



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

MÁSTER UNIVERSITARIO EN INGENIERÍA DEL SOFTWARE E INTELIGENCIA ARTIFICIAL

Mejora en la detección de objetos pequeños en secuencias de carreteras mediante redes neuronales convolucionales y super-resolución

Improved detection of small objects in road sequences using convolutional neural networks and super-resolution

Realizado por:

Iván García Aguilar

Tutorizado por:

Ezequiel López Rubio
Rafael Marcos Luque Baena

Departamento:

Lenguajes y Ciencias de la computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Junio, 2021

Resumen:

El aumento exponencial del uso de la tecnología en los sistemas de gestión de carreteras ha permitido disponer de información visual en tiempo real en miles de puntos de las redes de carreteras. Un paso previo a la prevención o detección de accidentes consiste en la detección de vehículos. La aplicación de las redes neuronales convolucionales aplicadas a la detección de objetos ha supuesto importantes mejoras en este campo. Así, la irrupción de algoritmos de detección de objetos basados en *Deep Learning* ha mejorado las técnicas clásicas basadas en visión por computador, aunque se ha demostrado que existen deficiencias por la baja tasa de detección que proporcionan los modelos pre-entrenados disponibles, especialmente para objetos pequeños. El principal inconveniente a la hora de mejorar dichas detecciones se basa en el requerimiento de realizar el etiquetado manual de los vehículos que aparecen en las imágenes de cada cámara IP ubicada en la red de carreteras para re-entrenar el modelo, tarea que no es factible si tenemos miles de cámaras distribuidas por la extensa red de carreteras de cada nación o estado.

El escenario actual está en constante evolución, y aparecen nuevos modelos y técnicas que intentan mejorar este campo. En particular, aparecen nuevos problemas e inconvenientes en cuanto a la detección de objetos pequeños, que corresponden principalmente a los vehículos que aparecen en las escenas de carretera. Todo ello hace que sean imprescindibles nuevas soluciones que traten de mejorar la baja tasa de detección de elementos pequeños. Entre las diferentes líneas de investigación emergentes, este trabajo se centra en la detección de objetos pequeños. En concreto, nuestra propuesta tiene como objetivo la detección de vehículos a partir de imágenes captadas por cámaras de videovigilancia.

En este trabajo, se propone un nuevo procedimiento automático para la detección de objetos a pequeña escala en secuencias de tráfico. En la primera etapa, los modelos de vehículos detectados a partir de un conjunto de fotografías se generan automáticamente y mediante un proceso Offline, la red neuronal pre-entrenada se integra con procesos de super-resolución con el objetivo de aumentar la resolución de las imágenes para mejorar el rendimiento de la detección de objetos. Posteriormente, el modelo de detección de objetos se re-entrena con los datos obtenidos previamente, adaptándolo así a la escena analizada. Finalmente, y ya en tiempo real, el modelo re-entrenado se utiliza sobre el resto de la secuencia de tráfico o sobre el *streaming* de vídeo generado por la cámara. Este marco ha sido probado con éxito en un repositorio público de secuencias de tráfico reales obtenidas del Departamento de Transporte de los Estados Unidos, comparando diferentes propuestas tanto cualitativa como cuantitativamente.

Palabras claves: Detección de objetos, Pequeña escala, Super-resolución, Redes neuronales convolucionales.

Abstract:

The exponential increase in the use of technology in road management systems has led to real-time visual information in thousands of locations on road networks. A previous step to the prevention or detection of accidents consists of the detection of vehicles. The application of convolutional neural networks applied to object detection has led to significant improvements in this field. Thus, the irruption of object detection algorithms based on *Deep Learning* has enhanced the classical techniques based on computer vision, although it has been demonstrated that there are deficiencies for the low detection rate provided by the available pre-trained models, specially for small objects. The main drawback in improving such detections is based on the requirement to manually labeling the vehicles that appear in the images of each IP camera located in the road network to retrain the model, a task that is not feasible if we have thousands of cameras distributed throughout the extensive road network of each nation or state.

The current scenario is constantly evolving, and new models and techniques are appearing trying to improve this field. In particular, new problems and drawbacks appear regarding detecting small objects, which correspond mainly to the vehicles that appear in the road scenes. All this means that new solutions that try to improve the low detection rate of small elements are essential. Among the different emerging research lines, this work focuses on the detection of small objects. In particular, our proposal aims to vehicle detection from images captured by video surveillance cameras.

In this work a new automatic procedure is developed for the detection of small-scale objects in traffic sequences. In the first stage, vehicle patterns detected from a set of frames are generated automatically and through an Offline process, the pre-trained neural network is integrated with super-resolution processes to increase image resolution to improve object detection performance. Subsequently, the object detection model is retrained with the previously obtained data, thus adapting it to the analyzed scene. Finally, and in real-time, the retrained model is used on the rest of the traffic sequence or the video streaming generated by the camera. This framework has been successfully tested on a public repository of real traffic sequences obtained from the U.S. Department of Transportation, comparing different proposals both qualitatively and quantitatively.

Keywords: Object detection, Small scale, Super-resolution, Convolutional neural networks.

Índice General:

1.	Capítulo 1 - Introducción:	8
1.1.	Objetivos:	8
1.2.	Descripción del problema a resolver:	9
1.3.	Situación de la cual se parte:	9
1.4.	Relevancia de la solución propuesta:	10
1.5.	Organización del resto del trabajo:	11
2.	Capítulo 2 - Antecedentes:	13
3.	Capítulo 3 - Descripción del problema:	17
4.	Capítulo 4 - Tecnología utilizada y detalles de la propuesta:	21
4.1.	Modelo pre-entrenado para aplicar SR:	21
4.2.	Redes Neuronales Convolucionales utilizadas:	24
4.3.	Detalles de la propuesta:	25
4.4.	Mejora tras aplicar super-resolución en la detección de objetos:	25
4.5.	Agilizar la generación de detecciones - Algoritmo de <i>Bron-Kerbosch</i> :	29
4.6.	<i>Fine-Tuning</i> mediante datos de entrenamiento etiquetados automáticamente:	31
5.	Capítulo 5 - Resultados:	33
5.1.	Generación del conjunto de datos destinado a las pruebas:	33
5.2.	Aplicación de procesos de disminución de ruido:	35
5.3.	Resultados cualitativos y cuantitativos en la fase off-line:	36
5.4.	Resultados cualitativos y cuantitativos en la fase on-line:	44
5.5.	Comparativa del tiempo requerido en realizar las detecciones:	48
6.	Capítulo 6 - Conclusiones y líneas futuras:	51

Capítulo 1 - Introducción:

Actualmente, la detección de objetos corresponde con una de las aplicaciones de visión por computador más populares dentro del ámbito del aprendizaje profundo. La proliferación de este tipo de aplicaciones viene provocada por el aumento de datos de vídeo obtenidos de diferentes fuentes, así como por la mejora en la potencia computacional del hardware, facilitando así la realización con éxito de esta tarea.

Cualquier área esencial como la agricultura, la industria o el transporte, ámbito sobre el cual se plantea dicho trabajo, se rige por los datos. Por ello, el análisis de los mismos puede ayudar a mejorar y optimizar los procesos internos de cada disciplina. En el ámbito del transporte, la red de carreteras de cada nación o estado ha instalado en los últimos años miles de cámaras *IP* en multitud de lugares, con el fin de vigilar el estado de las carreteras y prevenir atascos, accidentes o simplemente para contabilizar la densidad del tráfico en diferentes momentos del día.

A lo largo de este capítulo, en la subsección 1.1 se establecerán los objetivos que se han conseguido obtener con el desarrollo de dicho trabajo. En el punto 1.2 se enuncia el problema que se desea resolver. Posteriormente, en la subsección 1.3 se detalla el escenario sobre el cual se va desarrollar el trabajo, así como los modelos neuronales convolucionales pre-entrenados existentes junto con la tasa de detección que presentan los mismos. En la subsección 1.4 se establecen los ámbitos en los cuales dicho desarrollo puede ser de interés así como una breve introducción del flujo de trabajo desarrollado para conseguir dichos fines. Finalmente en el punto 1.5 se detalla como se ha organizado el trabajo.

1.1. Objetivos:

En esta subsección, se tratarán tanto el objetivo principal, así como los objetivos específicos llevados a cabo para la realización de dicho trabajo.

Objetivo principal:

El objetivo principal de este proyecto es conseguir mejorar la detección de elementos de pequeño tamaño de forma totalmente automática sin la necesidad de procesos manuales de etiquetado, mediante redes neuronales convolucionales profundas (*DCNN*), haciendo uso del *framework* de trabajo denominado como *Tensorflow 2*¹ y técnicas de pre-procesado de la imagen basadas en la disminución de ruido, así como la aplicación de super-resolución.

Objetivos específicos:

- Uno de los objetivos específicos de este trabajo se basa en el pre-procesado de la imagen dada como entrada a la red neuronal convolucional. Para la disminución de ruido se ha estudiado que función así como los hiperparámetros de la misma son los adecuados a utilizar atendiendo al contexto del área de transportes. En cuanto a la aplicación de super-resolución, se ha tenido en cuenta los modelos pre-entrenados disponibles y se ha realizado un estudio en base a la eficacia en la precisión

¹ https://github.com/tensorflow/models/tree/master/research/object_detection

y velocidad de los mismos para aplicarlo en escenarios en los cuales las detecciones deban realizarse en tiempo real.

- Para aplicar la propuesta desarrollada, será necesario realizar *Fine-Tuning* (Re-entrenamiento de las últimas capas que componen la red neuronal convolucional) de acuerdo con los resultados provistos tras aplicar el pre-procesado de la imagen dada como entrada.
- Desarrollar un conjunto de datos (*dataset*) manualmente, haciendo uso de una serie de vídeos recopilados de secuencias proporcionadas por el departamento de transporte de Estados Unidos.
- Se ha realizado un estudio detallado, descrito en la sección de resultados número 5, comparando el rendimiento de la solución propuesta con respecto a la precisión obtenida por el modelo original pre-entrenado sin modificar.

1.2. Descripción del problema a resolver:

La aplicación de redes neuronales convolucionales orientado a la detección de objetos presenta una serie de problemas en la actualidad a la hora de identificar objetos de pequeña escala. La red no detecta elementos de pequeño tamaño debido al escaso número de píxeles que estos poseen. En el ámbito del transporte, dado que los sistemas de videovigilancia se colocan en puntos elevados, es vital identificar la mayor cantidad de vehículos que circulan por una vía. Esta información, proporcionada por la transmisión de vídeo de estas cámaras pueden ser útil para detectar eventos anómalos o accidentes en las carreteras. Para ello, es crítico detectar correctamente los vehículos que circulan por ellas, una tarea que ha tenido un gran impacto con la irrupción de algoritmos de detección de objetos basados en *Deep Learning*.

Otro de los factores a tener en cuenta corresponde con la heterogeneidad de los escenarios debido a la posición de la cámara, el ángulo, la orientación y la distancia a la carretera. En la sección número 3 se describe con más detalles los problemas que se presentan en una imagen captada por sistemas de tráfico. De acuerdo con los problemas comentados anteriormente, se ha desarrollado un flujo de trabajo automático que permite mejorar el número de elementos detectados en una imagen, incluyendo por ello elementos de tamaño reducido, definidos como aquellos objetos que ocupan una pequeña región dentro de la imagen completa dada como entrada a la red. Además de ello, se mejora la puntuación de clase obtenida para los elementos detectados, haciendo más fiable los resultados obtenidos por la red para su posterior tratamiento.

1.3. Situación de la cual se parte:

En la actualidad, algunos modelos pre-entrenados presentan importantes problemas intrínsecos los cuales deben ser resueltos. Los modelos de detección de objetos determinan el número de características agregando la información de los píxeles en bruto a través de las capas que componen la red convolucional. La mayoría de ellos, reducen la resolución de las imágenes en las capas intermedias. Este hecho, provoca la pérdida de las características de los objetos pequeños, los cuales desaparecen durante el procesamiento realizado por la red, evitando así su detección. La baja tasa de detección de este tipo de elementos también está causada por el desorden de fondo genérico en las imágenes

a inferir, lo que hace más compleja la tarea de detección de nuevos elementos debido a la gran cantidad de ubicaciones potenciales de los objetos. Los elementos de tamaño reducido a detectar dentro del ámbito de las carreteras, poseen formas simples que no pueden descomponerse en partes o características más pequeñas. Otro punto a tener en cuenta es la calidad de la imagen, ya que existen sistemas de videovigilancia los cuales captan imágenes con baja resolución. Las imágenes de baja calidad, afectan negativamente al rendimiento de los métodos de detección de objetos, ya que este tipo de secuencias, especialmente en el caso de los vídeos de tráfico, son densas, con vehículos pequeños y con cierta oclusión en los mismos. Por otra parte, algunos objetos son similares en cuanto a características, formas, color o patrón se refiere. De acuerdo con estos hechos comentados, los modelos pre-entrenados existentes en la actualidad no pueden distinguir con alta precisión estos elementos de forma eficaz.

Actualmente, existen algunos modelos destinados a la detección de estos elementos mediante dos flujos de trabajo. El primero de ellos sigue el flujo habitual en el que se generan una serie de regiones candidatas para realizar posteriormente la clasificación de cada área propuesta. El segundo método, establece la detección de objetos como un problema de regresión o clasificación, para adoptar un marco de trabajo que permita alcanzar los resultados finales. Entre los métodos basados en propuestas regionales, encontramos principalmente *R-CNN* o *Faster R-CNN* [1]. En los métodos de regresión, cabe mencionar *SSD* [2] y *YOLO* [3]. Como ejemplo, el modelo definido como *EfficientDet* [4] cuenta con una tasa de detección de media del 51 % para elementos de tamaño medio. En cambio, elementos más pequeños solo se detectan en el 12 % de los casos. También hay que tener en cuenta que no existen conjuntos de datos dedicados especialmente a la detección de objetos pequeños, por lo que la mayoría de las detecciones obtenidas corresponden a elementos de mayor tamaño. Al aplicar modelos existentes como *Faster R-CNN*, se observa que pasa por alto varios objetos pequeños en parte debido al tamaño de las cajas de anclaje de los elementos detectados en la imagen.

1.4. Relevancia de la solución propuesta:

El problema de la detección de objetos pequeños es relevante en muchos ámbitos. Entre las aplicaciones directas se encuentran la mejora en la clasificación y detección de elementos a través de imágenes captadas por satélites, la mejora en la videovigilancia a través del tratamiento de las imágenes proporcionadas por las cámaras de seguridad establecidas en puntos elevados, así como el aumento en la detección de peatones o tráfico entre otros. Este trabajo se centra en la última de estas aplicaciones directas. Gracias a esta nueva solución, será posible detectar un mayor número de elementos y mejorar la confianza de cada detección, todo ello realizado de forma automática. Es por ello por lo que esta solución puede ser ventajosa para los sistemas de control de tráfico.

La propuesta planteada en este artículo se basa en el diseño, implementación y experimentación de una técnica automática de re-entrenamiento del modelo basado en redes neuronales convolucionales, capaz de detectar elementos de pequeña escala y mejore la inferencia de clase. Para lograr estos objetivos, se han modificado los hiperparámetros de la red para reducir los falsos negativos. Posteriormente, se han aplicado procesos de super-resolución a la imagen de entrada para generar una serie de imágenes centradas en cada uno de los elementos detectados para así aumentar su resolución y detectar elementos próximos que, de otro modo, pasarían desapercibidos. Utilizando este método, implementado junto con el pre-procesamiento de la imagen mediante un filtro de eliminación de ruido, el modelo mejora notoriamente en cuanto al número de elementos detectados y

su puntuación de clase. Este proceso se realiza *Offline* y solo una vez por escena. Con ello lo que se ha conseguido ha sido realizar un ajuste fino (*Fine-Tuning*) de la red sin aumentar el tiempo de ejecución, permitiendo adaptarse automáticamente a la escena de tráfico sin intervención humana. Una vez finalizado el entrenamiento, la red se adapta para detectar mejor los objetos con la distancia de la cámara y la perspectiva captada.

Para llevar a cabo la fase de experimentación, ha sido necesario crear un nuevo conjunto de datos para realizar las pruebas pertinentes con el fin de evaluar la mejora cualitativa alcanzada por la técnica desarrollada. La mayoría de los conjuntos de datos existentes no son adecuados. Algunos como *KITTI* [5], se compone de una serie de imágenes recopiladas por cámaras situadas en vehículos móviles. Alguna de las imágenes que conforman este conjunto de datos poseen vehículos pero de gran tamaño. Es por ello por lo que se ha desarrollado un nuevo conjunto de datos de detección de vehículos, a partir de una serie de vídeos de tráfico capturados por cámaras y sistemas de vigilancias destinados a tal fin. Se han recopilado tres conjuntos de prueba filmados en diferentes lugares y con una serie de retos como la calidad de la imagen, la inferencia de la luz y el desenfoque del movimiento entre otros. Este conjunto de datos consta de 476 imágenes con un total de 14557 detecciones.

Bajo las premisas descritas anteriormente, es posible determinar que existen carencias importantes en los métodos existentes en la actualidad, para la detección de elementos de pequeña escala, dada la baja tasa de detección de los modelos actuales y los escasos procedimientos de mejora en el rendimiento.

1.5. Organización del resto del trabajo:

El resto del artículo se organiza de la siguiente forma. En la sección 2 se exponen los antecedentes de acuerdo con la propuesta desarrollada para este trabajo. A lo largo de la sección 3 se explica en profundidad el problema. En la sección 4 se detalla la solución desarrollada. La sección 5 recoge las pruebas realizadas junto con sus respectivos resultados con el fin de garantizar el correcto funcionamiento de la solución establecida. Finalmente, en la sección 6, se exponen las conclusiones y los trabajos futuros a desarrollar.

Capítulo 2 - Antecedentes:

Dado el contexto de nuestra propuesta, se requiere de un procesamiento de imágenes computacionalmente intensivo. Trabajos como [6, 7] han identificado nuevos métodos que aceleran significativamente la detección de elementos visuales, permitiendo así la inferencia de imágenes a alta velocidad. Aunque la potencia de cálculo necesaria para la inferencia es menor que para el entrenamiento, el uso de unidades de procesamiento gráfico (GPU) dedicadas suele ser esencial para aplicar el método en un tiempo razonable. Existen métodos que proponen algoritmos de detección enfocados a minimizar el tiempo de computación y así acelerar la detección [8, 9], pero su uso en aplicaciones que requieren una alta velocidad de respuesta (al menos 15 fotogramas por segundo) sigue requiriendo una potencia de cálculo considerable. Aunque existe una intensa línea de investigación en la adaptación de este tipo de métodos para trabajar utilizando hardware de menor potencia y coste [10, 11], actualmente no existe una solución general que trivialice el tiempo de computación necesario para aplicar la detección de objetos a una imagen.

En la actualidad, existen modelos pre-entrenados como por ejemplo *YOLO*. Este modelo divide la imagen en una cuadrícula para posteriormente, predecir las etiquetas de clase de cada cuadro delimitador. Otro modelo utilizado con frecuencia es *CenterNet* [12], el cual detecta cada uno de los objetos como un triplete en lugar de como un par, mejorando así la precisión en la detección y puntuación de la clase de cada elemento.

Según estudios anteriores, se ha establecido que los métodos basados en propuestas regionales obtienen una tasa de aciertos mayor en cuanto a la detección de elementos en una imagen. Por otro lado, cabe destacar que existen algunos detectores basados en *RCNs* como [13, 14], sin embargo, se hace referencia a la detección de elementos de pequeña escala sin entrar en grandes detalles. Otros trabajos como [15, 16] estudian la modificación de la arquitectura del modelo para detectar elementos de pequeño tamaño como caras y peatones respectivamente. Precisamente en la detección de elementos pequeños como los peatones, existen una serie de avances y desarrollos interesantes. Por ejemplo, *Zhang et al.* [17] proponen un detector de peatones que opera con representaciones neuronales multicapa para predecir la ubicación de los peatones con precisión. *Chen et al.* [18] presenta un nuevo marco de conocimiento para un detector de peatones el cual reduce el coste computacional asociado a las detecciones, manteniendo una alta precisión.

La videovigilancia inteligente aplicada al análisis del tráfico ha sido un tema de investigación recurrente en los últimos años, con multitud de trabajos relacionados con la detección de vehículos [19, 20], el seguimiento de objetos a lo largo de la escena [21] o el modelado de su comportamiento [22]. Las técnicas de aprendizaje profundo han reavivado el interés y el desarrollo de nuevas propuestas en el área. El paso de reconocer qué objeto muestra una imagen (problema conocido como clasificación de imágenes) [23, 24, 25, 26] a detectar varios objetos dentro de la imagen, sus clases y sus posiciones (problema conocido como detección de objetos), es un salto cualitativo que ha llevado a una revolución en el campo de la visión por computador con una gran cantidad de propuestas de procesamiento de imagen y vídeos capaces de beneficiarse de la identificación de objetos incluyendo estos métodos como parte clave [27, 28]. Las prestaciones de ambos problemas (clasificación de imágenes y detección de objetos) han experimentado una espectacular mejora al generalizarse el uso de las redes neuronales convolucionales profundas (*DCNN*). Por lo tanto, los métodos de detección de objetos basados en *DCNN* han sido un campo de intensa exploración en los últimos

años [29, 30, 31]. Sus principales retos son las clases detectables, el tiempo de procesamiento y la potencia computacional que requieren, así como su fiabilidad cuando se aplica en condiciones no ideales. *Kim et al.* [32] propone una técnica que reconoce la dirección del vehículo en la imagen de entrada. Este método puede ser adecuado para poder clasificar el modelo de vehículo basándose en su apariencia, sin embargo, a medida que la distancia del vehículo es mayor, la calidad de la imagen se deteriora, lo que se traduce en una peor precisión en el reconocimiento de elementos pequeños.

Por otro lado, en cuanto a los procesos de super-resolución de imágenes, se establecen una serie de técnicas para aumentar la resolución de una imagen de entrada. Existen varios avances y desarrollos para los procesos de super-resolución aplicados a una sola imagen. Por ejemplo, *Dong et al.* [33] propuso una red basada en procesos de super-resolución conocida como *SRCNN*. Esta red establece un mapeo de extremo a extremo entre las imágenes dadas como entrada con baja resolución y las procesadas por el modelo. *Jiwon Kim et al.* [34] proponen una red conocida como *VDSR* que apila una serie de capas convolucionales con aprendizaje de caracteres residuales. Asimismo, *Bee Lim et al.* [35] establecen una red denominada *EDSR* que consigue un alto rendimiento para los procesos de *SR* al eliminar las capas de normalización de lotes, dando lugar al modelo conocido como *SRResNet*.

Aunque estos procesos siguen estando en una fase primitiva y son complejos, hay avances significativos, principalmente centrados en la creación de algoritmos que realicen estas funciones de forma eficaz. Una de las principales aproximaciones en este campo es la presentada por *Dong et al.* [36], ya que propusieron algoritmos de super-resolución basados en el aprendizaje profundo. Una de las aplicaciones más destacadas en el ámbito de la super-resolución corresponde al procesamiento de imágenes de satélite, donde *Marc Bosch et al.* [37] establecen un análisis cuantitativo para determinar el éxito de las mejoras en la resolución de estas imágenes a través de la modificación de capas en el modelo de detección. La propuesta establecida por *Liujuan Cao et al.* [38] hace uso de imágenes aéreas de alta resolución para mejorar la detección de vehículos mediante un modelo lineal simple, obteniendo una mejora significativa de los resultados gracias a procesos de super-resolución. Trabajos como [39] abogan por la generación de imágenes de baja resolución pre-procesadas para aumentar la cobertura en las detecciones inferidas por el modelo.

El conjunto de clases detectables con un rendimiento razonable suele estar condicionado a la existencia de un conjunto de datos etiquetados lo suficientemente grande como para realizar el entrenamiento supervisado necesario. La obtención de estos datos suele requerir de intervención humana y el entrenamiento de la red con éxito es muy exigente en términos de tiempo y potencia de cálculo, por lo que obtener una red entrenada desde cero para un dominio específico no es trivial. Estas razones motivan a trabajar a partir de una red pre-entrenada y entrenada con algunos de los conjuntos de datos más comunes (por ejemplo, Common Objects in Context, también conocido como COCO [40]) y, sólo si es necesario, realizar algún tipo de ajuste fino con un conjunto de datos más pequeño y específico del dominio para adaptar la red al dominio de aplicación.

La creación y adaptación de métodos de detección de objetos basados en DCNN para hacer frente a condiciones que distan de ser ideales es también un campo de investigación intenso. Uno de los retos más típicos es utilizar la detección de objetos con una gran distancia entre la cámara y los objetos. Este problema se conoce como Detección de Objetos Pequeños y su importancia queda demostrada por la cantidad de enfoques recientes aplicados para resolverlo: métodos basados

en redes Adversariales Generativas [41], métodos que adaptan el funcionamiento de los métodos existentes [42] o estudios sobre cómo afectan las funciones de pérdida a este problema [43]. La mayoría de los enfoques para aumentar la efectividad en la detección de objetos pequeños implican aumentar el tiempo de procesamiento.

Capítulo 3 - Descripción del problema:

Con este trabajo de investigación lo que se pretende es determinar la mejora obtenida tras aplicar procesos de super-resolución en elementos de tamaño reducido. Tal y como se ha establecido en la introducción, estos elementos son aquellos que componen aproximadamente el 1% de la imagen al completo.

El contexto de la detección de objetos presentan dos problemas que las redes neuronales convolucionales pretenden resolver. En primer lugar, encontramos el problema de la clasificación de imágenes. En este caso, al modelo se le da una imagen como entrada con el objetivo de que infiera en la clase del elemento que conforma la misma. En este caso, la inferencia se realiza atendiendo a rasgos del elemento en cuestión como puede ser las formas, colores o proporciones. Una imagen se traduce en el contexto de las redes neuronales convolucionales a un vector de características, determinando por ello una clase en particular. Cuando se detecta un elemento, se infiere de tal modo que se establezcan una serie de clases en base a aquellas utilizadas en el entrenamiento del modelo. Cada una de ellas se determinará con una probabilidad en concreto, estableciendo la clase final como aquella con un mayor porcentaje de precisión.

Otro de los problemas que encontramos relacionado con el contexto de dicho trabajo corresponde con la localización. Tras dar como entrada una imagen, es posible que en la misma esté conformada por múltiples elementos distribuidos a lo largo de la misma. De acuerdo al ámbito del proyecto, será necesario detectar vehículos a través de vídeos captados por sistemas de videovigilancia, es por ello por lo que en un mismo fotograma aparecerán múltiples elementos que deben ser detectados.

En la actualidad, existen una serie de modelos pre-entrenados, cada uno de ellos con una arquitectura definida, sin embargo, la inferencia de los mismos es muy baja a la hora de detectar elementos pequeños. Para determinar por qué se produce este hecho, es necesario establecer el funcionamiento de una red neuronal convolucional.

Las redes neuronales convolucionales están compuestas por una serie de capas. Las capas se organizan en tres dimensiones atendiendo a la anchura, el alto y la profundidad de las imágenes dadas como entrada. Con respecto al funcionamiento de las redes neuronales regulares, al introducir una imagen esta se transforma a un vector plano de píxeles, obviando por ello la importancia que tiene la posición de los elementos dentro de la imagen. La independencia de píxeles no es habitual en una imagen, y debe ser tenida en cuenta para la extracción de patrones y formas.

En las redes neuronales convolucionales, la imagen inicial se va comprimiendo espacialmente, hecho el cual corresponde con una disminución de la resolución al mismo tiempo que el número de mapas de características que va detectando aumenta. En la figura número 2 podemos ver a grandes rasgos como se conforman las arquitecturas de estas redes neuronales convolucionales, determinando como conforme se va avanzando en el número de capas convolucionales, se va reduciendo el número de píxeles a procesar.

Volviendo al contexto del ámbito sobre el cual se basa este trabajo, las imágenes captadas por sistemas de videovigilancia poseen una serie de retos que deben ser tenidos en cuenta para mejorar las detecciones establecidas por el modelo. Los principales retos que caben destacar en primer lugar

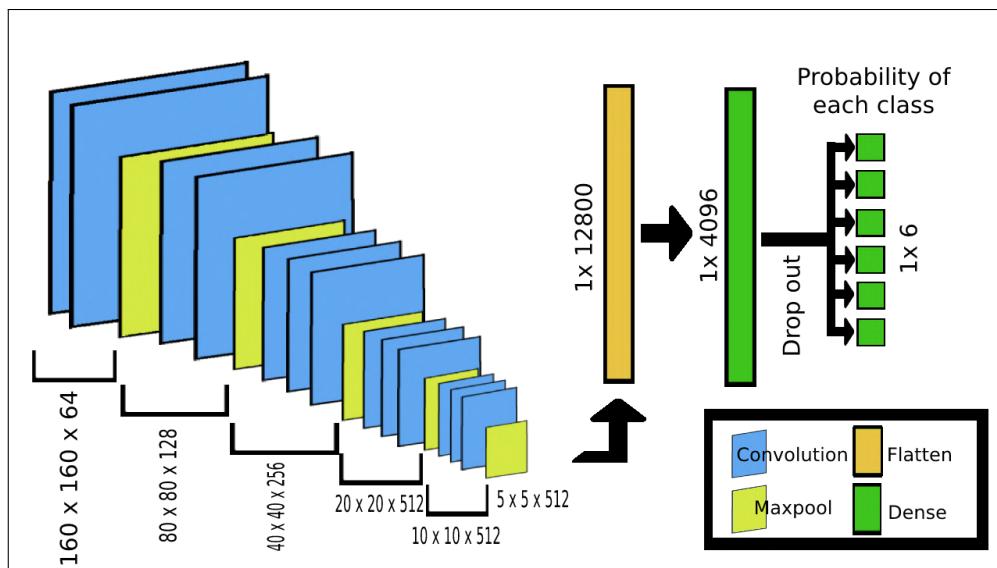


Figura 2: Ejemplo de arquitectura de una red neuronal convolucional

es la pequeña proporción en cuanto al número de píxeles se refiere en elementos de pequeño tamaño. Por otro lado, también hay que tener en cuenta la baja calidad con la que son captados los mismos conforme se van alejando del sistema el cual capta dichas imágenes. Finalmente otro factor que repercute corresponde con el desenfoque de los elementos, pues en el contexto de los vehículos estos avanzan a una determinada velocidad. En la figura 3 se definen los mismos.



Figura 3: Retos de las imágenes de entrada

Una vez definido el problema del cual partimos con respecto a la detección de vehículos de pequeños tamaños debido a la forma de procesar las imágenes las redes neuronales convolucionales, así como los retos los cuales son necesarios tener en cuenta, en el punto 4 se desarrollará la propuesta establecida para mejorar la precisión ante dicho problema.

Capítulo 4 - Tecnología utilizada y detalles de la propuesta:

A lo largo de este punto, se detallarán las tecnologías utilizadas para llevar a cabo la propuesta. En primer lugar, en el punto 4.1 se realiza una comparativa determinando los diversos modelos destinados para la realización de *SR*. En el punto 4.2 se establecen las redes que han sido utilizadas para llevar a cabo las pruebas. Posteriormente, se detalla la propuesta en el punto 4.3. Se describe como se ha establecido el flujo de trabajo desarrollado a lo largo del punto 4.4 y se desarrolla una versión para agilizar este proceso en el punto 4.5. Finalmente, en la subsección 4.6 se detalla como se ha llevado a cabo el ajuste fino.

4.1. Modelo pre-entrenado para aplicar SR:

En este trabajo, se ha hecho uso de un modelo pre-entrenado que aplique super-resolución a la imagen inicial dada como entrada, es por ello por lo que se va a detallar qué es y cómo funciona el modelo seleccionado. Existen algunos modelos pre-entrenados para la ejecución de los procesos que aumentan la resolución inicial de una imagen y generan resultados similares, tal y como se puede ver en la Figura 4. Los modelos considerados son los siguientes:

- Fsrcnn: *Fast Super-Resolution Convolutional Neural Network* [44].
- Edsr: *Enhanced Deep Residual Networks Single Image Super-Resolution* [35].
- Espcn: *Efficient Sub-Pixel Convolutional Neural Network* [45].
- Lapsrn: *The Laplacian Pyramid Super-Resolution Network* [46].



Figura 4: Comparación de los diversos modelos de SR.

En primer lugar, es necesario determinar que tipo de modelo se ajusta a las necesidades destinadas para este trabajo. Uno de los aspectos a considerar es el tiempo, es por ello por lo que de acuerdo

con el conjunto de datos desarrollado, se ha hallado en la Tabla 1 la media, mediana y desviación estándar para cada uno de ellos con el objetivo de determinar cual es más conveniente a utilizar en escenarios donde el tiempo sea decisivo. En segundo lugar, se ha realizado una comparativa establecida en la Tabla 2, destinada al estudio de los tiempos de procesamiento, así como los resultados obtenidos para la proporción máxima de señal a ruido denotado como *PSNR* y la similitud estructural *SSIM*.

Modelo	Media	Mediana	Desviación estándar
EDSR	504.098	503.936	± 2.033
ESPCN	0.706	0.703	± 0.0154
FSRCNN	0.745	0.742	± 0.0169
LAPSRN	6.892	6.901	± 0.0714

Tabla 1: Media, mediana y desviación estándar definida en segundos para cada uno de los modelos de super-resolución. En negro se resaltan los mejores tiempos obtenidos.

En la Figura 5 es posible determinar el tiempo requerido para el modelo *FSRCNN* en comparación con uno de los más actualizados, correspondiéndose con *SRCNN*.

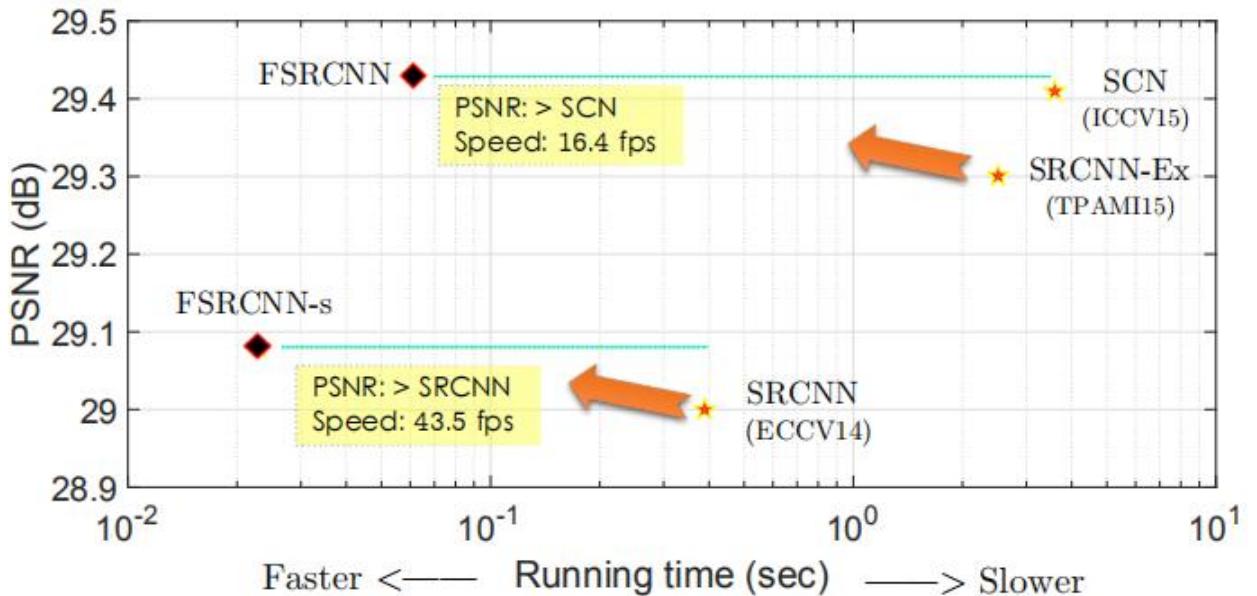


Figura 5: Velocidad del modelo *FSRCNN* [44].

Tal y como se muestra en la tabla 1, modelos como *EDSR* o *LAPSRN* quedan totalmente descartados del ámbito del trabajo pues requiere de gran cantidad de tiempo para procesar la imagen super-resuelta. Por otro lado, vemos como los tiempos obtenidos tanto por *FSRCNN* como *EDSR* son similares, siendo este último el más rápido. Realizando una comparación con otros estudios realizados sobre el rendimiento de los modelos de super-resolución bajo el conjunto de datos denominado como *General100 dataset* [44] destinado para medir el rendimiento especialmente del modelo *FSRCNN*, podemos comprobar efectivamente como estos últimos modelos son los más rápidos. Adicionalmente, debemos tener en cuenta valores como la máxima señal a ruido así como la similitud estructural. Realizando la comparativa sobre el conjunto de datos *General100 dataset*, los resultados obtenidos son los siguientes:

Modelo	Avg PSNR	Avg SSIM
EDSR	34.1300	0.9447
ESPCN	32.7059	0.9276
FSRCNN	32.8886	0.9301
LAPSRN	32.2681	0.9248

Tabla 2: Máxima señal de ruido media así como la similitud estructural media obtenida para el conjunto de datos General 100 con los modelos de super-resolución utilizados. En negro se resaltan los mejores tiempos obtenidos.

En la tabla 2, podemos determinar que los mejores resultados para un factor de ampliación x2 son provistos por el modelo *EDSR* sin embargo este queda fuera del ámbito de contextos en los que el tiempo sea un factor clave. Es por ello por lo que realizando una comparativa entre tiempos y resultados obtenidos, se ha seleccionado como candidato el modelo denotado como *FSRCNN* para la realización de las sucesivas pruebas.

FSRCNN hace uso de imágenes de baja resolución como entrada, defiriendo de otros modelos como por ejemplo *SRCNN*. Alguna de las características de este modelo se basan en el establecimiento de *upsampling* en las capas finales que componen la red, no existe un mapeo no lineal y el filtro seleccionado para el procesamiento de la imagen es de pequeño tamaño, con una estructura de red más profunda. Tal y como se puede ver en la Figura 6, este modelo está compuesto por una serie de capas que se detallan a continuación.

Extracción de características: A diferencia de otros modelos, *FSRCNN* procesa la imagen de baja resolución directamente dada como entrada. Es por ello por lo que hace uso de un tamaño de *Kernel* (núcleo) menor.

Reducción: En esta capa se produce la reducción de la dimensionalidad. En base a las características obtenidas de la imagen de baja resolución, el coste computacional será elevado, el por ello por lo que el tamaño de núcleo de nuevo se reduce, siendo el óptimo 1*1.

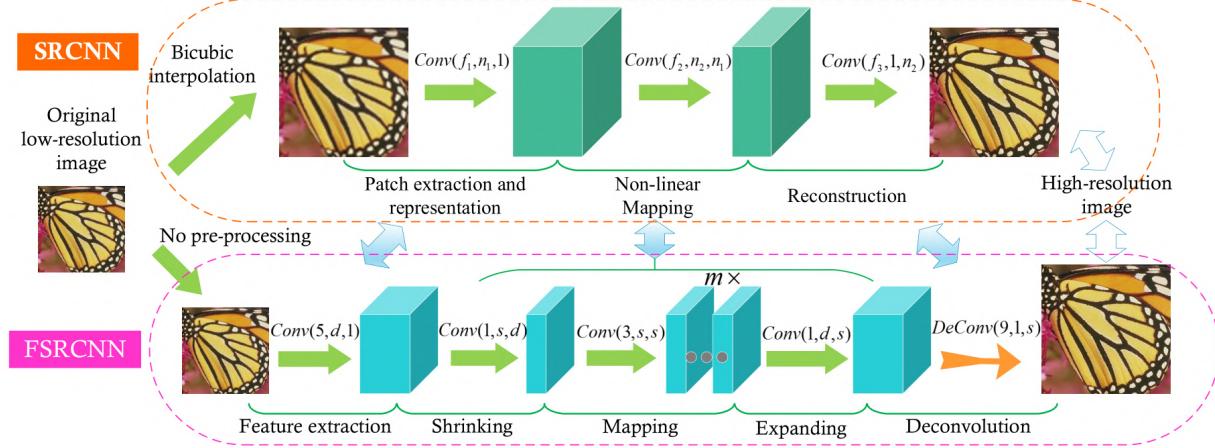


Figura 6: Estructura del modelo *FSRCNN* [44].

Mapeo no lineal: En esta capa se selecciona un kernel de 3×3 y se conectará a la capa de convolución que compone el modelo.

Expansión: El autor del modelo establece que la fase de reconstrucción directamente sobre la imagen de baja resolución dada como entrada no era adecuada pues la calidad de la imagen es demasiado pobre.

Deconvolución y PReLU: Se lleva a cabo una operación inversa a la de convolución con el fin de ampliar el mapa de características y finalmente se lleva a cabo la función de activación denotada como *PReLU*.

4.2. Redes Neuronales Convolucionales utilizadas:

La propuesta desarrollada puede ser aplicada utilizando cualquier modelo neuronal de detección de objetos basado en *DCNN* siempre y cuando las detecciones establecidas inicialmente sean correctas. Para realizar las pruebas, se han utilizado diversas variantes del modelo *EfficientNet* y *CenterNet*. Con respecto a *EfficientNet*, existe una serie de variantes que van desde la cero a la siete, donde cada una de ellas representan una serie de alternativas de acuerdo con la eficiencia y la precisión según una escala determinada. Gracias a esta heurística de escalado, es posible que el modelo base denotado como *D0* supere a los modelos de cada escala, evitando además una extensa búsqueda en la red de hiperparámetros. En un principio, puede dar la impresión de que *EfficientNet* constituye una familia continua de modelos definidos por una elección arbitraria de la escala, sin embargo, se contemplan otros aspectos como la resolución, la anchura y la profundidad. En cuanto a la resolución, en los modelos de baja escala como *D0* o *D1* se aplica un relleno de cero cerca de los límites de ciertas capas, con lo que se desperdician recursos computacionales. En cuanto a la profundidad y la anchura, el tamaño de los canales debe ser múltiplo de ocho. Respecto a *CenterNet*, se han utilizado tres variantes, cada una de ellas con una arquitectura definida, obteniendo como principales diferencias el tamaño de la imagen dada como entrada, así como la forma de procesar cada una de las detecciones, puesto que se ha probado con variantes basadas en *Keypoints* así como

modelos tradicionales. Estos modelos pre-entrenados han sido extraídos de *Tensorflow Model Zoo*².

4.3. Detalles de la propuesta:

El sistema propuesto para la mejora del rendimiento de la detección de objetos mediante redes neuronales convolucionales profundas se detalla a continuación. Nuestra propuesta se compone de dos subsistemas. En primer lugar, la salida de una red neuronal convolucional profunda original utilizada para la detección de objetos se refina por super-resolución para obtener detecciones de alta calidad. Este subsistema se describe en la subsección 4.4. Se presenta además una forma de agilizar las detecciones iniciales en la subsección 4.5. A continuación, las detecciones de alta calidad se emplean para construir un nuevo conjunto de entrenamiento para la red profunda de detección de objetos, con la finalidad de realizar un ajuste fino de la red el cual permita mejorar su rendimiento, como se explica en la subsección 4.6. El flujo de trabajo se presenta en la figura 7. El código desarrollado se encuentra subido al siguiente repositorio de *GitHub*³.

4.4. Mejora tras aplicar super-resolución en la detección de objetos:

Esta sección está dedicada a la descripción del procedimiento desarrollado para mejorar la calidad de la detección de objetos de una red neuronal convolucional profunda. Nuestro punto de partida es una red convolucional profunda original destinada a la detección de objetos, cuya entrada es una imagen \mathbf{X} y produce un conjunto de detecciones S como salida:

$$S = \mathcal{F}(\mathbf{X}) \quad (1)$$

$$S = \{(a_i, b_i, c_i, d_i, q_i, r_i) \mid i \in \{1, \dots, N\}\} \quad (2)$$

Donde N representa el número de detecciones, $(a_i, b_i) \in R^2$ son las coordenadas de la esquina superior izquierda de la i -ésima detección dentro de la imagen, \mathbf{X} , $(c_i, d_i) \in R^2$ corresponde con las coordenadas de la esquina inferior derecha de la detección i -ésima dentro de \mathbf{X} , q_i es la etiqueta de clase de la detección y $r_i \in R$ denota la puntuación de la clase de la detección. Cuanto mayor sea r_i , mayor es la confianza en la cual se detecte un objeto de la clase q_i en dicha detección. Se supone que el origen del sistema de coordenadas de todas las imágenes se encuentra en el centro de la imagen. A continuación, se realiza la selección de cada una de las detecciones establecidas inicialmente. La segunda tarea a realizar en nuestro procedimiento es procesar la entrada dada una imagen de entrada de baja resolución \mathbf{X}_{LR} con la red profunda como se indica en el paso dos del flujo de trabajo 7. Tras realizar la detección inicial de la imagen dada como entrada, se obtiene un conjunto de detecciones tentativas S_{LR} :

$$S_{LR} = \mathcal{F}(\mathbf{X}_{LR}) \quad (3)$$

² https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

³ https://github.com/IvanGarcia7/SR-FT_ENHANCEMENT

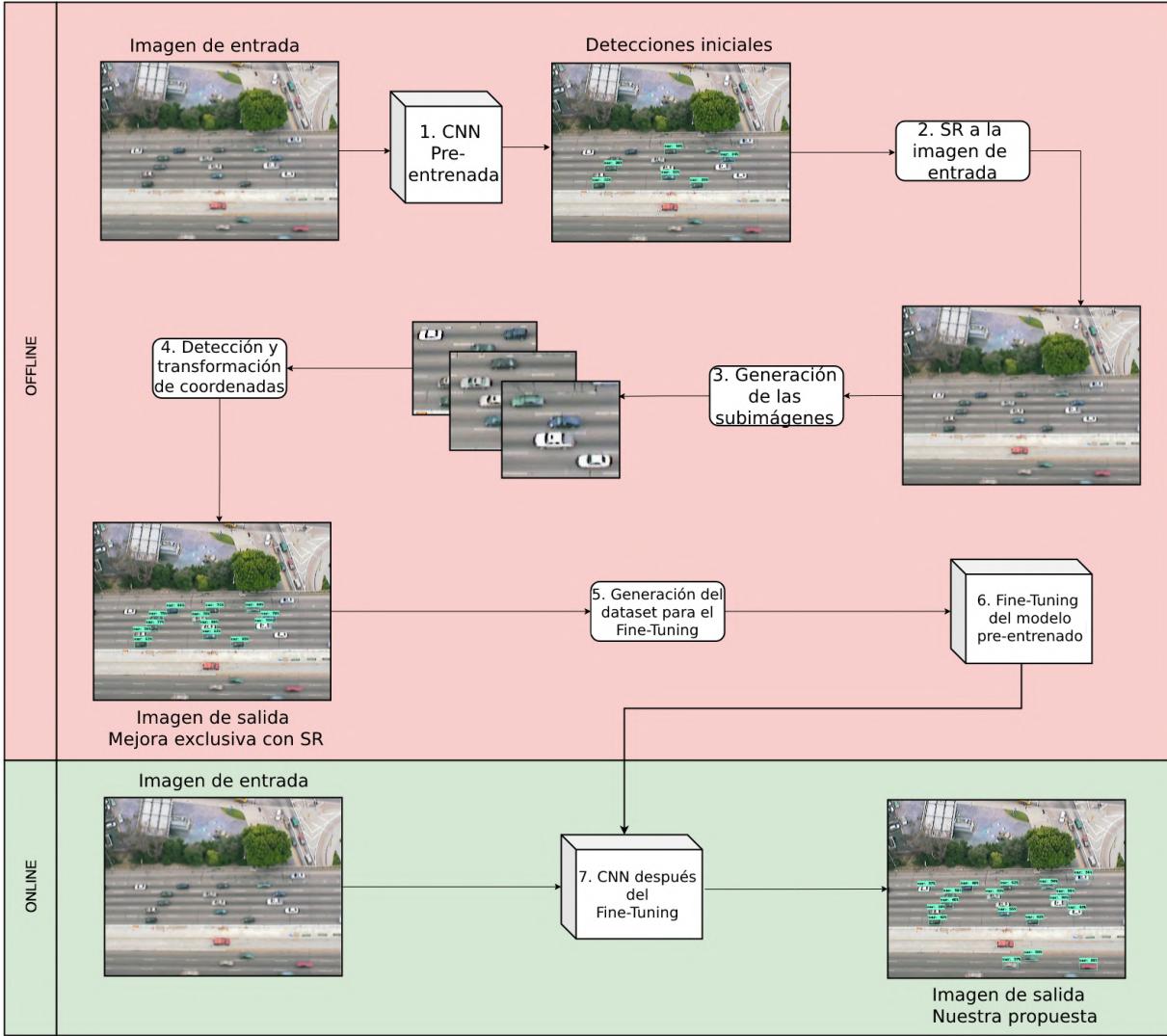


Figura 7: Flujo de trabajo de la técnica propuesta. La parte *Offline* se compone de la aplicación de *Super-Resolución (SR)*, además de la generación del conjunto de datos para el *Fine-Tuning*. La parte *Online* se compone de la detección realizada por el modelo pre-entrenado.

Posteriormente, como se detalla en el tercer paso del flujo de trabajo, se genera una imagen aplicando procesos de super-resolución y *denoising* (disminución de ruido), a la imagen inicial dada como entrada.

Se utiliza una red profunda de super-resolución para obtener una imagen de alta resolución, denominada como \mathbf{X}_{HR} con un factor zoom Z a partir de la imagen de entrada de baja resolución \mathbf{X}_{LR} . En la figura 8 se muestra un ejemplo del procedimiento aplicado a la imagen de entrada. Para implementar este proceso de super-resolución, se ha utilizado la librería *OpenCV*. Tal y como se ha visto en la subsección 4.1, existen multitud de modelos pre-entrenados destinados a aplicar super-resolución.

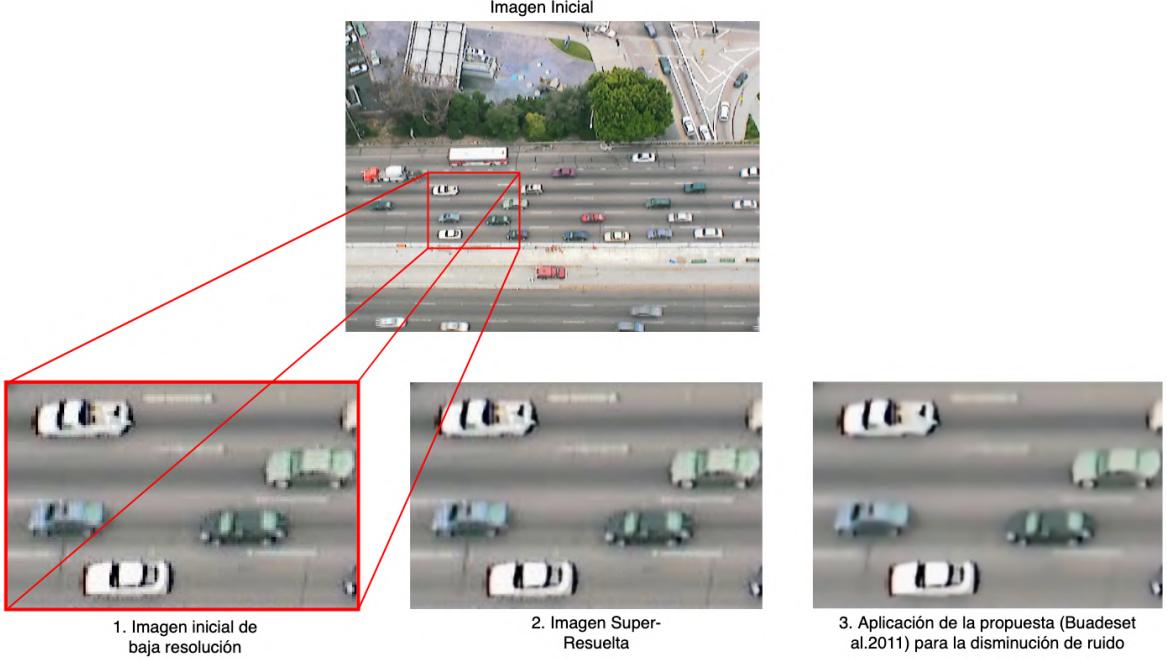


Figura 8: Procesamiento de la imagen

Atendiendo a la calidad final de la imagen obtenida por cada uno de estos modelos, así como al tiempo necesario para realizar el proceso de procesamiento, se ha seleccionado el modelo *Fsrcnn*, por ser uno de los que procesan las imágenes con mayor rapidez y calidad. Posteriormente, para mejorar las detecciones por parte de la red, se realiza una eliminación de ruido. Este proceso se lleva a cabo mediante el uso del algoritmo de eliminación de ruido no local [47], haciendo uso de varias optimizaciones computacionales. Este método se aplica cuando se espera un ruido blanco *gaussiano* (punto 3 de la figura 8). Tras un estudio empírico, se concluyó con que este tipo de procesamiento mejora el número de elementos detectados.

Entonces, para cada detección en S_{LR} , una subimagen \mathbf{X}_i con el mismo tamaño que \mathbf{X}_{LR} se extrae de \mathbf{X}_{HR} . El centro de la ventana \mathbf{X}_i coincide con el centro de la detección:

$$\mathbf{y}_i = \left(\frac{a_i + c_i}{2}, \frac{b_i + d_i}{2} \right) \quad (4)$$

$$\hat{\mathbf{y}}_i = Z\mathbf{y}_i \quad (5)$$

Donde \mathbf{y}_i es el centro de \mathbf{X}_i expresado en coordenadas de \mathbf{X}_{LR} , mientras que $\hat{\mathbf{y}}_i$ corresponde con el centro de \mathbf{X}_i expresado en coordenadas de \mathbf{X}_{HR} . La red de detección de objetos se aplica a \mathbf{X}_i para producir un nuevo conjunto de detecciones, lo que lleva al paso cuatro del flujo de trabajo.

$$S_i = \mathcal{F}(\mathbf{X}_i) \quad (6)$$

Las detecciones de S_i se expresan en coordenadas de \mathbf{X}_i , es por ello por lo que es necesario trasladar las posiciones de los elementos al sistema de coordenadas de \mathbf{X}_{LR} . Para ello, se hace uso de la siguiente ecuación encargada de trasladar un punto $\tilde{\mathbf{h}}$ en el sistema de coordenadas de \mathbf{X}_i , a coordenadas \mathbf{h} de \mathbf{X}_{LR} :

$$\mathbf{h} = \mathbf{y}_i + \frac{1}{Z}\tilde{\mathbf{h}} \quad (7)$$

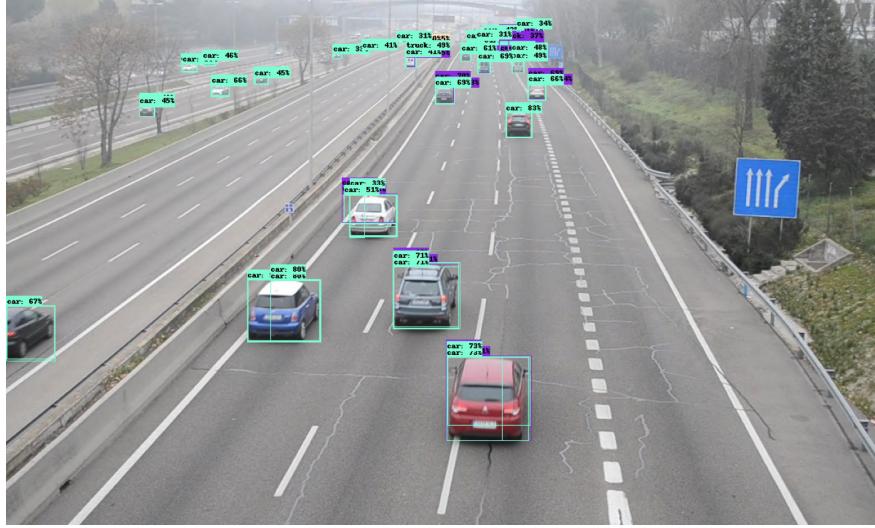


Figura 9: Fotograma de vídeo antes de filtrar las detecciones coincidentes

Tal y como se muestra en la Figura 9, para un mismo objeto habrá varias detecciones posibles. Esto ocurre debido a las múltiples detecciones almacenadas por cada una de las subimágenes generadas. Es por ello por lo que por cada pasada, se almacena una tripla T conformada por los *bounding boxes* de los objetos detectados, la clase a la que pertenecen así como la puntuación, obteniendo por ello la tripla $T_i = (\text{bounding Boxes}, \text{clase}, \text{puntuación})$. Al aplicar la red neuronal convolucional a cada una de las subimágenes generadas a partir de las detecciones iniciales, se almacenará por ello una lista de triples $L = [T_0, T_1, \dots, T_n]$.

A continuación, se crea un cluster de objetos, para ello, mientras se encuentren elementos en la lista de triples L , se itera sobre la misma seleccionando el objeto O_k de la tripla L_k con mayor puntuación, devolviendo la posición del objeto O_k así como la lista a la cual pertenece L_k . Acto seguido, se elimina de la tripla almacenada en la lista y se incluye en el cluster de objetos K conformado por las *bounding boxes*, su clase y su respectiva puntuación.

Posteriormente, se itera sobre el resto de triples contenida en la lista a excepción de L_k con el fin de buscar detecciones pertenecientes al objeto O_k detectadas en las múltiples pasadas realizadas. Para ello, atendiendo a un umbral θ establecido, se realiza la operación de IoU . Se considera que las detecciones D_j y D_k corresponden al mismo objeto, siempre que $IOU > \theta$.

$$IOU = \frac{Area(D_j \cap D_k)}{Area(D_j \cup D_k)} \quad (8)$$

Tras eliminar todos los posibles objetos de la lista de triples L , se devuelve el cluster de elementos, del cual se seleccionará de cada cluster el elemento con mayor puntuación. Al final de este proceso, se obtendrá una imagen con un mayor número de detecciones y con una mejora en la inferencia de clase de cada elemento, véase por ello la Figura 10. Estas detecciones se almacenarán para posteriormente realizar el *Fine-Tuning* al modelo neuronal convolucional pre-entrenado.



Figura 10: Resultado final tras filtrar las detecciones coincidentes

4.5. Agilizar la generación de detecciones - Algoritmo de *Bron-Kerbosch*:

La primera versión desarrollada, basada en la generación de subimágenes por cada una de las detecciones iniciales, obtiene un mayor número de detecciones así como una mayor precisión en cuanto a la clase de los elementos que conforman la imagen, sin embargo es necesario destacar que se debe realizar una inferencia por cada elemento inicialmente detectado, es por ello por lo que el tiempo aumenta considerablemente. En esta segunda versión, se ha definido una variante del flujo de trabajo con el fin de optimizar el tiempo de procesamiento para obtener mejores resultados con respecto a los obtenidos por el modelo inicial en un menor tiempo.

Para llevar a cabo esta segunda versión, se ha modificado por ello el punto 3 del flujo de trabajo. En lugar de generar una imagen por cada elemento detectado y con centro en el mismo, en esta ocasión se buscan ventanas dentro del fotograma super-resuelto que contengan la mayor cantidad de posibles vehículos detectados en base al fotograma inicial. Consideremos por ello el grafo no dirigido denotado como $G = (V, E)$, donde cada nodo V referencia a cada uno de los vehículos detectadas en el fotograma inicial mientras que E corresponde con la distancia que separa cada uno de estos elementos detectados. Dos vehículos están conectados en el grafo, si y solo si sus *bounding boxes* caben dentro de una misma ventana en el fotograma super-resuelto. Posteriormente, se obtiene la

lista de los *cliques* sobre el grafo no dirigido con el algoritmo de *Bron–Kerbosch*. El pseudo-código de dicho algoritmo se detalla a continuación:

```

1 algorithm BronKerbosch1(R, P, X) is
2     if P and X are both empty then
3         report R as a maximal clique
4     for each vertex v in P do
5         BronKerbosch1(R ∪ {v}, P ∩ N(v), X ∩ N(v))
6         P := P \ {v}
7         X := X ∪ {v}

```

El funcionamiento es el siguiente, se van cogiendo los *cliques* y se calcula el centroide de los centroide de los *bounding boxes* de los vehículos del *clique*. A partir de ese centroide, se comprueba qué vehículos quedan completamente dentro de la ventana definida por ese centroide, y se señalan esos vehículos como procesados. Se siguen cogiendo cliques hasta que todos los vehículos estén señalados como procesados. En la Figura 11 se muestra el flujo de trabajo definido haciendo uso del algoritmo presentado.

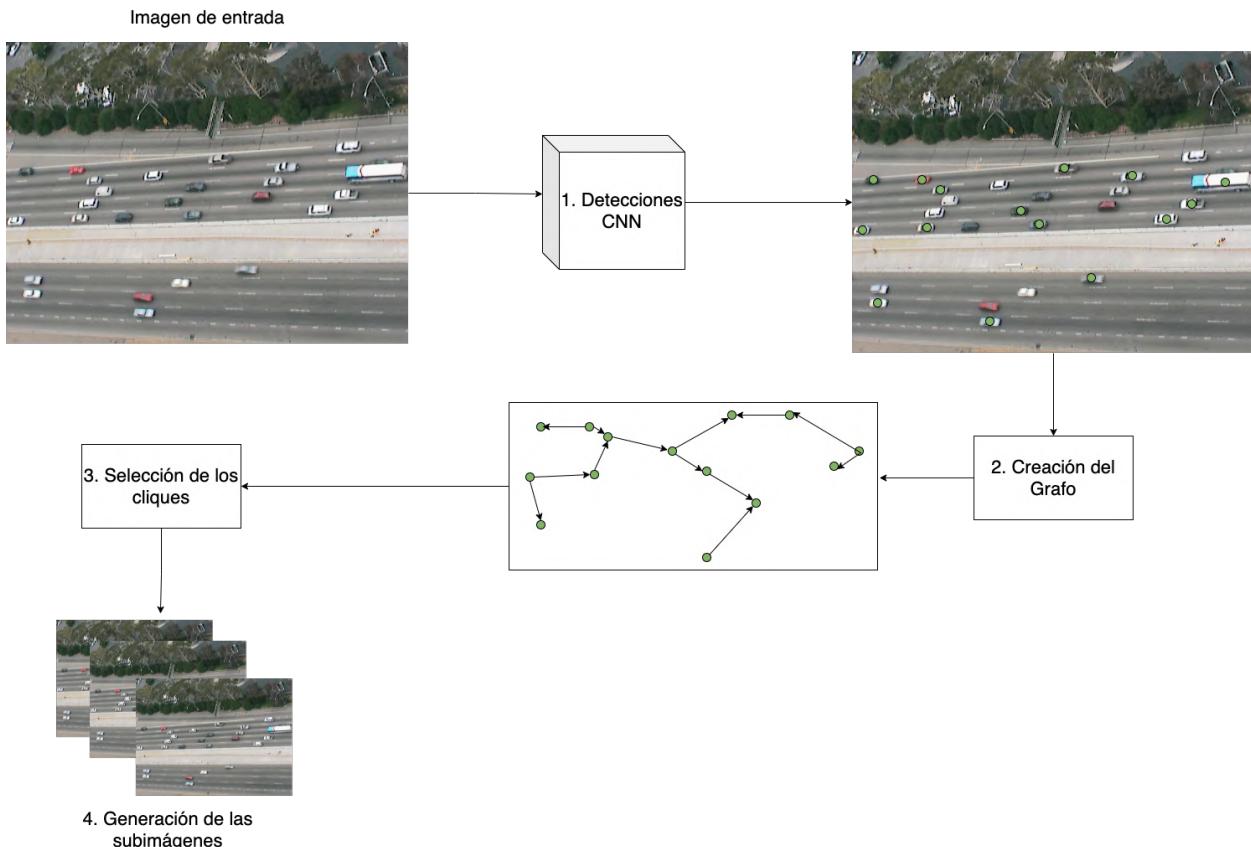


Figura 11: Flujo de trabajo del algoritmo descrito con el algoritmo de *Bron–Kerbosch*.

4.6. *Fine-Tuning* mediante datos de entrenamiento etiquetados automáticamente:

Posteriormente, tal y como se detalla en el quinto paso del flujo de trabajo de la Figura 7, se propone un procedimiento automatizado para generar un conjunto de datos de entrenamiento etiquetados para llevar a cabo el *Fine-Tuning* de una red de detección de objetos. Como se indica en el sexto paso del flujo de trabajo, una vez que una serie de detecciones de alta calidad W_{final} se obtiene del subsistema descrito en la subsección 4.4, se emplea como conjunto de entrenamiento para afinar la red de detección de objetos. La razón de ello es que el conjunto de detecciones de alta calidad contiene mayor número de detecciones con una mayor confianza con respecto al conjunto de detecciones obtenidas por la red de detección de objetos original. Por lo tanto, se espera que el *Fine-Tuning* de la red con W_{final} , dé lugar a una mayor precisión de la red de detección de objetos ajustada, como puede verse en la imagen dada como resultado por el modelo después de ser re-entrenado en el paso número siete del flujo de trabajo.

Capítulo 5 - Resultados:

En este punto se detalla como se ha creado el conjunto de trabajo destinado para la realización de las pruebas a lo largo de la subsección 5.1, así como la implicación que lleva a cabo procesos de reducción de ruido para la mejora de los resultados obtenidos en cuanto a términos de precisión se refiere dentro del punto 5.2. Para ello, se ha realizado una comparativa determinando la diferencia en términos de precisión que se obtienen al llevar a cabo este proceso. Se detalla un estudio realizado determinando la mejora tanto de carácter cualitativo a través de una serie de *frames* o imágenes sobre los cuales se han llevado a cabo el flujo de trabajo presentado, así como resultados cuantitativos en términos de precisión utilizando la métrica provista por *COCO* con el fin de determinar la precisión, véase por ello el punto 5.3 y 5.4. Finalmente, se ha realizado una comparativa entre las dos versiones presentadas en el punto 5.5. Con ello lo se ha pretendido ha sido mostrar de forma visual como mejora el tiempo necesario para llevar a cabo el procesamiento de la imagen, sacrificando aspectos como la precisión, ya que esta segunda versión es menos precisa pero más rápida.

5.1. Generación del conjunto de datos destinado a las pruebas:

Para la realización de las pruebas, en la actualidad no existen conjuntos de datos etiquetados dentro del ámbito de la detección de vehículos de tamaño reducido. Conjuntos de datos como *KITTI* poseen imágenes con multitud de vehículos, sin embargo, el tamaño de los mismos no es adecuado para el ámbito sobre el cual gira dicho trabajo.

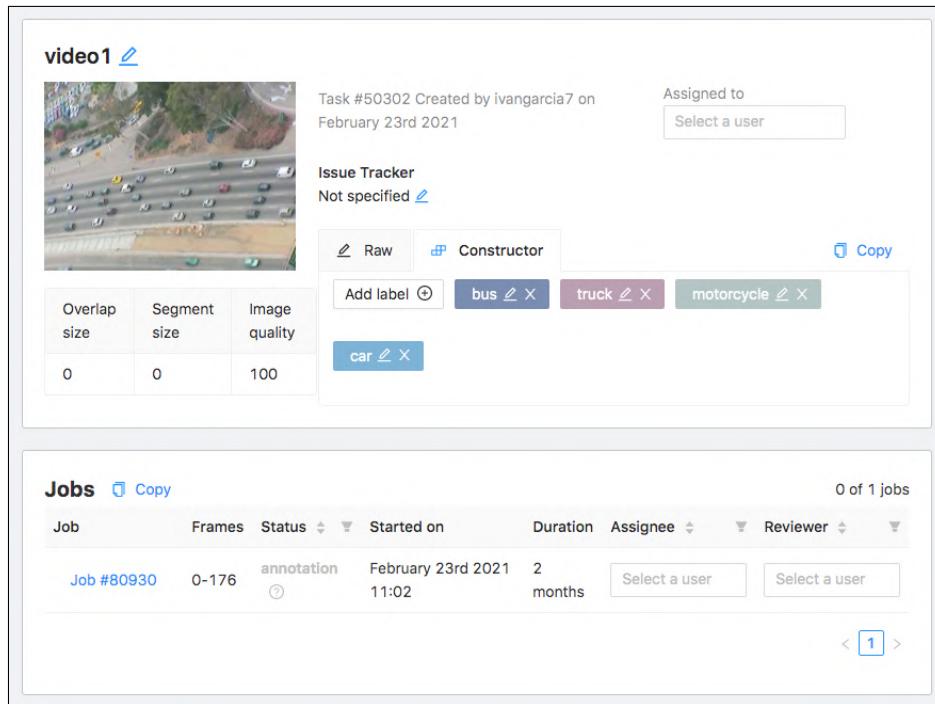


Figura 12: Creación de un nuevo proyecto para el vídeo uno.

Para probar nuestra propuesta, se han utilizado tres secuencias de vídeo de *US Highway 101 Dataset*⁴. Los vídeos toman los nombres *sb-camera1-0820am-0835am*, *sb-camera3-0820am-0835am* y *sb-camera4-0820am-0835am* respectivamente y han sido capturados por sistemas de videovigilancia en autopistas con gran perspectiva. En la Figura 12 se muestra como se ha definido el proyecto una vez creado. Estos vídeos pertenecen al departamento de transporte de Estados Unidos. Los sistemas recogen una serie de secuencias con un gran número de vehículos pequeños. Las secuencias tienen una duración total de unos 15 minutos cada una de ellas. Como se puede ver en el nombre de cada uno de los vídeos, están formados por el número de la cámara a la cual hace referencia, así como por la hora a la que se tomaron las imágenes. Para la realización de este conjunto de datos, se ha hecho uso de la herramienta denominada como *CVAT (Computer Vision Annotation Tool)*. En la figura 13 se muestra uno de los *frames* anotados con dicha herramienta.

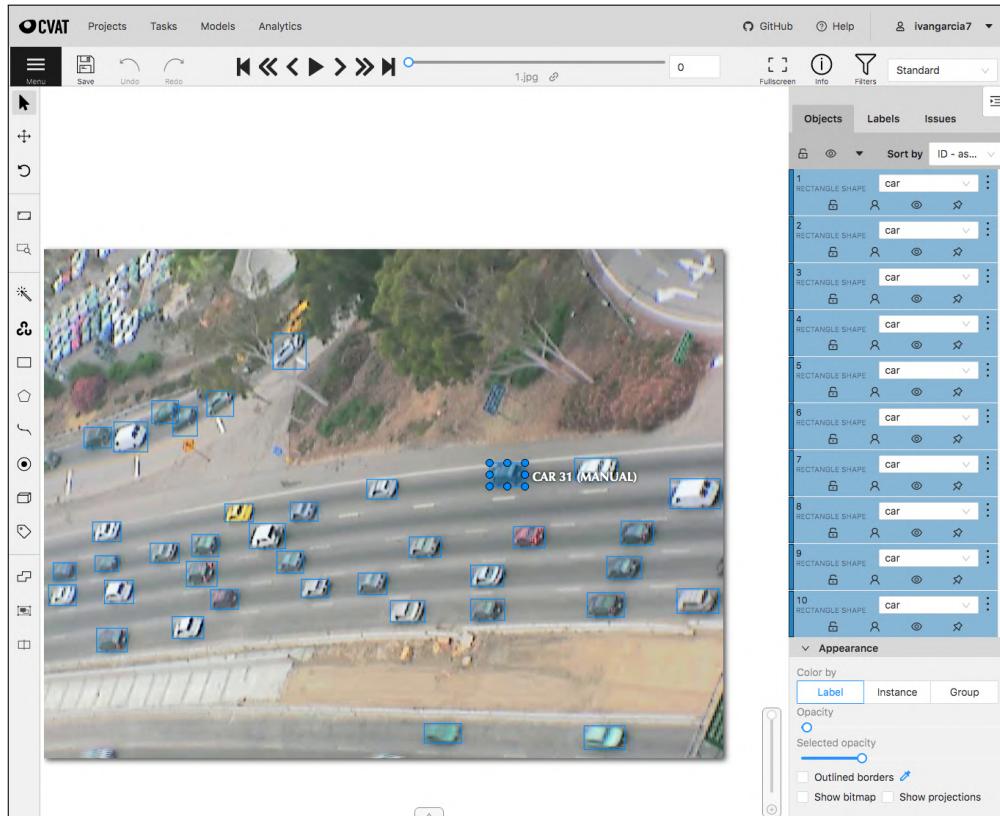


Figura 13: Ejemplo de uno de los *frames* anotados para el vídeo uno.

Desde el inicio de cada secuencia de vídeo se han anotado manualmente 1 minuto, dando a un total de **476** imágenes etiquetadas, con un total de **14557** vehículos. Es posible descargar el conjunto de datos anotados del siguiente repositorio de *GitHub*⁵. Con el objetivo de evaluar nuestra pro-

⁴ <https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm>

⁵ <https://github.com/IvanGarcia7/NGSIM-Dataset-Annotations>

puesta, se ha obtenido la Precisión Media (*mAP*) una vez aplicado el *Fine-Tuning*. Los siguientes 30 segundos de vídeo se han descartado para garantizar que un mismo vehículo perteneciente al conjunto de evaluación, no aparezca dentro del conjunto de datos destinado al entrenamiento. A partir de los siguientes fotogramas, se genera el conjunto de datos de entrenamiento aplicando la mejora de *SR*. Se han seleccionado cuatro categorías específicas relacionadas con la detección de vehículos en carreteras, es por ello por lo que se ha definido las etiquetas coche, camión, moto y autobús para realizar un análisis de los resultados. El conjunto de datos está compuesto por una serie de imágenes con ciertas características interesantes para el estudio de los resultados obtenidos, destacando principalmente los siguientes aspectos:

- Los vehículos establecidos en cada fotograma ocupan un área pequeña de la imagen en total, es por ello por lo que podemos determinar estos elementos como pequeños.
- Existe un desequilibrio de clases: Hay categorías de elementos más abundantes, como los coches frente a las motos, que aparecen en un número limitado de fotogramas.

5.2. Aplicación de procesos de disminución de ruido:

Tal y como se ha comentado en puntos anteriores, la aplicación de eliminación de ruido a una imagen repercute en el número de elementos que detecta, así como la inferencia que determina sobre cada uno de estos. En este caso, se ha propuesto la aplicación de eliminación de ruido *Gaussiano* con el objetivo de determinar si existe una posible mejora. Para realizar las pruebas pertinentes, se ha seleccionado dos de los vídeos previamente etiquetados desarrollados en la sección número 5.1. Para ello, se va a aplicar el proceso de super-resolución junto a la disminución de ruido para determinar si verdaderamente tiene implicación en el resultado final obtenido.



Figura 14: Ejemplo aplicado al vídeo 2 del *dataset* definido con el objetivo de determinar la implicación en el procesamiento de reducción de ruido de la imagen inicial. A la izquierda se muestra la imagen sin la aplicación de la disminución de ruido mientras que la derecha corresponde a la salida obtenida tras ser aplicado.

Tras aplicar el proceso de disminución de ruido por ejemplo al vídeo 2 y 3, obtenemos una mejoría notable, pues en un principio, se detectan multitud de elementos sin embargo la inferencia de clase no es correcta, categorizando por ello a coches como camiones. Al aplicar la disminución de ruido vemos como ahora si las categorías si corresponden con los elementos detectados en las Figuras 14 y 15.



Figura 15: Ejemplo aplicado al vídeo 3 del *dataset* definido con el objetivo de determinar la implicación en el procesamiento de reducción de ruido de la imagen inicial. A la izquierda se muestra la imagen sin la aplicación de la disminución de ruido mientras que la derecha corresponde a la salida obtenida tras ser aplicado.

Por ello, podemos determinar por tanto que la aplicación de este tipo de procesos depende directamente del tipo de imagen que se aplique a la red neuronal convolucional como entrada. A lo largo de las pruebas realizadas en la sección 5.3 se mostrarán la mejora obtenida aplicando en cada uno de los vídeos captados por diversos sistemas de videovigilancia procesos de disminución de ruido con el objetivo de obtener los mejores resultados en cada caso.

5.3. Resultados cualitativos y cuantitativos en la fase off-line:

Tras determinar el modelo de super-resolución seleccionado para la realización de las pruebas, siendo en este caso *FSRCNN* y determinar que la aplicación de procesamiento de imágenes basado en la disminución de ruido en una imagen mejora los resultados obtenidos, se ha procedido a realizar las pruebas sobre el conjunto de datos desarrollado en el punto 5.1. En primer lugar, es necesario establecer una serie de parámetros de la red como son el número máximo de objetos detectados, la inferencia mínima a tener en cuenta de dicho objeto, así como el umbral θ para el índice de intersección sobre la unión (*IoU*) con el fin de eliminar detecciones simultáneas para un mismo objeto. Estos parámetros se establecen en la tabla 3 y han sido seleccionados según los mejores resultados obtenidos tras la aplicación de una serie de pruebas.

Parámetro	Valor
Máximo número de detecciones	100
Mínimo porcentaje de inferencia	0.3
Umbral de IoU θ	0.1

Tabla 3: Tabla con los hiperparámetros seleccionados para la aplicación de la técnica descrita.

A continuación, se muestran una serie de ejemplos en las figuras 16,17 y 18 en las cuales se mejora el resultado proporcionado con la técnica comentada en este trabajo sobre el conjunto de datos desarrollado. Para comprobar los resultados obtenidos por la técnica propuesta, además de sobre las imágenes correspondientes a los vídeos seleccionados, se han probado sobre una serie de fotogramas de los cuales se disponía el *Ground truth* para verificar la eficacia de la técnica propuesta tal y como se puede ver en las figuras 19, 20 , 21 y 22.

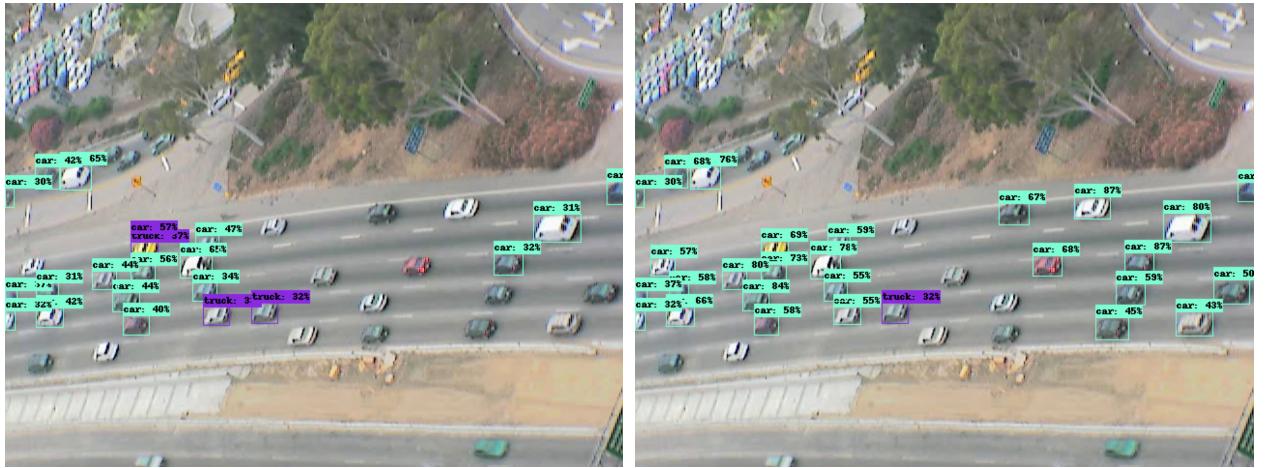


Figura 16: Ejemplo aplicado al *frame* 1 del vídeo denotado como *sb-camera1-0820am-0835am*. El lado izquierdo muestra los resultados obtenidos por el modelo sin modificar mientras que el lado derecho muestra la precisión de las detecciones tras aplicar la técnica desarrollada en este trabajo usando el modelo base CenterNetHourGlass104Keypoints.

Para obtener resultados cuantitativos de rendimiento, se ha utilizado el evaluador desarrollado por COCO. El cuadro delimitador de una predicción se considera una detección correcta en función del solapamiento con el cuadro delimitador del *ground truth*, medido por la intersección sobre la unión conocida como *IoU*. Además, es necesario que la inferencia de clase sea correcta para dicho elemento. Como salida, se obtienen una serie de precisiones medidas en función del tamaño de los objetos, así como el porcentaje de inferencia de los elementos detectados. Según las pruebas realizadas, el uso de esta técnica resuelve una serie de problemas atendiendo al contexto de la imagen. En primer lugar, detecta un mayor número de elementos que no fueron detectados a priori por el modelo en la primera pasada. En segundo lugar, en la inferencia inicial podemos observar

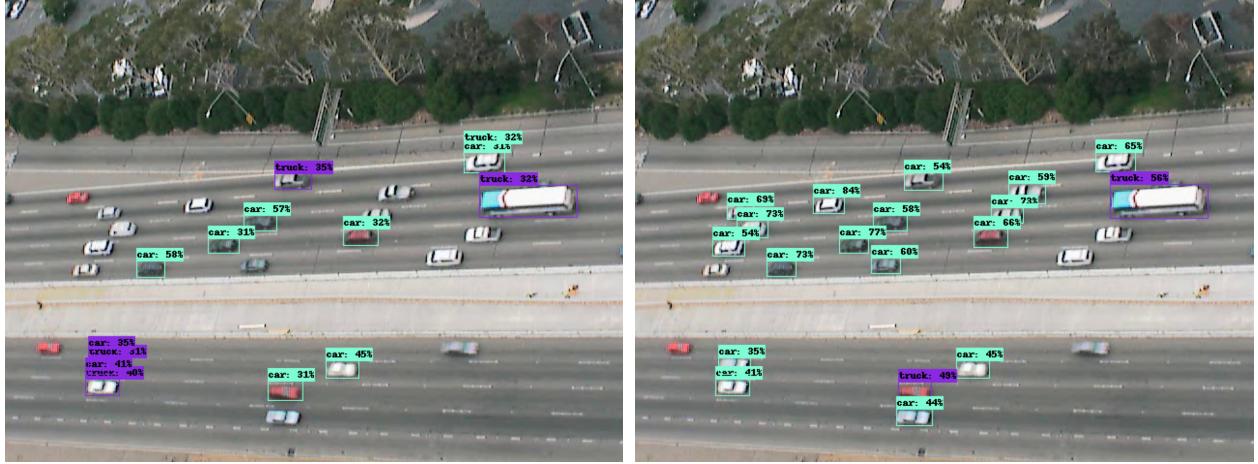


Figura 17: Ejemplo aplicado al *frame* 6 del vídeo denotado como *sb-camera3-0820am-0835am*. El lado izquierdo muestra los resultados obtenidos por el modelo sin modificar mientras que el lado derecho muestra la precisión de las detecciones tras aplicar la técnica desarrollada en este trabajo usando el modelo base *CenterNetHourGlass104Keypoints*.

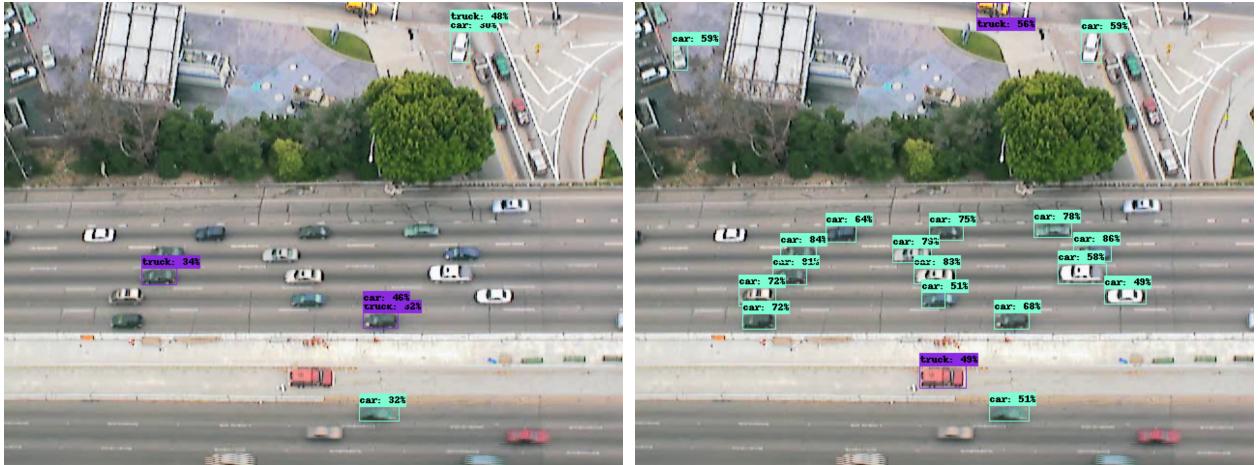


Figura 18: Ejemplo aplicado al *frame* 136 del vídeo denotado como *sb-camera4-0820am-0835am*. El lado izquierdo muestra los resultados obtenidos por el modelo sin modificar mientras que el lado derecho muestra la precisión de las detecciones tras aplicar la técnica desarrollada en este trabajo usando el modelo base *CenterNetHourGlass104Keypoints*.

en determinadas zonas que la clase de los elementos no es correcta y tras aplicar esta técnica, este problema queda resuelto.

Sin embargo, hay que tener en cuenta que esta solución no es infalible y dependerá del contexto al que se aplique, ya que por ejemplo en la figura 22 podemos ver como se aumenta el número de elementos, sin embargo, también obtenemos falsos positivos como el del centro de la imagen o la



Figura 19: Ejemplo aplicado al *frame* 5837 del dataset denominado como *M-30-HD* [48]. El lado izquierdo muestra los resultados obtenidos por el modelo sin modificar mientras que el lado derecho muestra la precisión de las detecciones tras aplicar la técnica desarrollada en este trabajo usando el modelo base *CenterNetHourGlass104Keypoints*.

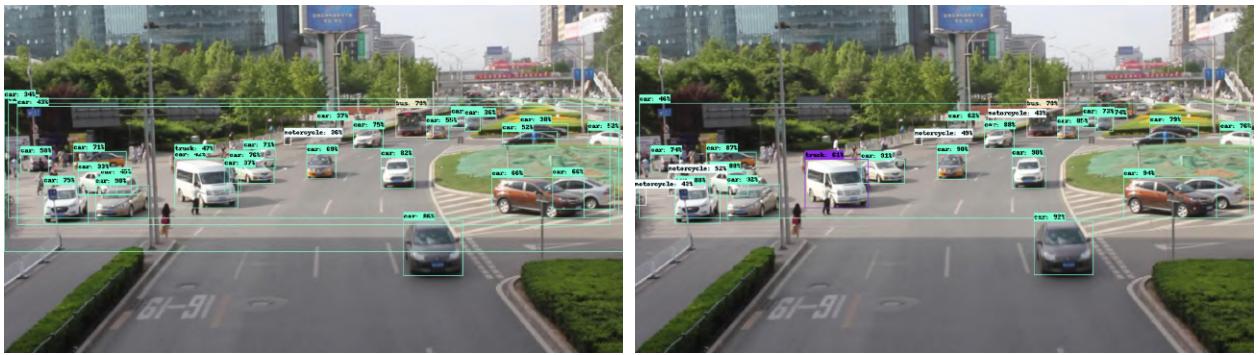


Figura 20: Ejemplo aplicado al *frame* 2 del dataset denominado como *UA-DETRAC MVI_39311* [49, 50, 51]. El lado izquierdo muestra los resultados obtenidos por el modelo sin modificar mientras que el lado derecho muestra la precisión de las detecciones tras aplicar la técnica desarrollada en este trabajo usando el modelo base *CenterNetHourGlass104Keypoints*

inferencia correcta de la moto situada en la parte central izquierda. Este problema se puede paliar aumentando el porcentaje mínimo de inferencia para tener en cuenta la detección.

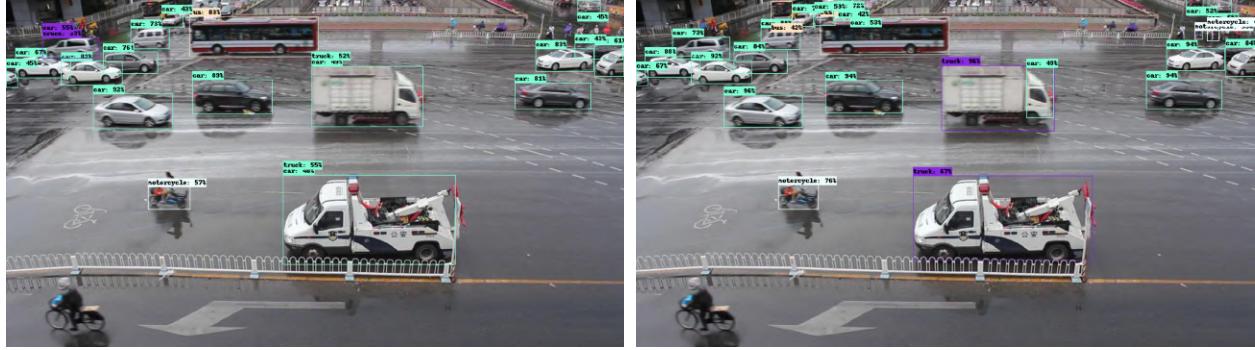


Figura 21: Ejemplo aplicado al *frame* 1 del dataset denominado como *UA-DETRAC MVI_40903* [49, 50, 51]. El lado izquierdo muestra los resultados obtenidos por el modelo sin modificar mientras que el lado derecho muestra la precisión de las detecciones tras aplicar la técnica desarrollada en este trabajo usando el modelo base *CenterNetHourGlass104Keypoints*

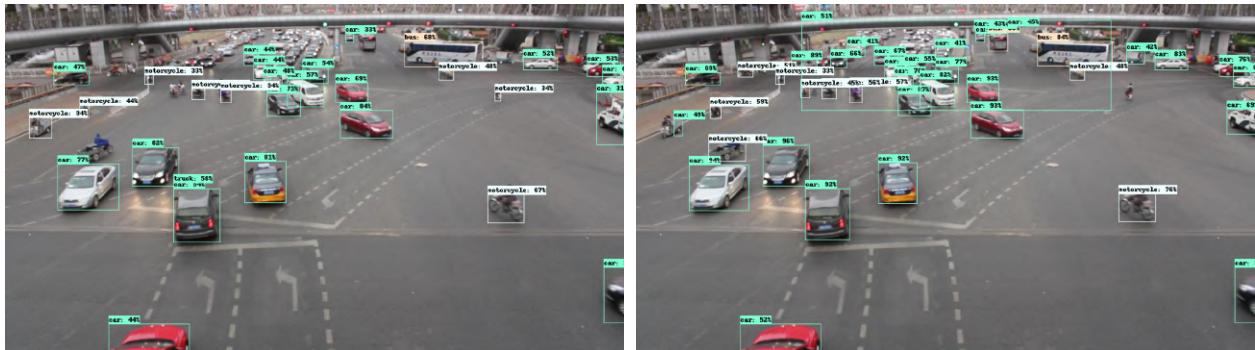


Figura 22: Ejemplo aplicado al *frame* 5 del dataset denominado como *UA-DETRAC MVI_40852* [49, 50, 51]. El lado izquierdo muestra los resultados obtenidos por el modelo sin modificar mientras que el lado derecho muestra la precisión de las detecciones tras aplicar la técnica desarrollada en este trabajo usando el modelo base *CenterNetHourGlass104Keypoints*

A continuación, se compara el rendimiento de las detecciones obtenidas por el modelo inicial y las obtenidas tras aplicar la técnica desarrollada a lo largo de dicho trabajo con el objetivo de medir el rendimiento de la fase *Offline* en la detección de coches en el conjunto de datos desarrollado manualmente con imágenes de carreteras transitadas por una serie de vehículos. Para ello, se ha establecido únicamente la clase número 3, correspondiente en *COCO* con el elemento coche, obteniendo así las tablas 4, 5 y 6. En cada una de estas tablas se determinan las métricas obtenidas por *COCO* tanto para las detecciones iniciales como para los resultados obtenidos tras aplicar la técnica descrita. Las tres primeras columnas corresponden a precisión general obtenida para las detecciones de elementos de cualquier tamaño con un umbral de precisión del 0.50-0.95, mayor que 0.50 y mayor que 0.75. Las columnas cuatro y cinco en cambio están enfocadas directamente en la precisión obtenida para elementos de tamaño reducido.

Video 1 - sb-camera1-0820am-0835am					
Modelo	0.50:0.95—all	>0.50—all	>0.75—all	0.50:0.95—Small	>0.50—Medium
CenterNet HourGlass104 Kp	0.244 / 0.361	0.420 / 0.628	0.263 / 0.381	0.258 / 0.375	0.193 / 0.283
CenterNet MobileNetV2 FPN Kp	0.083 / 0.161	0.191 / 0.358	0.053 / 0.121	0.085 / 0.163	0.081 / 0.122
CenterNet Resnet101 V1 FPN	0.273 / 0.350	0.512 / 0.656	0.248 / 0.318	0.274 / 0.355	0.244 / 0.273
EfficientDet D3	0.159 / 0.336	0.276 / 0.585	0.169 / 0.364	0.159 / 0.343	0.154 / 0.247
EfficientDet D4	0.236 / 0.442	0.424 / 0.790	0.237 / 0.459	0.243 / 0.451	0.168 / 0.298
EfficientDet D5	0.245 / 0.378	0.433 / 0.672	0.252 / 0.389	0.251 / 0.395	0.158 / 0.164

Tabla 4: Resultados obtenidos para el primer vídeo. En la parte izquierda se muestra la precisión obtenida por el modelo base y en la parte derecha la precisión obtenida por la técnica descrita en este trabajo. Cuanto mayor es la precisión, mejor se considera la técnica. Los mejores resultados están marcados en **negrita**.

Video 2 - sb-camera3-0820am-0835am					
Modelo	0.50:0.95—all	>0.50—all	>0.75—area—all	0.50:0.95—Small	>0.50—Medium
CenterNet HourGlass104 Kp	0.090 / 0.186	0.176 / 0.363	0.074 / 0.154	0.092 / 0.191	0.043 / 0.112
CenterNet MobileNetV2 FPN Kp	0.117 / 0.158	0.309 / 0.390	0.047 / 0.083	0.119 / 0.159	0.039 / 0.085
CenterNet Resnet101 V1 FPN	0.111 / 0.188	0.242 / 0.403	0.072 / 0.124	0.111 / 0.191	0.096 / 0.093
EfficientDet D3	0.092 / 0.294	0.177 / 0.590	0.073 / 0.224	0.091 / 0.297	0.105 / 0.171
EfficientDet D4	0.236 / 0.345	0.480 / 0.689	0.171 / 0.261	0.237 / 0.350	0.232 / 0.244
EfficientDet D5	0.178 / 0.292	0.345 / 0.588	0.145 / 0.219	0.178 / 0.295	0.207 / 0.221

Tabla 5: Resultados obtenidos para el segundo vídeo. En la parte izquierda se muestra la precisión obtenida por el modelo base y en la parte derecha la precisión obtenida por la técnica descrita en este trabajo. Cuanto mayor es la precisión, mejor se considera la técnica. Los mejores resultados están marcados en **negrita**.

Se han omitido las métricas obtenidas para objetos de gran tamaño, ya que no existen muestras en el conjunto de datos establecido. Como se puede observar en cada una de las tablas establecidas para

Video 3 - sb-camera4-0820am-0835am						
Modelo	0.50:0.95—all	>0.50—all	>0.75—area=all	0.50:0.95—Small	>0.50—Medium	
CenterNet HourGlass104 Kp	0.074 / 0.219	0.137 / 0.394	0.072 / 0.217	0.075 / 0.227	0.054 / 0.060	
CenterNet MobileNetV2 FPN Kp	0.060 / 0.102	0.159 / 0.237	0.026 / 0.064	0.066 / 0.108	0.009 / 0.014	
CenterNet Resnet101 V1 FPN	0.039 / 0.082	0.075 / 0.160	0.028 / 0.065	0.040 / 0.084	0.016 / 0.015	
EfficientDet D3	0.029 / 0.163	0.050 / 0.292	0.022 / 0.164	0.029 / 0.168	0.012 / 0.040	
EfficientDet D4	0.129 / 0.280	0.236 / 0.518	0.118 / 0.264	0.130 / 0.287	0.068 / 0.151	
EfficientDet D5	0.056 / 0.204	0.099 / 0.384	0.051 / 0.184	0.058 / 0.207	0.006 / 0.082	

Tabla 6: Resultados obtenidos para el tercer video. En la parte izquierda se muestra la precisión obtenida por el modelo base y en la parte derecha la precisión obtenida por la técnica descrita en este trabajo. Cuanto mayor es la precisión, mejor se considera la técnica. Los mejores resultados están marcados en **negrita**.

las imágenes que componen el conjunto de datos definido, existe una clara mejora en la precisión obtenida por la métrica *COCO*⁶. Para ilustrar mejor el aumento en el número de detecciones, se ha seleccionado para esta prueba el modelo *CenterNet Hourglass 104 Keypoints*, ya que es uno de los modelos con mejor rendimiento en comparación con el resto de modelos utilizados para esta comparativa en las Figuras 23, 24 y 25.

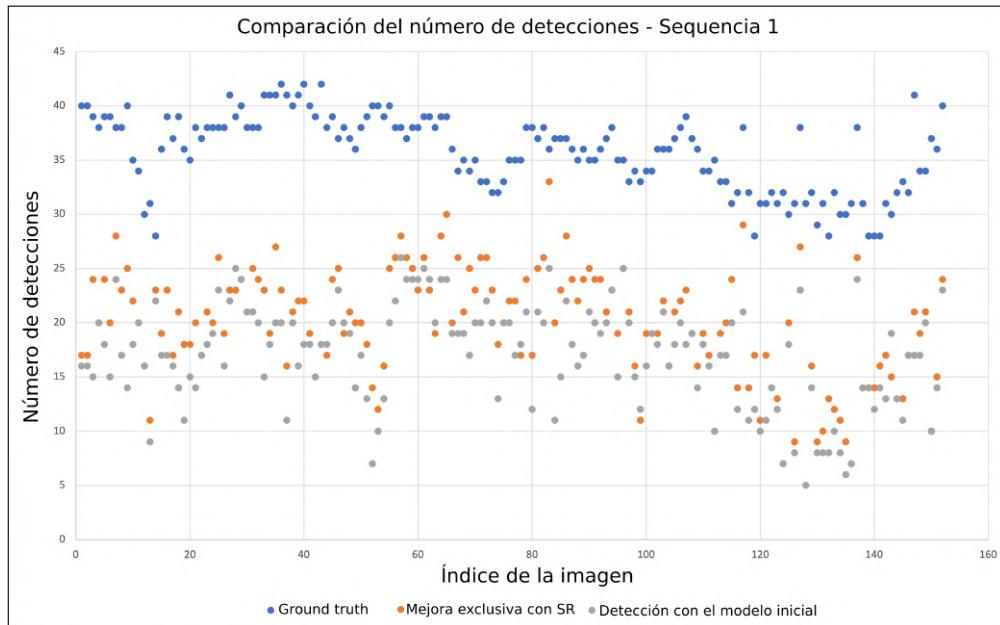


Figura 23: Comparación del número de detecciones para la primera secuencia.

⁶ <https://github.com/cocodataset/cocoapi/blob/master/PythonAPI/pycocotools/cocoeval.py>

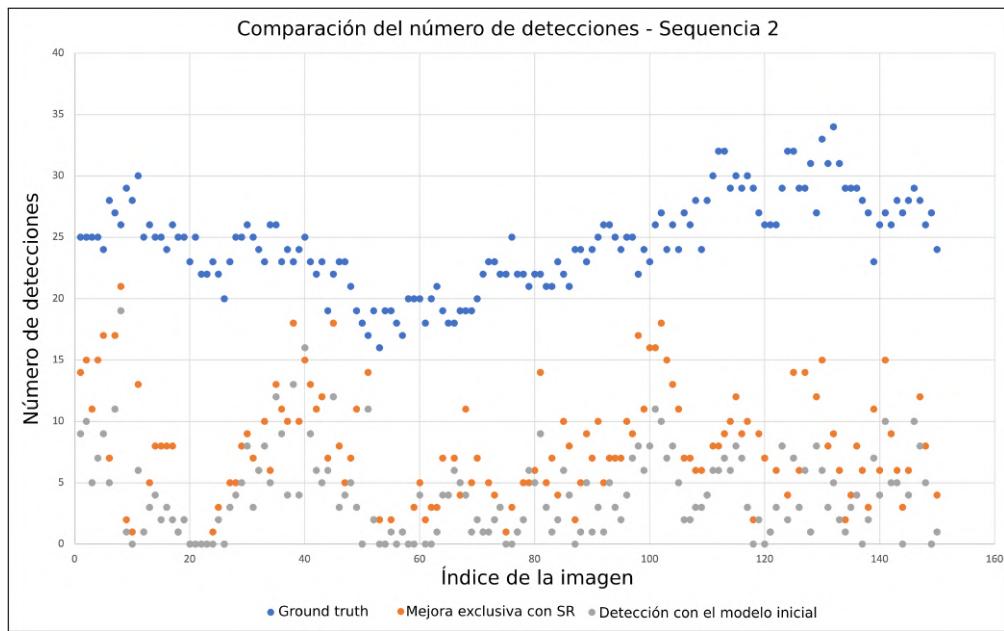


Figura 24: Comparación del número de detecciones para la segunda secuencia.

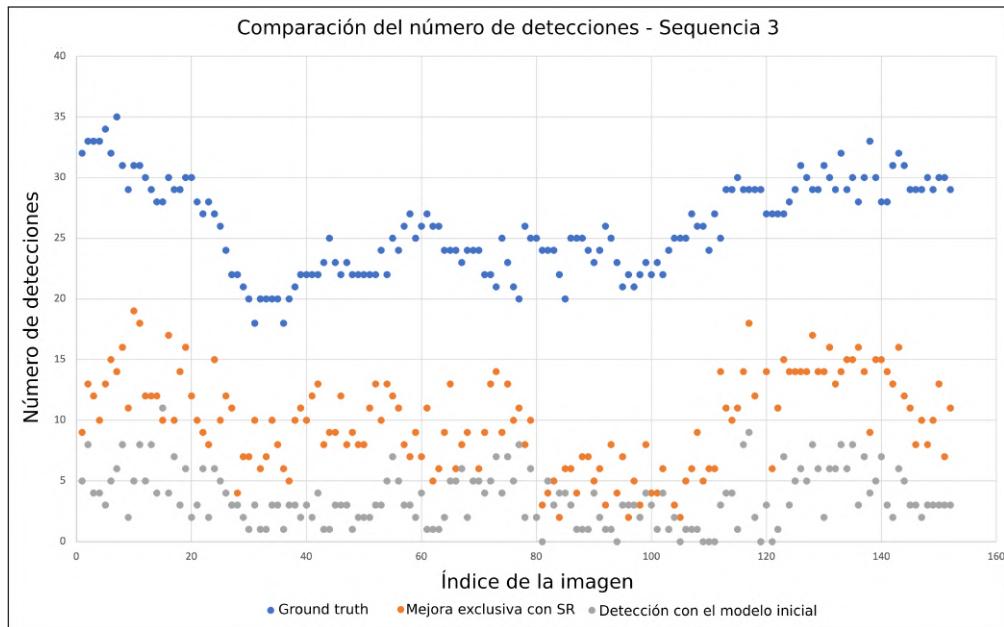


Figura 25: Comparación del número de detecciones para la tercera secuencia.

5.4. Resultados cualitativos y cuantitativos en la fase on-line:

Tal y como se ha podido demostrar en vista de los resultados descritos en la fase 5.3, es posible determinar que tanto el número de elementos detectados como la inferencia mejora, sin embargo es vital desarrollar soluciones enfocadas a detecciones en tiempo real para el ámbito de la red de transportes. La aplicación de super-resolución para el posterior procesamiento de las diversas sub-imágenes generadas incrementa considerablemente el tiempo de procesamiento requerido tal y como se muestra en la sección 5.5, es por ello por lo que con las salidas obtenidas de esta técnica, se realiza el *Fine-Tuning* del modelo de forma totalmente automática con el fin de que este aprenda de los nuevos objetos no detectados a priori y que en una sola pasada detecte a los mismos de manera rápida y eficaz.

Con el fin de determinar la efectividad de la técnica propuesta tras realizar el *Fine Tuning* del modelo al llevar a cabo el entrenamiento de forma totalmente automática, se ha realizado una comparación entre los siguientes métodos:

- Modelo Original: El modelo de detección de objetos sin modificar.
- Aplicación exclusiva de SR: La aplicación del meta-modelo haciendo uso de SR.
- Fine-Tuning exclusivo: Re-entrenamiento del modelo original con las salidas provistas por el mismo.
- Técnica propuesta: La técnica propuesta está basada en el re-entrenamiento del modelo, haciendo uso de las salidas provistas por la mejora tras aplicar SR.

A partir de las secuencias de vídeo seleccionadas (Subsección 5.1), a la hora de realizar el ajuste fino del modelo pre-entrenado se ha restringido el número de clases de las 90 clases iniciales del conjunto de datos COCO [40] a 4 (coche, camión, moto y autobús) ya que son las clases más probables en las escenas de tráfico. Para llevar a cabo la fase de experimentación, se seleccionó la clase coche ya que el número de vehículos correspondientes a esta clase era mayor en las secuencias de vídeo seleccionadas.

Como medida para comparar la eficacia de la propuesta presentada desde un punto de vista cuantitativo, se ha establecido el factor de aceleración obtenido por cada una de las técnicas establecidas en comparación con el modelo original en la Tabla 7. Además, se ha determinado el número de elementos que componen las secuencias seleccionadas, así como el número medio de detecciones por fotograma y la inferencia de clase obtenida para las mismas. Este proceso se ha realizado para las tres secuencias de vídeo elegidas. Esta comparación se realiza en la Tabla 8 de la página 45. Las pruebas se han realizado con una Nvidia Geforce GTX Titan X.

Técnica	Factor de Aceleración
Modelo Original	$1x$
Aplicación exclusiva SR	$42,4x$
Fine-Tuning exclusivo	$1x$
Técnica propuesta	$1x$

Tabla 7: Factor de aceleración de las técnicas presentadas basadas en el modelo original para la secuencia 1.

		EfficientNet D4 - Clase Coche (ID:3)		
		Número total de elementos	Media de detecciones	Media puntuación de clase
Seq. 1	Ground Truth	6261	-	-
	Modelo Original	2672	15.096 ± 3.016	0.354 ± 0.013
	Aplicación exclusiva SR	5210	29.435 ± 3.117	0.502 ± 0.043
	Fine-Tuning exclusivo	3757	21.226 ± 3.707	0.531 ± 0.032
	Técnica propuesta	6073	34.311 ± 2.808	0.703 ± 0.022
Seq. 2	Ground Truth	3680	-	-
	Modelo Original	1815	12.1 ± 2.949	0.381 ± 0.02
	Aplicación exclusiva SR	2709	18.181 ± 4.611	0.468 ± 0.046
	Fine-Tuning exclusivo	1867	12.447 ± 2.851	0.472 ± 0.034
	Técnica propuesta	3564	23.76 ± 3.885	0.621 ± 0.036
Seq. 3	Ground Truth	3947	-	-
	Modelo Original	943	6.245 ± 2.551	0.359 ± 0.023
	Aplicación exclusiva SR	2160	14.211 ± 4.16	0.491 ± 0.07
	Fine-Tuning exclusivo	1052	6.921 ± 2.531	0.429 ± 0.041
	Técnica propuesta	2811	18.493 ± 3.12	0.493 ± 0.027

Tabla 8: Número total de elementos en la secuencia, media de las detecciones, media de la puntuación de clase y desviación estándar por cada imagen procesada por las diversas técnicas propuestas usando el modelo *EfficientNet D4*.

Finalmente, se ha considerado la evaluación de COCO⁷ para determinar la mAP (Average Precision) de las detecciones obtenidas por cada secuencia de vídeo. Comparando con ella la precisión obtenida para elementos de cualquier tamaño, elementos de tamaño reducido y tamaño medio, respectivamente. La tabla 9 de la página 46 muestra una comparación de la precisión media (mAP) obtenida para las secuencias 1, 2 y 3. Para cada una de ellas se muestran los resultados obtenidos por el modelo tras realizar una detección en bruto, la precisión obtenida con la propuesta de SR estableciendo múltiples pasos para el mismo fotograma, el modelo tras realizar el ajuste fino con las salidas proporcionadas por el modelo en bruto, y finalmente el modelo tras realizar el ajuste fino automático con las salidas proporcionadas por la propuesta de SR.

⁷ <https://github.com/cocodataset/cocoapi/blob/master/PythonAPI/pycocotools/cocoeval.py>

		Mean Average Precision (mAP) - EfficientNet D4				
		IoU=0.50:0.95 area=all	IoU>0.50 area=all	IoU>0.75 area=all	IoU=0.50:0.95 area=Small	IoU>0.50 area=Medium
Seq. 1	Modelo Original	0.236	0.424	0.237	0.243	0.168
	Aplicación exclusiva SR	0.442	0.790	0.459	0.451	0.298
	Fine-Tuning exclusivo	0.320	0.592	0.294	0.330	0.152
	Técnica propuesta	0.489	0.915	0.440	0.497	0.360
Seq. 2	Modelo Original	0.236	0.480	0.171	0.237	0.232
	Aplicación exclusiva SR	0.345	0.689	0.261	0.350	0.244
	Fine-Tuning exclusivo	0.224	0.503	0.126	0.224	0.213
	Técnica propuesta	0.399	0.920	0.202	0.401	0.314
Seq. 3	Modelo Original	0.129	0.236	0.118	0.130	0.068
	Aplicación exclusiva SR	0.280	0.518	0.264	0.287	0.151
	Fine-Tuning exclusivo	0.137	0.266	0.108	0.140	0.044
	Técnica propuesta	0.339	0.678	0.251	0.343	0.179

Tabla 9: Mean average precision (mAP) para las tres secuencias probadas. (Cuanto mayores sean los resultados mejoran serán los mismos). Los mejores resultados están marcados en **negrita**. El *Fine-Tuning* se ha llevado a cabo con las salidas provistas por el modelo *EfficientNet D4*.

Según la Tabla 8, podemos afirmar que nuestra propuesta detecta un mayor número de elementos, siendo más del doble que el modelo original. Como se puede observar, el número de detecciones medias establecidas para cada fotograma es mayor. Por ejemplo, para la secuencia uno, hemos obtenido una media de 34,311 vehículos, frente a los 15,096 detectados por el modelo preentrenado.

Con la propuesta establecida, la puntuación de clase es mucho mayor, obteniendo una fiabilidad media del 70 % en los elementos detectados para la secuencia uno. Este hecho se repite en el resto de secuencias en las que se ha realizado la fase de experimentación.

Otro aspecto a tener en cuenta es el tiempo de inferencia requerido por cada una de las técnicas comparadas. En la Tabla 7, como se puede observar, la propuesta inicial basada en la aplicación exclusiva de la super-resolución no es válida en escenarios en los que se requiere la detección de elementos en tiempo real. La aplicación de la SR, al requerir múltiples detecciones en las distintas subimágenes generadas, incrementa considerablemente el tiempo requerido. A partir de las pruebas realizadas sobre la secuencia uno, se puede determinar que se requiere el procesamiento de una media de 43 imágenes por cada fotograma que compone el vídeo. Cabe destacar que la propuesta presentada no requiere más tiempo de procesamiento.

Aplicando el evaluador COCO, se obtiene la tabla 9. Podemos determinar que para elementos de cualquier tamaño, la métrica obtenida por la propuesta supera al resto. Efectivamente, hay campos en los que la aplicación exclusiva de la super-resolución aporta mejores resultados. Sin embargo, enlazando con el tiempo mencionado anteriormente, éste se multiplica en función del número de elementos detectados inicialmente. Al mismo tiempo, la propuesta presentada no aumenta el tiempo necesario para aplicar el modelo de detección de objetos. Según la relación entre el número de objetos detectados y el GT (Ground-truth) establecida en la Tabla 8, estos resultados están relacionados con el índice obtenido en la columna IOU (Intersection Over Union)>0,50 de la Tabla número 9. Esto se debe principalmente al pequeño tamaño de los vehículos, que representan aproximadamente el 1 % de toda la imagen. Utilizando la secuencia número dos como ejemplo, podemos determinar que el número de elementos detectados por nuestra propuesta se acerca al número de elementos definidos por el GT. Por ello, pasamos de un 48 % de precisión por el modelo bruto a un 92 % obtenido tras aplicar la propuesta bajo el índice IoU>0,50. En el artículo, [52] se establece cómo se realiza la

evaluación en términos de métricas de rendimiento y precisión. Ejemplos cualitativos de uno de los fotogramas de las secuencias 1, 2 y 3, utilizados para realizar la fase experimental, se muestran en las Figuras 26, 27 y 28 en las que se muestran los resultados obtenidos con la técnica propuesta para algunas instancias del conjunto de datos considerado. El número de elementos detectados en la imagen es mayor. Además, cabe destacar que la inferencia de clase establecida para cada uno de estos elementos también aumenta.



Figura 26: Ejemplo aplicado al fotograma 33 del primer vídeo denotado como sb-camera1-0820am-0835am. La parte izquierda muestra los resultados obtenidos por el ajuste fino con el modelo sin procesar, mientras que la parte derecha muestra las detecciones tras aplicar el ajuste fino con los resultados de la técnica descrita utilizando la secuencia uno y el modelo EfficientNet D4.



Figura 27: Ejemplo aplicado al fotograma 26 del segundo vídeo denotado como sb-camera3-0820am-0835am. La parte izquierda muestra los resultados obtenidos por el ajuste fino con el modelo sin procesar, mientras que la parte derecha muestra las detecciones tras aplicar el ajuste fino con los resultados de la técnica descrita utilizando la secuencia dos y el modelo EfficientNet D4.



Figura 28: Ejemplo aplicado al fotograma 115 del tercer vídeo denotado como sb-camera4-0820am-0835am. La parte izquierda muestra los resultados obtenidos por el ajuste fino con el modelo sin procesar, mientras que la parte derecha muestra las detecciones tras aplicar el ajuste fino con los resultados de la técnica descrita utilizando la secuencia tres y el modelo EfficientNet D4.

En base a los resultados presentados, podemos afirmar que la aplicación de la técnica descrita en este trabajo no solo mejora la precisión de los elementos inicialmente detectados por el modelo, sino que además detecta objetos que a priori no estaban identificados sin aumentar el tiempo requerido para llevar a cabo la detección de elementos. Por ello, podemos corroborar que en todos y cada uno de los fotogramas que componen los vídeos de prueba hemos obtenido un mayor número de elementos. A través del re-entrenamiento con las salidas automáticas provistas por la aplicación de SR, se ha mejorado por tanto las detecciones provistas por sistemas de videovigilancia detectando la mayoría de vehículos.

5.5. Comparativa del tiempo requerido en realizar las detecciones:

A lo largo de este punto, tomaremos como ejemplo uno de los vídeos que componen el *dataset* propuesto en el punto 5.1 con el fin de realizar una comparativa entre el tiempo requerido para el procesamiento de las imágenes.

Comenzando por el tiempo de procesamiento, para el vídeo 1 del *dataset* desarrollado, se ha recopilado los tiempos obtenidos por el modelo *EfficientDet D4*. En la gráfica 29 se muestra una comparativa de los tiempos requeridos para cada una de las propuestas definidas en este trabajo, mientras que en la Tabla 10 se representa el número medio de pasadas requeridas por cada técnica, así como la media y desviación estándar en segundos.

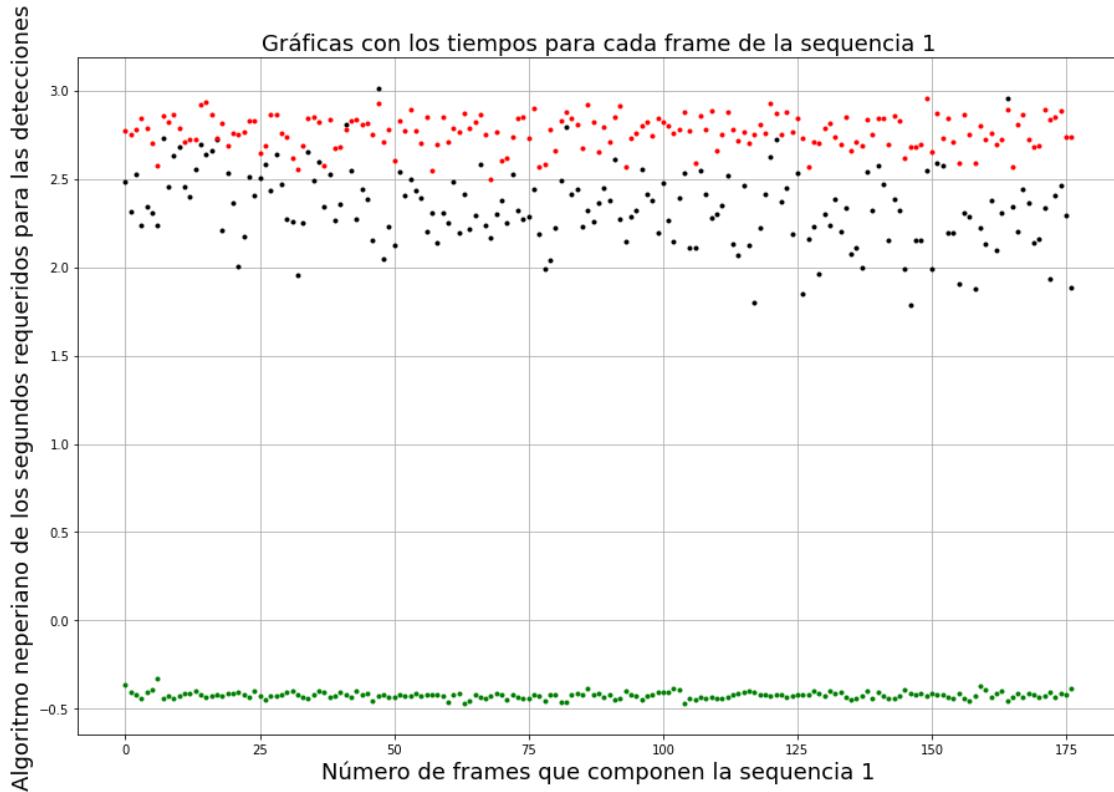


Figura 29: Gráfico mostrando el logaritmo neperiano de los tiempos obtenidos para cada detección. Los tiempos obtenidos por la propuesta de aplicación exclusiva de SR se ha representado en rojo, el algoritmo de *Bron-Kerbosch* en negro mientras que la propuesta de SR + fine-tuning en verde.

Técnica	Número medio de pasadas requeridas	Media	Desviación estándar
Múlt. pasadas por cada detección - 4.4	43	15.93	± 1.44
Algoritmo de <i>Bron-Kerbosch</i> - 4.5	25	10.55	± 2.29
Fine Tuning del modelo con la propuesta 4.6	1	0.66	± 0.03

Tabla 10: Número medio de pasadas, media y desviación estándar definida en segundos para cada uno de las propuestas desarrolladas con el fin de detectar nuevos elementos y mejorar su inferencia. En negro se resaltan los mejores tiempos obtenidos.

Capítulo 6 - Conclusiones y líneas futuras:

En este trabajo se propone una metodología para adaptar automáticamente un método de detección de objetos a una escena específica con objetos pequeños sin intervención humana. En primer lugar se aplica una estrategia de super-resolución *Offline* (denominada SR Enhancement) para encontrar ejemplos de detección de la escena aunque el método de detección de objetos no sea capaz de detectarlos a partir de la imagen original. A continuación, los nuevos ejemplos se utilizan como datos de entrenamiento para llevar a cabo un proceso de ajuste fino a partir de un método de detección de objetos preentrenado de propósito general para mejorar su rendimiento en esa escena específica.

Para probar la propuesta, se han realizado experimentos con tres secuencias de vídeo junto con estudios de ablación para comprobar la utilidad de cada sección del método. Los resultados apoyan nuestra propuesta mostrando una mejora sobresaliente en los valores de Precisión Media cuando se aplica sin un aumento del tiempo de procesamiento asociado. Es importante destacar que esta propuesta es un metamétodo aplicable cuando se utiliza cualquier método de detección de objetos para aumentar su efectividad al tratar con objetos pequeños manteniendo el tiempo de procesamiento por lo que lo consideramos una estrategia recomendable cuando se trata de problemas en tiempo real utilizando cámaras distantes. Recordemos que, en todos los casos, las imágenes que componen el modelo sobre el que se han validado los resultados corresponden a sistemas de videovigilancia situados en puntos altos.

Como trabajo futuro, se podrían realizar estudios con otros métodos de detección de objetos basados en CNN. Dado que el método de recolección de datos inicial basado en la mejora de la SR es Offline, se podrían utilizar técnicas de aumento en tiempo de prueba o un conjunto de métodos de detección de objetos para obtener un mejor conjunto de datos de entrenamiento de ajuste. Otra de las posibles técnicas a realizar sería el establecimiento de un *cluster* de modelos que realizaran la detección inicial sobre la imagen dada como entrada con el objetivo de poder establecer de forma más efectiva y segura los elementos detectados inicialmente por estos modelos pre-entrenados para posteriormente realizar el *Fine-Tuning* sobre aquel que mejores resultados obtenga. Una de las líneas en las cuales ya se está trabajando actualmente se basa en la aplicación del flujo de trabajo presentado aplicado a modelos de segmentación como pueden ser *Mask-RCNN* con el fin de determinar si mejora la detección en cuanto a las máscaras de los elementos se refiere.

Referencias:

- [1] Ross Girshick. “Fast R-CNN”. En: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, págs. 1440-1448. DOI: 10.1109/ICCV.2015.169.
- [2] Wei Liu y col. “SSD: Single Shot MultiBox Detector”. En: *Lecture Notes in Computer Science* (2016), págs. 21-37. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46448-0_2. URL: http://dx.doi.org/10.1007/978-3-319-46448-0_2.
- [3] Joseph Redmon y col. “You Only Look Once: Unified, Real-Time Object Detection”. En: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, págs. 779-788. DOI: 10.1109/CVPR.2016.91.
- [4] Mingxing Tan, Ruoming Pang y Quoc V. Le. “EfficientDet: Scalable and Efficient Object Detection”. En: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, págs. 10778-10787. DOI: 10.1109/CVPR42600.2020.01079.
- [5] Andreas Geiger y col. “Vision meets Robotics: The KITTI Dataset”. En: *International Journal of Robotics Research (IJRR)* (2013).
- [6] Ross Girshick. “Fast R-CNN”. En: *CoRR* abs/1504.08083 (2015).
- [7] Y. Byeon y K. Kwak. “A Performance Comparison of Pedestrian Detection Using Faster RCNN and ACF”. En: *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. Los Alamitos, CA, USA: IEEE Computer Society, jul. de 2017, págs. 858-863. DOI: 10.1109/IIAI-AAI.2017.196. URL: <https://doi.ieee.org/10.1109/IIAI-AAI.2017.196>.
- [8] Wei Liu y col. “SSD: Single Shot MultiBox Detector”. En: *Lecture Notes in Computer Science* (2016), págs. 21-37. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46448-0_2.
- [9] Joseph Redmon y col. “You Only Look Once: Unified, Real-Time Object Detection”. En: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, págs. 779-788. DOI: 10.1109/CVPR.2016.91.
- [10] Youngwan Lee y col. “An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection”. En: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, págs. 752-760.
- [11] Jesús Benito-Picazo y col. “Deep learning-based video surveillance system managed by low cost hardware and panoramic cameras”. En: *Integrated Computer-Aided Engineering 27* (mayo de 2020), págs. 1-15. DOI: 10.3233/ICA-200632.
- [12] Kaiwen Duan y col. *CenterNet: Keypoint Triplets for Object Detection*. 2019. arXiv: 1904.08189 [cs.CV].
- [13] Tao Kong y col. “HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection”. En: *CoRR* abs/1604.00600 (2016).
- [14] Fan Yang, Wongun Choi y Yuanqing Lin. “Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers”. En: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Jun. de 2016, págs. 2129-2137. DOI: 10.1109/CVPR.2016.234.
- [15] Christian Eggert y col. “Improving Small Object Proposals for Company Logo Detection”. En: *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval* (jun. de 2017). DOI: 10.1145/3078971.3078990. URL: <http://dx.doi.org/10.1145/3078971.3078990>.
- [16] Peiyun Hu y Deva Ramanan. “Finding Tiny Faces”. En: *CoRR* abs/1612.04402 (2017).

- [17] X. Zhang y col. “Too Far to See? Not Really!—Pedestrian Detection With Scale-Aware Localization Policy”. En: *IEEE Transactions on Image Processing* 27.8 (2018), págs. 3703-3715. DOI: 10.1109/TIP.2018.2818018.
- [18] R. Chen y col. “Learning Lightweight Pedestrian Detector with Hierarchical Knowledge Distillation”. En: *2019 IEEE International Conference on Image Processing (ICIP)*. 2019, págs. 1645-1649. DOI: 10.1109/ICIP.2019.8803079.
- [19] M.A. Molina-Cabello y col. “Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images”. En: *Integrated Computer-Aided Engineering* 25.4 (2018), págs. 321-333. DOI: 10.3233/ICA-180577.
- [20] R.M. Luque y col. “A neural network approach for video object segmentation in traffic surveillance”. En: *Lecture Notes in Computer Science* 5112 LNCS (2008), págs. 151-158.
- [21] S. Sivaraman y M.M. Trivedi. “Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis”. En: *IEEE Transactions on Intelligent Transportation Systems* 14.4 (2013), págs. 1773-1795. DOI: 10.1109/TITS.2013.2266661.
- [22] W. Hu y col. “A survey on visual surveillance of object motion and behaviors”. En: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 34.3 (2004), págs. 334-352.
- [23] Alex Krizhevsky, Ilya Sutskever y Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. En: *Commun. ACM* 60.6 (mayo de 2017), págs. 84-90. ISSN: 0001-0782. DOI: 10.1145/3065386.
- [24] Karen Simonyan y Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. En: *3rd International Conference on Learning Representations, ICLR 2015*. Ed. por Yoshua Bengio y Yann LeCun. 2015.
- [25] Kaiming He y col. “Deep Residual Learning for Image Recognition”. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Jun. de 2016.
- [26] Christian Szegedy y col. “Going deeper with convolutions”. En: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, págs. 1-9. DOI: 10.1109/CVPR.2015.7298594.
- [27] Gongjin Lan y col. “Real-Time Robot Vision on Low-Performance Computing Hardware”. En: dic. de 2018. DOI: 10.1109/ICARCV.2018.8581288.
- [28] Jorge García-González y col. “Foreground Detection by Probabilistic Mixture Models Using Semantic Information from Deep Networks”. En: *24th European Conference on Artificial Intelligence, ECAI*. Vol. 325. 2020, págs. 2696-2703. DOI: 10.3233/FAIA200408.
- [29] Ross Girshick y col. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Jun. de 2014.
- [30] Ross Girshick. “Fast R-CNN”. En: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dic. de 2015.
- [31] S. Ren y col. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), págs. 1137-1149.
- [32] JooYoun Kim y col. “Vehicle Model Recognition Using SRGAN for Low-Resolution Vehicle Images”. En: AIPR ’19. Beijing, China: Association for Computing Machinery, 2019. ISBN: 9781450372299. DOI: 10.1145/3357254.3357284. URL: <https://doi.org/10.1145/3357254.3357284>.

- [33] Chao Dong y col. “Learning a Deep Convolutional Network for Image Super-Resolution”. En: *Computer Vision – ECCV 2014*. Ed. por David Fleet y col. Cham: Springer International Publishing, 2014, págs. 184-199.
- [34] Jiwon Kim, Jung Kwon Lee y Kyoung Mu Lee. “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. En: *CoRR* abs/1511.04587 (2016).
- [35] Bee Lim y col. “Enhanced Deep Residual Networks for Single Image Super-Resolution”. En: *CoRR* abs/1707.02921 (2017).
- [36] C. Dong y col. “Image Super-Resolution Using Deep Convolutional Networks”. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (2016), págs. 295-307. DOI: 10.1109/TPAMI.2015.2439281.
- [37] Marc Bosch, Christopher M. Gifford y Pedro A. Rodriguez. “Super-Resolution for Overhead Imagery Using DenseNets and Adversarial Learning”. En: *CoRR* abs/1711.10312 (2017).
- [38] Liujuan Cao, C. Wang y J. Li. “Vehicle detection from highway satellite images via transfer learning”. En: *Inf. Sci.* 366 (2016), págs. 177-187.
- [39] Yong Xu, Lin Lin y Deyu Meng. “Learning-Based Sub-Pixel Change Detection Using Coarse Resolution Satellite Imagery”. En: *Remote Sensing* 9 (jul. de 2017), pág. 709. DOI: 10.3390/rs9070709.
- [40] Tsung-Yi Lin y col. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].
- [41] Jakaria Rabbi y col. “Small-Object Detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network”. En: *Remote Sensing* 12.9 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12091432.
- [42] Guimei Cao y col. “Feature-fused SSD: fast detection for small objects”. En: *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*. Vol. 10615. 2018, págs. 381-388. DOI: 10.1117/12.2304811.
- [43] Xue Yang y col. *Rethinking Rotated Object Detection with Gaussian Wasserstein Distance Loss*. 2021. arXiv: 2101.11952 [cs.CV].
- [44] Chao Dong, Chen Change Loy y Xiaoou Tang. “Accelerating the Super-Resolution Convolutional Neural Network”. En: *CoRR* abs/1608.00367 (2016). arXiv: 1608.00367. URL: <http://arxiv.org/abs/1608.00367>.
- [45] Wenzhe Shi y col. “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”. En: *CoRR* abs/1609.05158 (2016). arXiv: 1609.05158. URL: <http://arxiv.org/abs/1609.05158>.
- [46] Wei-Sheng Lai y col. “Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution”. En: *CoRR* abs/1704.03915 (2017). arXiv: 1704.03915. URL: <http://arxiv.org/abs/1704.03915>.
- [47] Antoni Buades, Bartomeu Coll y Jean-Michel Morel. “Non-Local Means Denoising”. En: *Image Processing On Line* 1 (2011), págs. 208-212. DOI: 10.5201/ipol.2011.bcm_nlm.
- [48] R. Guerrero-Gomez-Olmedo y col. “Vehicle Tracking by Simultaneous Detection and Viewpoint Estimation”. En: *IWINAC 2013, Part II, LNCS 7931*. 2013, págs. 306-316.
- [49] Longyin Wen y col. “UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking”. En: *Computer Vision and Image Understanding* 193 (2020), pág. 102907.
- [50] Siwei Lyu y col. “UA-DETRAC 2018: Report of AVSS2018 & IWT4S challenge on advanced traffic monitoring”. En: *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. 2018, págs. 1-6.

- [51] Siwei Lyu y col. “UA-DETRAC 2017: Report of AVSS2017 & IWT4S Challenge on Advanced Traffic Monitoring”. En: *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE. 2017, págs. 1-7.
- [52] Rafael Padilla y col. “A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit”. En: *Electronics* 10.3 (2021). ISSN: 2079-9292. URL: <https://www.mdpi.com/2079-9292/10/3/279>.