RIC REVISTA DE INICIACIÓN CIENTÍFICA
Portal de revistas: http://revistas.utp.ac.pa

Red neuronal artificial para detección de armas de fuego y armas blancas en video vigilancia

Artificial neural network for firearms and cutting weapons detection in video surveillance

Miguel A. Campos[1], *Jacqueline Sánchez*[2*]
[1]Licenciatura en Ingeniería de Sistemas y Computación, Facultad de Ingeniería de Sistemas Computacionales, Universidad Tecnológica de Panamá,
[2] Facultad de Ingeniería de Sistemas Computacionales, Universidad Tecnológica de Panamá

***Autor de correspondencia:** jacqueline.sanchez@utp.ac.pa

Resumen. Con el crecimiento de la población en América Latina, las ciudades incrementan el uso de video vigilancia para monitorear áreas con el objetivo de detectar incidentes de violencia y/o delictivos para tomar acción oportunamente. Actualmente el proceso de video vigilancia es desarrollado por personal humano revisando innumerables cantidades de señales de video al mismo tiempo, la actual solución es propensa a errores, no es escalable y plantea un desafío. En esta contribución se propone construir una red neuronal convolucional CNN para la detección de armas de fuego y armas blanca en imágenes con el objetivo de automatizar y optimizar el proceso de monitorización de señales de video. Se especificó una arquitectura de red neuronal artificial que fue entrenada con un conjunto de datos (construido a medida) y evaluada para dar solución a problemática. Se logra construir el conjunto de datos objetivo y la arquitectura SSD con red base Inception V3. La arquitectura logró la detección satisfactoria de las características propuestas luego de ser entrenada con el conjunto de datos, y se discuten ciertos elementos que podrían ser mejorados en futuras experiencias.

Palabras clave. Aprendizaje profundo, CNN, detección de objetos, red neuronal artificial, SSD, vigilancia inteligente, visión computacional.

Abstract. With the growth of the population in Latin America, cities increase the use of video surveillance to monitor areas in order to detect incidents of violence and/or crime to take timely action. Currently the video surveillance process is developed by human personnel reviewing countless video signals at the same time, the current solution is error prone, not scalable and challenging. In this contribution, it is proposed to build a convolutional neural network CNN for the detection of firearms and cutting weapons in images for automating and optimizing the surveillance process. An artificial neural network architecture was specified and trained with a dataset (custom built) and tested to solve the problem. It was possible to build the dataset and the SSD architecture using Inception V3 as base network. The architecture achieved the satisfactory detection of the proposed characteristics after being trained with the dataset, and some elements that could be improved in future experiences are discussed.

Keywords. Deep Learning, CNN, Object detection, artificial neural network, SSD, smart surveillance, computer vision.

1. Introducción

La República de Panamá y Latinoamérica en general, cada día se enfrenta a problemas de violencia y criminalidad más severos. Hoy en día la región es considerada la más violenta del mundo [1], con serias agrupaciones dedicadas al crimen organizado, tráfico de drogas y armas. Por esta razón, los países de la región sufren una amenaza a la gobernabilidad democrática y la seguridad pública, además de la desmejora progresiva de los habitantes en su calidad de vida. Ciertamente, América Latina está inundada de armas. Dos tercios de todos los asesinatos en Centroamérica se cometen con armas de fuego, esto en contraste al treinta y dos por ciento que es el promedio global [2].

Como consecuencia, de lo anterior, las autoridades de muchos países han recurrido a incrementar la presencia de cámaras de video vigilancia, las cuales son dispositivos muy económicos, pero el proceso de monitorización hace el proceso de vigilancia costoso e ineficiente, debido a que es realizado por seres humanos [3].

Muchas veces los sistemas de video vigilancia cumplen un rol de documentación de hechos al cual se recurre cuando ya se ha reportado el incidente, si se usaran los sistemas de video vigilancia de forma activa e inteligente para detectar eventos que requieren atención, entonces se podría optimizar el personal necesario para hacer el proceso de monitorización [3].

Entonces, surge la interrogante: ¿será posible la implementación de un *software* para la detección inteligente de armas de fuego y/o blanca en señales de video? Algunas investigaciones previas han desarrollado algunos algoritmos para mitigar esta problemática anteriormente como se detalla en la siguiente sección.

1.1 Antecedentes

Previo al inicio de esta investigación se han desarrollado algoritmos que detectan armas de fuego y armas blancas, se pueden destacar dos investigaciones principalmente.

El primer enfoque fue el problema en dos grandes partes, en una parte se desarrolló la detección de cuchillos, y en otra la detección de armas de fuego. Se planteó la detección de cuchillos basada en la detección de descriptores visuales y el empleo de *machine learning* [4]. El algoritmo identifica las manos y/o el abdomen de los seres humanos en la imagen, luego usando un mecanismo de ventanas deslizantes sobre la imagen que extraen características propias de los cuchillos mediante el empleo de una clasificación con el algoritmo SVM siglas de *Support Vector Machine* en inglés (Maquinas de vector de soporte) [4].

El enfoque planteado para la detección de armas de fuego es diferente al empleado para la detección de cuchillos, debido a la dificultad de crear un algoritmo que llene todos los requerimientos, los autores de la investigación se enfocaron en la detección de pistolas de entre todos los elementos del conjunto de armas de fuego [4].

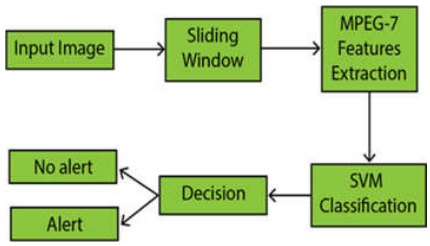

Figura 1. Algoritmo de detección de cuchillos.
Fuente: Automated Detection of Firearms and Knives in a CCTV Image. [4]

El segundo grupo de algoritmos plantea la extracción de características con el algoritmo de histogramas de gradientes orientados (HOG). Este algoritmo fue escogido por el autor ya que se demostró que es el algoritmo que mejor describe los bordes de la hoja de un cuchillo [5]. La clasificación es realizada con una red neuronal artificial formada con una capa de entrada, dos capas intermedias (con 50 y 30 neuronas artificial respectivamente) y una capa de salida con tres neuronas artificiales [5].

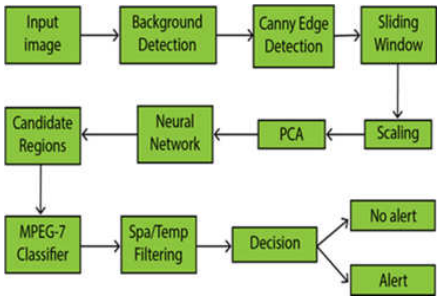

Figura 2. Algoritmo de detección de armas de fuego.
Fuente: Automated Detection of Firearms and Knives in a CCTV Image. [4]

En *machine learning*, especialmente en aprendizaje supervisado con redes neuronales, es frecuente el empleo de optimizadores matemáticos para aproximar el comportamiento del algoritmo a el conjunto de soluciones objetivo. Entre algunos de los optimizadores más conocidos en el campo, está el descenso de gradiente estocástico (SGD en adelante), el cual minimiza los parámetros de una función [6] (aprendizaje) mediante el uso del tradicional algoritmo de descenso del gradiente (DG en adelante) aplicado en lotes (subconjunto del conjunto de datos de entrenamiento) cuyas instancias son seleccionadas de forma completamente aleatoria. El algoritmo SGD representa una ventaja sobre el algoritmo GD, justificado en una rápida convergencia de la función objetivo, y la minimización significativa del espacio ocupado por el algoritmo con respecto a su predecesor (descenso del gradiente).

El algoritmo de estimación de momento adaptativo (ADAM en adelante) es un optimizador matemático derivado del descenso de gradiente estocástico que introduce estimaciones adaptativas de momento [7], es decir, el cálculo de diferentes tasas de aprendizajes para cada conjunto de parámetros entrenables [6]. Los autores, demostraron de forma experimental la efectividad del algoritmo ADAM, obteniendo tasas de convergencia mas rápidas que SGD [7].

Dentro del mundo de la detección de objetos en visión computacional existen muchos enfoques para la detección de objetos de interés, algunos otros enfoques emplean el uso de *machine learning* para procesos de clasificación y en algunos otros enfoques se emplea el uso de *machine learning* para

clasificación y ubicación del objeto de interés. La arquitectura *single shot multibox detection* (SSD en adelante) es una arquitectura que se enfoca en la ubicación y clasificación de objetos de interés [8]. La arquitectura SSD emplea el uso de una red base para la extracción de características y la aplicación de convoluciones (*multibox*) para la detección de los objetos de interés (ubicación y predicción de categoría). En una ultima etapa, se aplica un proceso para seleccionar las predicciones mas destacadas (según un parámetro configurable) y suprime la detección sobrepuesta o repetidas (mediante operaciones de conjuntos), esta etapa se lleva a cabo en la capa "*non-maximum suppression*" [8].

1.2 Objetivos, relevancia y contribución

La meta fue construir un algoritmo que sea capaz de reconocer armas de fuego y armas blancas al mismo tiempo. Para ello, primero, hubo que recolectar los datos para construir el conjunto de datos que sirvieron para entrenar y evaluar el algoritmo; segundo, la construcción del algoritmo (definir y construir la arquitectura de red neuronal) para luego, tercero, entrenar y evaluar la arquitectura de red neuronal con el conjunto de datos recolectado.

Una vez estos objetivos fueron realizados, se pudo disponer de un algoritmo para poder detectar armas de fuego o armas blancas en múltiples sitios video vigilados por un equipo de seguridad de forma automática e inteligente. Un algoritmo como el planteado en esta contribución podría significar la disminución del personal de video vigilancia revisando constantemente señales de video donde no se está desarrollando un hecho delictivo con armas, y emplear el tiempo y/o recursos humanos en otras actividades no automatizadas.

Se han desarrollado otras investigaciones relacionadas a la detección de armas de fuego o armas blancas, pero el dominio de este trabajo se enfoca en un algoritmo que pueda detectar ambos grupos de objetos y no una detección parcial o por separado.

2. Método

2.1 Construcción del conjunto de datos

Todo algoritmo de *machine learning* necesita de datos para poder ser entrenado, los algoritmos que se pretenden desarrollar en las siguientes secciones no son la excepción, por lo tanto, la necesidad de un conjunto de datos que describan acertadamente las características que representan cada una de las categorías a detectar es de suma importancia.

2.1.1 Definición de objetos a detectar

La idea de un sistema detección de objetos es cumplir con una serie de propiedades, entre ellas reconocer muchos tipos de objetos y categorizarlos [6, p. 540]. Pero ¿cuáles son esos objetos a detectar? La red neuronal se entrenó para detectar un conjunto especifico de armas (fuego y blanca).

Un arma de fuego la podemos definir como un dispositivo con la capacidad de provocar la ignición de algún combustible que impulse un (unos) proyectil (es) no explosivo (s), y para efectos de esta investigación lo delimitamos a armas de fuego capaces de ser acarreadas por seres humanos sin ayuda extra; las armas de fuego se pueden clasificar a su vez en dos categorías, cortas y largas, y se refiere a la longitud del cañón y de los accesorios que por su tamaño (del arma) forman parte del arma en sí.

Por otro lado, un arma blanca es una herramienta capaz de producir cortes o punzar, mediante bordes o puntas afilados (as), para efectos de esta investigación se delimitó a armas blancas capaces de ser acarreadas por seres humanos sin ayuda extra.

Se desarrolló un algoritmo que detecta esas tres categorías, entonces las categorías a detectar son:

- Arma de fuego corta.
- Arma de fuego larga.
- Arma blanca.

Con el objetivo de explicar claramente cada una de las categorías a detectar se listan algunos ejemplos de armas para cada categoría. Entre las armas de fuego (cortas) se pueden destacar ejemplos:

- Pistolas.
- Sub fusiles.
- Pistolas con silenciadores.

Entre las armas de fuego (largas) se pueden destacar como ejemplos:

- Ametralladoras.
- Rifles.
- Escopetas.
- Rifles de asalto.
- Rifles francotirador.

Entre las armas blancas podemos destacar ejemplos:

- Cuchillos.
- Navajas.
- Espadas.
- Machetes.
- Katanas (espada japonesa).
- Puñal.
- Sables.

2.1.2 Etiquetado de objetos en imágenes

Luego de recolectadas las imágenes con base en los objetos a detectar definidos previamente, se procedió a hacer el etiquetado de objetos sobre cada una de las imágenes.

El etiquetado de objetos en imágenes (*image labeling* en inglés) es el proceso de identificar los objetos presentes en las imágenes matemáticamente.

Cada uno de los objetos debe ser identificado con un recuadro independiente, cada recuadro representa una tupla de 4 números, posición (x, y), altura y ancho. Existen ciertos *softwares* que no hacen uso de altura y ancho, en su defecto se usa un segundo par de coordenadas (x2, y2), este enfoque junto con el enfoque anterior se puede usar indistintamente siempre y cuando todo el conjunto de datos este bajo un solo formato.

El *software* utilizado para el etiquetado de imágenes es un *software* de código abierto llamado "LabelImg".

2.1.3 Aumento de datos

El aumento de datos se refiere a la optimización interactiva a través de la introducción de muestras de datos no observados [7]. La idea es diversificar el conjunto de datos para generalizar mejor el conocimiento del modelo al momento de ser aplicado el optimizador matemático. El aumento de datos se decidió a implementarse de forma dinámica en código fuente en la fase de entrenamiento de entrenamiento, es decir, no se aumentaron los datos replicando una y otra vez el mismo archivo con diferentes transformaciones, más bien, tales transformaciones se aplicaron en tiempo de ejecución.

En el experimento se hizo un aumento de datos bajo ciertos criterios:

- Enfoque gradual sobre los recuadros únicamente (objetos de interés).
- Reflejo de la imagen en forma horizontal.

Para el entrenamiento se han aplicado todos los criterios en conjunto de forma aleatoria sobre las instancias de datos, aumentando el conjunto de datos entre 9 y 16 veces el tamaño original.

2.2 Modelo

Una vez se construyó el conjunto de datos de acuerdo con las necesidades de cada uno de los experimentos; con el conjunto de datos definido se necesitó el algoritmo de *machine learning* capaz de clasificar las características de cada una de las categorías a detectar y generar las ubicaciones (x, y, w, h) de las que se han hablado anteriormente.

2.2.1 Arquitectura SSD para detección de objetos

Se escogió la arquitectura *single shot multibox detector* SSD como algoritmo para la detección de objetos en imágenes, la red base de este detector es *Inception* v3 como extractor de características.

En la figura 3 se describe la arquitectura SSD [8] con red base *Inception* v3 [9]. Esta nueva arquitectura está basada en módulos *Inception* (bloques de convoluciones nombradas "Incp") que son convoluciones paralelas aplicadas al mismo mapa de características (*feature map* en inglés) [9].

En términos simples, se tomó la arquitectura original SSD y se le ha reemplazado el extractor de características VGG por todas las capas de convoluciones hasta el séptimo grupo de módulos *inception* en la arquitectura *inception* v3, adicionalmente se retiró la convolución multibox descrita en la arquitectura SSD original que toma como mapa de características los productos del grupo de convoluciones número cinco del mapa de características VGG.

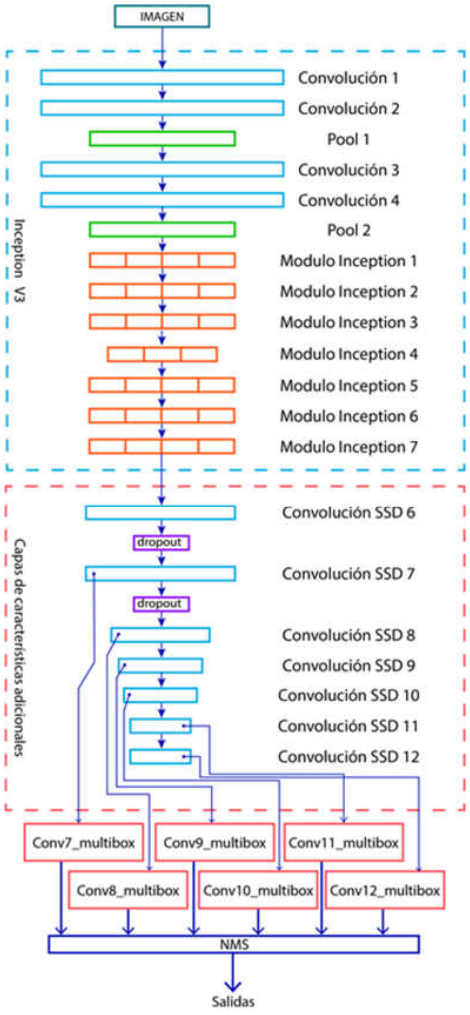

Figura 3. La arquitectura SSD con red base *Inception* v3.

2.3 Entrenamiento

2.3.1 Función de costo

Para poder medir el error del algoritmo (y para poder realizar el proceso de optimización la cual es la que produce el aprendizaje automático) se necesita computar la función de costo, y es que el producto de la función de costo es la tasa de error cuyo valor se encuentra en el rango un número real entre cero y uno [10, p. 101]. Existen muchas funciones de costo de uso general, pero, la arquitectura SSD nos brinda una función de costo específica para el cálculo del rendimiento de cada una de las capas *multibox* [8].

Para la computación de la función de costo específica de la arquitectura SSD, se calculan básicamente dos grandes funciones de costo, la primera función de costo es de confianza y la segunda función de costo es de localización [8] como se muestra en (1).

$$L(x,c,l,g) = 1/N\left[L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\right] \quad (1)$$

Donde N es el número de recuadros por defecto, g son los recuadros de prueba (teóricos), l son los recuadros generados, x categorías generadas, c categorías teóricas, α es una constante de valor 1, L_{loc} es la función de costo de la ubicación de recuadros, L_{conf} es la función softmax entre categorías generadas y las teóricas.

2.3.2 Optimizador

El proceso de optimización consiste en encontrar la mejor hipótesis que se ajuste a la experiencia (conjunto de datos) [11, p. 722]. Con la arquitectura implementada, se debe definir cuál de los optimizadores matemáticos deberán ser configurados para el entrenamiento de la red neuronal convolucional.

ADAM utiliza el concepto de "*momentum*" para hacer converger más rápido la red neuronal convolucional, el cual simplemente significa que se agrega una fracción de la actualización anterior a la actualización actual, de modo que las actualizaciones repetidas en un compuesto de dirección particular acumulamos impulso, moviéndonos más y más rápido en esa dirección [12]. ADAM utiliza la filosofía de seleccionar adaptativamente una tasa de aprendizaje por separado para cada parámetro. Los parámetros que normalmente recibirían actualizaciones más pequeñas o menos frecuentes recibirán actualizaciones más grandes con ADAM, esto acelera el aprendizaje en los casos en que las tasas de aprendizaje adecuadas varían según los parámetros [12].

2.3.3 Hiperparámetros

El comportamiento del algoritmo de aprendizaje es controlado mediante el empleo de hiperparámetros [10], los hiperparámetros común mente son utilizados en algoritmos de *machine learning* para modificar la forma en que el algoritmo opera o es entrenado.

En el proceso de experimentación se plantea utilizar como optimizador ADAM con una tasa de aprendizaje entre (0.1, 0.01, 0.001), un coeficiente de regulación L2 de 0.0005; una mínima tasa de aprendizaje de 0.0000001; y un coeficiente de *dropout* del 85%.

Específicamente del algoritmo ADAM se emplea una beta1 de 0.9 y una beta2 de 0.999, todo esto junto a un valor de *momentum* 0.9.

Se configuró el tamaño del lote a entrenar (muestra del conjunto de datos) entre 10 y 35 muestras aleatorias por ciclo; y específicamente del algoritmo SSD se configura un conjunto de cajas de detección por defecto con los siguientes valores [(21.0, 45.0), (45.0, 99.0), (99.0, 153.0), (153.0, 207.0), (207.0, 261.0), (261., 315.0)].

3. Resultados

La arquitectura SSD con *Inception* V3 configurado con los hiperparámetros especificados se ha entrenado se puede apreciar a simple vista que la arquitectura llega a converger, esto con base en la forma exponencial invertida con asíntota en y = 0 que tiene la gráfica de la función de costo mostrada en la figura 4.

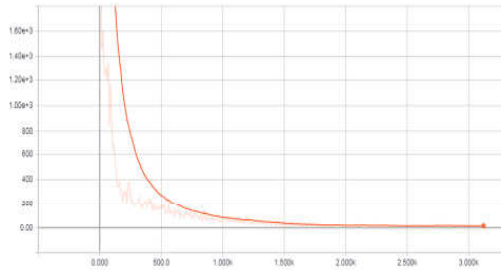

Figura 4. Gráfica de función de costo total del experimento.

Los hallazgos principales demuestran que sí es posible detectar armas de fuego y armas bancas en videos extraídos de video vigilancia, en el domino de este trabajo, se demuestra que las redes neuronales convolucionales demuestran su utilidad para el procesamiento de imágenes y sus aplicaciones en detección de armas de fuego y arma blanca en un mismo algoritmo.

Como segundo hallazgo, se puede destacar una ventaja de la arquitectura SSD con *Inception* V3 comparada con la

arquitectura SSD con VGG en la cantidad de parámetros necesarios para su entrenamiento.

En la figura 5, se puede apreciar una disminución considerable en el número de parámetros de SSD *Inception* V3 con respecto a SSD VGG (propuesta inicialmente en la arquitectura SSD [8]), la disminución representa un 26% menos parámetros.

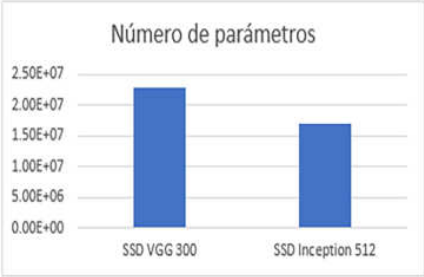

Figura 5. Gráfica de comparación del número de parámetros entre las arquitecturas SSD VGG y SSD *Inception* V3.

También, halló dificultad para la detección de objetos de interés pequeños o muy pequeños, esto dentro de la imagen ajustada a 512 pixeles (en una relación de aspecto cuadrado), todo objeto que ocupe un área de pixeles menor a 4,000 pixeles cuadrados (64^2pixeles cuadrados) es muy complejo de detectar para el modelo.

Otra limitación que se observó en los resultados radica en las categorías que se intentan detectar, dichas categorías son muy complejas en cuanto a características, y muy diversas en cuanto a los tipos subyacentes a cada categoría, por lo tanto, existen objetos que pueden ser detectados correctamente en sus localizaciones, pero incorrectamente clasificados.

4. Conclusiones

Los enfoques previos a este trabajo plantearon la utilización de múltiples algoritmos para resolver tareas independientes, extracción de características, desplazamiento de ventanas, algoritmos de clasificación como SVM y en un caso el uso de una red neuronal.

Con los nuevos resultados se incorpora el enfoque de redes neuronales convolucionales para la detección de objetos, y se demuestra el uso de un único algoritmo para detectar armas de fuego y armas blancas.

Existen algunas limitaciones en el algoritmo y en el desarrollo de este trabajo como la detección de objetos pequeños, que es un aspecto que se debe mejorar en futuros trabajos. Otra limitante de esta investigación se presenta en el conjunto de datos, debido a su reducido tamaño. Este aspecto debe ser mejorado para futuros trabajos con el objetivo de provocar una generalización del aprendizaje, disminuir el error, mitigar el sobreajuste, evitar la memorización del conjunto de datos; para esto se puede solicitar apoyo de las autoridades locales quienes tienen acceso a datos del dominio de esta temática.

AGRADECIMIENTOS

Al Profesor Víctor López por su insistencia y revisión de este trabajo.

CONFLICTO DE INTERESES

Los autores declaran no tener ningún conflicto de interés.

REFERENCIAS

[1] T. Phillips, «'Breathtaking homicidal violence': Latin America in grip of murder crisis,» 26 April 2018. [En línea]. Available: https://www.theguardian.com/world/2018/apr/26/latin-america-murder-crisis-violence-homicide-report

[2] A. Erickson, "Latin America is the World's most violent region. A new report investigates why.," 25 April 2018. [Online]. Available: https://www.washingtonpost.com/news/worldviews/wp/2018/04/25/latin-america-is-the-worlds-most-violent-region-a-new-report-investigates-why/

[3] T. Ko, «A Survey on Behavior Analysis in Video Surveillance for Homeland Security Applications,» IEEE xplore, 2008.

[4] M. Grega, A. Matiolański, P. Guzik y M. Leszczuk, «Automated Detection of Firearms and Knives in a CCTV Image,» MDPI Open Access Journals, 2016.

[5] R. Vajhala, R. Maddineni y P. R. Yeruva, «Weapon Detection In Surveillance Camera Images,» 2016.

[6] D. Forsyth y J. Ponce, Computer Vision A Modern Approach 2nd Edition, Pearson.

[7] D. A. Van Dyk y X.-L. Meng, «The Art of Data Augmentation,» Journal of computation and grafical statistics.

[8] W. Liu, A. Dumitru, C. Szegedy, S. Reed, C.-Y. Fu y A. C. Berg, «SSD: Single Shot Multibox Detector,» Lecture Notes in Computer Science, 2016.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke y A. Rabinovich, «Going Deeper with convolutions,» IEEE Xplore, 2015.

[10] I. Goodfellow, Y. Bengio y A. Courvill, Deep Learning, Cambridge, Massachusetts, London England: The MIT Press, 2016.

[11] S. J. Russell y P. Norvig, Artificial Intelligence A Modern Approach 3rd Edition, Pearson, 2010.

[12] D. P. Kingma y J. Lei Ba, «ADAM: A Method For Stochastic Optimization,» ICLR, 2015.