2 MARCO TEÓRICO	12
2.1 Detección y clasificación de vehículos mediante Inteligencia y Visión Artificial	12
2.1.1 Inteligencia Artificial.....	12
2.1.1.1 Red neuronal convolucional (CNN).	12
2.1.1.2 El algoritmo You Only Look Once (YOLO).	12
2.1.1.3 El algoritmo You Only Look Once (YOLO).	12
2.1.1.4 You Only Look Once (YOLOv5x).....	13
2.1.1.5 Optimización de cuadros delimitadores.	13
2.1.1.6 Seguimiento de objetos.	15
2.1.1.7 Diseño e implementación.	15
2.1.1.8 Entrenamiento de redes neuronales.....	15
2.1.1.9 Detección de un solo objetivo.	17
2.1.1.10 Detección de objetos múltiples.	17
2.1.1.11 Seguimiento de objetos múltiples.	17
2.1.1.12 Métricas de desempeño.	17
2.1.2 Visión Artificial.....	17
2.1.2.1 Definición.....	17
2.1.2.2 Sistemas de visión artificial.	18
2.1.2.3 Funcionamiento de un sistema de visión artificial.	18
2.1.2.4 Aplicaciones de la visión artificial.	18
3 METODOLOGÍA	19
3.1. Procedimiento para la ejecución de los códigos	19
3.1.1 Ejecución de YoloV5x.....	19
3.1.2 Código Inteligencia Artificial.....	21
3.1.3 Código Visión Artificial.	22
4 RESULTADOS Y DISCUSIÓN.....	23
4.1. Método Inteligencia Artificial.....	23

4.2.	Método Inteligencia Artificial.....	24
4.3.	Estudio In Situ	24
4.4.	Comparación de Resultados.....	25
4.5.	Discusión	28
4.5.1	De la Inteligencia Artificial.	28
4.5.2	De la Visión Artificial.	28
5	CONCLUSIONES Y RECOMENDACIONES	29
	REFERENCIAS	31

ÍNDICE DE TABLAS

Tabla 1 Resultados de clasificación y conteo con Inteligencia Artificial.....	23
Tabla 2 Resultados de clasificación y conteo con Visión Artificial.....	24
Tabla 3 Resultados del Estudio In Situ.....	25
Tabla 4 Comparación de resultados.....	25

ÍNDICE DE FIGURAS

Figura 1 Descripción general de la arquitectura de CNN.....	13
Figura 2 Regresión de cuadro de anclaje.....	14
Figura 3 Red piramidal de características.....	14
Figura 4 Imágenes de muestra del conjunto de datos de vehículos de entrenamiento.....	15
Figura 5 Diagrama de flujo del entrenamiento de redes neuronales.....	16
Figura 6 Imágenes recolectadas para el entrenamiento.....	19
Figura 7 Línea de código.....	20
Figura 8 Resultado de precisión de entrenamiento de red neuronal.....	20
Figura 9 Código inteligencia artificial.....	21
Figura 10 Delimitación de áreas, código visión artificial.....	22
Figura 11 Resultado lectura de código inteligencia artificial.....	23
Figura 12 Resultado código visión artificial.....	24
Figura 13 Comparación de resultados: Conteo IA vs IS.....	26
Figura 14 Comparación de resultados: % IA vs %IS vs %Error.....	26
Figura 15 Comparación de resultados: Conteo VA vs IS.....	27
Figura 16 Comparación de resultados: % VA vs %IS vs %Error.....	27

Aplicación de inteligencia y visión artificial para la obtención del aforo vehicular

RESUMEN:

Para la obtención del aforo vehicular actualmente se usan métodos manuales tradicionales los cuales generan un trabajo extenso y tiempo prolongado para realizar los estudios previos para el diseño de un pavimento, planificación del tránsito, proyecciones de vías, proyectos de señalización e iluminación, entre otros. Es por ello que se requiere automatizar los datos de conteo y clasificación vehicular. Por esta razón, se aplicaron dos métodos tecnológicos que pueden ayudarnos a mecanizar la información obtenida a través de videgrabaciones o en tiempo real. El primer método basado en inteligencia artificial con el uso de redes neuronales convolucionales y el segundo con visión artificial mediante el uso de librerías virtuales que nos ayudaran en la detección vehicular. El objetivo de esta investigación es realizar una comparación de ambos métodos contrastados con un estudio manual in situ para encontrar la variación de precisión y error a través de porcentajes, tomando en cuenta que el estudio en campo tendrá el 100% de precisión. Se realizó una prueba con ambos métodos con una videgrabación real de flujo vehicular, considerando 6 clases de vehículos (motocicletas, mototaxis, automóviles, camionetas, microbuses y vehículos pesados). Después de haber ejecutado ambas metodologías para los fines pertinentes se presentan los mejores resultados obtenidos con ambos métodos: Un 95.60% de precisión del conteo y clasificación de microbuses con inteligencia artificial y un 97.80% de precisión de microbuses con el uso de visión artificial. Finalmente se concluye con un sistema efectivo con capacidad aceptable de detección, conteo y clasificación vehicular con el uso de Inteligencia Artificial como método más apropiado para el propósito de la investigación ya que este puede reemplazar las mismas capacidades que el cerebro humano en la obtención del aforo vehicular, la visión artificial nos brindó datos más precisos sin embargo este se basa en áreas de detección lo que no es muy confiable usar en cualquier zona de estudio por la diferencia de áreas de acuerdo al ángulo de enfoque; se recomienda para tener mejor precisión en los resultados usando inteligencia artificial, entrenar la red neuronal con una base de datos más amplia y contar con una tarjeta gráfica óptima del ordenador para la lectura del código.

Palabras clave: Aforo vehicular, inteligencia artificial, visión artificial.

Application of intelligence and artificial vision to obtain vehicle capacity

ABSTRACT:

To obtain traffic capacity, traditional manual methods are currently used, which generate extensive work and prolonged time to carry out preliminary studies for the design of a pavement, traffic planning, road projections, signaling and lighting projects, among others. That is why it is required to automate vehicle counting and classification data. For this reason, two technological methods were applied that can help us mechanize the information obtained through video recordings or in real time. The first method based on artificial intelligence with the use of convolutional neural networks and the second with artificial vision through the use of virtual libraries that will help us in vehicle detection. The objective of this research is to make a comparison of both methods contrasted with a manual study in situ to find the variation of precision and error through percentages, taking into account that the study in the field will have 100% precision. A test was carried out with both methods with a real video recording of vehicular flow, considering 6 types of vehicles (motorcycles, motorcycle taxis, cars, vans, minibuses and heavy vehicles). After having executed both methodologies for the pertinent purposes, the best results obtained with both methods are presented: 95.60% accuracy of counting and classification of buses with artificial intelligence and 97.80% accuracy of buses with the use of artificial vision. Finally, it is concluded with an effective system with acceptable detection, counting and vehicle classification capacity with the use of Artificial Intelligence as the most appropriate method for the purpose of the investigation since it can replace the same capacities as the human brain in obtaining capacity. vehicle, artificial vision gave us more precise data, however this is based on detection areas, which is not very reliable to use in any study area due to the difference in areas according to the angle of focus; It is recommended to have better precision in the results using artificial intelligence, to train the neural network with a larger database and to have an optimal computer graphics card for reading the code.

Keywords: Vehicle capacity, artificial intelligence, artificial visión.

1 INTRODUCCIÓN

Realizar un estudio de tránsito es de vital importancia porque nos permite hacer un diagnóstico de demanda vehicular, los resultados de esta evaluación nos sirven para realizar una planificación de tránsito con proyecciones futuras para manifestar un mejor desarrollo social y económico, de esta manera también tomar acciones para evitar el congestionamiento vehicular de acuerdo a los resultados obtenidos.

La determinación del aforo y clasificación vehicular es de considerable importancia en el campo de la ingeniería civil, dicha información ha sido de gran utilidad durante muchos años para el diseño de pavimentos de las carreteras, planificación del tráfico, entre otros.

Un estudio de tránsito comprende varias partes, una de ellas es la determinación del aforo vehicular, en el Perú actualmente de acuerdo al Ministerio de Transportes y Comunicaciones se realiza manualmente el conteo y clasificación vehicular, esto se utiliza fundamentalmente para el planeamiento: proyección de vías, programas de acondicionamiento de pavimento, diseño del pavimento de una carretera, proyectos de señalización e iluminación, estudios medioambientales y de impacto ambiental, entre otros.

En el lado de la tecnología se fueron innovando algunas técnicas tradicionales de vanguardia para la detección y clasificación de vehículos los cuales fueron el histograma de gradiente orientado (HOG) y la máquina de vectores de soporte (SVM), según (N. & D.N., 2021) que implica pasos multifacéticos y una mayor participación de personas.

Estos métodos mencionados anteriormente no son prácticos para el escenario actual y es tal el motivo de llevar esta investigación mediante las metodologías de las redes neuronales convolucionales (CNN) y métodos con visión artificial para la comparación correspondiente, según las investigaciones de (Yin et al., 2021), son técnicas basadas en la detección y clasificación de vehículos que producen un rendimiento mejorado en comparación con el método tradicional.

La aplicación de herramientas tecnológicas puede permitirnos automatizar los datos del aforo vehicular, por lo tanto, el desarrollo de la presente investigación tiene un enfoque novedoso debido a que pretende utilizar visión e inteligencia artificial para obtener los mismos resultados que hoy en día obtenemos manualmente.

Durante estos años se realizaron investigaciones utilizando las herramientas de la IA. La investigación de (Mandal et al., 2020) demuestra los resultados donde se implementaron R-CNN, YOLO y CenterNet obteniendo los resultados de 90.5% de precisión mientras que la máxima precisión alcanzada por YOLO fue del 93,7%. La discrepancia en la detección correcta de objetos se debió principalmente a la mala calidad de la imagen. (Bugeja et al., 2020) Indican que los resultados que muestran YOLOv5x y RCNN son modelos más rápidos y equilibrados.

De la misma manera (Song et al., 2019) realizan una investigación de un sistema de detección de vehículos basado en aprendizaje profundo en escenas donde lograron que el método propuesto de detección y seguimiento de vehículos para escenas de video de vigilancia de carreteras tiene un buen rendimiento y viabilidad. Comparado con el método tradicional de monitorear el tráfico de vehículos por hardware, el método de este documento es de bajo costo y alta estabilidad y no requiere trabajos de construcción o instalación a gran escala en el equipo de monitoreo existente.

Proponer un método automatizado para la recopilación de estos datos beneficiará a los proyectos en su etapa de evaluación y estudios previos con respecto al tiempo y trabajo que se requiere para realizar dicho análisis de acuerdo a su fin. Además, el presente estudio dará inicio a nuevas investigaciones para proponer la implementación de sistemas inteligentes de transporte, monitoreo automático del flujo vehicular y con ello el progreso para futuras ciudades inteligentes.

En esta investigación se propone dos métodos tecnológicos para la detección de vehículos basada en las redes neuronales convolucionales y bibliotecas virtuales, para la clasificación entre motocicletas, mototaxis, automóviles, camionetas, microbuses y vehículos de carga pesada. La prueba realizada se capturó con una cámara digital en una autopista ubicada en una región sureña del Perú con flujo vehicular constante y polifacético durante 1 hora de prueba. En este caso el primer modelo utilizado trabaja con los modelos de YOLOv5x y el segundo con bibliotecas virtuales de la visión artificial. Además, se realizará un análisis comparativo de ambos métodos contrastado con la obtención del aforo vehicular mediante un estudio manual in situ, de acuerdo a la precisión de la extracción de características en video utilizando la Inteligencia y Visión Artificial.

2 MARCO TEÓRICO

2.1 Detección y clasificación de vehículos mediante Inteligencia y Visión Artificial

2.1.1 Inteligencia Artificial.

2.1.1.1 Red neuronal convolucional (CNN).

Su principio se basa en la extracción de los puntos de interés correspondientes a piezas predefinidas sobre la imagen del vehículo. Estos puntos se filtrarán y compararán con los puntos del modelo 3D.

CNN es una arquitectura de red neuronal ampliamente utilizada para tareas relacionadas con la visión por computadora. La ventaja de CNN es que realiza automáticamente la extracción de características en las imágenes, es decir, las características importantes son detectadas por la propia red.

2.1.1.2 El algoritmo *You Only Look Once (YOLO)*.

Fue desarrollado por J.Redmon y otros en 2015. Este algoritmo se usa para la detección de objetos, se basa en dos etapas; el primero es detectar objetos mediante redes neuronales convolucionales, el segundo paso hace la cuadrícula de la imagen y hace la predicción de la clase de objeto detectado si existe.

El modelo de red neuronal de convolución (CNN) está diseñado para el conjunto de datos de vehículos urbanos para la detección de un solo objeto y YOLOv5x para la detección de múltiples objetos en el conjunto de datos COCO.

La eficiencia de la red de neuronas convolucionales CNN depende en gran medida de la calidad del conjunto de datos de entrenamiento; la red producirá buenos resultados solo si los datos de entrenamiento utilizados contienen suficientes características importantes para que puedan producir nuevas predicciones.

2.1.1.3 El algoritmo *You Only Look Once (YOLO)*.

Existe una amplia gama de tareas de visión por computadora que benefician a la sociedad, como clasificación, detección, rastreo, conteo de objetos, segmentación semántica, subtítulos de imágenes, etc. El proceso de identificar objetos en una imagen y encontrar su posición se conoce como detección de objetos.

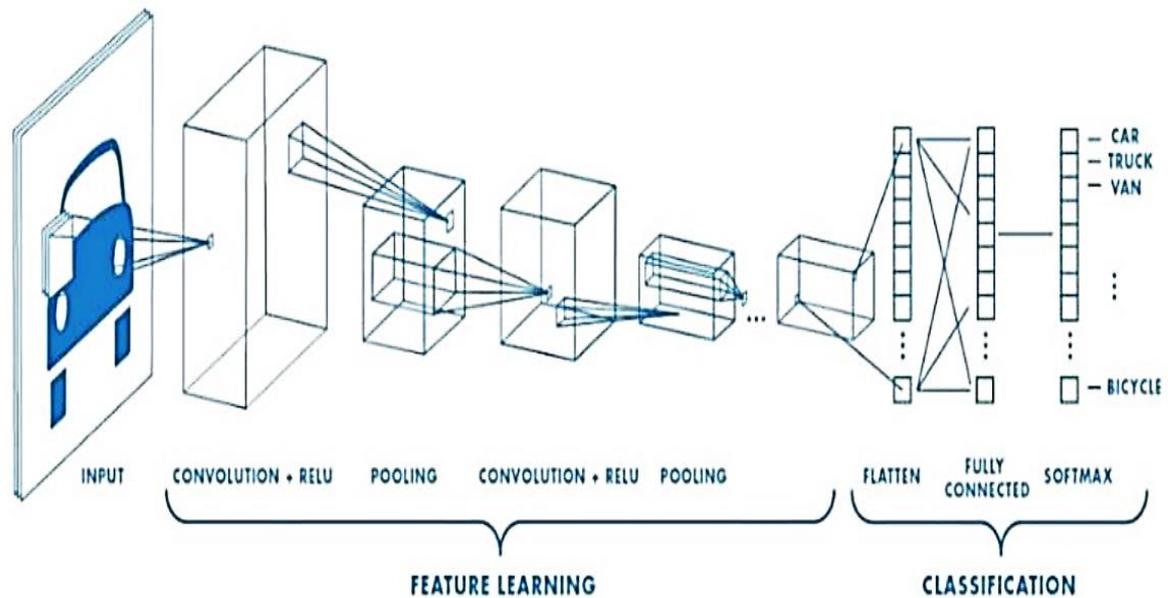


Figura 1. Descripción general de la arquitectura de CNN

Fuente: Tareas de visión por computadora moridpour, s. (2021).

CNN se compone de tres componentes importantes llamados Capa convolucional, Capa de agrupación, Capa completamente conectada como se muestra en la Fig. 1.

2.1.1.4 You Only Look Once (YOLOv5x).

Las versiones anteriores de YOLO aplican funciones softmax que convierten la puntuación en probabilidades. Este enfoque es factible cuando los objetos solo se excluyen mutuamente. YOLOv5x emplea una clasificación de etiquetas múltiples mejoradas. El clasificador logístico independiente se utiliza para calcular la probabilidad de que la entrada pertenezca a una etiqueta específica.

2.1.1.5 Optimización de cuadros delimitadores.

Al utilizar Logistic, regresión YOLOv5x predice la puntuación de presencia de objeto. Se define un cuadro de verdad del suelo para todos los objetos, si el cuadro de anclaje se superpone más con el cuadro de verdad del suelo, se dice que la puntuación de objetividad es 1.

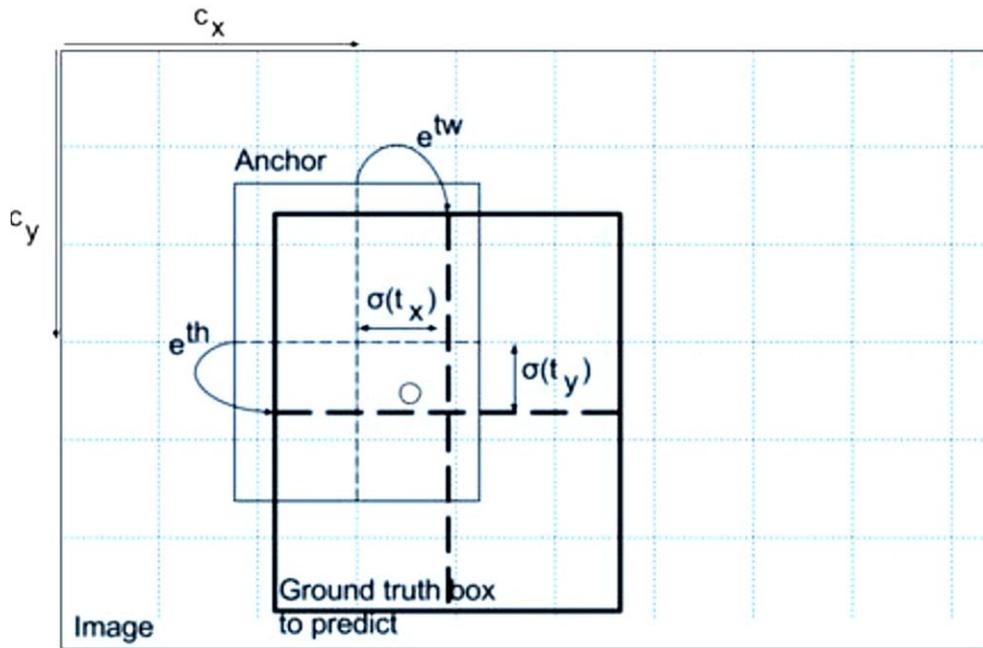


Figura. 2. Regresión de cuadro de anclaje.

Fuente: YOLOv5X definición de cuadros. Seychell, D. (2020).

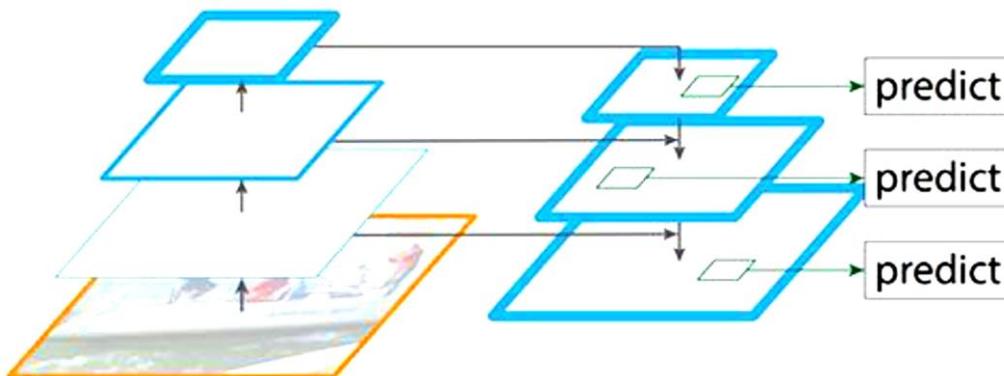


Figura. 3. Red piramidal de características.

Fuente: YOLOv5X pirámide de predicción por imágenes. Duran, Z. (2020).

Las predicciones se realizan en 3 escalas diferentes, como en la Fig. 3. La predicción inicial se realiza en la última capa del mapa de características. Luego, el mapa de características se muestra en un factor de 2. YOLOv5x fusiona el mapa de características con la característica muestreada mediante la adición de elementos. Se aplica una capa convolucional para obtener segundas predicciones. La repetición de la segunda predicción producirá una alta información semántica.

2.1.1.6 Seguimiento de objetos.

Internet es la red principal que conecta a millones de personas en el mundo. El principal factor de entretenimiento y la fuente de mayor conocimiento es el video. El video es una colección de fotogramas. El intervalo de tiempo insignificante entre los fotogramas hace que el flujo de fotos parezca un flujo de escenas. Al diseñar un algoritmo para el procesamiento de video. Los videos se clasifican en dos clases. La transmisión de video es un proceso continuo para el análisis de video. El procesador no está al tanto de futuros fotogramas.

2.1.1.7 Diseño e implementación.

Una CNN está diseñada y entrenada en el conjunto de datos de vehículos urbanos, que es un conjunto de datos indígena para aplicaciones de vigilancia del tráfico. Se tabulan según las especificaciones del conjunto de datos de vehículos urbanos o del conjunto de datos de vehículos de carretera.



Figura. 4. Imágenes de muestra del conjunto de datos de vehículos de entrenamiento

Fuente: Diseño CNN Gelvez, S. (2019).

2.1.1.8 Entrenamiento de redes neuronales.

El diagrama de flujo del entrenamiento de la red neuronal es como se muestra en la Fig. 5. El primer paso en el entrenamiento de una red usando el aprendizaje profundo para una aplicación es preparar un conjunto de datos apropiado y hacer el Tren-Prueba Split según los datos disponibles. Se diseña o selecciona una red adecuada (en el caso del aprendizaje por transferencia) para la formación.

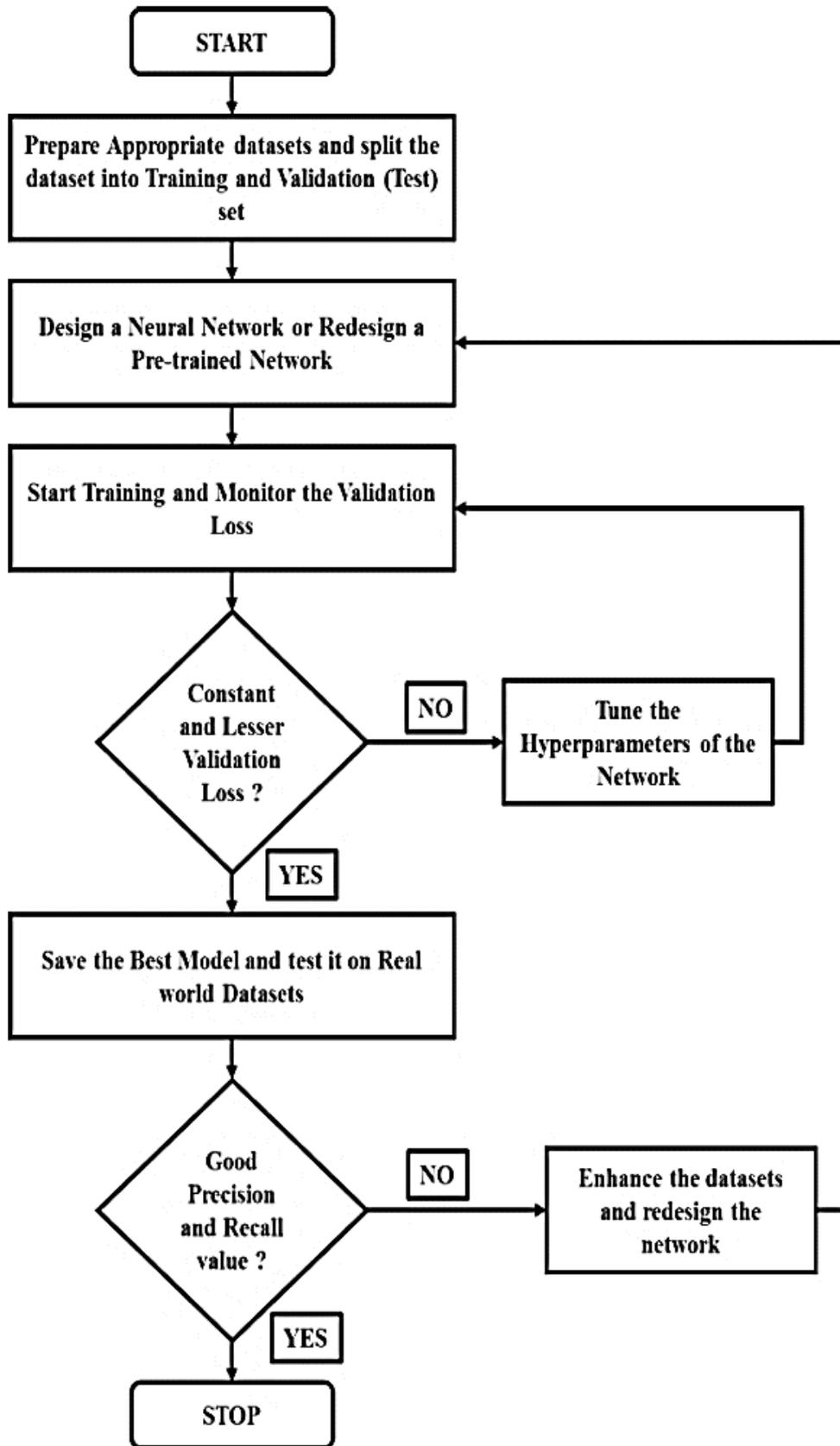


Figura. 5. Diagrama de flujo del entrenamiento de redes neuronales.

Fuente: Diagrama de entrenamiento Gelvez, S. (2019).

2.1.1.9 Detección de un solo objetivo.

La figura 5 muestra un diagrama de flujo de la detección de un solo objeto. Las bibliotecas necesarias se importan primero y los datos de entrenamiento se ingresan a través de la unidad de Google. Se utilizó Google-Colab, una herramienta de simulación en línea para algoritmos de Python y Tensor-flow. Luego, el algoritmo compila datos y aprende a formarlos de manera supervisada.

2.1.1.10 Detección de objetos múltiples.

Se proporciona una imagen como entrada al algoritmo y la transformación se realiza mediante CNN. Estas transformaciones se realizan para que la imagen de entrada sea compatible con las especificaciones del algoritmo. A continuación, se realiza la operación de aplanamiento. Aplanamiento es convertir datos en una matriz unidimensional para ingresarlos en la siguiente capa. El aplanamiento de la salida de capas convolucionales consiste en crear un único vector de características largas y está conectado al modelo de clasificación final, que se denomina capa completamente conectada. Al cambiar el umbral de puntuación, se puede ajustar cómo el modelo ML asigna estas etiquetas.

2.1.1.11 Seguimiento de objetos múltiples.

En el seguimiento de múltiples objetos, entrene al rastreador de vehículos utilizando YOLOv5x y métodos de aprendizaje profundo y optimice la tasa de éxito del detector proporcionando resultados eficientes de detección de vehículos probando el detector de vehículos entrenado en datos de prueba. Consta de seis fases, como la carga del conjunto de datos, el diseño de YOLOv5x, la configuración de las opciones de entrenamiento, el entrenamiento del rastreador de objetos y la evaluación del rastreador, respectivamente.

2.1.1.12 Métricas de desempeño.

El modelo entrenado que utiliza el aprendizaje profundo debe evaluarse por su rendimiento en datos invisibles denominados como conjunto de datos de prueba. La elección de métricas de rendimiento influirá en el análisis de algoritmos. Esto ayuda a identificar las razones de las clasificaciones erróneas para que pueda corregirse tomando las medidas necesarias.

2.1.2 Visión Artificial.

2.1.2.1 Definición.

La visión artificial es una disciplina científica que incluye métodos para adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por una máquina.

2.1.2.2 Sistemas de visión artificial.

Un sistema de visión artificial consta de varios componentes, desde la cámara que capta una imagen para su inspección hasta el propio motor de procesamiento que renderiza y comunica el resultado, en detalle: Sensores de visión (incluye los diferentes tipos de cámaras y sensores de luz en todas las frecuencias, sensores 2D y 3D), cámaras inteligentes (incluyen un procesador capaz de analizar las imágenes y proporcionar resultados con los que tomar decisiones) y sistemas de visión avanzados (son los dispositivos, por ejemplo, los cobots, que integran los sistemas de visión artificial para realizar tareas específicas).

2.1.2.3 Funcionamiento de un sistema de visión artificial.

Para que un sistema de visión artificial funcione correctamente es necesario que se cumplan una serie de condiciones y requisitos: Iluminación, puesta en escena, cámaras, procesador de imagen, comunicación.

2.1.2.4 Aplicaciones de la visión artificial.

Para la detección de objetos (comparación de una imagen con patrones conocidos y almacenados en una base de datos para detectar objetos), reconocimiento óptico de caracteres (OCR), robots con visión artificial.

3 METODOLOGÍA

3.1. Procedimiento para la ejecución de los códigos

3.1.1 Ejecución de YoloV5x.

You Only Look Once (YOLO) es un método utilizado para la detección de objetos en tiempo real. Cuando YOLO obtiene una imagen de entrada, hace que se divida la imagen en cuadrícula $S \times S$ donde S es un valor aleatorio para la cantidad de cuadrículas, y donde cada cuadrícula de celda produce cuadros delimitadores y les da una puntuación de confianza, cada celda también hace una predicción de clase.

El estudio se realizó en la nube de Google Colab, posteriormente se editó el código para la lectura en PyCharm. El algoritmo, tomado del conjunto de datos COCO de pesos que se entrenan en 80,000 imágenes de entrenamiento, el cual fue modificado de acuerdo al propósito de nuestra investigación, así que se creó una carpeta “data” la cual se entrenó con las clases de vehículos que se seleccionaron (motocicleta, mototaxi, automóvil, camioneta, microbús y vehículos pesados).

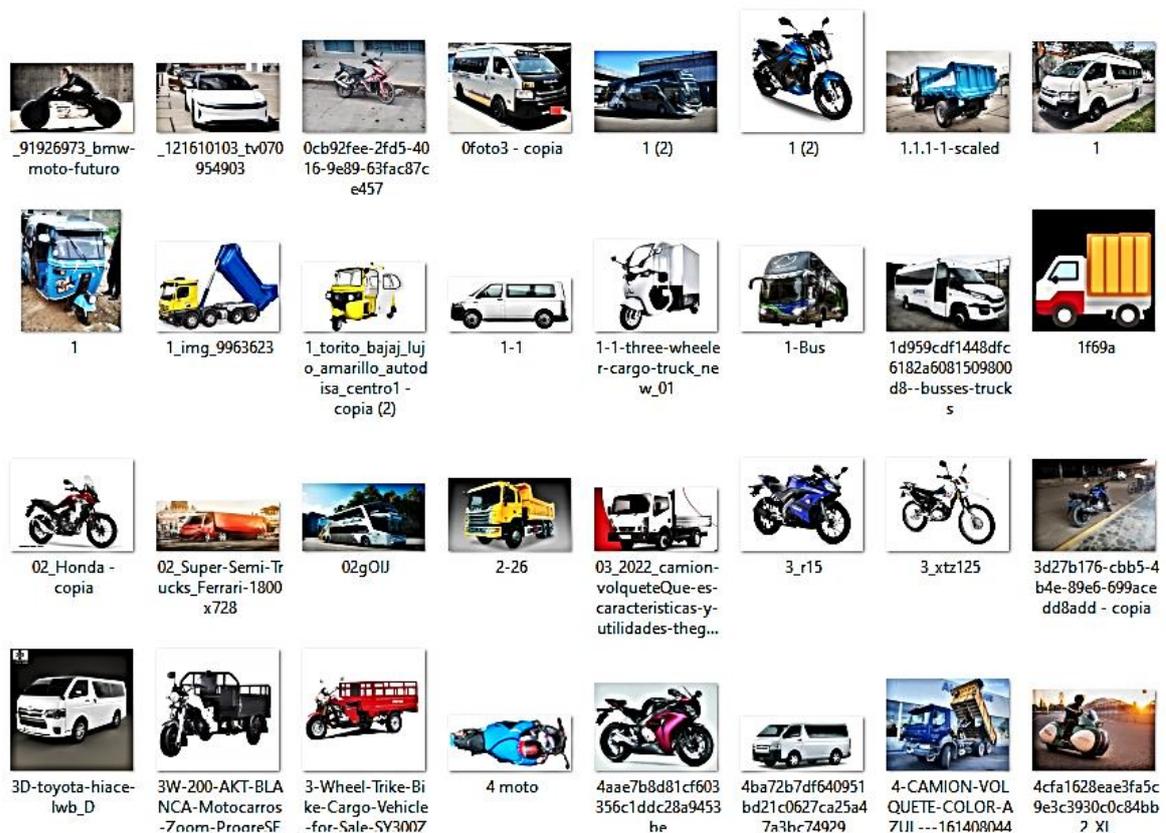


Figura 6: Imágenes recolectadas para el entrenamiento.

Fuente: Elaboración propia.

Parte del trabajo fue recolectar imágenes tomando fotos de motocicletas, mototaxis, automóviles, camionetas, microbuses y vehículos pesados para volver a entrenar y tener un algoritmo que los identifique. Estas imágenes se ingresaron en un solo formato y para este proceso de selección, se trabajó con Jupyter que mediante una línea de código ayuda a clasificar las imágenes en formato JPEG. Que normalmente es el formato indicado para este tipo de trabajos. De la misma manera en este trabajo se utilizó el programa virtual Make Sense el cual permite seleccionar imágenes y poner etiqueta según sea la imagen y de esta manera podemos indicar el tipo de imagen a reconocer en una predicción o clasificación. Con la data preparada y lista procedemos a ejecutar el entrenamiento con la siguiente línea de código, usando la red neuronal YOLOv5x en Google Colab, con los detalles que se muestran:

```
[ ] !python train.py --img 416 --batch 16 --epochs 50 --data /content/yolov5/yolov5/data/custom.yaml --weights yolov5x.pt --d
```

Figura 7: Línea de código.

Fuente: Elaboración propia.

Luego de haberse ejecutado nos indicara los resultados del entrenamiento con su respectiva precisión por tipo de vehículo, los cuales se muestran a continuación:

```
Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 444 layers, 86207059 parameters, 0 grad
  Class      Images  Labels  P
  all        1200    1200    0.909
  motocicleta 1200    203     0.965
  mototaxi    1200    200     0.963
  automovil   1200    192     0.861
  camioneta   1200    206     0.911
  microbus    1200    169     0.817
  veh.pesados 1200    230     0.935
Results saved to runs/train/exp
```

Figura 8: Resultado de precisión de entrenamiento de red neuronal.

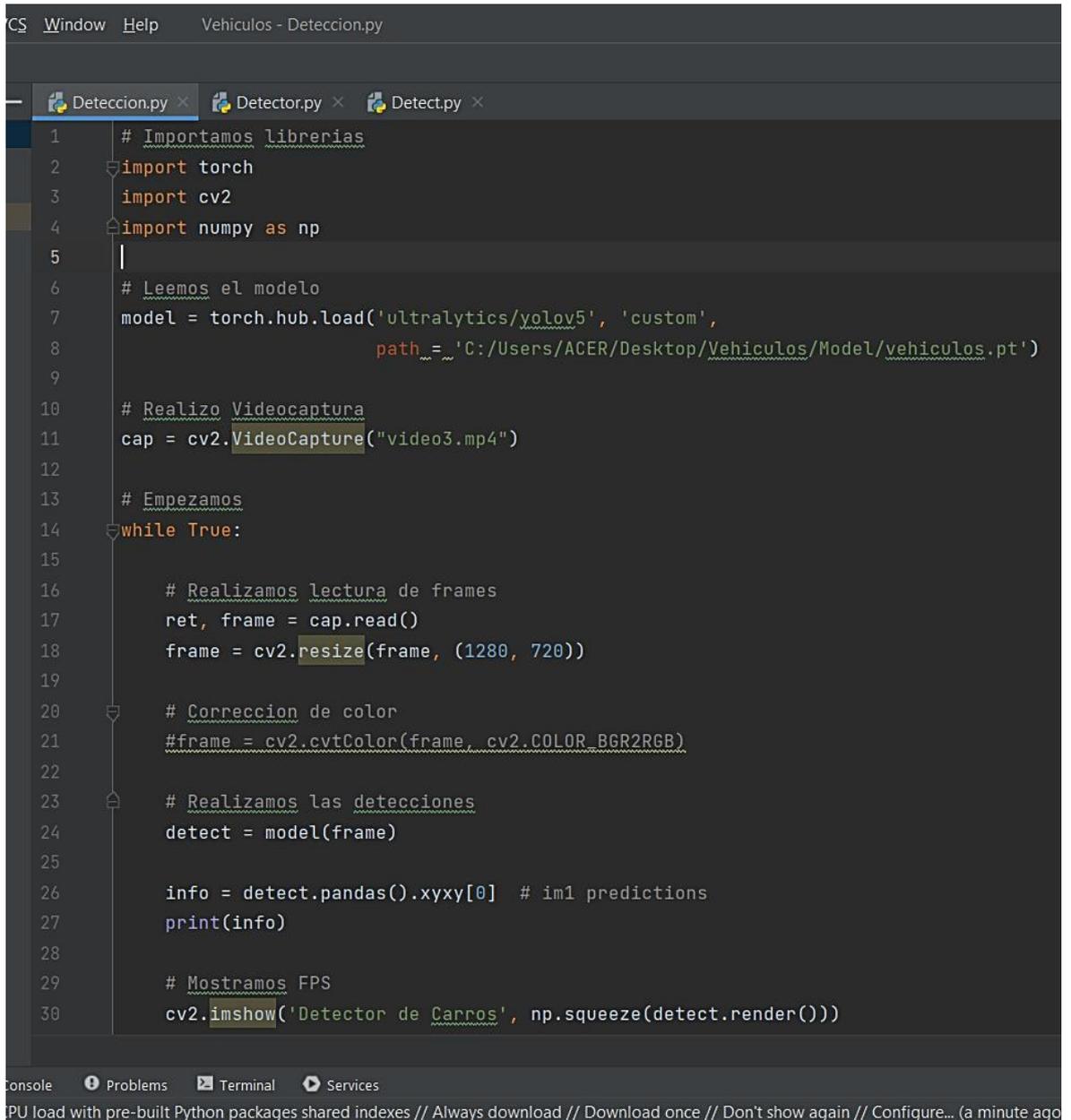
Fuente: Elaboración propia.

Como podemos observar, la precisión del entrenamiento por tipo de vehículo es alta, lo que nos indica que nuestro entrenamiento está listo para poder ejecutarlo con un código que nos permita leer y poner a prueba el video que deseamos comparar. Por ello procedemos a descargar nuestros resultados de entrenamiento y preparar nuestro código fuente para lecturarlo.

3.1.2 Código Inteligencia Artificial.

Para ejecutar nuestro entrenamiento debemos hacerlo con un código que podamos utilizar para insertar nuestro video y que tenga todas las librerías importadas, estas son: Torch, Cv2 y Numpy.

Procedemos a leer el modelo que entrenamos en Google Colab, para posteriormente ejecutar nuestro video; escribimos diferentes líneas de código de acuerdo al propósito del estudio para luego correrlo. A continuación, se muestra el código utilizado:



```
CS Window Help Vehiculos - Deteccion.py
Deteccion.py x Detector.py x Detect.py x
1 # Importamos librerias
2 import torch
3 import cv2
4 import numpy as np
5
6 # Leemos el modelo
7 model = torch.hub.load('ultralytics/yolov5', 'custom',
8                        path_='./C:/Users/ACER/Desktop/Vehiculos/Model/vehiculos.pt')
9
10 # Realizo Videocaptura
11 cap = cv2.VideoCapture("video3.mp4")
12
13 # Empezamos
14 while True:
15
16     # Realizamos lectura de frames
17     ret, frame = cap.read()
18     frame = cv2.resize(frame, (1280, 720))
19
20     # Correccion de color
21     #frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
22
23     # Realizamos las detecciones
24     detect = model(frame)
25
26     info = detect.pandas().xyxy[0] # im1 predictions
27     print(info)
28
29     # Mostramos FPS
30     cv2.imshow('Detector de Carros', np.squeeze(detect.render()))
console Problems Terminal Services
GPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (a minute ago)
```

Figura 9: Código inteligencia artificial.

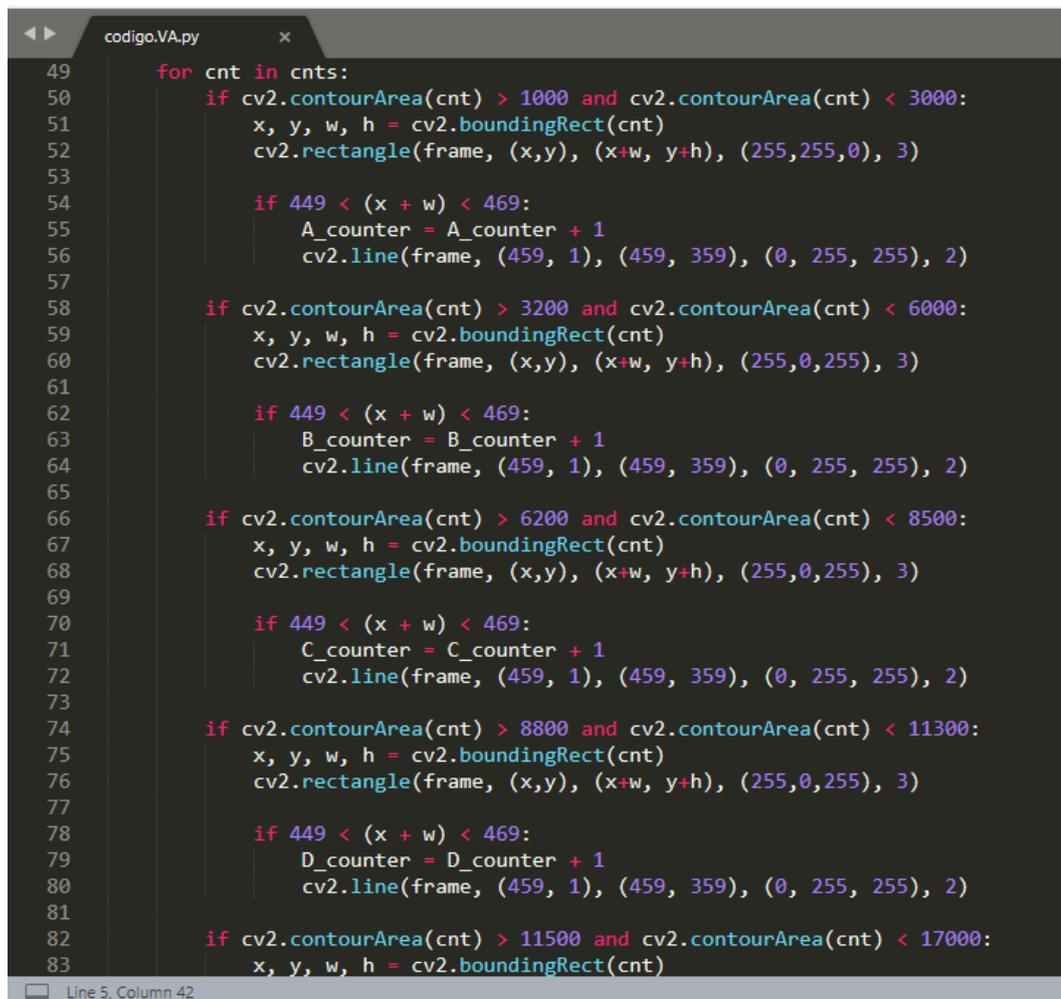
Fuente: Elaboración propia.

3.1.3 Código Visión Artificial.

Para el conteo y clasificación mediante visión artificial usaremos bibliotecas y/o librerías virtuales, para lo cual instalamos nuestro lenguaje de programación Python y un editor de texto.

Previo a la edición de nuestro código se grabó un video el cual compararemos, dicho video se grabó a una altura lo suficientemente alta para poder capturar los vehículos con claridad y completos, ya que algunos vehículos no entraban en el foco de la cámara digital se optó por hacer una remasterización del video grabado, esto porque nuestro método consiste en realizar la detección por áreas.

Con el video listo procedemos a instalar las librerías: OpenCv2, Numpy e Imutils, para luego lecturar el video a poner a prueba; se definieron diferentes áreas de acuerdo al tipo de vehículo como se muestra en la siguiente imagen:



```
49     for cnt in cnts:
50         if cv2.contourArea(cnt) > 1000 and cv2.contourArea(cnt) < 3000:
51             x, y, w, h = cv2.boundingRect(cnt)
52             cv2.rectangle(frame, (x,y), (x+w, y+h), (255,255,0), 3)
53
54             if 449 < (x + w) < 469:
55                 A_counter = A_counter + 1
56                 cv2.line(frame, (459, 1), (459, 359), (0, 255, 255), 2)
57
58         if cv2.contourArea(cnt) > 3200 and cv2.contourArea(cnt) < 6000:
59             x, y, w, h = cv2.boundingRect(cnt)
60             cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,255), 3)
61
62             if 449 < (x + w) < 469:
63                 B_counter = B_counter + 1
64                 cv2.line(frame, (459, 1), (459, 359), (0, 255, 255), 2)
65
66         if cv2.contourArea(cnt) > 6200 and cv2.contourArea(cnt) < 8500:
67             x, y, w, h = cv2.boundingRect(cnt)
68             cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,255), 3)
69
70             if 449 < (x + w) < 469:
71                 C_counter = C_counter + 1
72                 cv2.line(frame, (459, 1), (459, 359), (0, 255, 255), 2)
73
74         if cv2.contourArea(cnt) > 8800 and cv2.contourArea(cnt) < 11300:
75             x, y, w, h = cv2.boundingRect(cnt)
76             cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,255), 3)
77
78             if 449 < (x + w) < 469:
79                 D_counter = D_counter + 1
80                 cv2.line(frame, (459, 1), (459, 359), (0, 255, 255), 2)
81
82         if cv2.contourArea(cnt) > 11500 and cv2.contourArea(cnt) < 17000:
83             x, y, w, h = cv2.boundingRect(cnt)
```

Figura 10: Delimitación de áreas, código visión artificial.

Fuente: Elaboración propia.

4 RESULTADOS Y DISCUSIÓN

4.1. Método Inteligencia Artificial

Se obtuvieron resultados con una precisión alta, sin embargo, existe una variación la cual analizaremos en la comparación; el código no nos proporciona una base de resultados, por lo cual estos fueron sacados durante la ejecución, los cuales se muestran a continuación:

Tabla 1.

Resultados de clasificación y conteo con Inteligencia Artificial.

Clases de Vehículos	Cuantificación
Motocicleta	45
Mototaxi	27
Automóvil	92
Camioneta	83
Microbús	87
Vehículos Pesados	26

Nota: Fuente: Elaboración propia.

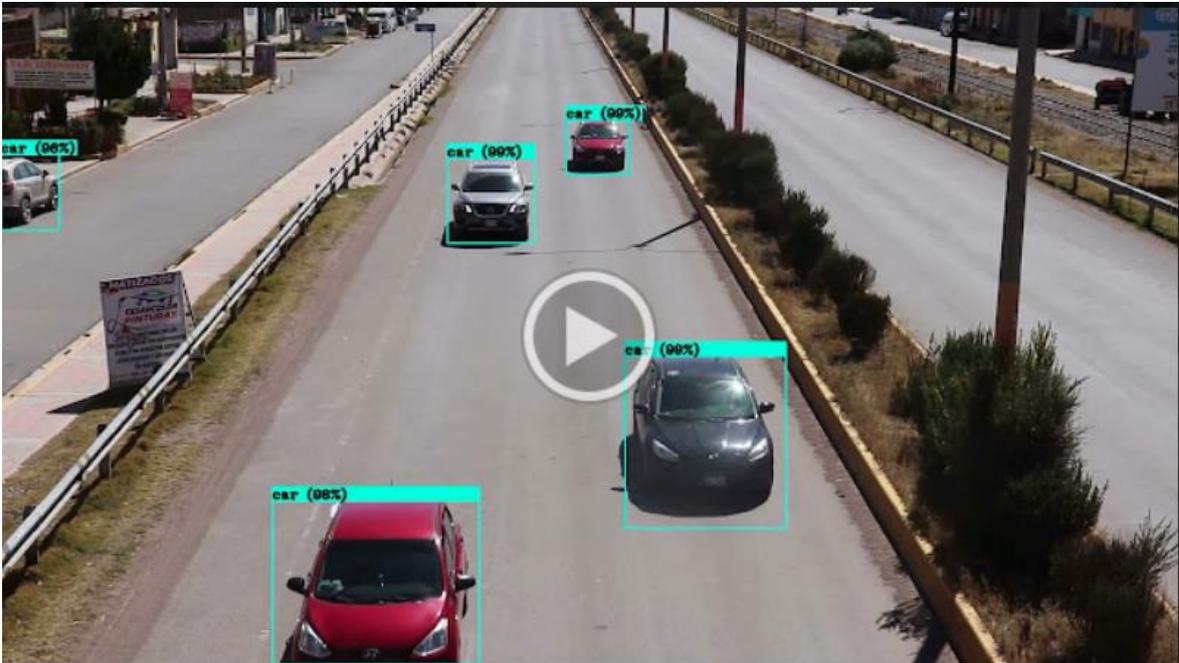


Figura 11: Resultado lectura de código inteligencia artificial.

Fuente: Elaboración propia.

4.2. Método Inteligencia Artificial

Con este método también se obtuvieron resultados con una precisión alta, sin embargo, también existe una variación la cual analizaremos en la comparación; el código en su pantalla de ejecución nos muestra el conteo y clasificación mientras estos se van contando, los resultados se muestran a continuación:

Tabla 2.

Resultados de clasificación y conteo con Visión Artificial.

Clases de Vehículos	Cuantificación
Motocicleta	44
Mototaxi	28
Automóvil	116
Camioneta	81
Microbús	89
Vehículos Pesados	30

Nota: Fuente: Elaboración propia.

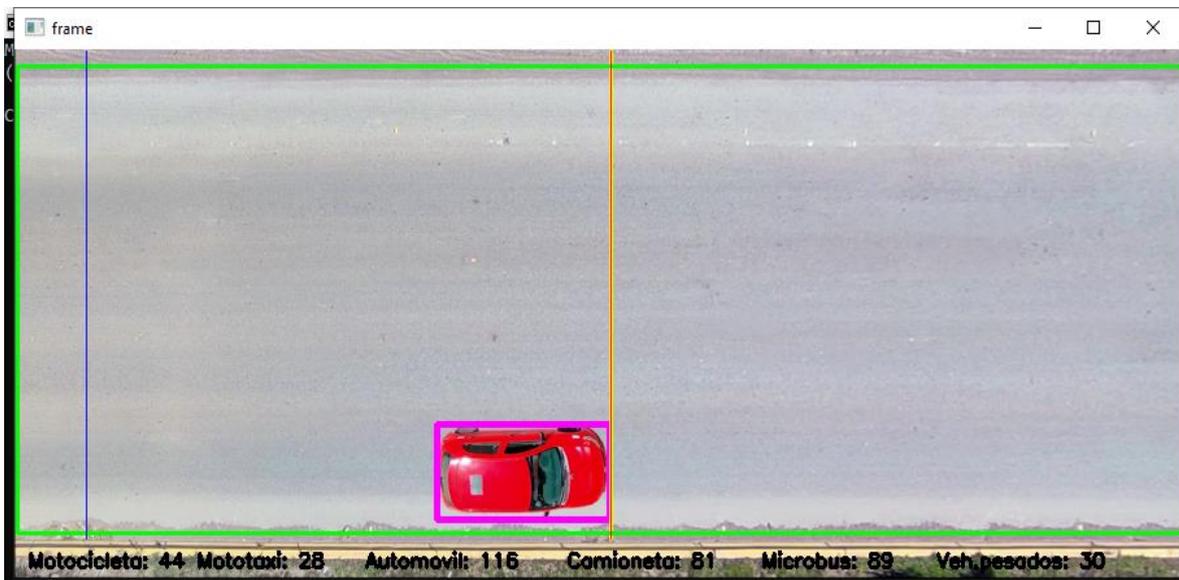


Figura 12: Resultado código visión artificial.

Fuente: Elaboración propia.

4.3. Estudio In Situ

Se realizó la detección, conteo y clasificación de los vehículos con el método tradicional, para posteriormente realizar una comparación con los métodos investigados. Estos estudios

se realizaron en el lapso de una hora en un punto de prueba el cual tiene un amplio espacio para poder grabar videos de altura sin las distracciones o interrupciones de otro tipo de contenido.

Los resultados de este estudio se observan a continuación:

Tabla 3.

Resultados del Estudio In Situ

<i>Clases de Vehículos</i>	<i>Cuantificación</i>
Motocicleta	48
Mototaxi	29
Automóvil	108
Camioneta	90
Microbús	91
Vehículos Pesados	28

Nota: Fuente: Elaboración propia.

4.4. Comparación de Resultados

Tabla 4.

Comparación de resultados.

<i>Clases</i>	<i>Inteligencia Artificial (IA)</i>			<i>Visión Artificial (VA)</i>			<i>In Situ (IS)</i>	
	<i>Conteo IA</i>	<i>% IA</i>	<i>%Error IA</i>	<i>Conteo VA</i>	<i>% VA</i>	<i>%Error VA</i>	<i>Conteo IS</i>	<i>% IS</i>
Motocicleta	45	93.75%	6.25%	44	91.67%	8.33%	48	100%
Mototaxi	27	93.10%	6.90%	28	96.55%	3.45%	29	100%
Automóvil	92	85.19%	14.81%	116	107.41%	-7.41%	108	100%
Camioneta	83	92.22%	7.78%	81	90.00%	10.00%	90	100%
Microbús	87	95.60%	4.40%	89	97.80%	2.20%	91	100%
Vehículos Pesados	26	92.86%	7.14%	30	107.14%	-7.14%	28	100%
TOTAL	360	91.37%	8.63%	388	98.48%	1.52%	394	100%

Nota: Fuente: Elaboración propia.

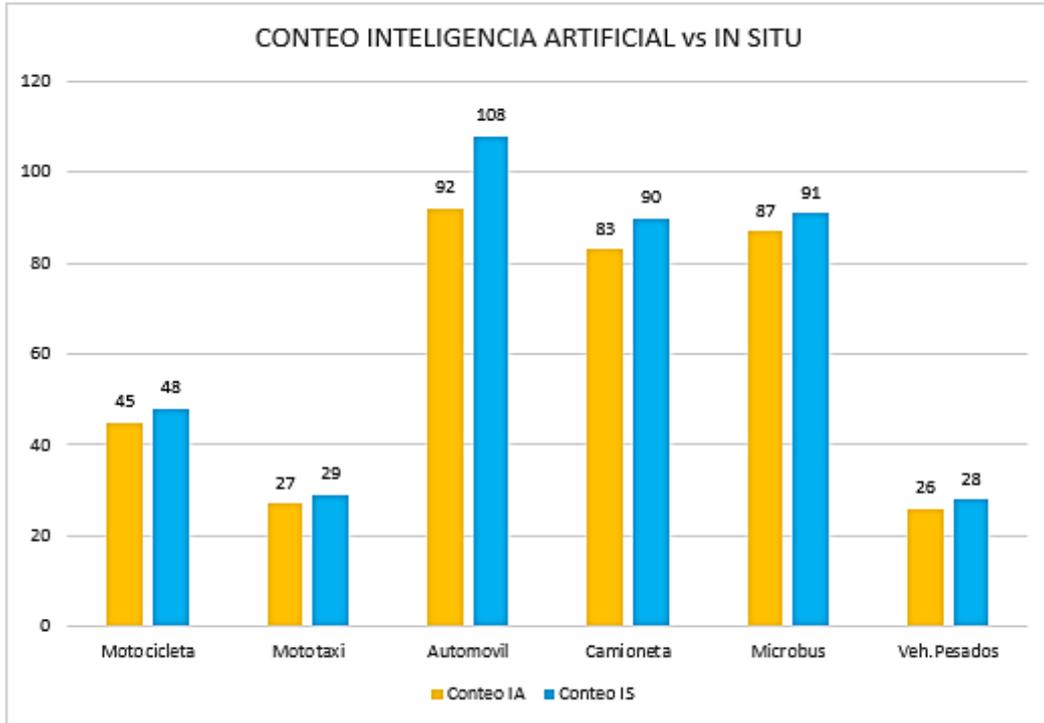


Figura 13: Comparación de resultados: Conteo IA vs IS.

Fuente: Elaboración Propia.

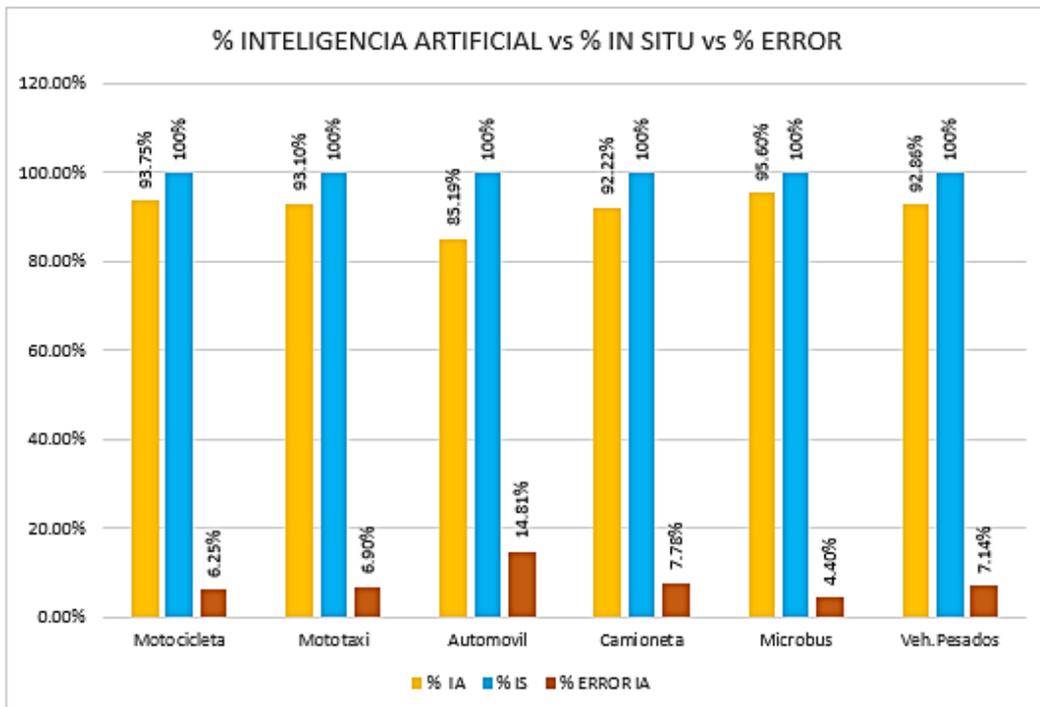


Figura 14: Comparación de resultados: % IA vs %IS vs %Error

Fuente: Elaboración Propia.

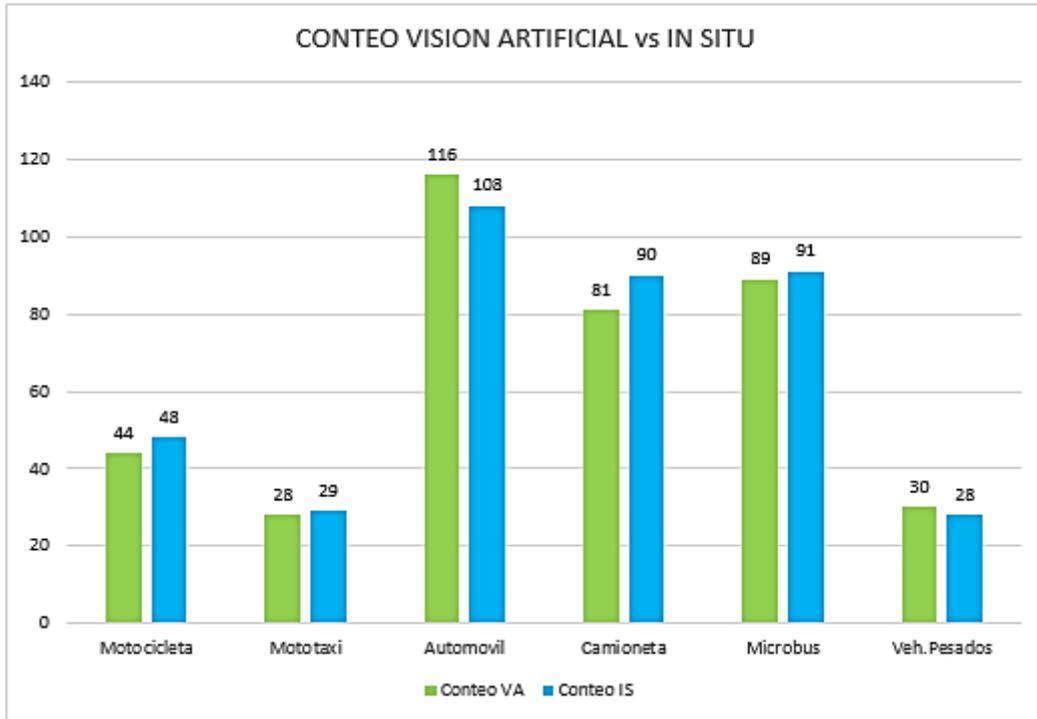


Figura 15: Comparación de resultados: Conteo VA vs IS.

Fuente: Elaboración Propia.

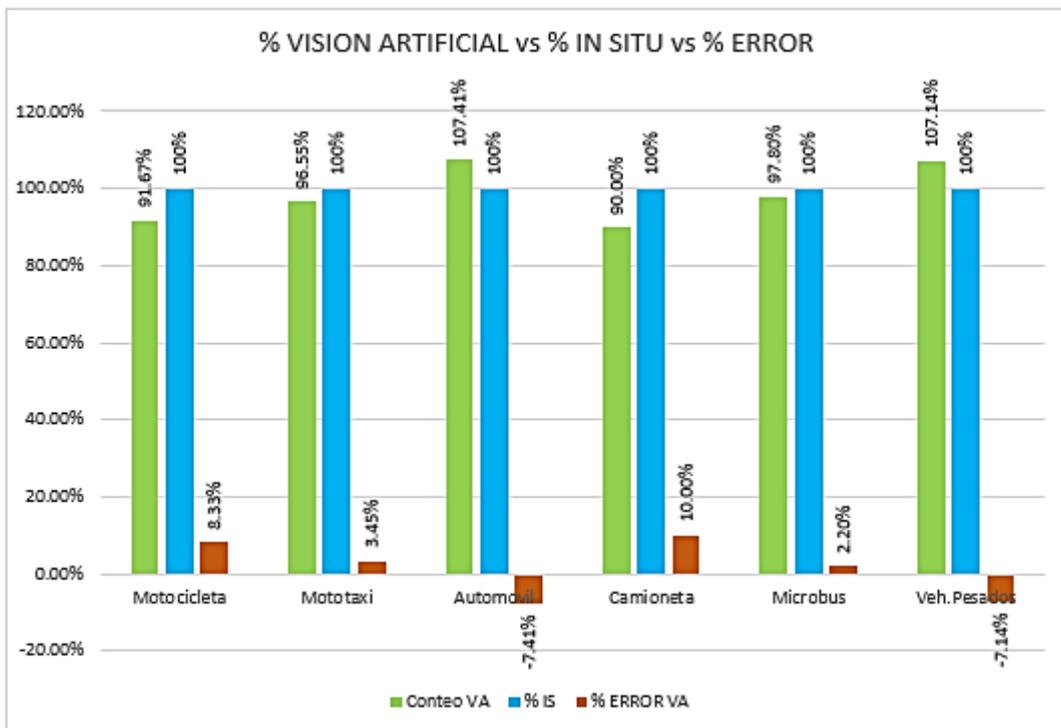


Figura 16: Comparación de resultados: % VA vs %IS vs %Error

Fuente: Elaboración Propia.

4.5. Discusión

4.5.1 De la Inteligencia Artificial.

De los resultados del método usado con IA podemos observar que de acuerdo al entrenamiento que hicimos la precisión es alta y podría mejorar si se entrena el algoritmo con una base de datos más amplia, por otro lado, se tuvieron inconvenientes en la ejecución del modelo con nuestro video prueba, pudimos identificar que este problema se dio por el ordenador que usamos, ya que mientras nuestro código se va ejecutando se necesita tener una tarjeta gráfica más optima porque de lo contrario la reproducción del video es muy lenta; por tal caso, antes de implementar este método en campo debemos asegurarnos de tener un mejor ordenador.

4.5.2 De la Visión Artificial.

De los resultados del método usado con VA podemos deducir que aparentemente es el método más optimo, sin embargo, para la clasificación vehicular con esta metodología debemos definir áreas para que el código realice la detección, lo cual no es tan recomendable usar en lugares donde hay movimiento de otros objetos como por ejemplo: personas, animales, entre otros; porque al realizar una detección y clasificación por áreas, el código podría reconocer otro objeto que tenga el mismo área que nuestro vehículo, además para usar este método debemos tener cámaras digitales fijas ya que ante algún movimiento el código falla y detecta otros objetos; en el presente estudio se fijó la cámara digital a una altura necesaria para que los áreas de los vehículos no varíe de acuerdo al ángulo de enfoque, lo que nos ayudó a tener resultados más precisos.

5 CONCLUSIONES Y RECOMENDACIONES

Se ha desarrollado dos métodos inteligentes para la obtención del aforo vehicular, el primer método basado en inteligencia artificial ha propuesto un algoritmo de detección y seguimiento vehicular, que puede detectar y rastrear vehículos en tiempo real y en video, este puede mejorar efectivamente la precisión de la extracción de características en video y también mejorar en gran medida la calidad de precisión de detección. En este artículo, las características de los vehículos se extraen y compilan utilizando la red neuronal YOLOv5x. Así mismo, se aplicó la visión artificial mediante el uso de un código para clasificar y contar los vehículos de acuerdo a sus áreas promedio respectivas, se importaron librerías como cv2, numpy e imutils, las cuales eran necesarias para la ejecución del código; la aplicación de este método tiene deficiencias con respecto a la forma de detección, ya que si se pone a prueba en lugares donde hay presencia de otros objetos en movimiento el reconocimiento fallaría y podría identificar otros objetos que tengan el mismo área planteada para cada vehículo; además, se debe tener en cuenta que para ejecutar este método la cámara digital a usar debe estar estática sin presenciar ningún movimiento para que no varíe el ángulo de enfoque.

En resumen, los resultados obtenidos con inteligencia artificial respecto al estudio realizado in situ fueron buenos alcanzando un 95.60% en el conteo y clasificación de microbuses, con un porcentaje de error del 4.40%, mientras en los otros tipos de vehículos hubo un porcentaje menor, sin embargo, no muy lejos del más alto; los resultados obtenidos con visión artificial respecto al estudio hecho en campo fueron buenos también, alcanzando un 97.80% en el conteo y detección de microbuses, con un porcentaje de error del 2.20%; los demás vehículos alcanzaron un porcentaje menor o incluso mayor por algunos errores en la lectura del código. Del conteo general mediante el estudio manual in situ se tiene por resultado la cuantificación de 394 vehículos el resultado equivale al 100% del flujo vehicular; con el método de inteligencia artificial se contaron 360 vehículos que equivalen al 91.37% respecto al estudio in situ y con el método de visión artificial se contaron 388 vehículos que equivale al 98.48% respecto al estudio in situ. De esta manera se tiene con el método de inteligencia artificial un 8.63% de error en el conteo respecto al estudio in situ y con el método de visión artificial un 1.52% de error en el conteo general respecto al estudio in situ.

Se recomienda probar la metodología de inteligencia artificial en diferentes ambientes climáticos y lugares con menor iluminación ya que esto podría dificultar el procesamiento

de datos durante la ejecución del código entrenado con redes neuronales convolucionales. Además, probar este método con diferentes ángulos de enfoque de la cámara digital. En unas próximas investigaciones se recomienda realizar un entrenamiento de la red neuronal con una base de datos más amplia para una mejor precisión y contar con una tarjeta gráfica más óptima para la ejecución del código basado en Inteligencia Artificial, también probar la detección y clasificación en tiempo real. Otro de los temas podría ser una investigación en cuanto a la comparación de bajos costos entre estas metodologías y las tradicionales.

REFERENCIAS

- Aguirre, E. (2017). Medicion de trafico mediante Iot y SIG. Caldas, Colombia.
- Akhtar, M. (2021). A Review of Traffic Congestion Prediction Using Artificial Intelligence. *Journal of Advanced Transportation*.
- Atoche, W. (2013). Transito en una carretera usando simulacion. *LACCEI Latin American and Caribbean Conference for Engineering and Technology*.
- Bugeja, D. A. (2020). Comparison of Vehicle Detection Techniques applied to IP Camera Video Feeds for use in Intelligent Transport Systems. *Transportation Research Procedia*, 45, 971-978.
- Cepni, A. &. (2020). Vehicle detection using different deep learning algorithms from image sequence. *Baltic Journal of Modern Computing*, 347-358.
- Gelvez. (2019). Redes neuronales convolucionales y redes neuronales recurrentes en la transcripcion automatica. *July*, 1-8.
- Mandal, M. J.-g. (2020). Sustainability Artificial Intelligence-Enabled Traffic Monitoring System. *Mpdi*.
- Milutinovic, J. M. (2016). Platform dependent efficiency of a monte carlo code for tissue neutron dose assessment. *RAD Conference Proceedings*, 21-25.
- N. K, &. D. (2021). Optimized YOLOv2 based vehicle classification and tracking for intelligent transportation system. *Results in Control and Optimization*, 2.
- Ocampo, B. V. (2012). Metodos de modelamiento y prediccion de trafico orientados a plataformas de transmision de video e IPTV usando series de tiempo. *Revista cientifica ISSN 0124 2253*,12.
- Redmon, D. G. (2016). YOLOv1. *Cvpr*, 779-788.
- Rosado, Q. &. (2014). Desarrollo ágil de software aplicando programacion extrema. *Revista Ingenio Universidad Francisco de Paula Santander Ocaña*, 24-29.
- Singh, M. M. (2019). Multi-layer pruning framework for compressing Single Shot MultiBox detector. *Winter Conference on Applications of Computer Vision*, 1318-1327.
- Song, L. L. (2019). Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*.
- Yin, W. W. (2021). Deep Learning on Traffic Prediction: Methods, Analysis and Future Directions. *IEEE Transactions on Intelligent Transportation Systems*, 1-16.