¿ES POSIBLE ENTRENAR MODELOS DE APRENDIZAJE PROFUNDO CON DATOS SINTÉTICOS?

Noelia Vallez, Alberto Velasco-Mata, Juan Jose Corroto y Oscar Deniz

VISILAB, ETSI Industriales, Avda Camilo Jose Cela sn, 13071 Ciudad Real, Spain
Noelia.Vallez@uclm.es

Resumen

La demanda de datos para el entrenamiento de las nuevas técnicas de aprendizaje profundo se ha incrementado durante los últimos años. Aunque se ha creado una comunidad extensa alrededor del intercambio de datos, e incluso muchos de los conjuntos de datos de grandes empresas se han publicado de forma gratuita, continúa habiendo problemas específicos para los que no se dispone de conjuntos específicos para el entrenamiento de los modelos que los resuelvan. Este es el caso de la detección de armas en escenas videovigiladas donde la detección temprana de situaciones y objetos peligrosos es de vital importancia. Varias han sido las soluciones propuestas en los últimos años al respecto pero la adquisición de los datos necesarios para su desarrollo sigue siendo un problema. Por ello, en este trabajo se propone generar imágenes de videovigilancia con un motor gráfico y comprobar si estos datos sintéticos pueden sustituir la captura y el etiquetado de imágenes reales.

Palabras clave: Detección de objetos, aprendizaje profundo, datos sintéticos.

1. INTRODUCCIÓN

La base de la capacidad de reconocimiento del aprendizaje máquina está en los datos que se utilizaron para entrenar. Los datos se han convertido, por tanto, en la moneda de cambio para la obtención de modelos de utilidad en distintas aplicaciones cotidianas. Este hecho ha cobrado aún más importancia con el desarrollo de las técnicas de aprendizaje profundo en las que suele ser necesario contar con gran cantidad de datos (del orden de cientos de GBs en algunos casos). Actualmente solo algunas empresas como Google, Microsoft o Facebook tienen acceso directo a tal cantidad de datos y cuentan con los recursos necesarios para prepararlos y utilizarlos.

Aunque se ha creado una extensa comunidad alrededor del aprendizaje automático y de los datos, e incluso muchos de los conjuntos de datos de grandes empresas han sido puestos a disposición de la comunidad, continúa habiendo problemas específicos para los que no se dispone de los datos necesarios. Este es el caso de la vigilancia automática donde no se cuenta con conjuntos de datos para el entrenamiento de algoritmos de detección de armas en estancias videovigiladas por cámaras de circuito cerrado de televisión (CCTV) [7].

La detección temprana de situaciones y objetos peligrosos es de vital importancia en seguridad. Hasta el momento este tipo de sistemas ha requerido la supervisión constante por parte de de una persona de las imágenes capturadas por las cámaras del sistema de CCTV. Durante los últimos años se ha tratado de automatizar estos sistemas mediante la detección de objetos peligrosos y eventos sospechosos en secuencias de vídeo obteniéndose resultados prometedores con el uso de las nuevas técnicas de aprendizaje profundo [9, 7, 17].

Debido a la necesidad de contar con datos para entrenar nuevos modelos se han llevado a cabo algunos estudios sobre cómo generarlos artificialmente [8]. Quizá el caso más representativo es el uso de videojuegos para generar los datos necesarios para el entrenamiento de los modelos que incorporan algunos coches autónomos [16]. En cualquier caso, los datos sintéticos aportan algunas otras ventajas como son la generación de situaciones difíciles de ocurrir en un entorno real o la ausencia de problemas de privacidad.

A pesar de lo expuesto anteriormente, es necesario comprobar si los modelos entrenados con datos exclusivamente sintéticos son capaces de obtener buenos resultados con datos reales. Con esta finalidad, proponemos la creación de un escenario videovigilado con el motor gráfico de Unreal [15]. Dicho escenario representa el pasillo de un instituto por el que transita gente. De todo lo que puede implicar un sistema de videovigilancia nos hemos centrado en el problema de detección de objetos peligrosos, más concretamente en la detección de pistolas. Tres detectores de pistolas han sido entrenados, utilizando las arquitecturas YOLO, Tiny-YOLO y VGG-SSD, con las imágenes sintéticas y probados con imágenes reales con resultados que hacen pensar que no es posible utilizar estos datos de forma general.

https://doi.org/10.17979/spudc.9788497497169.859

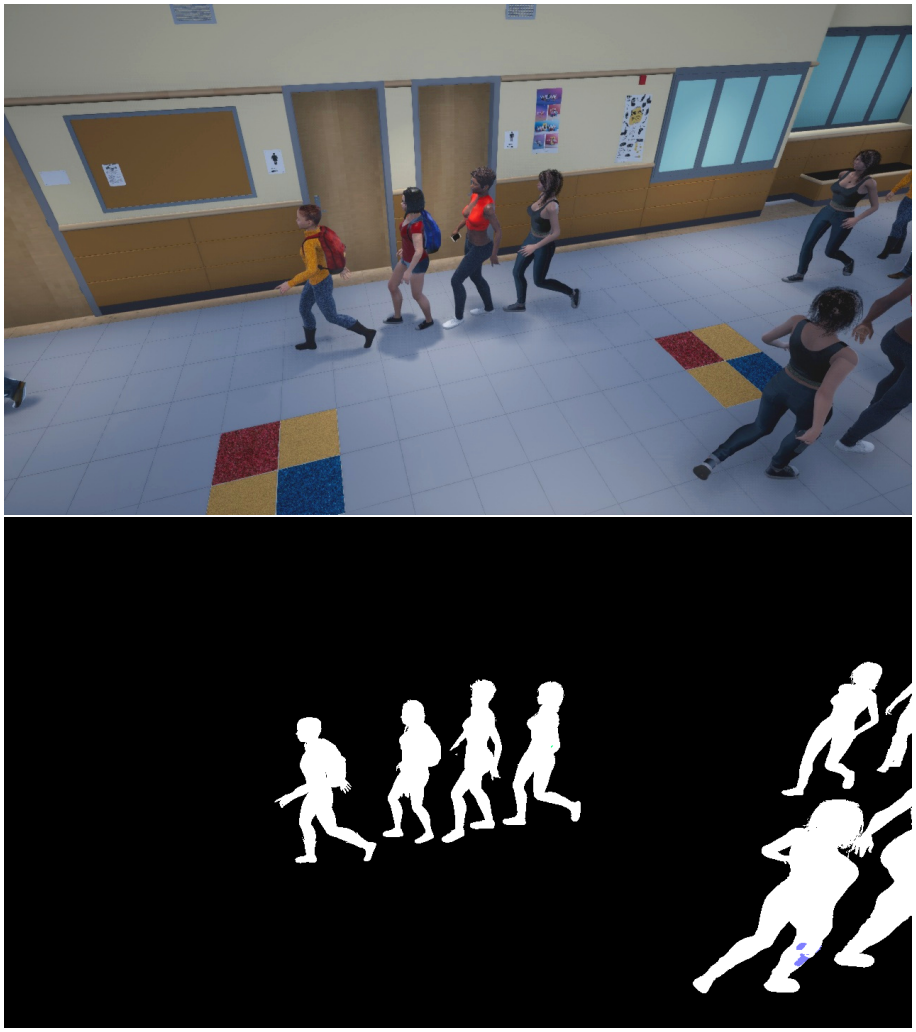

Figura 1: Escenario sintético y máscara con las pistolas. En este caso el arma se encuentra marcada en la esquina inferior derecha.

El resto del documento se organiza como sigue. En la Sección 2.1 se explica todo lo referente a la generación de las imágenes artificiales y el resto de imágenes utilizadas. La sección 3 expone los detectores utilizados y entrenados con el conjunto de datos artificial. Por último, la Sección 4 muestra los resultados obtenidos y la Sección 5 comenta las principales conclusiones del trabajo.

2. CONJUNTOS DE DATOS

2.1. IMÁGENES SINTÉTICAS

Tanto la recogida como el etiquetado de los datos necesarios para entrenar modelos de aprendizaje profundo son tareas que requieren mucho tiempo y esfuerzo. Estas tareas son aún más complejas en problemas de detección o de segmentación en los que alguien debe seleccionar la zona de la imagen en la que se encuentra el objeto, o el contorno exacto de éste, además de la categoría a la que pertenece. Una posible solución a este problema es el uso de conjuntos de datos públicos pero, dependiendo del problema en cuestión, no siempre es posible contar con ellos.

La generación de imágenes sintéticas facilita el trabajo de etiquetado y la generación de grandes conjuntos de datos. Para este trabajo se ha generado un conjunto de datos totalmente sintético con Unreal Engine 4 [15] a partir de un escenario que representa un pasillo de instituto y desde el punto de vista de una cámara de seguridad. Existen alternativas a este motor gráfico como pueden ser Unity [14] o Lumberyard/CryEngine [6] que también pueden utilizarse con la misma finalidad.

Las imágenes se obtienen, por tanto, de un vídeo generado a partir de la representación del escenario virtual. Mientras que algunas de las personas del escenario portan objetos cotidianos en las manos, como pueden ser teléfonos móviles, otras llevan pistolas o no llevan nada (Figura 1). Como la

generación de los datos está controlada totalmente por el usuario, es posible generar también de forma automática una imagen de máscara con las pistolas para cada una de las imágenes del vídeo (Figura 1).

Una vez obtenidas las máscaras se extrae toda la información relativa a las coordenadas de la caja que contiene cada arma y se guardan las anotaciones en archivos XML con el formato definido por el concurso *Pascal VOC 2012 Challenge* por ser uno de los formatos más extendidos [3] (ver Figura 2).

```
<annotation>
        <filename>
            image_99681.jpg
        </filename>
        <size>
                <width>1280</width>
                <height>720</height>
                <depth>3</depth>
        </size>
        <object>
                <name>Weapon</name>
                <bndbox>
                        <xmin>1114</xmin>
                        <ymin>600</ymin>
                        <xmax>1152</xmax>
                        <ymax>632</ymax>
                </bndbox>
        </object>
</annotation>
```

Figura 2: Archivo XML de etiquetado de la imagen de la Figura 1

En total se han generado 100.000 imágenes con 147.838 armas. La resolución de estas imágenes es de 1280×720. De todo el conjunto, 70.000 imágenes se utilizaron para el entrenamiento de todos los modelos, 10.000 para su validación y las 20.000 restantes para test.

2.2. IMÁGENES REALES

Para comprobar el funcionamiento de los detectores entrenados con el conjunto de datos sintético se cuenta con vídeos reales grabados y etiquetados por la Universidad de Sevilla dentro del proyecto Victory [1]. Dichos vídeos fueron grabados en un pasillo y en un laboratorio de la propia Universidad (Figura 3). La perspectiva de estas imágenes es la de una cámara de seguridad ubicada en el techo, de forma similar al conjunto de datos artificial. Este conjunto de datos cuenta con 871 imágenes que contienen 262 armas anotadas.

Figura 3: Escenario real

3. DETECTORES DE PISTOLAS

Para comprobar si los detectores entrenados con el conjunto de datos sintético se comportan igual con los datos reales es necesario primero entrenar algunos de ellos. Tradicionalmente se han venido aplicando métodos clásicos de aprendizaje máquina como son el *keypoint matching* o la extracción de características y su posterior clasificación para la localización de objetos peligrosos en imágenes RGB tomadas de cámaras CCTV [7]. La mayoría de esos métodos utilizan el enfoque de "ventana deslizante" con el que es posible convertir el problema de detección en un problema de clasificación. Este enfoque no solo funciona con los métodos tradicionales sino que puede ser aplicado también a las nuevas técnicas de aprendizaje profundo. Las redes neuronales convolucionales (CNN en inglés) se pueden utilizar de la misma forma que un clasificador lineal o un clasificador en cascada pero sin tener que representar la imagen a través de un vector de características para realizar la clasificación.

El problema de la ventana deslizante radica en la variabilidad tanto en las ubicaciones de los objetos dentro de las imágenes como en la relacción de aspecto de éstos. Es por ello que el número de regiones que debe ser analizado suele ser muy grande. Una posible solución a este problema se basa en la selección de diferentes regiones de interés mediante un método de búsqueda selectiva que elige esas "regiones candidatas" [4]. Dentro del aprendizaje profundo, este método es utilizado por las arquitecturas R-CNN, Fast R-CNN y Faster R-CNN [12]. Sin embargo, existen otras arquitecturas que no requieren la búsqueda previa de las regiones candidatas como son YOLO (You Only Look Once), Tiny-YOLO [10] y SSD (Single-Shot Detector) [5]. Estas redes son capaces de predecir las coordenadas y las clases de varios objetos en la imagen examinándola en una única pasada.

YOLO divide la imagen en regiones y predice las posibles localizaciones y probabilidades de obje-

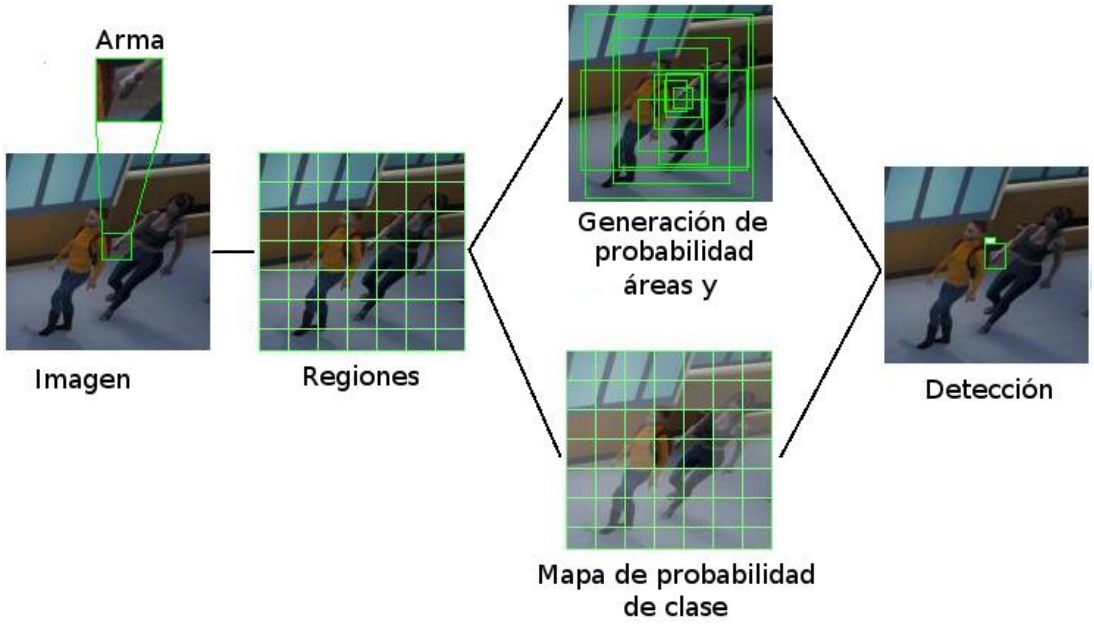

Figura 4: Funcionamiento de la red YOLOv3

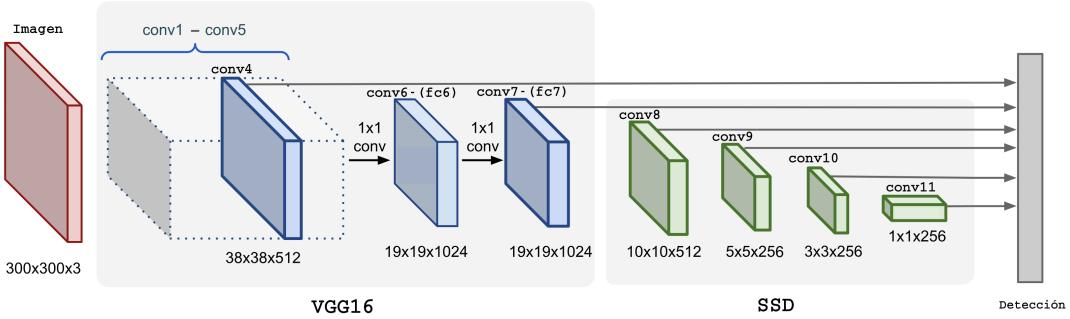

Figura 5: Arquitectura de la red VGG-SSD

to para cada una de ellas (Figura 4). La ventaja principal de este modelo es la incorporación del contexto global de la imagen a la información de la región. Además, al contrario que otras arquitecturas como la R-CNN, YOLO es capaz de realizar inferencias con una única evaluación, lo que reduce el tiempo computacional de forma considerable (1000x más rápida que R-CNN y 100x más que Fast R-CNN según los autores [10]). Existe una variante de esta red llamada Tiny-YOLO. Esta última no es más que una versión reducida de la anterior para sistemas con grandes restricciones de tiempo y memoria. El número de convoluciones y parámetros es mucho menor en esta última.

SSD es otra de las arquitecturas de detección que ha aparecido en los últimos años para cubrir las deficiencias de tiempo computacional de las R-CNN. Al contrario que YOLO, SSD requiere de una arquitectura auxiliar que extraiga el mapa de características (Figura 5). Algunas de las combinaciones más utilizadas son VGG-SSD y MobileNet-SSD [**?**, **?**]. Tanto la probabilidad de localización como la probabilidad de la clase a la que pertenece el objeto se obtienen mediante la aplicación de pequeños filtros convolucionales.

En este caso se han entrenado tres detectores de pistolas con el conjunto de imágenes sintéticas y las arquitecturas YOLOv3 [11], Tiny-YOLOv3 y VGG-SSD.

4. RESULTADOS

Los tres detectores, YOLOv3, Tiny-YOLOv3 y VGG-SSD, han sido entrenados con el 70 % de las imágenes del conjunto de datos generados de forma artificial con Unreal Engine 4. Dichas imágenes contenían un total de 103.742 armas anotadas. Los modelos YOLO y Tiny-YOLO fueron obtenidos utilizando su implementación original en el entorno Darknet [2]. Mientras que YOLO necesitó 36.000 iteraciones para llegar al modelo final, Tiny-YOLO lo hizo en 147.000. En ambos casos

se utilizó una NVIDIA Quadro P4000 con 8GB de GRAM. Por otro lado, el modelo de VGG-SSD se obtuvo entrenando en una NVIDIA GeForce GTX 1080 con 6GB de GRAM con una implementación en Keras y TensorFlow [13] y requirió 75 épocas. El entrenamiento de los tres detectores se realizó inicializando la red con un modelo pre-entrenado y tardó alrededor de 5 días en todos los casos.

Durante el entrenamiento se mantuvo un conjunto de validación con el 10 % de las imágenes del conjunto de datos original. Este conjunto sirvió para seleccionar el mejor modelo y evitar el posible sobreajuste que se comete al seleccionarlo en base a la pérdida obtenida sobre el conjunto de entrenamiento. Este conjunto de imágenes contenía 14.709 pistolas anotadas.

Además del conjunto de validación, se reservó el 20 % restante de las imágenes para realizar una fase de test y ver qué tal se comportaba el modelo sobre datos que no han sido utilizados ni para su entrenamiento ni para para su selección. Estas imágenes contienen un total de 29.387 armas.

Finalmente, se aplicaron todos los modelos obtenidos sobre el conjunto de datos reales formado por 871 imágenes con 262 armas anotadas.

En todos los casos, se obtuvieron una serie de medidas que nos permiten conocer la capacidad de detección de los modelos obtenidos. Estas medidas son:

- Número de verdaderos positivos o detecciones correctas (VP).
- Número de falsos positivos o detecciones incorrectas (FP)
- La media de la precisión promedio del detector (mAP - *mean average precision*) [3].

En la obtención de todas las medidas se utilizó un umbral de solapamiento entre el área de la detección y el área del objeto real, IoU (*Intersection over Union*), de 0,5. Las Tablas 1, 2 y 3 muestran los resultados numéricos obtenidos en los distintos conjuntos de datos evaluados.

Cuadro 1: Resultados de los detectores sobre el conjunto de validación.

Validación				
	Armas	VP	FP	mAP
YOLO		12947	780	89,83 %
Tiny-YOLO	14709	13211	2356	88,15 %
VGG-SSD		14250	1670	90,30 %

Cuadro 2: Resultados de los detectores sobre el conjunto de test.

Test				
	Armas	VP	FP	mAP
YOLO		26020	1515	93,73 %
Tiny-YOLO	29387	26519	4545	88,46 %
VGG-SSD		28531	3362	90,20 %

Cuadro 3: Resultados de los detectores sobre el conjunto de imágenes reales.

Reales				
	Armas	VP	FP	mAP
YOLO		0	8	0 %
Tiny-YOLO	262	0	706	0 %
VGG-SSD		5	6835	0 %

Como se observa en los resultados, los mejores valores se obtuvieron con la arquitectura YOLOv3 con un 89,83 % de mAP en validación y un 93,73 % en test aunque las tres opciones obtienen valores muy altos. Sin embargo, esto cambia drásticamente al aplicar dichos modelos sobre el conjunto de imágenes reales. Los tres modelos fallan al detectar las armas en el conjunto de datos real obteniéndose tan solo 5 detecciones correctas sobre 262 con VGG-SSD en el mejor de los casos. Esto parece indicar que no es posible obtener modelos con datos generados de forma artificial y esperar que se comporten de la misma forma en datos reales.

5. CONCLUSIONES

La generación de datos de forma artificial no es nueva. Cuando el conjunto de datos con el que se cuenta es pequeño, se han venido haciendo modificaciones de las imágenes reales mediante técnicas de aumento de datos que incluyen escalados, traslaciones, rotaciones, etc. Estas técnicas han dado buenos resultados aumentando la capacidad de generalización de los modelos. Sin embargo, estos métodos no contemplan nuevos escenarios ni la generación de datos que sean totalmente artificiales.

En la mayoría de los casos en los que se han utilizado datos generados de forma artificial ha sido para añadir casos que no estaban contenidos en el conjunto de datos real y que debían ser modelados. Esto puede ayudar cuando el conjunto de datos no representa todos los posibles escenarios del problema que se desea resolver pero no elimina la necesidad de recopilar y etiquetar un conjunto de datos de entrenamiento.

En este trabajo se ha comprobado si es posible

utilizar un conjunto de datos generado de forma artificial para entrenar modelos que serán aplicados después en situaciones reales. Para ello se ha creado el escenario de una zona videovigilada mediante el uso de un motor gráfico que representa el pasillo de un instituto por el que transita gente. En este caso, los objetos peligros que deben ser detectados para dar una señal de alarma son las pistolas. Una vez generado, se ha utilizado para entrenar tres detectores diferentes obteniendo valores de mAP de alrededor del 90 % en todos los casos. A pesar de estos resultados, cuando los detectores son aplicados sobre el conjunto de imágenes reales éstos fallan drásticamente, lo que parece indicar que no es posible utilizar únicamente imágenes sintéticas para el entrenamiento de este tipo de detectores.

La ausencia de datos continúa siendo una desventaja en el entrenamiento de modelos de aprendizaje profundo pero el efecto de la incorporación de datos sintéticos aún debe ser estudiado. Otros métodos, como la incorporación de solo una parte de las imágenes sintéticas al conjunto de entrenamiento, deben ser evaluados para conocer el efecto final que tiene realizar esta acción sobre el modelo final.

Agradecimientos

Agradecemos al profesor Dr. J.A. Álvarez por las imágenes reales suministradas para comprobar el funcionamiento de los detectores.

Este trabajo ha sido parcialmente financiado por los proyectos TIN2017-82113-C2-2-R del Ministerio Español de Economía y Empresa y SBPLY/17/180501/000543 del gobierno autonómico de Castilla-La Mancha y del ERDF.

English summary

IS IT POSSIBLE TO TRAIN DEEP LEARNING MODELS WITH SYNTHETIC DATA?

Abstract

With the development of the new deep learning techniques, the data demand for training these models has increased. Although a large community has been created around data and even big companies have release their own datasets free of charge, there are specific problems for which training datasets are not available. This is the case of weapon detection in video-surveillance where the early detection of dangerous situations and objects is of vital importance. Several solutions have been proposed in the last years but the data barrier is still a problem. Therefore, in this work we propose to generate video surveillance images with a graphical engine and check if the synthetic data generated can replace collecting and labeling real images.

Keywords: Object detection, aprendizaje profundo, datos sintéticos.

Referencias

[1] Fernando Enriquez de Salamanca Ros, Luis Miguel Soria-Morillo, Juan Antonio Alvarez Garcia, Fernando Sancho Caparrini, Francisco Velasco Morente, Oscar Deniz, and Noelia Vallez. Vision and crowdsensing technology for an optimal response in physical-security. In *SmartSys – ICCS 2018*, 2019.

[2] Darknet. `https://pjreddie.com/`. Accessed: 2019-05-30.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. `http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html`.

[4] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[6] Lumberyard. `https://aws.amazon.com/es/lumberyard`. Accessed: 2019-04-09.

[7] Roberto Olmos, Siham Tabik, and Francisco Herrera. Automatic handgun detection alarm in videos using deep learning. *CoRR*, abs/1702.05147, 2017.

[8] N. Patki, R. Wedge, and K. Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, Oct 2016.

[9] Apoorva Raghunandan, Mohana Mohana, Raghav Pakala, and Ravish Aradhya H V. Object detection algorithms for video surveillance applications. In *IEEE - 7th International Conference on Communication and Signal Processing*, 04 2018.

[10] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[11] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[12] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[13] Ssd en keras y tensorflow. `https://github.com/pierluigiferrari/ssd_keras`. Accessed: 2019-05-30.

[14] Unity. `https://unity.com`. Accessed: 2019-04-09.

[15] Unreal Engine 4. `https://www.unrealengine.com`. Accessed: 2019-04-09.

[16] Pros and cons of machine learning algorithms with fake data. `https://www.ingedata.net/blog/machine-learning-algorithms-fake-data`. Accessed: 2019-05-29.

[17] Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117 – 127, 2017. Image and Video Understanding in Big Data.