UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

TRABAJO DE INVESTIGACIÓN
AÑO 2021

TRABAJO DE INVESTIGACIÓN PARA OPTAR EL GRADO ACADÉMICO DE BACHILLER EN INGENIERÍA ELECTRÓNICA

TÍTULO	: IMPLEMENTACIÓN DE UNA RED NEURONAL ARTIFICIAL CONVOLUCIONAL PARA LA DETECCIÓN Y CONTEO DE VEHÍCULOS EN UNA RECREACIÓN DE ESTACIONAMIENTO ABIERTO
LÍNEA DE INVESTIGACIÓN	: PROCESAMIENTO DIGITAL, VISIÓN E INTELIGENCIA ARTIFICIAL
NOMBRES Y APELLIDOS DEL ESTUDIANTE	: GIAN FRANCO GONZALES DEL VALLE ROMERO WALTER JESÚS NEYRA ESPINOZA
CÓDIGO DEL ESTUDIANTE	: 201610345 201611217
CICLO DE INGRESO	: 2016-1 2016-1
DOCENTE DE LA ASIGNATURA	: PEDRO FREDDY HUAMANÍ NAVARRETE

_____ _____

FIRMA DEL ESTUDIANTE **FIRMA DEL DOCENTE**

_____

VºBº DIRECTOR DE ESCUELA

ÍNDICE

INTRODUCCIÓN

En la actualidad, el área de aplicación de las redes neuronales artificiales se ha extendido considerablemente. Esto se debe al avance tecnológico que permite desarrollar algoritmos más complejos, los que conlleva a la creación de nuevos tipos de redes neuronales artificiales, teniendo como resultado la especialización de estas redes para ciertas aplicaciones en concreto. Una de ellas son las llamadas redes neuronales convolucionales, que utilizan filtros de convolucion que les permiten detectar, reconocer y generalizar patrones y formas dentro de una imagen imitando a la corteza visual del cerebro, pudiéndole dar la capacidad de "ver" y reconocer objetos a una máquina, esto se le conoce como visión computacional o artificial. Esta capacidad que ahora se les puede dar a las máquinas ha sido explotada de distintas maneras, por ejemplo en sistemas de seguridad, control de calidad de productos, conducción autónoma de vehículos, detección temprana de enfermedades, etc. En el presente trabajo de investigación se utiliza la visión artificial como una alternativa para un sistema de detección y conteo de vehículos en un estacionamiento abierto.

1. PLANTEAMIENTO DEL PROBLEMA

En muchos estacionamientos, no está implementado un sistema de control de parqueo con la finalidad de controlar la entrada y salida de estos, en especial en donde hay una gran afluencia de vehículos y en donde la espera en la entrada o salida de estos aparcamientos pueden generar atascos.

La finalidad de este trabajo es implementar una red neuronal convolucional que permita detectar y contar los vehículos estacionados para obtener la cantidad de espacios libres de estacionamiento.

1.1. Formulación y determinación del problema

1.1.1. Problema General:

¿Cómo implementar una red neuronal artificial convolucional para la detección y conteo de vehículos en un estacionamiento abierto?

1.1.2. Problemas Específicos:

- ¿Qué técnicas de procesamiento de imágenes utilizar en la implementación de la red neuronal convolucional que permita una detección y reconocimiento óptimo?
- ¿Es posible implementar una red neuronal convolucional utilizando la arquitectura de YOLO para la detección de vehículos?
- ¿Cómo programar un buffer de datos que permita contar los vehículos detectados?

1.2. Importancia y justificación del trabajo de investigación

En referencia al presente trabajo de investigación, su importancia radica en implementar una red neuronal convolucional que pueda detectar y contar vehículos dentro de un estacionamiento abierto y que pueda ser utilizado a futuro en una interfaz gráfica para el aviso de disponibilidad de espacios del estacionamiento. La justificación reside en que, gracias a la red neuronal se ahorraría tiempo a los conductores y evitaría congestionamiento en la entrada y la salida del estacionamiento.

1.3 Limitaciones del estudio

En la presente investigación se tomó un video de una sección estacionamiento de espacio abierto, debido a que solo se dispondrá de una cámara y la condición de luz del día. Debido a la limitación en hardware de los equipos utilizados, se redujo el área de estacionamiento a solo 8 espacios.

1.4. Objetivos

1.4.1. Objetivo General:

Implementar una red neuronal artificial para la detección y conteo de vehículos en un estacionamiento abierto.

1.4.2. Objetivos Específicos:

- Utilizar técnicas de procesamiento de imágenes en la implementación de la red neuronal convolucional para una detección y reconocimiento óptimo.
- Implementar una red neuronal convolucional utilizando la arquitectura de YOLOv3 para la detección de vehículos.
- Programar un buffer de datos para la operación de conteo de los vehículos detectados

2. MARCO TEÓRICO

2.1. Marco conceptual

a) <u>Procesamiento de imágenes:</u>

Redimensionamiento de imágenes:

Es una técnica de procesamiento de imagen donde se manipulan los píxeles que la conforman de tal manera que se incremanta o se disminuye su escala, es decir, la relación del número de pixeles que tiene por ancho entre la cantidad que tiene de alto. Para lograr esto, el algoritmo añade o quita píxeles a la imagen, que puede dar como resultado un emborronamiento o emprobrecimiento de la calidad de los detalles de la imagen.

Filtro de medias no locales para reducción del ruido:

El filtro de medias no locales utiliza todas las posibles autopredicciones y autosimilitudes que puede proporcionar la imagen para determinar las ponderaciones de los píxeles para filtrar la imagen ruidosa, partiendo de la base de que la imagen contiene una gran cantidad de autosimilitudes. Como los píxeles están muy correlacionados y el ruido suele estar distribuido de forma independiente e idéntica, al promediar estos píxeles se suprime el ruido y se obtiene un píxel similar a su valor original. El filtro de medias no locales elimina el ruido y limpia los bordes sin perder demasiada estructura y detalles finos [2].

Una imagen con algún tipo de ruido se representa de la siguiente manera:

$$v(i) = u(i) + n(i)$$

Donde $v(i)$ es lo que se observa en la imagen o los valores de los pixeles apreciados, $u(i)$ es el valor verdadero y $n(i)$ representa al ruido que contiene una imagen.
Dada una imagen con ruido $v(i)$, el valor estimado del algoritmo de filtro de medias no locales para un pixel i, se calcula como un promedio ponderado de todos los píxeles dentro de la imagen:

$$NL[v](i) = \sum_{j \in I} w(i,j)v(j)$$

Donde el grupo de pesos $\{w(i,j)\}_j$ depende de la similitud de los pixeles i y j. Esta similitud de estos dos píxeles depende de la similitud de los vectores de nivel de intensidad de gris $v(N_i)$ y $v(N_j)$, donde N_k se refiere a una proximidad cuadrada de tamaño fijo y centrada en un píxel k. Para una imagen a color se realiza el mismo procedimiento para cada uno de los tres canales que contiene el pixel.

Figura 1: Filtro de medias no locales. Donde w(p,q_i) son los pesos y p, q_i son píxeles.
Fuente: https:// https://www.researchgate.net/

Esta similitud se mide como una función decreciente de la distancia euclidiana ponderada [3]:

$$\left\| v(N_i) - v(N_j) \right||^2_{2,a}$$

Donde $a > 0$ es la desviación estandar del núcleo gaussiano. Mientras que los pesos están definidios mediante la siguiente ecuación:

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i)-v(N_j)||^2_{2,a}}{h^2}}$$

Donde $Z(i)$ es una constante de normalización:

$$Z(i) = \sum_j e^{-\frac{\|v(N_i)-v(N_j)||^2_{2,a}}{h^2}}$$

Y el parámetro h funje como un grado de filtración. Controla el decaimiento de la función exponencial y, en consecuencia, el decaimiento de los pesos en función de las distancias euclidianas, de esta manera los píxeles con una mayor similitud en su valor a $v(N_i)$ tendrán pesos más grandes en el promedio ponderado, mientras píxeles con valores más alejados tendrán menos pesos e influenciarán menos en el cálculo del algortimo. En la figura 1 se observa una imagen con pesos w(p,q_1), w(p,q_2) y w(p,q_3), donde el último píxel q_3 al tener una valor más alejado del pixel p tendrá un peso más pequeño.

Normalización:

La normalización es un proceso que cambia el rango de valores de intensidad de pixeles. Entre las aplicaciones deseadas están las imágenes con poco contraste. La normalización se denomina también estiramiento de contraste o estiramiento de histograma. En campos más generales como en procesamiento digital de señales se le denomina expansión de rango dinámico.

El realce de contraste o "*Contrast Stretching"* es un proceso de imagen digital muy usado, sirve para acentuar diferencias sutiles en los valores de datos y mejorar la interpretacion visual de la imagen.

El realce de contraste es llamada una operación por punto porque aplica independientemente un brillo o transformacon de color a cada pixel en la imagen [4].

La normalización transforma una imagen en escala de grises n-dimensional $I: \{X \subseteq R^n\} \rightarrow \{Min, \dots, Max\}$ con valores de rango (Min, Max) en una nueva imagen con rango de $I_N: \{X \subseteq R^n\} \rightarrow \{newMin, \dots, newMax\}$

La normalización lineal de una imagen digital en escala de grises se realiza con la fórmula:

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin$$

Por ejemplo, si el rango de intensidad de la imagen es de 50 a 180 y el rango deseado es de 0 a 255, el proceso implica restar 50 de cada uno de la intensidad de píxeles, haciendo que el rango sea de 0 a 130. A continuación, cada intensidad de píxel se multiplica por 255/130, haciendo que el rango de 0 a 255.

La normalización también puede ser no lineal, esto sucede cuando no hay una relación lineal entre I y I_N . Un ejemplo de normalización no lineal es cuando la normalización sigue una función sigmoidea, en ese caso, la imagen normalizada se calcula de acuerdo con la fórmula

$$I_N = (newMax - newMin)\frac{1}{1 + e^{\frac{-I-\beta}{\alpha}}} + newMin$$

Dónde α define el ancho del rango de intensidad de entrada, y β define la intensidad alrededor de la cual se centra el rango.
Los valores de α y β no deben superar los 0 o 255 respectivamente debido a que pasan por un canal de 8 bits.

b) <u>Red neuronal artificial convolucional o CNN:</u>

Una red neuronal artificial es un algoritmo computacional con el fin de imitar el comportamiento de su equivalente biológico, es decir, las redes neuronales del cerebro. Una CNN es un tipo de red neuronal artificial utilizada para aprendizaje profundo, cuyo fuerte es la clasificación y reconocimiento de imágenes, dándole la capacidad a un ordenador, o dispositivo embebido, de ver; a esta capacidad se le conoce como visión computacional.

Estructura de una red neuronal convolucional:

Una CNN simple está compuesta por cinco capas: convolución, activación y agrupación (*pooling*), como se ve en la figura 2. Se pueden tener más de estas capas en una red.

A continuación, se describirá de manera resumida la función de cada capa, según Mathworks [5]:

- Convolución: Es la capa asignada de pasar por toda la imagen unos filtros de convolución también llamados *kernels*. La función de estos filtros es la de extraer las características de la imagen de entrada.
- Activación: Se encargan de eliminar valores negativos dejados por los filtros de convolución de la capa anterior y solo dejar pasar valores determinados.
- Agrupación o *pooling*: Proceso de reducir el tamaño de la imagen dejada por la capa anterior, con el fin de ahorrar capacidad de procesamiento.

- *Flatten* o “aplanamiento”: Como su nombre lo indica, consiste en reducir las dimensiones de la imagen reducida y formar una matriz de n filas y 1 columna.

- Softmax: Se encarga de producir los valores de probabilidad de una nueva entrada y enviarlas a sus respectivas neuronas de salida.

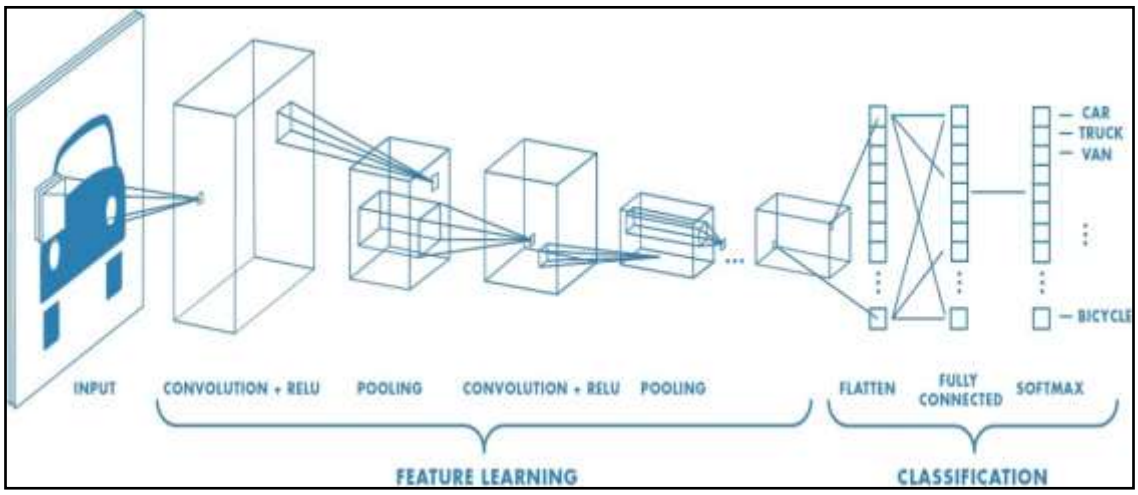

Figura 2: Ejemplo de una CNN con más de una capa convolucional.
Fuente: https://la.mathworks.com/

c) <u>Algoritmo YOLOv3:</u>

YOLO v3 es la tercera versión de un algortimo de detección de objetos en tiempo real creada por Joseph y Ali Farhadi en año 2018, capaz de detectar e identificar objetos en video, trasmisiones en vivo o en imágenes.

El algoritmo YOLOv3 separa primero una imagen en una cuadrícula. Cada celda de la cuadrícula predice un cierto número de cuadros delimitadores (a veces denominados cuadros de anclaje) alrededor de los objetos que obtienen una alta puntuación con las clases predefinidas antes mencionadas. Cada cuadro delimitador tiene una puntuación de confianza respectiva sobre la exactitud de la predicción, y sólo detecta un objeto por cuadro delimitador. Los recuadros delimitadores se generan agrupando las dimensiones de los recuadros reales del conjunto de datos original para encontrar las formas y los tamaños más comunes. YOLO está entrenado para realizar la clasificación y la regresión de bounding box al mismo tiempo [6].
YOLOv3 utiliza Darknet-53, que es una CNN de 53 capas convolucionales, como *backbone* para la extracción de las características de las imágenes de entrada. En 2018, J. Redmon y A. Farhadi [7]. En la tabla 1 se muestra como compararon Darknet-53 con otras *backbones*, demostrando que es más prescisa e igual de eficiente.

d) <u>TensorFlow:</u>

Descrito en [15], es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema flexible de herramientas, bibliotecas y

recursos permite a los investigadores impulsar un aprendizaje automático innovador y, a desarrolladores, compilar e implementar con facilidad aplicaciones con tecnología de AA.

Tabla1: Comparación de *backbones*, donde se muestra la presición, billones de operación (Ops), billones de operaciones de punto flotante por segundo (BFLOP/s) y los cuadros por segundo (FPS).

Backbone	Top-1	Top-5	Ops	BFLOP/s	FPS
Darknet-19	74.1	91.8	7.29	1246	**171**
ResNet-101	77.1	93.7	19.7	1039	53
ResNet-152	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

Fuente: https://viso.ai/deep-learning/yolov3-overview/

2.2. Estado del arte

En 2019, Q. Mao, H. Sun, Y. Liu y R. Jia [8] desarrollaron una versión mejorada del "framework" para reconocimiento y detección de objetos YOLOv3 llamado Mini YOLOv3. Consiste en una versión para dispositivos embebidos, ya que la versión original (YOLOv3) requiere más capacidad de procesamiento. La reestructuración de la arquitectura de la red provee un mejor rendimiento debido a dos principales logros: Se redujo el tamaño de la red principal dividiendo las capas de convolución en dos, la primera filtra cada canal de entrada, la segunda une la salida de la primera capa; y la implementación de una red piramidal de escalas múltiples, que se forma a partir de diferentes mapas de características extraídas de la red principal, ahorrando tiempo de procesamiento. En la conclusión, se alcanzaron detecciones y reconocimientos comparables a YOLOv3 pero que solo requiere la mitad de tiempo.

En 2019, G. Oltean, C. Florea, R. Orghidan y V. Oltean [9] desarrollaron un sistema de conteo de vehículos en tiempo real usando YOLO-tiny. El trabajo cuenta con un algoritmo que almacena la identificación de los vehículos detectados, de esta manera el vehículo solo es detectado una vez aun si se le detecta en múltiples fotogramas o incluso si no se le detecta en un fotograma durante su recorrido.

En 2016, G. Krasner y E. Katz [10] Aplicaron el algoritmo Oriented Fast and Rotated BRIEF (ORB) para lograr la identificación de plazas de estacionamiento filtrando las imágenes. Las características detectadas por el ORB son evaluadas según el modelo de color HSV (Matiz, saturación y valor), si las características están en una sombra, su saturación será baja por lo que se la ignora para que no interfiera con el proceso de detección de la plaza de aparcamiento.
En 2019, Yen-Ting Huang, Yan-Tsung Peng y Wen-Hung Liao [11] desarrollaron una red neuronal como un módulo de pre-procesamiento de imágenes con la capacidad de restaurar imágenes que tienen niveles de luminosidad no uniformes.

Jokela Jussi, en su tesis para grado de bachiller [12], realizó una investigación sobre el aprendizaje profundo y sus diversos *frameworks* y algoritmos y los comparó diseñando una pequeña red neuronal para el conteo de personas. En su tesis señala que el algoritmo de YOLOv3 es superior en cuanto a velocidad en comparación a Faster R-CNN y más exacto que su versión anterior, YOLOv2.

Edgar Guerra, en su tesis para obtener el título de ingeniero en electrónica y redes de comunicación [13], desarrolló un sistema web para la detección y conteo de vehículos para la cuidad de Quito. Su prototipo consta de 4 partes: adquisición y trasmisión de video, recepción y procesamiento, almacenamiento y reportes. Este sistema tiene como objetivo de ser la base para futuros proyectos que permitan medir el flujo del vehículos utilizando el Sistema de Vigilancia de Tránsito (STV) ya existente.

En otro artículo hecho por Emil Bjørlykhaug y Olav Egeland [14], aplicaron la visión artificial y el procesamiento de imágenes en máquinas limpiadoras de pescado de sus plantas industriales. Este método consistía en la detección de ciertos patrones y colores en el pescado que caracterizaba un factor de calidad aceptable para su empaquetado. El procesamiento de imágenes fue aplicado para la difuminación y brillo de la imagen.

3. PLANTEAMIENTO DE LA SOLUCIÓN

3.1. Descripción y características de la solución

El sistema de detección y monitoreo se representará en el siguiente diagrama de bloques en la figura 3:

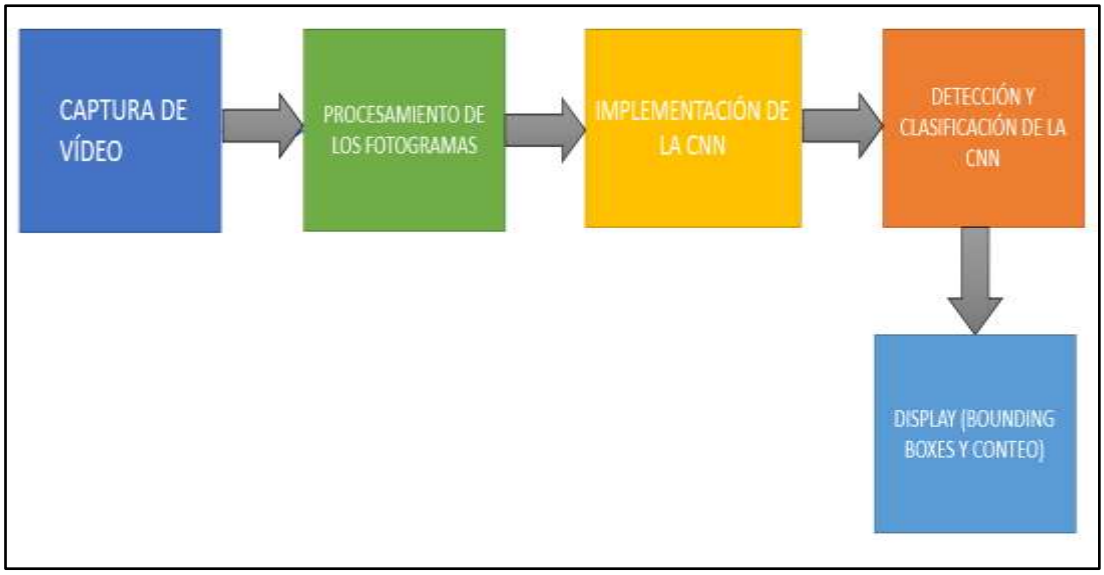

Figura 3: Diagrama de bloques del sistema de detección y monitoreo.
Fuente: Elaboración propia.

3.2. Metodología de la solución

CAPTURA DE VIDEO:

Para el primer apartado, se hizo la respectiva captura de un video en formato mp4 mediante una cámara de video o por un video pregrabado.

PROCESAMIENTO DE FOTOGRAMAS:

En el segundo apartado, se hizo la lectura de cada fotograma del video mediante la librería de software de fuente abierta OPEN CV, en el cual se llevaron a cabo los procesos de normalización, reducción del dimensionado de pixeles mediante algoritmos de interpolación y filtrado del ruido mediante medios no locales.

IMPLEMENTACIÓN DE LA CNN:

Para el tercer y cuarto apartado, se pasaron cada fotograma del video a través de la CNN pre-entrenada YOLOv3, para la detección y clasificación de los objetos en pantalla.

DISPLAY DE BOUNDING BOXES Y CONTEO:

En el último apartado, la red neuronal encerró a cada vehículo detectado del video dentro de un cuadro, al que es llamado *bounding box*. Para evitar que la CNN, recuente los mismos vehículos en cada fotograma, se les otorgó un ID, que mediante el algoritmo *DeepSort (Simple Online and Realtime Tracking with a Deep Association Metric)*, otorgó al programa la capacidad de diferenciar y hacer seguimiento a cada vehículo; de esta manera se logró un conteo óptimo de los vehículos estacionados. Por último se procedió a mostrar las *bounding boxes* de los objetos detectados con su respectiva ID, así como el conteo en la parte superior izquierda de la ventana.

3.3. Diseño de la solución

En el sistema propuesto se lograron las respectivas capturas de un fotograma del video de entrada, y antes de que este ingrese a la red neuronal convolucional preentrenada se realizaron los siguientes procesados de imágenes.

NORMALIZACIÓN:

Se realizó un realce de contraste modificando el canal de bits por pixel.
En la figura 4 se muestra un fotograma del video original con una normalización de 0 a 250 mientras que en la figura 5 se realizó una normalización de 0 a 400 del mismo fotograma.

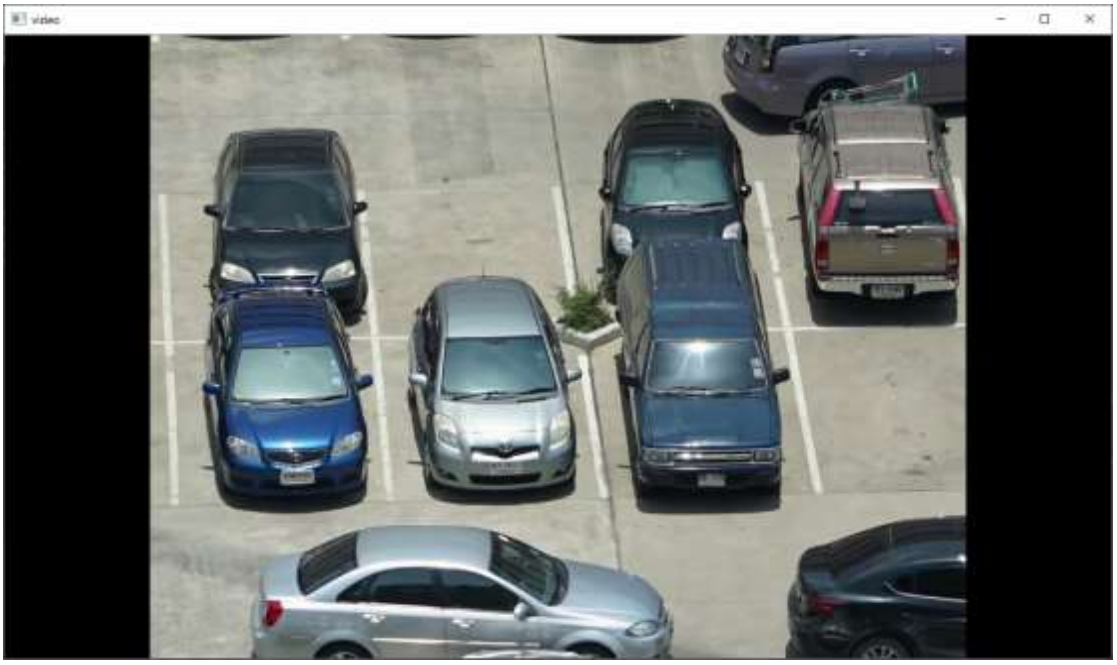

Figura 4: Video para una codificación de 8 bits de 0 a 255.
Fuente: Elaboración propia.

Figura 5: Realce de contraste para una codificación de bits superior a 8 bits de 0 a 400.
Fuente: Elaboración propia.

FILTROS DE MEDIAS NO LOCALES:

Se debió tener en cuenta el valor del tamaño en pixeles del parche que se utilizó para calcular los nuevos pesos, cuyo valor recomendado fue de 7. También se determinó el tamaño en píxeles de la ventana que se utilizó para calcular el promedio ponderado del píxel cuyo valor recomendado fue de 21, esto puede afectar el rendimiento.

En la figura 6 y en la figura 7 se muestra un fotograma del video original y el mismo fotograma después de aplicarle el filtro de medias no locales respectivamente.

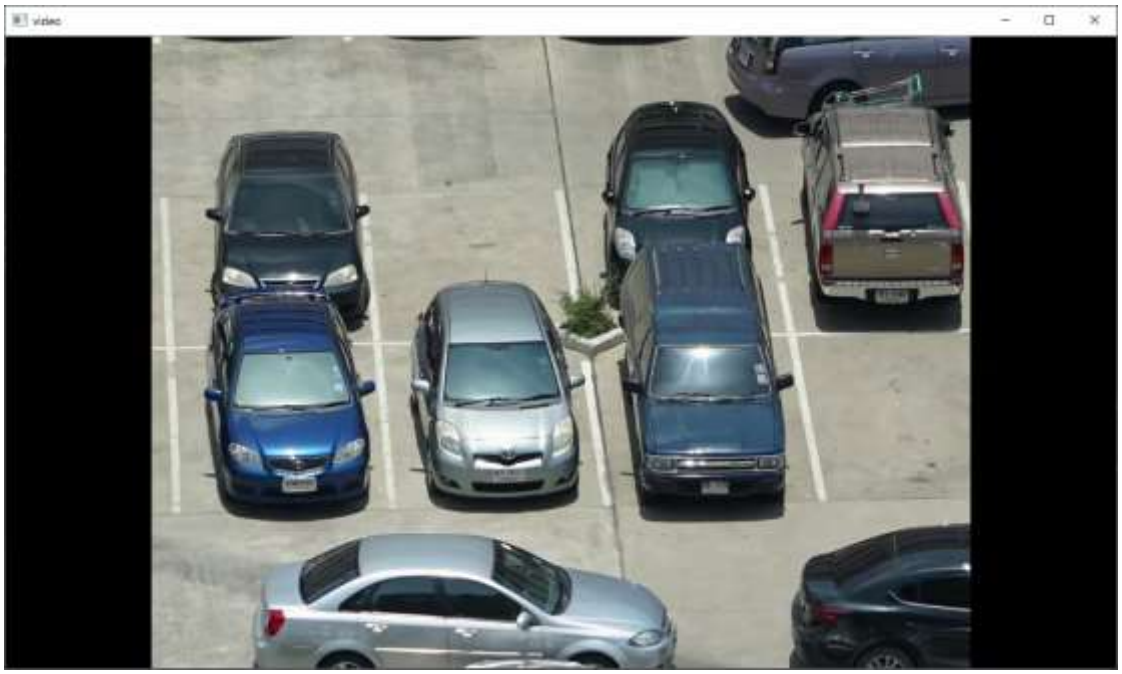

Figura 6: Video de forma nativa.
Fuente: Elaboración propia.

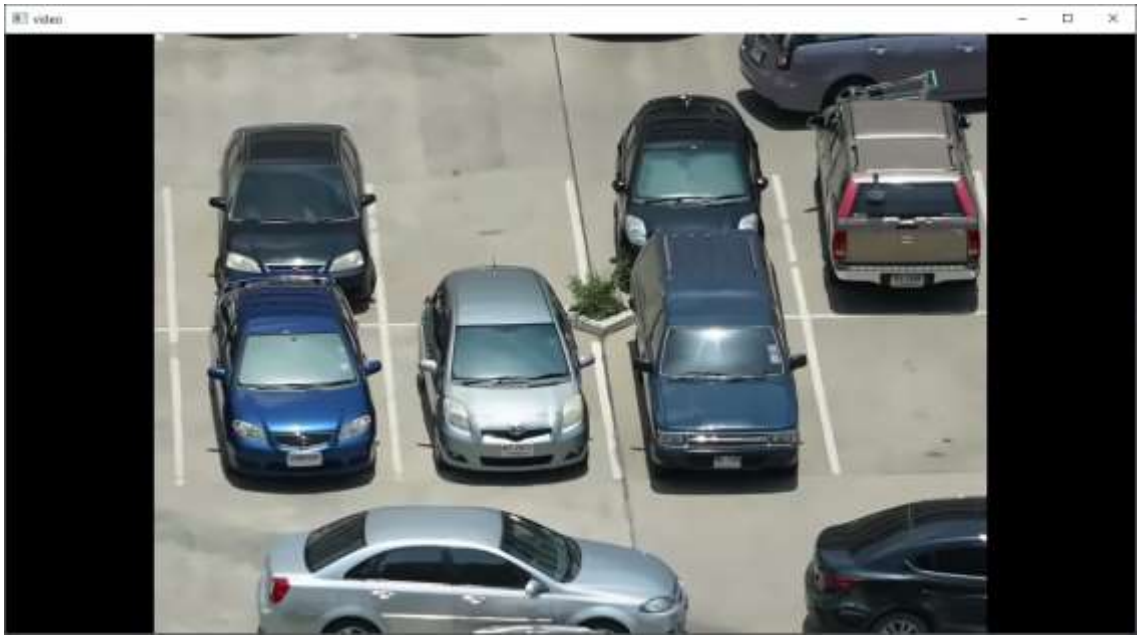

Figura 7: Video aplicando el filtro de medias no locales.
Fuente: Elaboración propia.

REDUCCIÓN DE PIXELES:

Las imágenes recopiladas fueron reducidas a una resolución menor a la nativa mediante el algoritmo de interpolación bicúbica debido a sus pocos artefactos de interpolación que se pueden producir, pero con mayor consumo computacional.

La disminución en la resolución se hizo debido a que la CNN Darknet-53 de YOLOv3 solo admite imágenes con una resolución de 416x416.

En la figura 8 se muestra un fotograma del video en su forma nativa a una resolución menor de 416x416 pixeles como se ve en la figura 9.

Figura 8: Video en su resolución nativa.
Fuente: Elaboración propia.

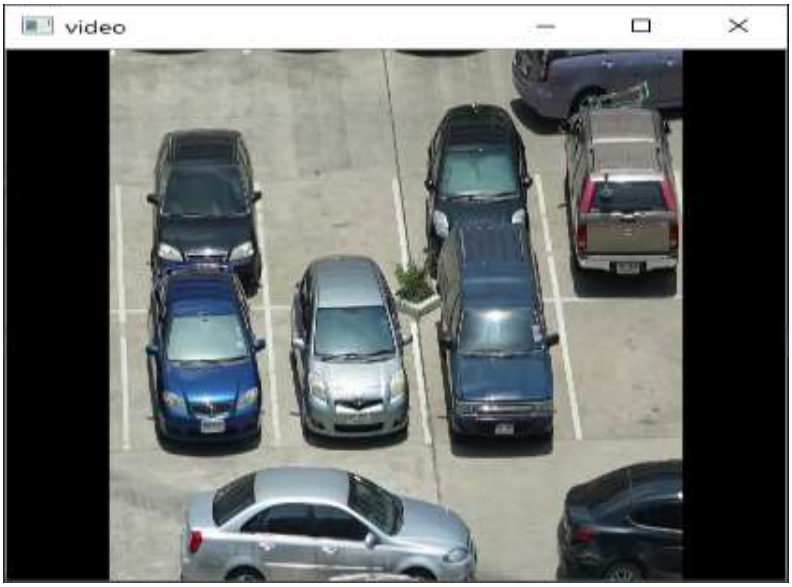

Figura 9: Video reducido a la resolución de 416x416 pixeles.
Fuente: Elaboración propia.

RED NEURONAL PREENTRENADA YOLOv3:

Los fotogramas procesados fueron pasados a la CNN YOLOv3 para su detección y reconocimiento. A cada vehículo se le asigna una *bounding box* y un ID, diciendo *"car"* o *"truck"* seguido respectivamente de un número, tal y como se muestra en la figura 10.

Luego, se agregaron dos líneas que sirven como “puertas” como se ve en la figura 11, de tal manera de que cada vez que el centro de una *bounding box* se encuentra entre ellas se aumenta el contador.

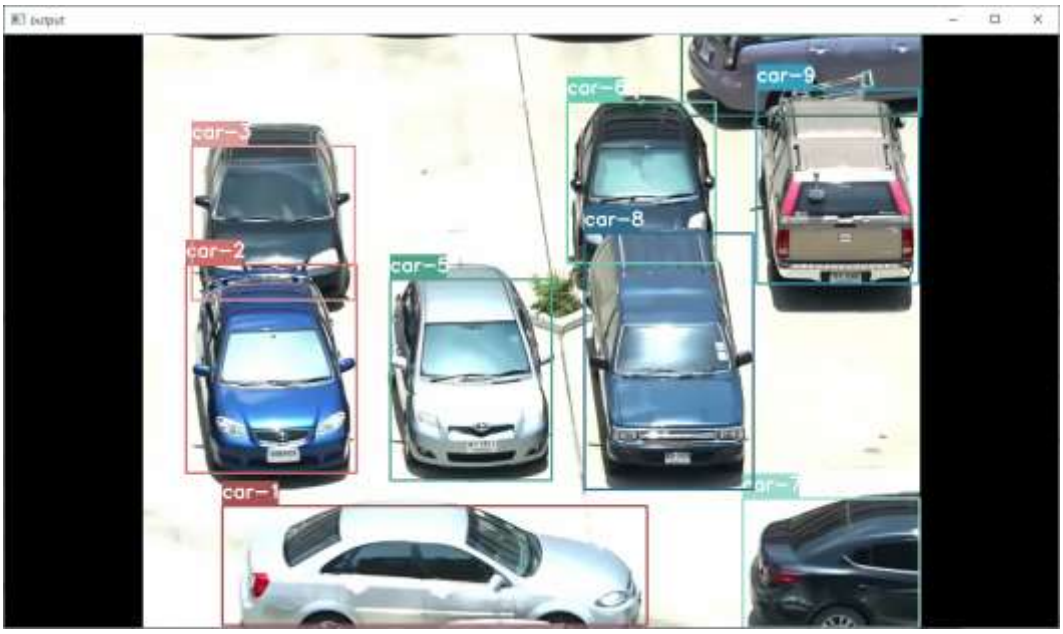

Figura 10: Detección de los vehículos en el video así como la asignación de sus ID.
Fuente: Elaboración propia.

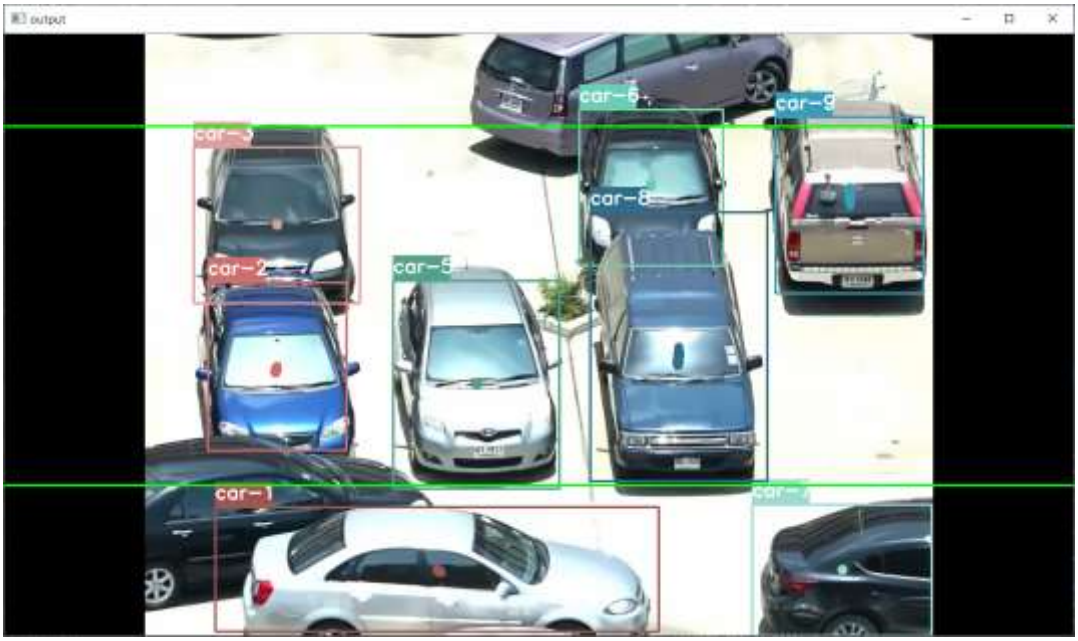

Figura 11: Líneas que sirvieron para el conteo de los vehículos.
Fuente: Elaboración propia.

4. RESULTADOS

Los resultados obtenidos al final del programa se visualizan en la figura 12, donde se muestran lo presentado en el capítulo 3 además de un texto señalando los vehículos estacionados así como los fotogramas por segundo (FPS) obteniéndose una tasa promedio aproximada de 2.8 FPS.

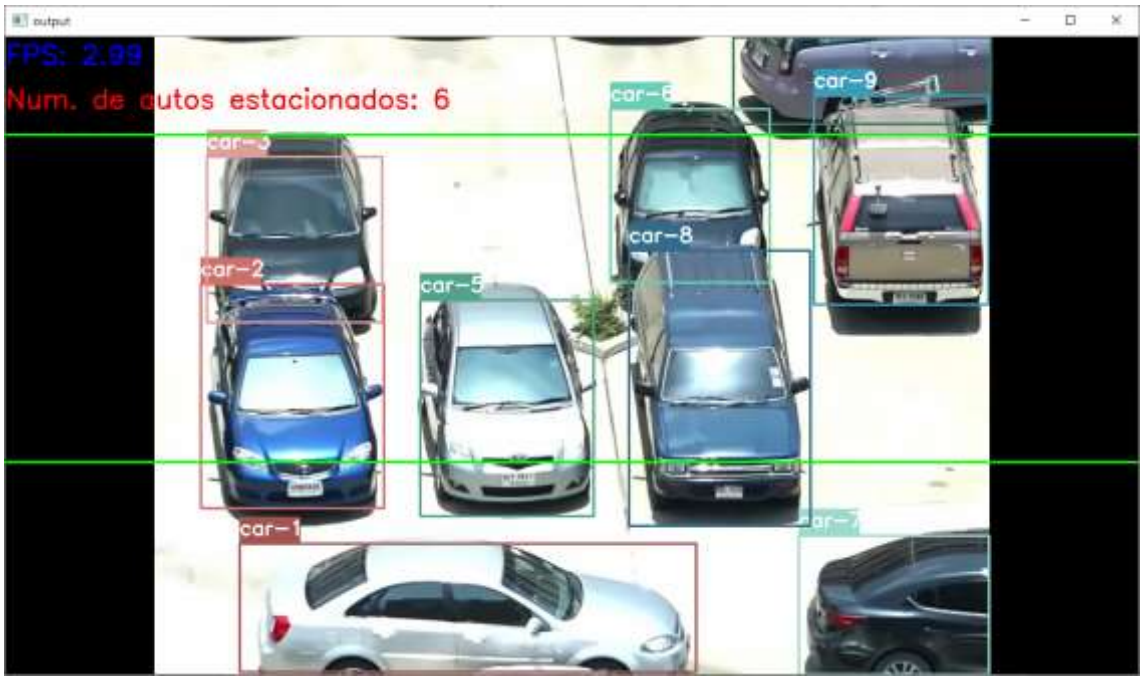

Figura 12: Identificación y conteo de los vehículos detectados.
Fuente: Elaboración propia.

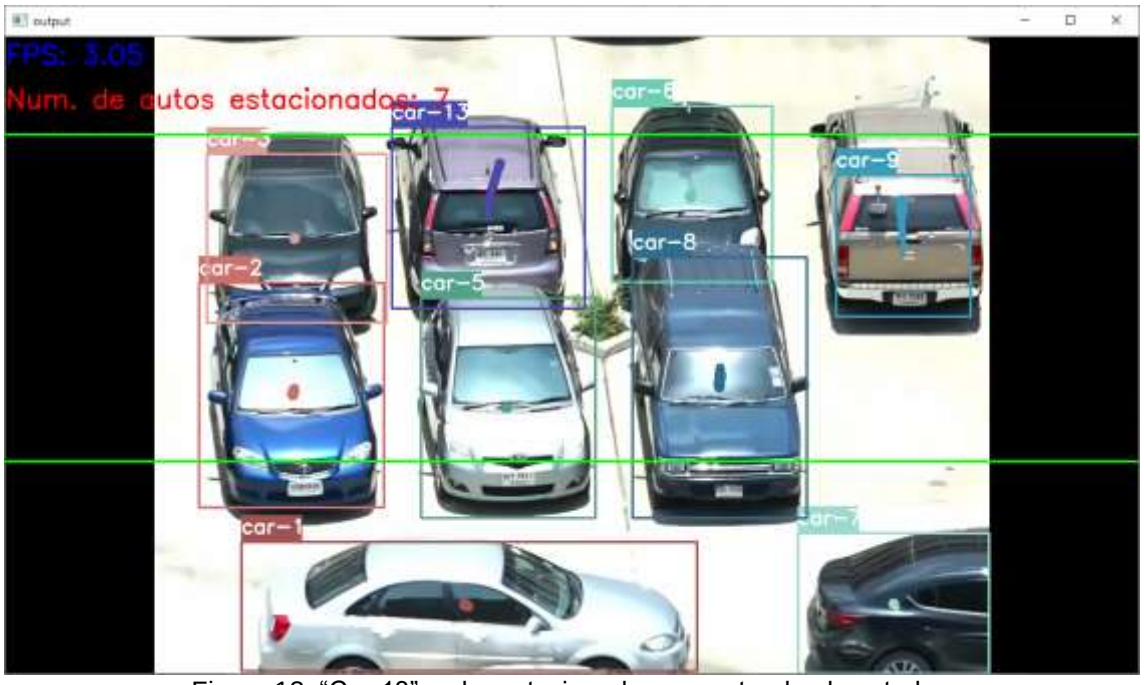

Figura 13: “Car-13” se ha estacionado, aumentando el contador.
Fuente: Elaboración propia.

En la figura 12 se observa que hay 6 vehículos estacionados, número que coincide con el mostrado en la parte superior izquierda de la ventana calculada por el programa.

En la figura 13 un vehículo con el ID de “car-13” se estaciona, aumentando el contador de 6 a 7 autos estacionados.

Ya con los 8 espacios de estacionamiento ocupados, como se ve en la figura 14, el vehículo con ID “car-61”, se empieza a retirar, saliendo el centro de su “*bounding box”* fuera del área entre las líneas verdes, disminuyendo el contador a 7, como se aprecia en la figura 15.

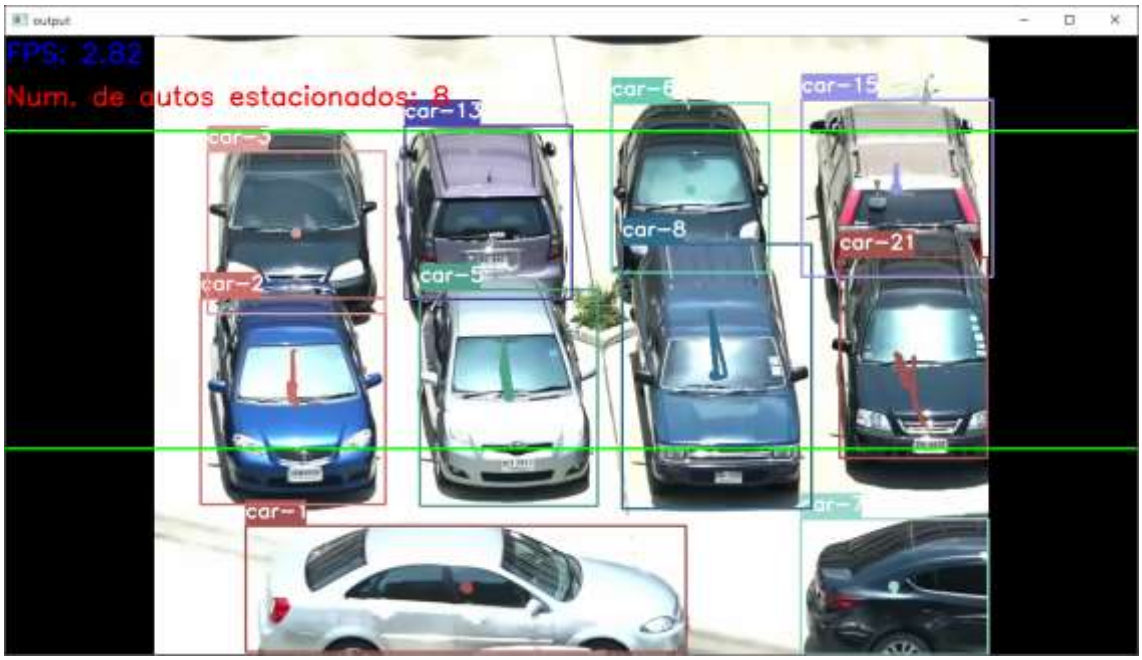

Figura 14: Los 8 espacios de estacionamientos ocupados.
Fuente: Elaboración propia.

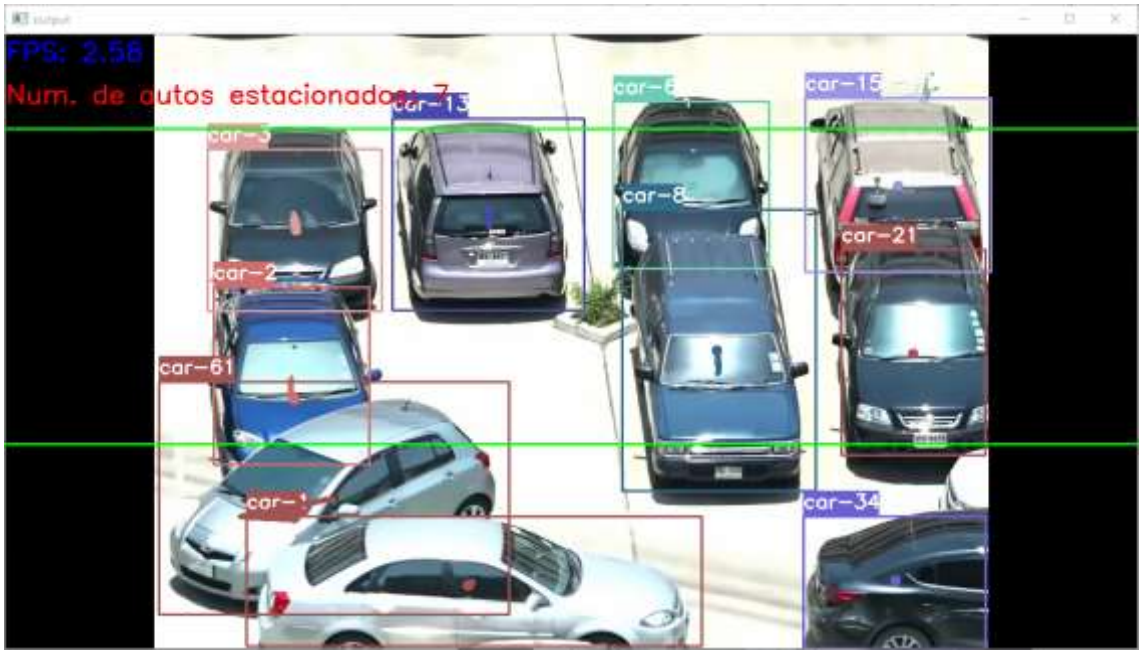

Figura 15: Vehículo “car-61” retirándose, el contador disminuye.
Fuente: Elaboración propia.

Los algoritmos de las redes neuronales convolucionales requieren, para sus cálculos, realizar numerosas operaciones matriciales, al igual que los algoritmos utilizados para el procesamiento de los fotogramas como el filtro de medias no locales; por tal motivo el programa del presente trabajo de investigación requirió bastante recursos computacionales, tal y como se muestra en la figura 16, donde el consumo de una CPU relativamente moderna (Intel Core i7-7500U) llegó al 100% durante todo el proceso.

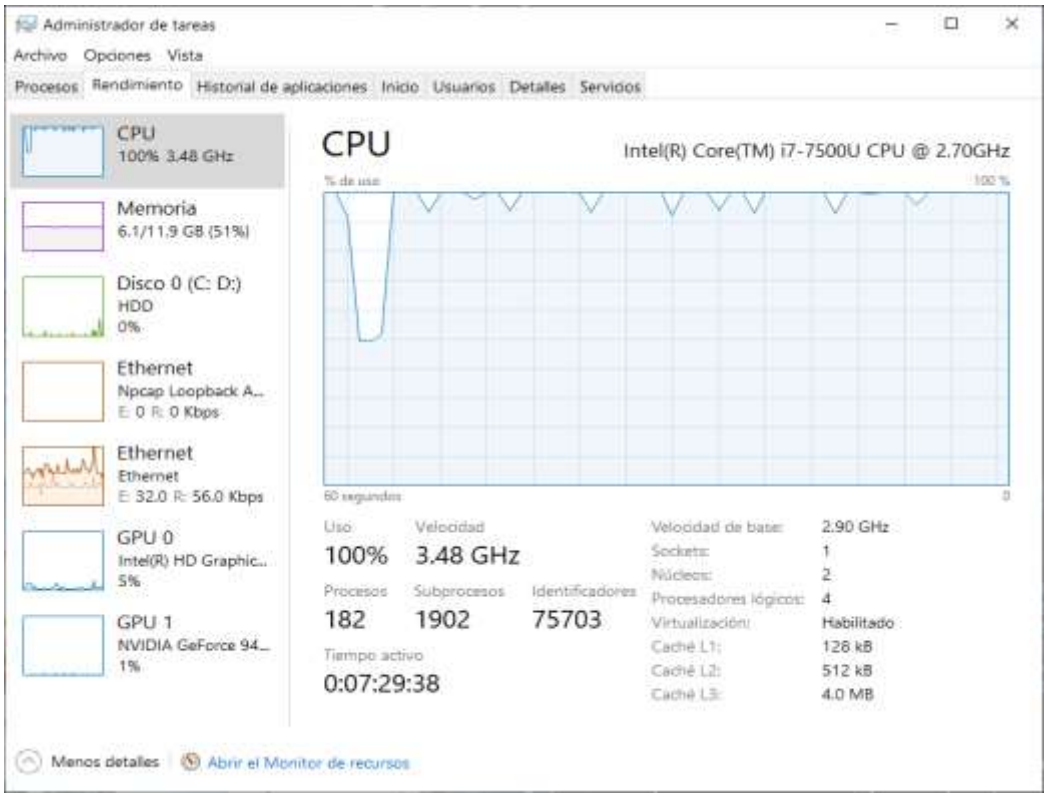

Figura 16: Consumo computacional del programa.
Fuente: Elaboración propia.

5. CONCLUSIONES

- El procesamiento de las imágenes capturadas del video, tanto la normalización y el suavizado mediante el filtro de medias no locales consumen un alta cantidad de recursos computacionales, lo cual reduce la cantidad de fotogramas mostrados en la pantalla.

- El uso de la red neuronal convolucional Darknet-53 de Yolo-v3 preentrenada, nos da una gran exactitud en la detección de los autos vistos en el estacionamiento, pero este requiere que solo sea aplicado a cierta zona de este para evitar detecciones innecesarias y evitar mayor uso de CPU.

- El buffer de datos para el conteo de vehículos llega a contar todos los vehículos dados en el área encerrada entre dos líneas, teniendo en cuenta solo los cuadros generados por la red neuronal convolucional.

6. RECOMENDACIONES

- Para un procesamiento más óptimo de las imágenes con un mayor número de fotogramas por segundo, se recomienda el uso de una CPU con un mínimo de 8 núcleos físicos o en último recurso , el uso de una GPU para el procesamineto del video.

- El reentrenamiento de la red neuronal (Transfer Learning) de la red YOLO-V3 para la detección unicamente de automóviles.

7. REFERENCIAS BIBLIOGRÁFICAS

[1] D. Valdés Amaro, "Procesamiento digital de imágenes LCC-594", Benemérita Universidad Autónoma de Puebla.

[2] B. Shreyamsha Kumar, "Image denoising based on non-local means filter and its method noise thresholding", *ResearchGate*, p. 2, 2013.

[3] B. Coll, J. Morel and A. Buades, "A non-local algorithm for image denoising", *Researchgate*, p. 4, 2014. [Accessed 30 June 2021].

[4] A. Roncero García, "La interpolación aplicada al procesamiento de imágenes digitales", *Repositori.uji.es*, 2019. [Online]. Available: http://repositori.uji.es/xmlui/bitstream/handle/10234/185159/TFG_AlbaRonceroGarcia__%281%29.pdf?sequence=1&isAllowed=y. [Accessed: 25- Jun- 2021].

[5] "Convolutional Neural Network", Mathworks.com, 2021. [Online]. Available: https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html. [Accessed: 10- Jun- 2021].

[6] "YOLOv3: Real-Time Object Detection Algorithm (What's New?) | viso.ai", *viso.ai*, 2021. [Online]. Available: https://viso.ai/deep-learning/yolov3-overview/. [Accessed: 11- Jun- 2021].

[7] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement", *Cornell University*, April 2018.

[8] Q. Mao, H. Sun, Y. Liu y R. Jia, "Mini-YOLOv3: Real-Time Object Detector for Embedded Applications," en *IEEE Access*, vol. 7, pp. 133529-133538, 2019, doi: 10.1109/ACCESS.2019.2941547.

[9] G. Oltean, C. Florea, R. Orghidan y V. Oltean, "Towards Real Time Vehicle Counting using YOLO-Tiny and Fast Motion Estimation," 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), Cluj-Napoca, Romania, 2019, pp. 240-243, doi: 10.1109/SIITME47687.2019.8990708.

[10] G. Krasner y E. Katz, "Automatic parking identification and vehicle guidance with road awareness," 2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE), Eilat, 2016, pp. 1-5, doi: 10.1109/ICSEE.2016.7806133.

[11] Y. Huang, Y. Peng y W. Liao, "Enhancing object detection in the dark using U-Net based restoration module," 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 2019, pp. 1-9, doi: 10.1109/AVSS.2019.8909820.

[12] J. Jokela, "Person Counter using Real-time Object Detection and a Small Neural Network", Bachiller, Turku University of Applied Sciences, 2018.

[13] E. Guerra Carapas, "Prototipo de un sistema web para el conteo de vehículos en tiempo real utilizando la librería de visión artificial OpenCV", Bachiller, Escuela Politécnica Nacional, 2016.

[14] Emil Bjørlykhaug, Olav Egeland, "Vision System for Quality Assessment of Robotic Cleaning of Fish Processing Plants Using CNN"," IEEE Access", Noruega, 2019.

[15] Tensorflow, «Tensorflow,» [En línea]. Available: https://www.tensorflow.org/lite?hl=es-419. [Último acceso: 12 Diciembre 2020].