ISSN 1870-4069

Construcción de un modelo automático para la detección de armas en el transporte público

Eduardo Enrique Roldán-Juárez, Gabriela Ramírez-de-la-Rosa, Carlos Joel Rivero-Moreno

Universidad Autónoma Metropolitana,
Departamento de Tecnologías de la Información,
México

{gramirez, crivero}@cua.uam.mx,
eerj1311@gmail.com

Resumen. El delito de asaltos a mano armada en el transporte público es un problema común en las grandes ciudades. La detección de estos eventos es importante en la respuesta policiaca. En este artículo se propone una metodología para detectar armas de fuego en imágenes obtenidas de video-vigilancia en el transporte público. La metodología propuesta incorpora: i) la utilización de una arquitectura existente basada en redes neuronales para la construcción del modelo de detección de armas; ii) un proceso de selección de imágenes cercanas al escenario real; y iii) dos opciones de aumento de datos durante la fase de entrenamiento. Cuando se consideró tanto la selección de imágenes como el aumento de dato en la creación del modelo se obtuvo un 81.33 % de mAP. A pesar del esempeño prometedor en este escenario complicado, es necesaria la obtención de imágenes específicas de este dominio.

Palabras clave: Detección de armas, detección de objetos, aumento de datos.

An Automatic Model for Firearms Detection in Public Transport

Abstract. Armed robbery on public transport is a common problem in large cities. Detecting these events promptly is important for quick police response. In this paper we proposed a methodology to detect firearms in images captured from video surveillance in public transport. The proposed methodology include: i) the use of an existing architecture based on neural networks to build a firearms detection's model; ii) a process of selecting images close to the real scenario; and iii) two options for data augmentation during the training phase. When considering both the selection of images and the increase in data for building an automatic firearms detection model we reach 81.33 % of mAP. Despite promising performance in this challenging scenario we believe domain-specific images are required.

Keywords: Firearms detection, object detection, data augmentation.

1. Introducción

En los últimos años, en México se ha experimentado un aumento gradual de violencia, delincuencia e inseguridad. Según el Índice Global de Paz 2020, México se ubica en el lugar 137 de seguridad y es considerado uno de los países más peligrosos del mundo [10]. La percepción de la población no es diferente a los índices globales. El 68.1 % de la población mexicana de 18 años o más considera que, en términos de delincuencia, vivir en su ciudad es inseguro [7].

Entre 2012 y 2019, se estima que en México hubo un total de 17.9 millones de delitos donde la víctima estuvo presente, de los cuales en un 42.9 % de los casos hubo un arma involucrada. Y en el 31 % de los casos, se trató de un arma de fuego [8, 9]. Datos anuales (del 2010 al 2019) del INEGI [9, 7] muestran que el delito asalto en transporte público o en la calle ha sido el delito de mayor incidencia en México.

El Área Metropolitana del Valle de México (que comprende Ciudad de México y municipios de Hidalgo, Tlaxcala y Estado de México) tiene una tasa de incidencia de más del doble que en el resto del país [9] en este delito. Afortunadamente, existen algunas acciones que se han tomado para aminorar los asaltos en transporte púbico o en la calle.

El organismo encargado de llevar a cabo esas acciones en la Ciudad de México es el Centro de Comando, Control, Cómputo, Comunicaciones y Contacto Ciudadano (o C5). El C5 realiza sus funciones de asistencia a la comunidad a través del vídeo monitoreo y de llamadas telefónicas [3].

Adicionalmente, una de las acciones implementadas por el gobierno de la Ciudad de México es la instalación de más de 15,000 cámaras de video-vigilancia con botones de auxilio, y más de 2000 cámaras que operan en el Sistema de Transporte Colectivo Metro.

También, en 2019 comenzó la implementación del proyecto de monitoreo integral y seguridad del transporte público vía GPS en la Ciudad de México [4]. Este programa consiste en instalar GPS, botón de pánico, contador de pasajeros y videocámaras.

Sin embargo, dado que el operador del transporte público es quien debe presionar el botón de pánico para que la cámara de la unidad sea enlazada directamente al C5, se pone en riesgo al operador.

Una posible alternativa a la activación manual de los botones de pánico es un sistema de monitoreo automático que detecte armas en el video capturado en las unidades de transporte e informe al C5 cuando se requiera atención específica. En este proyecto, nos enfocamos en la construcción de un modelo que permita identificar un arma en una imagen capturada por las cámaras de video-vigilancia.

La identificación de objetos (e.g., armas) en imágenes es una de las tareas más estudiadas y complejas en visión computacional [11]. Ayudado por el resurgimiento de las redes neuronales y la capacidad de cómputo actual, la arquitectura que más se ha utilizado recientemente en este problema son las redes neuronales convolucionales (CNN por sus siglas en inglés) [2, 11].

De manera general, el objetivo de la detección de objetos es que dado un conjunto predeterminado de objetos (o clases), éstos se puedan ubicar y etiquetar mediante una caja delimitadora (bounding box).

La literatura existente en el área se enfoca en el diseño o modificación-adaptación de redes neuronales profundas que funcionen en escenario donde existen un número grande de objetos a detectar o que sean más eficientes en la detección de objetos particulares [20, 2, 11, 18].

En este artículo, nos enfocamos en la detección de un tipo de objeto, i.e. arma de fuego. Particularmente, en la identificación de armas de fuego en imágenes obtenidas de videos del transporte público. Este problema trae consigo retos importantes: el sistema deberá ser robusto a i) las condiciones de captura de las imágenes, ii) entornos no controlados, y iii) ruido en las imágenes. Pues si bien existen cámaras dentro del transporte público, estas cámaras no tienen resoluciones altas, el ángulo de la toma no es controlado, puede obstruirse la cámara y generar cambios de contraste, entre otros problemas.

El objetivo principal de este trabajo es construir un modelo automático de detección de armas en imágenes dentro del transporte público. Nos enfocamos en la adecuación de la calidad de los datos de entrenamiento para que éstos datos sean lo más cercano al escenario de interés (armas usadas en asaltos en el transporte público).

Para ello se propone una metodología que incorpora: i) la utilización de una arquitectura existente basada en redes neuronales para la construcción del modelo de detección de armas, específicamente YOLO[15, 1]; ii) un proceso de selección de imágenes cercanas al escenario real; y iii) dos opciones de aumento de datos en el entrenamiento del modelo. La evaluación experimental muestran que la selección cuidadosa de un conjunto de imágenes de armas y el aumento de datos tanto interno como externo obtienen un desempeño por arriba del 81 % de mAP.

2. Trabajo relacionado

Agrupamos el trabajo relacionado en: i) aquellos que realizan modificaciones o adaptaciones de una arquitectura existente para el problema de clasificación de armas en imágenes [13], y ii) aquellos que utilizan aumento de datos como técnica para mejorar el desempeño del modelo generado [16, 6].

En [13] los autores proponen un sistema de detección de armas en videos con el objetivo de disminuir falsos positivos. Una de las contribuciones de este trabajo fue la construcción de conjuntos de datos. Los autores construyeron 5 conjuntos de datos, 3 de los cuales contienen únicamente dos clases: imágenes con armas e imágenes sin armas.

El mejor resultado (cero falsos positivos y una medida F1 de 0.91) lo obtuvieron usando un enfoque de región para buscar los objetos en las imágenes y la arquitectura Faster Region based-CNN (or Faster R-CNN) sobre el conjunto de datos de 3000 imágenes con 2 clases. Nosotros hacemos uso de este mismo corpus de 3000 imágenes (al que en la Sección 3.2 llamamos Corpus 1).

De los sistemas que hacen uso de técnicas de aumento de datos se encuentra, por un lado, el desarrollado por [16] que busca detectar armas cortas en videos de cámaras de seguridad. El autor construyó una base de datos de imágenes compuesta de más de 17 mil imágenes dividida en dos clases: imágenes con armas y sin armas (el conjunto de datos está balanceado). Se hizo un aumento manual de los datos utilizando rotaciones, re-dimensionamiento y volteo de la imagen.

Para el entrenamiento del modelo utilizaron dos arquitecturas de redes neuronales: VGG (Very Deep Convolutional Networks for Large-Scale Image Recognition) [17] y ZF Net [19]. Los mejores resultados los obtuvieron con VGG con una exactitud de 90 % utilizando imágenes en escala de grises para el entrenamiento. El uso de ejemplos generados por técnicas de aumento de datos resultó benéfico considerando el alto número de imágenes utilizado.

Por otro lado, en [6] realizaron un sistema para detectar pistolas en imágenes. Para el entrenamiento del modelo usaron la arquitectura YOLO v2 [15] y SSD (Single Shot Detection) [12]. Adicionalmente, los autores trabajaron con conjuntos de datos existentes (uno de los cuales es el mismo usado en [13]) más un conjunto de datos artificial que consiste en imágenes creadas a partir de técnicas de aumento de datos. El mejor resultado lo obtuvieron usando YOLO v2 sin aumento de datos, llegando a un mAP de 0.71.

Del análisis de estos trabajos previos podemos resaltar que el corpus de 3000 imágenes introducido por [13] y usado en [13, 16] es un buen punto de referencia. Usar este corpus nos permitirá realizar algunas comparaciones directas con estos trabajos. Por otro lado, el uso de técnicas de aumento de datos fue útil para el caso donde hay un número muy grande de imágenes iniciales.

No tanto para el caso donde se usan aproximadamente 3000 imágenes como origen; sin embargo, esta estrategia es prometedora. Finalmente, los dos últimos trabajos hacen uso de la arquitectura YOLO en sus métodos; mientras [16] lo usa como parte del filtrado de las imágenes a analizar, pues se entrena para identificar personas; [6] hace uso directo de YOLO para la identificación de armas, este enfoque también es prometedor.

A diferencia de estas investigaciones previas, nuestra metodología propone una etapa de selección de imágenes (analizar la composición del conjunto de datos para conservar imágenes lo más parecidas al problema a resolver) en combinación con dos tipos de aumento de datos bajo una arquitectura ampliamente usada en la detección de objetos.

3. Metodología propuesta

La metodología propuesta contempla el uso de un corpus que sea útil para el problema de la detección de armas a partir de imágenes obtenidas dentro del transporte público.

Por un lado, utilizamos una arquitectura basada en redes neuronales convolucionales, particularmente YOLO (que se describe en la Sección 3.1) para la construcción del modelo de detección de armas. Por otro lado, nuestra metodología enfatiza la importancia de los datos para entrenar un modelo. Por lo cual es relevante considerar una etapa de selección de imágenes para la generación de un nuevo corpus.

Finalmente, se propone la utilización de dos tipos de aumento de datos para enriquecer el conjunto de imágenes recolectadas en el paso anterior. Los dos tipos de aumento de datos (externo e interno) se describen en la Sección 3.3. La metodología que incorpora todas las etapas listadas se muestra en la Figura 1.

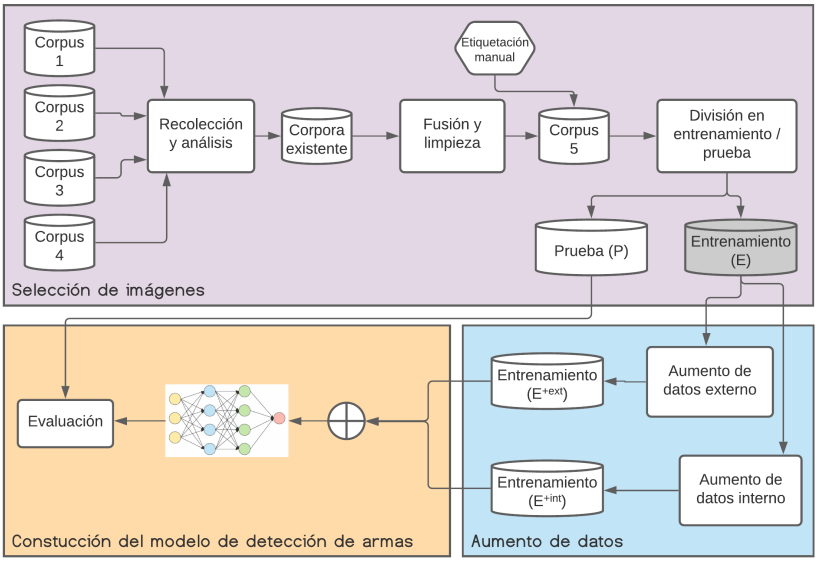

Fig. 1. Esquema general de la metodología propuesta.

3.1. Construcción del modelo de detección de armas

En la Figura 1 (esquina inferior izquierda) se muestran el esquema del proceso de construcción del modelo de detección de armas. De forma general, dado un conjunto de imágenes etiquetadas con armas, se entrena un modelo usando la arquitectura YOLO para posteriormente evaluar el desempeño del modelo usando un conjunto nunca antes visto de imágenes de prueba. A continuación, se describe la arquitectura utilizada.

YOLO: You only look once. YOLO [14] es un sistema de código abierto implementado para hacer detección de objetos en tiempo real. Hace uso de su red neuronal convolucional Darknet, a la que en contraste con otras arquitecturas, se le introducen los ejemplos sólo una vez. Esta característica hace que las detecciones se realicen de manera muy rápida. Además esta arquitectura es capaz de realizar varias detecciones de objetos en una sola imagen.

La red neuronal divide la entrada en regiones y después predice los cuadros delimitadores y probabilidades para cada región. Se optó por utilizar YOLO debido a que en el trabajo de [6], fue este modelo el que dio los mejores resultados, además de que la versión 4 de YOLO se encuentra muy bien calificada dentro del reto del COCO Dataset, siendo el mejor en cuanto velocidad de detección en tiempo real y desempeño [1].

Formato de etiquetado de YOLO. YOLO requiere un archivo de texto por cada elemento del corpus donde se incluirán las etiquetas. La estructura de este archivo contiene x, y, $width$ y $height$ correspondientes a las coordenadas donde se halla el objeto:

```
<clase del objeto>
    <x>,<y>,
    <width>,<height>
```

En la mayoría de casos de asaltos el delincuente lleva el arma en la mano; por lo tanto, en el etiquetado manual que se realiza en nuestra metodología se decidió tomar en cuenta *mano y arma*.

3.2. Selección de imágenes

Dado que no existe un corpus con las características específicas para el problema abordado en este trabajo (detección de armas en el transporte público), la metodología propuesta contempla la recolección y análisis, más la fusión y limpieza de datos. Estos procesos se describen a continuación y se ilustran en el primer renglón de la Figura 1.

Recolección y análisis de corpora existente Los corpora recolectados consisten en imágenes con alguna de las clases de interés: el objeto arma. Estos conjuntos de datos son públicos y han sido usados por la comunidad en trabajos previos.

- Corpus 1[1]. Consiste en 3000 imágenes de armas no etiquetadas. Aproximadamente la mitad son imágenes donde el arma se muestra en tamaños muy grandes, además presentan muchos ángulos repetitivos y sencillos (hacia la izquierda o derecha).
- Corpus 2[2]. Cuenta con 376 imágenes etiquetadas con una única clase (i.e., arma). Este corpus contiene muchos ejemplos de armas largas y armas que se muestran en la totalidad de la imagen. Sólo tiene 160 ejemplos de personas portando armas.
- Corpus 3[3]. Cuenta con 7829 imágenes etiquetadas en tres clases (armas grandes, armas cortas y fuego). Solamente 603 imágenes pertenecen a la clase armas cortas (que son el tipo de armas que nos interesa). De los 603 ejemplos de armas cortas, solo 146 presentan personas portando armas, el resto son solo armas.
- Corpus 4[4]. Cuenta con 333 imágenes no etiquetadas. Este corpus contiene una mayoría de ejemplos de personas con arma (241 imágenes).

Algunos ejemplos de imágenes de estos corpora (particularmente del Corpus 1) se muestra en la Figura 2a. Como puede observarse, los conjuntos de datos descritos son poco aptos para nuestro problema debido a que en el escenario de asaltos en transporte público las armas aparecen en ángulos complicados, el objeto de interés es pequeño y normalmente las imágenes tienen una resolución muy baja (ver Figura 2e para comparación). Ninguna de estas características aparecen en la mayoría de las imágenes incluidas en los corpora descritos.

[1] Obtenido de: https://sci2s.ugr.es/weapons-detection
[2] Obtenido de: https://www.kaggle.com/rogkesavan000/gun-dataset
[3] Obtenido de: https://www.kaggle.com/atulyakumar98/fire-and-gun-dataset
[4] Obtenido de https://www.kaggle.com/issaisasank/guns-objet-detection

(a) Imágenes en los corpora existentes.

(b) Imágenes en el Corpus 5.

(c) Imágenes ejemplo del proceso de aumento de datos externo.

(d) Imágenes ejemplo del proceso de aumento de datos interno (implementadas en YOLO).

(e) Ejemplos de imágenes reales obtenidas de la video vigilancia del transporte público.

Fig. 2. Ejemplos de los imágenes en las diferentes etapas de la metodología presentada.

Fusión y limpieza de corpora Para atender las limitaciones del corpora existente, se optó por eliminar: todas las imágenes donde el arma se presenta en ángulos sencillos o repetitivos, armas mostradas en una gran parte de la imagen (ejemplos de éstas se pueden observar en la Figura 2a).

Se dio prioridad a los ejemplos donde el arma tuviera algún portador (arma en mano) y armas en posiciones no triviales. Como resultado de esta limpieza se formó el *Corpus 5*[5] que contiene un total de 2044 imágenes (algunos ejemplos de imágenes en este corpus se puede observar en la Figura 2b).

[5] El Corpus 5 y el código fuente utilizado en este artículo está disponible en https://github.com/eduardo1311/Detector-de-armas-con-YOLO

Para utilizar el Corpus 5 en la construcción de un modelo de detección de armas es necesario etiquetar las armas presentes en cada imagen.

Por lo tanto, el siguiente paso es hacer el etiquetado manual. Esto es, se ubicó el arma que aparece en la imagen dentro de una caja delimitadora (bounding box). Para este proceso se usó el formato de etiquetado YOLO (ver Sección 3.1). Si el arma está siendo portada, se consideró parte de la mano y el arma en el etiquetado.

3.3. Aumento de datos

Consecuentemente, después del proceso de selección de imágenes, el conjunto resultante de ejemplos es pequeño (2044 en total). Por lo tanto, nuestra metodología propone la utilización de técnicas de aumento de datos. Como puede observarse en los ejemplos de la Figura 2b el conjunto de imágenes resultante en el Corpus 5, aunque son más parecidas al escenario real (asaltos) aún distan de las imágenes obtenidas en el transporte público (ver Figura 2e).

Así, el objetivo principal del aumento de datos es: i) incorporar variaciones a las imágenes tanto en posición como en calidad. Para atender a la variación de la posición se propone el tipo de aumento que denominamos *externo*; para atender a la variación de la calidad de las imágenes, se propone el tipo de aumento que denominamos *interno*.

En esta etapa se genera un conjunto de imágenes que contienen, dependiendo del tipo de aumento de datos utilizado, las imágenes originales más las adicionales (a estos conjuntos les nombramos E^{+ext} y E^{+int}, respectivamente).

Aumento de datos externo. Consta de rotaciones de 45 grados a cada elemento del conjunto de entrenamiento. Se obtienen 7 ejemplos nuevos por cada imagen. Un ejemplo de este tipo de aumento se puede observar en la Figura 2c.

Aumento de datos interno. YOLO implementa módulos de aumento de datos como parte del procesamiento que realiza antes de introducir los ejemplos a la red neuronal. A continuación, se describen las variantes de aumento proporcionado por YOLO que se tomaron en cuenta. Ejemplos de este tipo de aumento se puede observar en la Figura 2d.

- Saturación: Aleatoriamente cambia la saturación de la imagen.
- Exposición: Aleatoriamente cambia el brillo de la imagen.
- Matiz: Aleatoriamente cambia el matiz (color) de la imagen.
- Ruido Gaussiano: Añade ruido Gaussiano a la imagen.
- Flip: Invierte la imagen horizontal o verticalmente.
- Mosaico (Mejora añadida en YOLOv4): Combina varias imágenes en una sola.
- Difuminación (Mejora añadida en YOLOv4): Aplica un desenfoque aleatorio donde se borrará el fondo a excepción del objeto.

Tabla 1. Experimentos propuestos. Se marca con una x cuando se considera el aumento de datos externo o interno según lo indique el nombre de la columna.

Experimento	Aumento externo	Aumento interno
yolov2	-	-
yolov2-int	-	x
yolov2-ext	x	-
yolov2-int-ext	x	x
yolov4	-	-
yolov4-int	-	x
yolov4-ext	x	-
yolov4-int-ext	x	x

4. Evaluación experimental

Para la realización de los experimentos realizamos una división fija de las imágenes en el Corpus 5 de forma que se permita una comparación directa entre los diferentes modelos de detección de armas evaluados.

Así, se obtienen dos subconjuntos: entrenamiento (E) y prueba (P) (como se ilustra en el esquema general del método propuesto en la Figura 1). Para esta división se asignó el 75 % de ejemplos al conjunto de entrenamiento (1537 imágenes) y el 25 % a el conjunto de prueba (507 imágenes).

Se utilizó Colab de Google para hacer el entrenamiento ya que esta plataforma permite tener acceso remoto a equipos con GPU con un límite de 12 horas [5]. Para cada uno de los experimentos, se fijó el tamaño de lote en 64, la tasa de aprendizaje en 0.001, y se realizaron 10000 iteraciones.

Todos los experimentos fueron evaluados con un nivel de confianza de IoU (i.e., la intersección de la unión de las cajas delimitadoras reales y la detectada) de 50 % para medir el mAP (mean Avarage Precision, que mide el área bajo la curva de la relación precisión-recuerdo).

Con el objetivo de explorar la utilidad de las técnicas de aumento de datos utilizadas en la metodología presentada se proponen experimentos con las cuatro combinaciones generadas de usar o no usar el aumento de datos externos y el aumento de datos interno. La lista completa de experimentos realizados se muestra en la Tabla 1.

4.1. Resultados y discusión

En la Figura 3 se muestra una gráfica comparativa de los 8 experimentos realizados. Las líneas punteadas corresponden a los modelos entrenados con YOLOv2 y las líneas continuas a los entrenados con YOLOv4. En esta gráfica se puede notar que sí existe una brecha entre las versiones 2 y 4 de YOLO.

La versión 2 no supera el 75 % de mAP en ninguna de las variantes, mientras el comportamiento de la cuarta versión de YOLO es por arriba de ese rango. Por otra parte, observamos que al considerar el aumento de datos se obtienen mejores resultados que no usarlo.

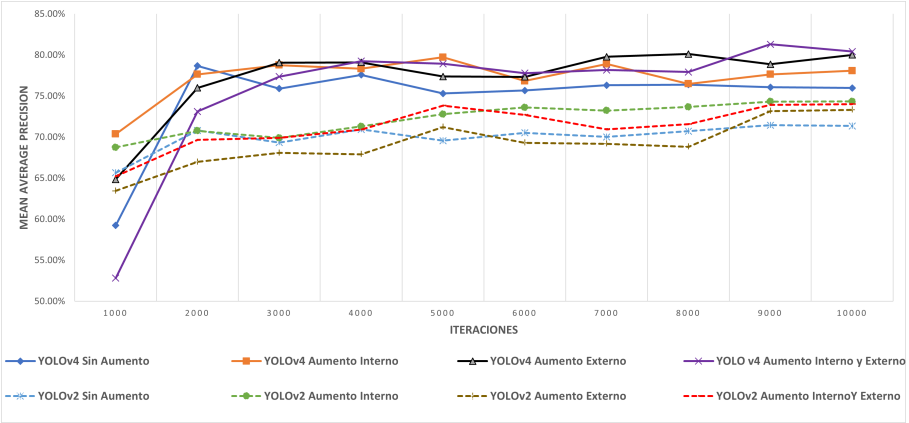

Fig. 3. Gráfica comparativa de resultados obtenidos. Se reporta la métrica de desempeño mAP. Las líneas punteadas corresponden al modelo entrenado sobre YOLOv2, y las líneas sólidas corresponden a desempeño de YOLOv4.

Tabla 2. Mejores resultados por cada experimento. La columna iteración indica el momento en el que el modelo obtuvo su mejor desempeño.

Experimento	iteración	mAP
yolov2	9000	71.44
yolov2-int	10000	74.33
yolov2-ext	10000	73.32
yolov2-int-ext	10000	74.00
yolov4	4000	77.58
yolov4-int	5000	79.71
yolov4-ext	8000	80.12
yolov4-int-ext	9000	**81.30**

Se obtuvieron los mejores resultados con YOLOv4 con aumento de datos interno y externo (yolov4-int-ext). Así mismo, el mejor resultado del modelo generado con YOLOv2 sin aumento de datos (yolov2) es el peor en comparación con los mejores resultados de los demás modelos, esto lo podemos notar en la Tabla 2.

De esta tabla se puede notar que la mayoría de los experimentos llegan a su punto máximo en las últimas iteraciones donde se disminuye la tasa de aprendizaje. También podemos notar que entre la línea de base (yolov2) y el experimento con el mejor rendimiento (yolov4-int-ext) hay casi un 10 % de diferencia.

4.2. Ajuste de los modelos

Para mejorar el desempeño de los modelos, se propusieron experimentos adicionales sobre los parámetros de la arquitectura de la red neural. En estos experimentos se modifica la tasa de aprendizaje (para considerar además de 0.001 a 0.01 y 0.0001) y el tamaño de lote (además de 64 considerar también 32 y 128).

(a) Falsos positivos.

(b) Falsos negativos.

(c) Verdaderos positivos.

Fig. 4. Ejemplos de imágenes en el conjunto de prueba y el resultado de la evaluación cualitativa. Bajo cada imagen se muestra la resolución en pixeles.

Para la realización de estos experimentos, se usó la configuración de datos del experimento yolo4-ext (i.e., solo aumento de datos externo) debido a que ese fue el modelo con el rendimiento más parecido al de los mejores resultados, pero que el tiempo de entrenamiento es la mitad.

Sin embargo, ninguna de estas modificaciones representó mejora alguna. El mejor resultado obtenido fue de 74 % de mAP.

4.3. Evaluación cualitativa

Con la finalidad de evaluar de forma cualitativa el modelo de detección de armas generado en los experimentos previos se eligió de forma manual una muestra del conjunto de prueba (`P`). Esta muestra consistió de 30 ejemplos de imágenes que consideramos eran más cercanas al escenario real (asaltos), ya que es este escenario en donde se pretende que funcione este sistema propuesto.

Para este análisis se usaron las métricas de evaluación tradicionales para sistemas de aprendizaje supervisado; esto es, falsos positivos, falsos negativos y verdaderos positivos.

Se aplicó el modelo yolov4-int-ext al conjunto de imágenes seleccionadas. De las 30 imágenes, el modelo detectó 19 verdaderos positivos, 13 falsos negativos y 5 falsos positivos.

De aquí se obtiene una precisión de 79 %, un recuerdo de 59 % y una medida-F de 0.629. En la Figura 4 se pueden observar tres ejemplos de imágenes con falsos positivos (se identificó un objeto diferente como arma), falsos negativos (no se identificaron armas donde había un arma), y verdaderos positivos (se identificaron armas de forma correcta).

Al analizar los resultados de las 30 imágenes y la Figura 4 se observa que las predicciones tienden a ser erróneas cuando la resolución de las imágenes es baja (por ejemplo la imagen de la izquierda y derecha en la Figura 4a y todas las imágenes en la Figura 4b).

También se pueden ver fallas en casos donde es demasiado complicado encontrar el arma incluso para el ojo humano (por ejemplo en las imágenes de la Figura 4b). Por otra parte, cuando el arma se presenta con fondos contrastantes a su alrededor, es más probable que se haga una detección correcta (por ejemplo las imagen de la izquierda y centro en la Figura 4c), pero si el fondo es oscuro o con un tono de color similar al del arma, será más fácil que el modelo se equivoque (ver las imágenes de la izquierda y la derecha de la Figura 4b).

5. Conclusiones

En este artículo se presentó una metodología para generar un modelo automático de detección de armas de fuego en imágenes obtenidas en el transporte público. La metodología propuesta consiste de tres partes, primero un proceso de selección de imágenes de corpora existente donde las imágenes seleccionadas se parezcan a nuestro escenario real.

Luego, la opción de utilizar dos tipos de técnicas de aumentos de datos para incorporar variaciones a las imágenes tanto en posición como en calidad. Finalmente, la metodología usa una arquitectura de redes neuronales convolucionales utilizada en trabajos previos para la detección de armas: YOLO.

Para determinar la utilidad de realizar una selección de datos y posterior automento de datos en un modelo de detección de armas, se propusieron y realizaron 8 experimentos con dos de las arquitecturas más utilizadas en la detección de objetos: YOLOv2 y YOLOv4. El mejor desempeño se obtuvo en el modelo entrenado con YOLOv4 y usando aumento de datos externo e interno, obteniendo un 81.33 % de mAP.

Después del análisis de los resultados se pudo observar que la calidad de las imágenes usado para entrenar un modelo es determinante para la correcta detección de armas de fuego. Por lo tanto, la limpieza de corpora ha ayudado a obtener mejores modelos en la detección de armas aplicada a ejemplos de asaltos reales, aunque esto disminuye considerablemente el tamaño del corpus.

Aumentar los datos de manera artificial tanto interna como externamente nos ha ayudado a ampliar la variabilidad de imágenes en el conjunto de datos usados para el entrenamiento y ha contribuido a mejorar el desempeño del modelo generado. Sin embargo, esto no resuelve todas las problemáticas del escenario real. Por ejemplo, el modelo tiende a equivocarse cuando la resolución de las imágenes es pequeña o cuando no existen suficiente contraste entre el arma y el fondo donde ésta aparece, dos de los aspectos que ocurren mucho en el escenario real.

Sin embargo, los resultados obtenidos en la evaluación del modelo son prometedores a pesar de haber utilizado una base de datos que contiene en su mayoría ejemplos diferentes a los que hay en el escenario real (como se pudo ver en la Figura 2). Estos resultados nos dan pauta a considerar la replicación de la metodología propuesta con un conjunto inicial de datos obtenidos en la vida real (tomados directamente de los videos que obtiene el C5, por ejemplo).

Como trabajo futuro se planea realizar un etiquetado manual diferente al realizado en este trabajo. Pues en la evaluación cualitativa se observan casos que cuando el modelo comete falsos positivos, es porque identifica una mano en una posición común a la forma de portar un arma que es usada para asaltar. Así mismo, se realizará una comparación exhaustiva con otras arquitecturas usadas en el estado del arte.

Agradecimientos. Los autores y autora agradecen a la Universidad Autónoma Metropolitana Unidad Cuajimalpa por el apoyo otorgado durante la realización de este trabajo.

Referencias

1. Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M.: Yolov4: Optimal speed and accuracy of object detection (2020) doi: 10.48550/arXiv.2004.10934
2. Dhillon, A., Verma, G. K.: Convolutional neural network: A review of models, methodologies and applications to object detection. Progress in Artificial Intelligence, vol. 9, no. 2, pp. 85–112 (2020) doi: 10.1007/s13748-019-00203-0
3. Gobierno de la Ciudad de México: El C5 de la CDMX (2021)
4. Gobierno de la Ciudad de México: Transporte público más seguro (2021)
5. Google Colaboraty: ¿Qué es colaboratory? (2021)
6. Gutiérrez-Lancho, C.: Detección de armas en vídeos mediante técnicas de deep learning. Master's thesis (2019)
7. INEGI: Encuesta nacional de seguridad pública urbana, pp. 1 (2020)
8. INEGI: Encuesta nacional de victimización y percepción sobre seguridad pública, pp. 38 (2020)
9. INEGI: Tasa de incidencia delictiva por entidad federativa de ocurrencia por cada cien mil habitantes (2021)
10. Institute for Economics and Peace: Global peace index 2020 briefing (2020)
11. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: A survey. International Journal of Computer Vision, vol. 128, no. 2, pp. 261–318 (2020) doi: 10.1007/s11263-019-01247-4
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C.: SSD: Single shot multibox detector. Lecture Notes in Computer Science, pp. 21–37 (2016) doi: 10.1007/978-3-319-46448-0_2

13. Olmos, R., Tabik, S., Herrera, F.: Automatic handgun detection alarm in videos using deep learning. Neurocomputing, vol. 275, pp. 66–72 (2018) doi: 10.1016/j.neucom.2017.05.012
14. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016) doi: 10.1109/CVPR.2016.91
15. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7263–7271 (2017) doi: 10.1109/CVPR.2017.690
16. Romero-Mogrovejo, D. O.: Desarrollo de un sistema de detección de armas de fuego cortas en el monitoreo de videos de cámaras de seguridad. Master's thesis (2018)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014) doi: 10.48550/arXiv.1409.1556
18. Verma, G. K., Dhillon, A.: A handheld gun detection using faster r-cnn deep learning. In: Proceedings of the 7th International Conference on Computer and Communication Technology, pp. 84–88 (2017) doi: 10.1145/3154979.3154988
19. Zeiler, M. D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833 (2014) doi: 10.1007/978-3-319-10590-1_53
20. Zhao, Z. Q., Zheng, P., Xu, S., Wu, X.: Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–3232 (2019) doi: 10.1109/TNNLS.2018.2876865