Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

Tipo de artículo: Artículos de revisión
Temática: Inteligencia artificial
Recibido: 25/06/2020 | Aceptado: 24/07/2020

Revisión de algoritmos de detección y seguimiento de objetos con redes profundas para videovigilancia inteligente

Algorithms for detection and tracking objects with deep networks for intelligent video surveillance: A review

David Ameijeiras Sánchez[1*] http://orcid.org/0000-0002-6734-7493
Héctor R. González Diez[1] http://orcid.org/0000-0002-7601-4201
Yanio Hernández Heredia[1] http://orcid.org/0000-0001-9433-5511

[1]Universidad de las Ciencias Informáticas (UCI). Km 2 y ½ Autopista La Habana – San Antonio de los Baños, La Habana, Cuba. dameijeiras@estudiantes, {hglez, yhernandezh}@uci.cu

*Autor para la correspondencia. (dameijeiras@estudiantes.uci.cu)

RESUMEN

En la actualidad cada vez más se utilizan las redes neuronales profundas para resolver problemas de visión por computadora, como el reconocimiento y seguimiento de personas a través de una red de cámaras. Se realizó una revisión de los principales algoritmos de rastreo y detección de objetos, basados en redes profundas que permitirían conformar la arquitectura de un sistema de videovigilancia inteligente. Se

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

determinó que: los algoritmos one-stage, son considerablemente más rápidos que los basados en propuestas de regiones, donde destaca SSD, y los algoritmos de rastreo offline tienen una mayor precisión comparado con los online, destacando a DeepSort como el más eficiente.

Palabras clave: Videovigilancia inteligente; Redes neuronales profundas; Clasificación de objetos; Seguimiento de objetos.

ABSTRACT

Today, deep neural networks are increasingly used to solve computer vision problems, such as recognizing and tracking people through a network of cameras. A review of the main algorithms for tracking and object detection, based on deep networks, was carried out, which would make it possible to shape the architecture of an intelligent video surveillance system. It was determined that: one-stage algorithms are considerably faster than those based on region proposals, where SSD stands out, and offline tracking algorithms have a higher accuracy compared to online ones, highlighting DeepSort as the most efficient.

Keywords: Intelligent video surveillance; Deep neural networks; Object classification; Object tracking.

Introducción

En un inicio los sistemas de vigilancia y seguridad solían caracterizarse por la existencia de múltiples cámaras de seguridad de tipo analógicas conectadas a un grupo de monitores, los cuales eran supervisados por uno/varios vigilantes en una única sala. Estos sistemas eran los llamados CCTV (Circuitos Cerrados de Televisión) y servían, por un lado, para tener un registro visual de las incidencias y por otro, para poder detectarlas en tiempo real. Dado que el operario debía vigilar varios escenarios simultáneamente y durante

varias horas seguidas, la probabilidad de que pasara por alto situaciones potencialmente peligrosas resultaba ser alta.

Con la llegada de las cámaras digitales, cambia completamente el mundo de la videovigilancia (video survillance). Disponer de una conexión IP, permite tratar los elementos de la red de videoseguridad como nodos, de esta forma, se pueden realizar transmisiones de video a modo streaming a un PC o servidor donde se almacena y se procesa la información; permitiendo la creación de sistemas inteligentes de videoseguridad. La videovigilancia tiene una amplia variedad de aplicaciones en entornos públicos y privados, como la seguridad nacional, prevención del delito, el control del tráfico, la predicción de accidentes, detección y monitoreo de pacientes, ancianos y niños en el hogar (Wang, 2013), determinar si se cumple el distanciamiento social en zonas públicas, como medida preventiva de la COVID-19. Esta se ha convertido en una de las más activas áreas de investigación de la visión por computadora. La meta es extraer eficientemente información útil de una gran cantidad de videos de cámaras de seguridad. (Sreenu y Saleem Dueai, 2019) definen los siguientes objetivos que ilustran la relevancia del tema: el monitoreo continuo es difícil y agotador para los humanos; se necesita la máxima precisión en la identificación de objetos y el reconocimiento de acciones; mejorar el análisis en multitudes de personas; el tiempo necesario para generar respuestas es muy importante en la situación del mundo real; la predicción de cierto movimiento, acción o violencia es muy útil en caso de emergencia.

Se puede definir para un sistema inteligente de reconocimiento los siguientes pasos : el pre procesamiento , la extracción de rasgos , clasificación y seguimiento de objetos y la respuesta según la comprensión (Sreenu y Saleem Dueai, 2019) siendo la clasificación según (Russakovsky, 2015) la salida de cuadros delimitadores con etiqueta para objetos individuales , y el seguimiento de objetos según (Duffy y Flynn, 2017) el rastreo de uno o varios objetos específicos de interés en una escena determinada, son unas de las herramientas más comunes que se pueden aplicar en los sistemas de videovigilancia (Vassilios y Tasos Dueai, 2018), junto con el reconocimiento de rostro y de acciones humanas.

Los enfoques de aprendizaje profundo se utilizan intensamente durante la última década para abordar algunos de los problemas más desafiantes con respecto al contenido visual, como son los relacionados con los objetos (identificación y rastreo). Las redes neuronales artificiales según (Pal y Shiu, 2004) se encuentran entre las herramientas computacionales que frecuentemente se adoptan para tareas de videovigilancia por sus ventajas

conocidas, como son la adaptabilidad, paralelismo, y aprendizaje; con ello es posible modificar el peso de las conexiones usando algoritmos de entrenamiento o reglas de aprendizaje y con la actualización de los pesos la red puede optimizar sus conexiones adaptándose a cambios en el ambiente (Petrosino y Maddalena, 2012), lo cual constituye una ventaja sobre los enfoques más tradicionales y menos precisos que proponen los autores (Gallehillos y Belongie, 2010); (Lowe y David, 2004); (Jia y Zhan, 2009); (Darwishalzughaibi, y otros, 2015); (Zhansheng, 2013); (E Qing, y otros, 2013). Las arquitecturas de las redes neuronales ahora van lejos del prototipo biológico, pero todavía producen resultados sobresalientes (Fomin y Bakhshiev, 2019).

En los últimos años, la mayoría de los algoritmos de detección de objetos de última generación, como FasterR-CNN (Ren, y otros, 2017), R-FCN (Dai, y otros, 2016), SSD (Liu, y otros, 2016), y YOLO (Redmon, y otros, 2016), han utilizado Redes Neuronales Convolucionales(CNN) y pueden ser desplegados en dispositivos móviles y productos de consumo. Para decidir qué detector se adapta mejor a una cierta aplicación de vigilancia, no solo son las métricas de precisión estándar como la precisión media promedio (mAP) son importantes, también otros factores, como la memoria, el consumo y los tiempos de funcionamiento, juegan un papel crítico (Arcos-García, y otros, 2018).

En la actualidad la tendencia es utilizar algoritmos basados en redes neuronales profundas para clasificar y rastrear objetos, aunque en algunos casos solo se proponen arquitecturas que resuelven uno de estos problemas, o combinan métodos tradicionales con otros de aprendizaje automáticos para lograr un mejor rendimiento.

El objetivo de este artículo es hacer un análisis de los principales algoritmos basados en redes neuronales profundas, que permitan conformar un sistema de videovigilancia inteligente en tiempo real centrándose en la detección y el seguimiento de objetos.

Métodos o Metodología Computacional

Como paso importante de los sistemas de reconocimiento planteados en (Screenu y Saleem Dueai, 2019) la detección y el rastreo de objetos son fundamentales. A pesar de que durante el proceso de rastreo de un determinado objeto es necesario localizarlo, se pueden emplear algoritmos distintos para resolver ambas cuestiones e

integrarlos en una misma arquitectura como la propuesta en (Bathija y Sharma, 2019). El reconocimiento de objetos puede integrar el proceso de seguimiento (Ankit, 2017), y aunque usando solo la detección podemos identificar objetos en cuadros de video, el sistema no será capaz de: conectar los objetos en el cuadro actual a los cuadros anteriores; el objeto que estaba rastreando podría salir de la vista de la cámara durante unos pocos fotogramas y aparecer otro; no se tiene forma de saber si es el mismo objeto.

Aprendizaje automático. Generalidades

La minería de datos como disciplina intenta reducir la brecha existente entre el volumen de información almacenado y el conocimiento que es capaz de extraerse de esta información. El proceso, idealmente, debe ser automático y el objetivo principal es que los patrones descubiertos deben tener algún significado en el sentido de que puedan ser utilizados como conocimiento útil. Dentro de la minería de datos el aprendizaje automático está considerado una disciplina que emplea información histórica formalmente estructurada sobre un fenómeno, para a través de modelos predecir el comportamiento de nuevas variables asociadas al problema (Witten y Frank, 2005); (Friedman, y otros, 2001).

En un escenario de aprendizaje automático convencional el problema queda caracterizado por un conjunto de variables generalmente cuantitativas, categóricas o mezclas de estas definidas en un rango o escala de medición. Estas variables se conocen como variables predictoras o variables de entradas y se caracterizan por ser variables independientes, idénticamente distribuidas iid por sus siglas en inglés. Por otra parte, en un problema de aprendizaje automático deseamos medir una cualidad o propiedad de dicho problema conocida como variable objetivo, variable de salida o simplemente clase. Esta variable objetivo presenta cierta dependencia respecto al conjunto de variables predictoras la cual puede ser descrita por un modelo de aprendizaje automático que sea capaz de caracterizar apropiadamente dicho problema. Para obtener un modelo de aprendizaje automático es necesario contar con datos históricos que caractericen el problema en estudio. Los problemas de aprendizaje automático más comunes se dividen en supervisados o no supervisados dependiendo de la información histórica recopilada sobre el problema en cuestión y el tipo de tarea de que se desee modelar. En el caso de los problemas supervisados necesitan de la información de la clase referida a la

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

muestra de datos históricos recopilados en dicho problema. Por otra parte, los métodos no supervisados son capaces de descubrir las estructuras subyacentes o los patrones que caracterizan los datos. La presente investigación se enmarca en el contexto de los métodos de aprendizaje supervisados.

Redes Neuronales Artificiales

El modelo simplificado y abstracto de las redes neuronales artificiales surge de intentar imitar el comportamiento de las neuronas que se encuentran en el cerebro. En las últimas décadas las Redes Neuronales Artificiales(ANN) han recibido un interés particular como una tecnología para minería de datos, puesto que ofrece los medios para modelar de manera efectiva y eficiente problemas grandes y complejos. Los modelos de ANN son dirigidos a partir de los datos, es decir, son capaces de encontrar relaciones (patrones) de forma inductiva por medio de los algoritmos de aprendizaje basado en los datos existentes más que requerir la ayuda de un modelador para especificar la forma funcional y sus interacciones.

Para explicar el funcionamiento de las redes neuronales se comenzará describiendo la red neuronal más simple que existe, una red formada por una única neurona y una única entrada. El esquema de esta red es el representado en la Figura 1.

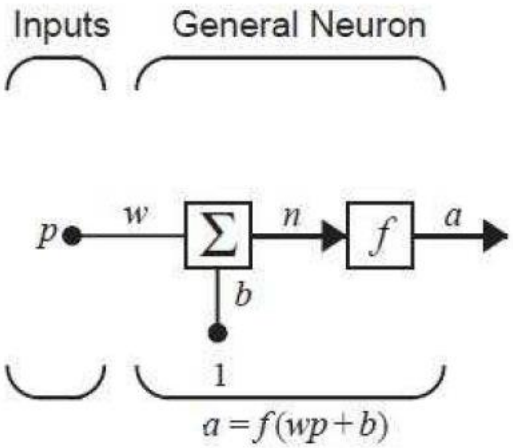

Fig. 1 – Red neuronal de una única entrada y una única salida.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

El siguiente proceso es simple. A la entra llega un escalar *p* que es multiplicado por su peso *w*, quedando $w * p$, que es una de las entradas de la sumatoria. La siguiente entrada siempre tiene valor 1 y es multiplicado por un bias. Una vez ambos se han sumado se obtiene la entrada *n* a la función de transferencia $w * p + b$. Finalmente se tiene la salida de la función *a* de la neurona:

$$a = f(wp + b)$$

Teniendo esto en cuenta, se puede observar que la salida del sistema depende del tipo de función de transferencia que se tenga. Es importante conocer que la función de transferencia la decide el diseñador y que los parámetros *w* y *b* son ajustables y calculados mediante una regla de aprendizaje (Suárez, 2017).

Usualmente con una única neurona no será suficiente para resolver la mayoría de problemas prácticos. Para resolver problemas más complejos se tendrá que hacer un uso conjunto de muchas de estas neuronas simples, dando lugar a una verdadera red neuronal, en la que se tendrán cientos o incluso miles de neuronas. Es ahí donde aparece el concepto de capa, que es la agrupación de todas estas neuronas en varios conjuntos dentro de la red neuronal completa (Suárez, 2017) como se muestra en la Figura 2.

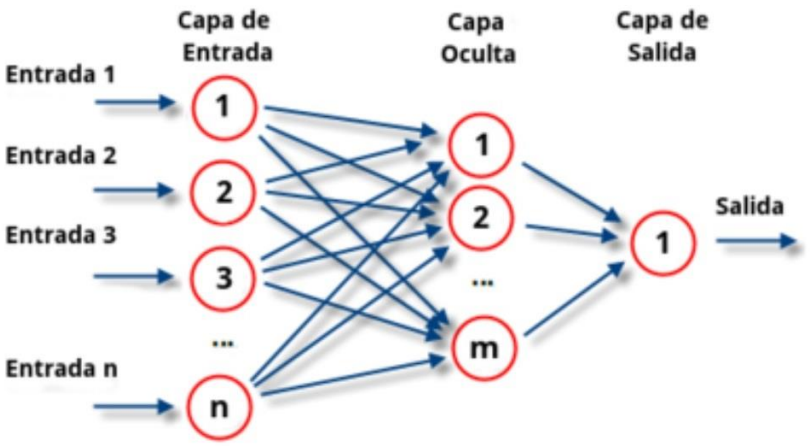

Fig. 2 – Esquema de una red neuronal de tres capas.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

El aprendizaje es la clave de la adaptabilidad de la red neuronal y esencialmente es el proceso en el que se adaptan las sinapsis, para que la red responda de un modo distinto a los estímulos del medio. Entrenar una red neuronal es un proceso que modifica los pesos y bias asociados a cada neurona, con el fin de que la red pueda a partir de datos de entrada, generar una salida.

Las ANN son un método de resolver problemas, de forma individual o combinadas con otros métodos, para aquellas tareas de clasificación, identificación, diagnóstico, optimización o predicción en las que el balance datos/conocimiento se inclina hacia los datos y donde, adicionalmente, puede haber la necesidad de aprendizaje en tiempo de ejecución y de cierta tolerancia a fallos. En estos casos las RNAs se adaptan dinámicamente reajustando constantemente los pesos de sus interconexiones (Salas, 2004).

Redes Neuronales Convolucionales

Una red neuronal convolucional según (Suárez, 2017) es un tipo de red multicapa que consta de diversas capas convolucionales y de pooling (submuestreo) alternadas, y al final tiene una serie de capas full-connected como una red perceptrón multicapa, como se muestra en la Figura 3.

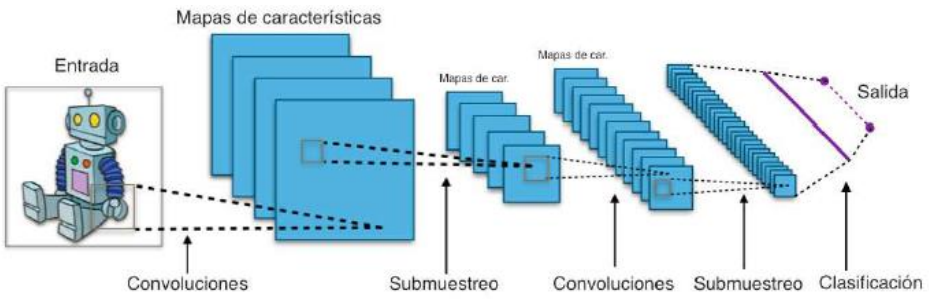

Fig. 3 – Esquema de una red neuronal convolucional.

La entrada de una red capa convolucional suele ser, generalmente, una imagen $m * m * x * r$, donde m es tanto la altura como el ancho de la imagen y r es el número de canales. Las capas convolucionales tienen l filtros (o kernels) cuyas dimensiones son $n * n * q$, donde n y q son elegidas por el diseñador (generalmente q suele

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

ser igual a r). Cada filtro genera mediante convolución un mapa de rasgos o características de tamaño $(m - n + 1) * (m - n + 1) * q$, siendo p el número de filtros que se desean usar. Después cada mapa es sub-muestreado en la capa de pooling con la operación mean pooling o max pooling sobre regiones contiguas de tamaño $p * p$ donde p puede tomar valores desde 2 para imágenes pequeñas hasta, comúnmente, no más de 5 para imágenes grandes. Antes o después del submuestreo, se aplica una función de activación sigmoidal más un sesgo para cada mapa de rasgos.

Las redes convolucionales están diseñadas suponiendo que la entrada a la red es una imagen; permite codificar ciertas propiedades en la arquitectura. Son muy potentes para el análisis de imágenes, debido a su capacidad de detectar rasgos simples como líneas y bordes.

Algoritmos de Reconocimiento de Objetos

La detección de objetos genéricos, también llamada detección de categorías de objetos genéricos, detección de clases de objetos o detección de categorías de objetos (Zhang, y otros, 2013), se define de la siguiente manera: dada una imagen, determinar si hay o no instancias de objetos de categorías predefinidas (normalmente muchas categorías, por ejemplo, 200 categorías en el desafío de detección de objetos del ILSVRC) y, si las hay, devolver la ubicación espacial y la extensión de cada instancia. Se hace mayor hincapié en la detección de una amplia gama de categorías naturales, en contraposición a la detección de categorías de objetos específicos, en la que solo puede estar presente una categoría de interés predefinida más estrecha (por ejemplo, rostros, peatones o automóviles). Aunque miles de objetos ocupan el mundo visual en el que vivimos, en la actualidad la comunidad de investigadores está interesada principalmente en la localización de objetos altamente estructurados (por ejemplo, automóviles, rostros, bicicletas y aviones) y objetos articulados (por ejemplo, seres humanos, vacas y caballos) más que en escenas no estructuradas (como el cielo, la hierba y las nubes) (Liu, y otros, 2020).

A continuación, se realiza un análisis de los principales algoritmos de reconocimiento de objetos basados en redes neuronales profundas, enfocado en el uso de los mismos para un sistema de videovigilancia.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

Regression/Classification Based Framework (YOLO, SDD)

Los marcos basados en propuestas regionales están compuestos por varias etapas correlacionadas, incluyendo la generación de propuestas de la región, extracción de características con CNN, clasificación y caja delimitadora de regresión, que normalmente se entrenan por separado. Incluso en recientes módulo de extremo a extremo, más rápido que R-CNN, como resultado, el tiempo pasado en la manipulación de diferentes componentes se convierte en el cuello de botella en tiempo real aplicación.

You only look once (Redmon, y otros, 2016)

YOLO consiste en una sola red convencional que predice simultáneamente múltiples cuadros delimitadores y probabilidades de clase para esos cuadros. Se entrena con imágenes completas. Es extremadamente rápido en comparación con otros métodos tradicionales, obteniendo aproximadamente 45 fps, en su primera versión (procesada en una Titan X GPU) y en una versión más rápida más de 150 fps. Esto significa que sería posible procesar la transmisión de video en tiempo real con menos de 25 milisegundos de latencia (Redmon, y otros, 2016) con óptimas prestaciones computacionales. Yolo logra también el doble de precisión de la media promedio de otros sistemas en tiempo real. Todo el código de entrenamiento y prueba es código abierto, y existe una variedad de modelos pre entrenados que pueden ser descargados.

El sistema (Redmon, y otros, 2016) modela la detección como un problema de regresión. Divide la imagen en una cuadrícula de S x S y para cada celda de la cuadrícula predice los cuadros delimitadores *B*, la confianza para cada uno de esos cuadros, y probabilidad de ser una clase C. Estas predicciones están codificadas como: S x S x ($5B + C$) tensor.

La arquitectura de la red del modelo está basada en GoogLeNet (modelo para la clasificación de imágenes (Santos, 2013). La red cuenta con 24 capas convolucionales seguidas de 2 capas con conecciones enteras (fully connected). Se utilizan 1 x 1 capas de reducción seguidas de $3 * 3$ capas convolucionales, similar a (Lin, y otros, 2016). También existe una versión más pequeña de YOLO pensado para lograr una mayor velocidad (con 9 capas en lugar de 24 y menos filtros en esas capas).

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

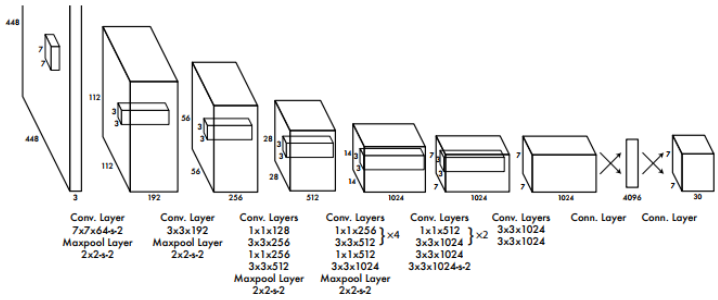

Fig. 4 – Arquitectura de red convolucional LeNet.

Se optimizó la suma de errores cuadrados para la salida del modelo. Se aumentó la pérdida de las cajas delimitadoras de predicción de coordenadas y se disminuyó la pérdida de las predicciones para las cajas que no contenían objetos.

El error de la suma al cuadrado también pondera los errores en grandes cajas y cajas pequeñas. La métrica de error debería reflejar que las pequeñas desviaciones en las grandes cajas importan menos que en las pequeñas cajas. Para abordar parcialmente esto se predice la raíz cuadrada de la anchura del cuadro delimitador en lugar de la anchura y la altura directamente (Redmon, y otros, 2016).

YOLO predice múltiples cajas delimitadoras por cada celda de la cuadrícula. En el entrenamiento un predictor de caja delimitadora es responsable de cada objeto. Se asigna un predictor basado en cuál es la predicción que tiene mayor valor IOU con la zona correcta. Esto lleva a la especialización entre los predictores de cada caja delimitadora. Cada pronosticador mejora en la predicción de ciertos tamaños, relaciones de aspecto o clases de objetos, mejorando en general (Redmon, y otros, 2016), siendo la función de pérdida:

$$\lambda coord \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x})^2 + (y_i - \hat{y}_i)^2 \right] + \lambda coord \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 + \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

Donde 1*obji* denota si el objeto aparece en la celda *i* y 1*objij* denota la *j* cuadro delimitador predictor en l 1celda *i*, que es responsable de esa predicción (Redmon, y otros, 2016).

-Existen cuatro versiones de dicho algoritmo: You Only Look Once: Unified, Real-Time Object Detection (Yolov1) (Redmon, y otros, 2016); YOLO9000: Better, Faster, Stronger (Yolo v2) (Redmon, y Farhadi, 2017); YOLOv3: An Incremental Improvement (Yolo v3) (Redmon, y Farhadi, 2018); YOLO v4: Optimal Speed and Accuracy for object detection (Yolo v4) (Bochkovskiy, y otros, 2020) y tres implementaciones:

1. Darknet (https://pjreddie.com/darknet/): Es la implementación oficial de YOLO, creada por las mismas personas detrás del algoritmo. Está escrito en C con CUDA, permitiendo el computo en GPU. Se puede usar para el trabajo con distintos modelos de redes neuronales, además de YOLO.
2. AlexeyAB/darknet (https://github.com/AlexeyAB/darknet): Es un fork de Darknet original para ser utilizarlo en Windows y Linux.
3. Darkflow (https://github.com/thtrieu/darkflow/): Es un port de Darknet a TensorFlow. Permite llevar la configuración y los pesos de una red en el formato original de YOLO a TensorFlow.

El modelo presenta las siguientes limitaciones según: YOLO funciona mal en objetos pequeños, particularmente aquellos que aparecen en grupos, ya que cada celda de la grilla predice solo B = 2 cuadros delimitadores de la misma clase (Aidouni y Manal, 2019). Aunque la función de pérdida predice la raíz cuadrada de la altura y el peso del cuadro delimitador en lugar de la altura y el peso directamente en un intento de resolver la ecualización de error para cuadros grandes y pequeños, esta solución soluciona parcialmente el problema, lo que resulta en un número significativo de errores de localización. Además, YOLO no detecta correctamente los objetos de formas nuevas o poco comunes, ya que no se generaliza mucho más allá de los cuadros delimitadores en el conjunto de entrenamiento. Para utilizar YOLO en un sistema de videovigilancia en tiempo real es necesario prestaciones computacionales por encima de la media, en su versión ligera la disminución de la precisión lo hace ineficiente en problemas de este tipo.

Como principales ventajas de YOLO: utiliza muchos menos cuadros delimitadores, solo 98 por imagen, en comparación con unos 2000 de Selective Search (Liu, y otros, 2020). Al predecir las coordenadas del cuadro delimitador directamente de las imágenes de entrada, YOLO trata la detección de objetos como un problema de regresión, a diferencia de los métodos basados en clasificadores (Redmon, y otros, 2016). La red refrescantemente simple de YOLO está entrenada conjuntamente y permite 45 fps de predicción de velocidad en tiempo real. Además, dado que predice simultáneamente cuadros delimitadores de toda la imagen en todas las clases, la red razona globalmente sobre todos los objetos en la imagen. Es capaz de detectar muchos objetos en una imagen. Existen implementaciones de YOLOv3 en otros frameworks.

Single Shot Detector (SDD) (Liu, y otros, 2016)

YOLO tiene dificultades para tratar con pequeños objetos en grupos, lo cual es causado por fuertes limitaciones espaciales impuestas a las predicciones de las cajas. Con el fin de resolver estos problemas, (Liu, y otros, 2016) propusieron un Single Shot Detector. Dado un mapa de características específicas, en lugar de las redes fijas adoptadas en YOLO, el SSD aprovecha un conjunto de cajas de anclaje con diferentes radios y escalas para discretizar el espacio de salida de las cajas delimitadoras. Para manejar objetos de varios tamaños, la red fusiona las predicciones de múltiples mapas de características con diferentes resoluciones. Dado la arquitectura de la red troncal VGG16, SSD añade varias características capas hasta el final de la red, estas son responsables de las predicciones para el balance de las cajas delimitadoras con distintas escalas y aspectos de radio, al igual que sus valores de confianza asociados.

Después de pasar por una serie de convolución para la extracción de características, obtenemos una capa de características de tamaño $m * n$ (número de ubicaciones) con canales p, como 8 x 8 o 4 x 4 arriba. Y se aplica una convolución de 3x3 en esta capa de extracción de características m x n x p (Liu, y otros, 2016).

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

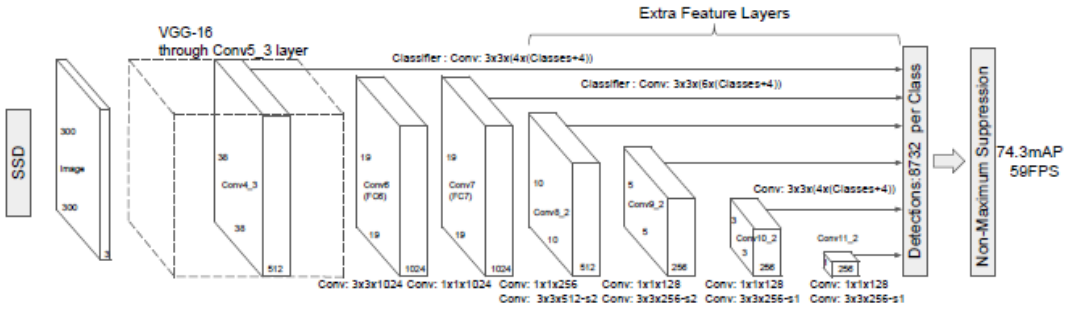

Fig. 5 – Arquitectura de SDD.

Para cada ubicación, tenemos cajas delimitadoras k. Estas tienen diferentes tamaños y relaciones de aspecto. El concepto es, tal vez un rectángulo vertical es más adecuado para el humano, y un rectángulo horizontales m´as adecuado para el objeto a detectar es un automóvil (Liu, y otros, 2016).

Para cada uno de los cuadros delimitadores, calcularemos las puntuaciones de la clase *c* y 4 desplazamientos relativos a la forma original del cuadro delimitador por defecto. Por lo tanto, tenemos (*c*+ 4)*kmn* elementos de salida (Liu, y otros, 2016).

La red es entrenada con una suma ponderada de pérdida de localización:

$$L_{(x,c,l,g)=\frac{1}{N}}(L_{conf}(x\,,c) + \alpha L_{loc}(x,l,g))$$

La función está integrada por dos términos: L_{conf} y L_{loc} donde N es la intersección de las cajas predeterminadas.

$$L_{loc}(x,l,g) = \sum_{ii\in Pos}^{N} \sum_{m\in\{cx,cy,w,h\}} \varphi_1(x_{ij}^k(l_i^m - \hat{g}_j^m))$$

donde las funciones de estimación ĝ quedarán expresadas como:

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

$$\hat{g}_j^w = \log(\frac{g_j^w}{d_i^w}) \qquad \hat{g}_j^h = \log(\frac{g_j^h}{d_i^h})$$

L_{loc} es la pérdida que corresponde a la localización que la pérdida suave de φ_1 entre los parámetros del cuadro de predicción (I) y los parámetros del cuadro correcto (g). Estos parámetros incluyen las compensaciones para el punto central (*cx*, *cy*), el ancho *w* y la altura *h* del cuadro delimitador. Esta pérdida es similar a la mejor en Faster R-CNN.

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p \log(\hat{c}_i^0) + \sum_{i \in Neg} log(c_i^0)$$

donde:

$$\hat{c}_i^p = \frac{exp\ (c_i^p)}{\Sigma_p \exp(c_i^p)}$$

El SSD supera significativamente al R-CNN más rápido en términos de precisión en PASCAL VOC y COCO, siendo tres veces más rápido. El SSD300 (el tamaño de la imagen de entrada es 300∗300) funciona a 59 FPS, que es más preciso y eficiente que el YOLO. Sin embargo, el SSD no es hábil en el manejo de objetos pequeños, que puede aliviarse adoptando un mejor extractor de características, por ejemplo, (ResNet101) (Zhao, y otros, 2019).

Al igual que YOLO tiene problemas en la detección de objetos, con resultados inferiores a otras arquitecturas en dos etapas como Faster RCNN y RFCN, pero resulta competitivo en objetos medianamente grandes.

Los detectores como YOLO (Redmon, y otros, 2016) y SDD (Liu, y otros, 2016) one-stage son generalmente más rápidos que los de dos etapas, debido a que evitan los algoritmos de preprocesamiento, realizan las predicciones con

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

menos regiones candidatas y hacen que la subred de clasificación sea totalmente convolutiva. En cualquier caso, la que más tiempo consume según (Law y Deng, 2018); (Ren, y otros, 2015) es el extractor de características.

Region Proposal Based Frameworks (RCNN, SPPnet, Fast R-CNN, Faster R-CNN, R-FCN, Mask R-CNN)

Region Proposal Based Framework, es un proceso de dos etapas, coincide con el mecanismo de atención del cerebro humano, lo que da una visión general de todo el escenario en primer lugar y luego se centra en las regiones de interés. Estos algoritmos son más lentos a la hora de procesar video en tiempo real comparados con algoritmos como YOLO y SSD, siendo prácticamente ineficientes para un sistema de video vigilancia en tiempo real, aunque en algunos escenarios de videoseguridad, podrían sacar más provecho como es el caso de (Fernandez-Carrobles, y otros, 2019); (Zhang, y otros, 2017); (Wei y Kehtarnavaz, 2019) por su alta precisión, lidiando mejor con objetos pequeños.

A continuación, se enuncian las ventajas y las desventajas de estos algoritmos con el extractor de características con el que se obtiene mayor velocidad a la hora de procesar video en tiempo real, todos los datos son a partir de las pruebas hechas sobre VOC2007, VOC2012 y COCO con una sola Nvidia Titan X GPU.

Faster-RCNN (Ren, y otros, 2015)

FasterRCNN es una red que hace detección de objetos. Como su nombre lo explica, es más rápida que sus descendientes RCNN y FastRCNN. Esta red tiene casos de uso en autos auto-conducción, manufactura, seguridad, e incluso se usa en Pinterest.

El algoritmo pasa la imagen por una CNN para obtener un mapa de características. Ejecuta el mapa de activación a través de una red separada (RPN), que produce regiones de interés. Para cada región de RPN se usan varias capas completamente conectadas a las salidas junto con la coordenada de la caja de enlace (Ren, y otros, 2015).

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

La arquitectura Fast R-CNN, esencialmente disminuye la carga computacional, respecto a CNN, y por esta razón disminuye el tiempo de detección que presenta la capa R-CNN. En consecuencia, Fast R-CNN junto con Búsqueda Selectiva, presenta una mejor calidad de detección. Sin embargo, ambos métodos necesitan de un generador de ROI externo y tienen problemas al momento de detectar objetos pequeños. Por ejemplo, no detectaría peatones a largas distancias (Galarza, y otros, 2018).

Para remediar estos inconvenientes se ha llegado a Faster R-CNN, que añade un generador de ROI basado en capas completamente conectadas RPN el cual comparte con Fast R-CNN, los mapas de características generados por la red convolucional. Por ende, se pueden implantar redes muy profundas debido a que la imagen total pasa una sola vez por la etapa CNN (Ren, y otros, 2015).

FasterRCNN resuelve el cuello de botella de tener que ejecutar la Búsqueda Selectiva para cada imagen como primer paso. La mayoría de las implementaciones de la Búsqueda Selectiva están en la CPU (lenta). La base de código implementa FasterRCNN tanto con Resnet101 como con VGG16 (Ren, y otros, 2015).

Para la regresión, se adoptan las parametrizaciones de las 4 coordenadas siguientes:

$$t_x = (x - x_a)/w_a \qquad t_y = (y - y)/h_a \qquad t_w = (log(w - w_a)) \quad t_h = (log(h - h_a))$$

$$t_x^* = (x^* - x_a)/w_a \qquad t_y^* = (y^* - y)/h_a \qquad t_w^* = (log(w^* - w_a)) \quad t_h^* = (log(h^* - h_a))$$

Donde *x, y, w,* y *h* denotan las dos coordenadas del centro de la región, ancho y alto. Variables *x, xa* y *x* son para la región de predicción, la región de anclaje y la región de aciertos respectivamente (igualmente para *y, w, h*). Esto puede verse como una regresión de la región delimitadora de una región de anclaje a una cercana región de acierto (Ren, y otros, 2015).

Unifica la RPN y la RCNN rápida en una sola red compartiendo capas de CONV. Obtuvo 73.2 mAP en VOC0770.4 en VOC12. Un orden de magnitud más rápida que la Faster-RCNN sin pérdida de rendimiento.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

Puede ejecutar pruebas a 5 FPS con VGG16. El entrenamiento es complejo. Su velocidad en tiempo real es inferior 5 FPS.

RFCN (Dai, y otros, 2016) con ResNet101

En el caso de los enfoques tradicionales de la red de propuestas regionales (RPN), como la R-CNN, la R-CNN rápida y la R-CNN mas rápida, las propuestas regionales se generan primero por la RPN. Luego se hace un pool de ROI, y se pasa a través de capas completamente conectadas (FC) para la clasificación y la regresión del cuadro delimitador (Dai, y otros, 2016).

El proceso (capas FC) después de la agrupación del ROI no se comparte entre el ROI, y lleva tiempo, lo que hace que los enfoques RPN sean lentos. Y las capas FC aumentan el número de conexiones (parámetros), lo que también aumenta la complejidad (Dai, y otros, 2016).

En el R-FCN, todavía tenemos RPN para obtener propuestas de regiones, pero a diferencia de la serie R-CNN, las capas de FC después de la puesta en común del ROI se eliminan. En su lugar, toda la complejidad principal se mueve antes de la agrupación de ROI para generar los mapas de puntuación. Todas las propuestas de regiones, después de la puesta en común del ROI, harán uso del mismo conjunto de mapas de puntuación para realizar la valoración media, que es un cálculo simple. Por lo tanto, no hay una capa de aprendizaje después de la capa de retorno de la inversión que es libre de costo. Como resultado, el R-FCN es incluso más rápido que el R-CNN más rápido (Dai, y otros, 2016).

Cuando se agrupa el ROI, $(C+ 1)$ (C categorías de objetos + 1 por el fondo) se producen mapas de características con el tamaño de k^2, es decir, $k^2(C+ 1)$. El agrupamiento se hace en el sentido de que se agrupan con la misma área y el mismo color en la figura. La media de la puntuación se realiza para generar $(C + 1)$ − *dimensión al vector*. Y finalmente se realiza el softmax en el vector (Dai, y otros, 2016).

En la capa convolucional $k^2(C+ 1)$ −d, se añade una capa convolucional paralela de $4k^2−d$. En este banco de mapas de $4k^2$ se realiza un pool de Rol sensible a la posición, produciendo un vector de $4k^2−d$ para cada uno.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

Luego se agrega en un vector $4-d$ por valoración media, que representa *tx*, *ty*, *tw*, *th* (posición y tamaño) del cuadro delimitador, que es el mismo que en el Fast R-CNN (Dai, y otros, 2016).

La función de pérdida es la siguiente:

$$L\left(s, t_{(x,y,w,h)}\right) = L_{cls}\left(s(c^*)\right) + \lambda[c^* > 0]L_{reg}(t, t^*)$$

L_{cls} es la pérdida de clasificación y L_{reg} es la pérdida de regresión del cuadro delimitador. De manera positiva esta red crea un conjunto de mapas de puntuación sensibles a la posición usando un banco de capas convolucionales especializadas; más rápido que la Faster-RCNN. Obtuvo 80.5 mAP en VOC0777.6 en VOC12. Pueden existir pequeñas fallas cuando se utiliza en tiempo real. Puede ejecutar pruebas a 5fps con VGG16. El entrenamiento es complejo. Su velocidad en tiempo real es inferior a 10 fps

RCNN (Girshick, 2015) con Alexnet

El primero en integrar la CNN con los métodos de RP. Mejora considerablemente el rendimiento sobre el estado del arte en ese momento. Publicado en CVRP14. Obtuvo 58.5 mAP en VOC07 53.3 en VOC12. Un proceso multietapa de entrenamiento secuencial (cálculo externo de RP, ajuste fino de CNN, cada RP modificado pasa por un entrenamiento de CNN, SVM y BBR). El entrenamiento es costoso en el espacio y en el tiempo; las pruebas son lentas. Su velocidad en tiempo real es inferior a 0.1 FPS.

SPPNet (He, y otros, 2014) con ZFnet

La primera en introducir el SPP en la arquitectura de la CNN. Permite compartir características convolucionales. Acelera la evaluación de la RCNN por órdenes de magnitud sin sacrificar el rendimiento. Obtuvo 60.9mAP en VOC07. Hereda las desventajas de la RCNN. No da como resultado mucha velocidad de entrenamiento. El ajuste fino no puede actualizar las capas de CONV antes de la capa de SPP. Su velocidad en tiempo real es inferior a 1 FPS.

Mask RCNN (He, y otros, 2017) con ResNet101

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

Un simple, flexible, y efectivo frameworks para segmentación. Hereda de Faster-RCNN y adiciona otra rama para la predicción de la máscara de objetos paralelos con una rama existente. Obtuvo 50.3 en COCO Result. Se utiliza Feature Pyramid Network (FPN). Presenta fallos pequeños en aplicaciones en tiempo real (He, y otros, 2017). Su velocidad en tiempo real es inferior a 5 FPS.

Algoritmos de Seguimiento de Objetos

El rastreo de objetos tiene una amplia gama de aplicaciones en la visión por computadora, como la vigilancia, la interacción entre el hombre y la computadora, y la obtención de imágenes médicas, la vigilancia del flujo de tráfico, el reconocimiento de la actividad humana, si la seguridad de un país quiere rastrear a un criminal que huye en un coche usando las cámaras de vigilancia de la ciudad.

Muchos algoritmos de rastreo tradicionales (no basados en el aprendizaje profundo) están integrados en la API de rastreo de OpenCV. La mayoría de estos rastreadores no son muy precisos comparativamente. Sin embargo, algunas veces pueden ser útiles para ejecutarse en un entorno con restricciones de recursos como un sistema integrado como lo son los propuestos por (Sun, 2012); (Liu, y otros, 2009); (Rong y Jilkov, 2005); (Ding y Li, 2013). Uno de los más usados de este tipo es *Kernelized Correlation Filters* (KCF) *tracker* (Henriques, y otros, 2015).

A continuación, se realiza un análisis de los principales algoritmos de seguimiento de objetos basados en redes neuronales profundas, enfocado en el uso de los mismos para un sistema de videovigilancia.

Offline Trackers

Los rastreadores fuera de línea se utilizan cuando se tiene que rastrear un objeto en un flujo grabado. Se pueden usar los fotogramas pasados y también los futuros para hacer predicciones de seguimiento más precisas.

El entrenamiento de estos rastreadores solo ocurre fuera de línea. A diferencia de los rastreadores de aprendizaje en línea, estos no aprenden nada durante el tiempo de ejecución. Estos rastreadores aprenden los

conceptos completos fuera de línea, es decir, podemos entrenar un rastreador para identificar personas. Entonces estos rastreadores pueden ser usados para rastrear continuamente a todas las personas en un flujo de video.

Para tareas de videovigilancia no serían factibles; al necesitar una secuencia de video completa no funcionarían en un sistema de videovigilancia en tiempo real.

Online Trackers

Los rastreadores en línea se utilizan cuando las predicciones están disponibles de inmediato y, por lo tanto, no pueden utilizar futuros marcos para mejorar los resultados.

Estos rastreadores normalmente aprenden sobre el objeto a rastrear usando el marco de inicialización y algunos marcos subsecuentes. Por lo tanto, estos rastreadores son más generales porque puedes dibujar un cuadro delimitador alrededor de cualquier objeto y rastrearlo. Por ejemplo, si se quiere seguir a una persona con camisa azul en el aeropuerto, se puedes dibujar un cuadro delimitador alrededor de esa persona en uno o varios cuadros. El rastreador aprendería sobre el objeto usando estos marcos y continuaría rastreando a esa persona; siendo estos algoritmos muy útiles para la creación de un sistema de videovigilancia en tiempo real.

Multi domain network (Nam y Han, 2013).

Multi domain network (MDnet) fue el algoritmo ganador del desafío VOT2015. Las redes neuronales convolucionales son computacionalmente muy costosas de entrenar, estos métodos tienen que utilizar una red más pequeña para entrenar a alta velocidad durante el despliegue. Sin embargo, las redes más pequeñas no tienen mucho poder de diferenciación. Una opción es que se entrene toda la red, pero durante la inferencia, se usan las primeras capas como extractor de características, es decir, solo se cambia el peso del último par de capas que se entrenan en línea. Así que se tiene a las CNN como extractor de características y las últimas capas pueden ser entrenadas rápidamente en línea. El objetivo es entrenar un CNN genérico multi dominio que pueda distinguir entre el objetivo y el fondo. Sin embargo, esto plantea un problema durante el

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 3, Mes Julio-Septiembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 165-195

entrenamiento, el objetivo en un video podría ser el fondo, situación que solo confundiría la red neural convolucional. Así que, MDNet hace algo inteligente: reorganiza la red en dos partes: la primera parte es la parte compartida y luego hay una parte que es independiente para cada dominio. Cada dominio significaría un video de entrenamiento independiente. Así que, la red se entrena primero en los dominios K de forma iterativa donde cada dominio clasifica entre su objetivo y su fondo. Esto ayuda a extraer la información independiente del video para aprender una mejor representación genérica del rastreador (Nam y Han, 2013).

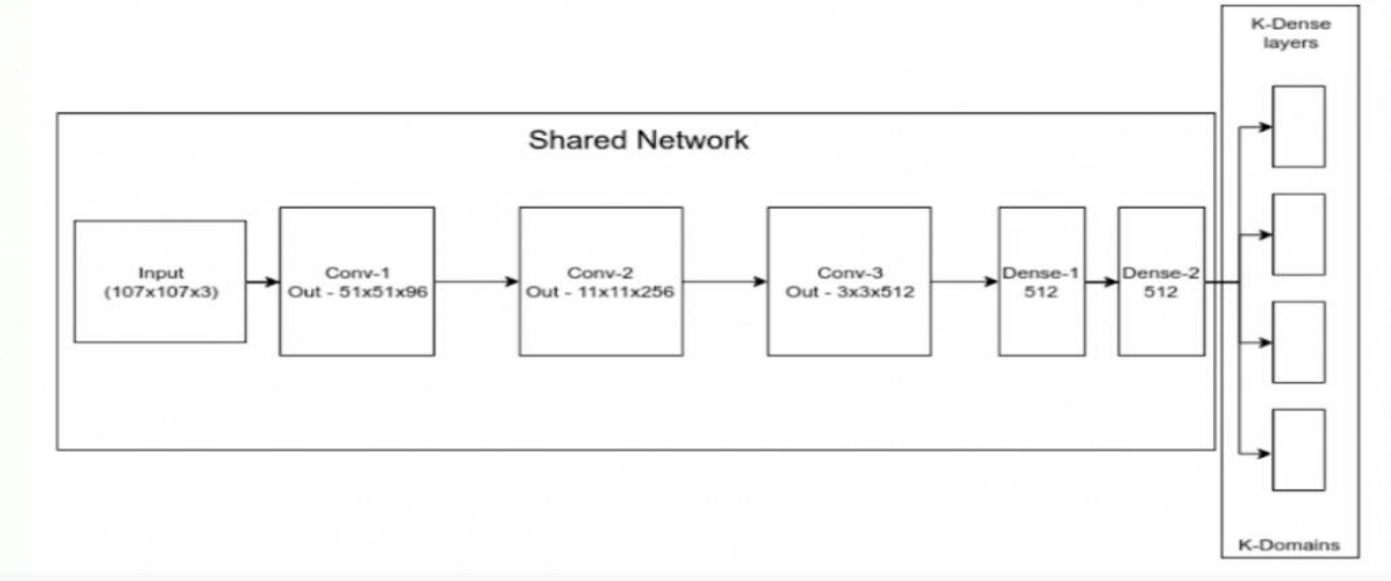

Fig. 6 – Arquitectura de Multi domain network.

Después del entrenamiento, se eliminan las capas binarias específicas del dominio y se obtiene un extractor de características que puede distinguir entre cualquier objeto objetivo y el fondo de forma genérica. Durante la Inferencia (producción), la parte compartida inicial se utiliza como extractor de características y se eliminan las capas específicas del dominio y se añade una capa de clasificación binaria encima del extractor de características. Esta capa de clasificación binaria se entrena en línea. En cada paso, se busca el objeto en la región alrededor del estado objetivo anterior mediante un muestreo aleatorio (Nam y Han, 2013). MDNet es uno de

los entrenamientos en línea basados en aprendizaje profundo más precisos, libre de detección, rastreador de un solo objeto.

Recurrent YOLO

El YOLO recurrente (ROLO) es un algoritmo de rastreo basado en la detección de un solo objeto, en línea. Utiliza la red YOLO para la detección de objetos y una red LSTM para encontrar la trayectoria del objeto objetivo. Hay dos razones por las que LSTM con CNN es una combinación muy eficiente.

1. Las redes de LSTM son particularmente buenas en el aprendizaje de patrones históricos, por lo que son particularmente adecuadas para el seguimiento visual de objetos.
2. b) Las redes LSTM no son muy costosas desde el punto de vista computacional, por lo que es posible construir muy rápidamente rastreadores del mundo real.

En ROLO los marcos de entrada van a través de la red YOLO. De la red YOLO se toman dos salidas diferentes (características de la imagen y coordenadas del cuadro delimitador). Estos dos productos se entregan a la red del LSTM. El LSTM emite las trayectorias, es decir, el cuadro delimitador del objeto a rastrear (Ning, y otros, 2017).

La interpretación preliminar de la ubicación de YOLO ayuda al LSTM a prestar atención a ciertos elementos visuales. ROLO explora la historia espacio-temporal, es decir, junto con la historia de la localización, ROLO también explota la historia de los elementos visuales. Incluso cuando la detección de YOLO es defectuosa debido a la borrosidad del movimiento, el seguimiento de ROLO se mantiene estable. Además, estos rastreadores son menos propensos a fallar cuando el objeto objetivo está ocluido (Ning, y otros, 2017).

The Simple Online and Realtime Trackinq (Bewley, y otros, 2016)

El algoritmo Simple Online and Realtime Tracking ha sido uno de los primeros conductos de MOT en aprovechar redes neuronales convolutivas para la detección de peatones. (Bewley, y otros, 2016) mostraron que la sustitución de las detecciones obtenidas usando las Características de Canal Agregado (ACF) (Piotr, y otros, 2014) con detecciones calculadas por el R-CNN más rápido podría mejorar la puntuación del MOTA en un 18,9 %

(cambio absoluto) en el conjunto de datos de la MOT15 (Leal-Taixe, y otros, 2015). Se utilizó un método relativamente simple que consistía en predecir el movimiento de los objetos usando el filtro de Kalman (Kalman, 1960) y luego asociar las detecciones junto con la ayuda del algoritmo Hungarian (HaroldWKuhn, 1955), usando distancias de intersección-sobre-unión (IoU) para calcular la matriz de costos. En el momento de la publicación, SORT fue clasificado como el algoritmo de código abierto de mejor rendimiento en el conjunto de datos MOT15 (Ciaparrone, y otros, 2019).

The Simple Online and Realtime Tracking with a Deep Association Metric (DeepSort) (Nicolai y Bewley, 2017).

Es una extensión del algoritmo Simple Online and Realtime Tracking, donde se le integra una asociación métrica. Debido a esto el algoritmo es capaz de rastrear objetos a través de un período de oclusión reduciendo el número de cambios de identidad. Durante la aplicación en línea se establece una medida para rastrear usando el vecino más cercano. Los experimentos mostraron una reducción del número de cambios de identidad en un 45 por ciento (Nicolai y Bewley, 2017).

Conclusiones

Los algoritmos one-stage, son considerablemente más rápidos que los basados en propuestas de regiones, aunque tienen dificultades para detectar objetos pequeños, sobre todo si están agrupados. Entre los analizados SSD supera a YOLO en velocidad y precisión en todos los escenarios (incluyendo objetos pequeños). Serían más factibles en sistemas de vigilancia en entornos controlados como pasillos, cuartos, almacenes, bancos y parqueos.

Algoritmos como YOLO y SSD tienen más adaptabilidad; pueden ser usados con menos recursos computacionales (YOLO en su versión reducida y SSD utilizando Mobilenet como extractor de

características), en estos casos, sacrificando precisión, pudiéndose ejecutar en dispositivos libres de GPU, como celulares de gama media baja y Rasperry Pi.

Algoritmos como Faster-RCNN y RFCN son muy precisos, dependiendo del extractor de características pueden utilizarse para funciones de videovigilancia, especialmente en el preprocesamiento de video, por su capacidad de detectar objetos pequeños, aunque no alcanzan velocidades que le permitan ser utilizados en sistemas en tiempo real, son muy eficientes detectando posibles amenazas en imágenes, por ejemplo, identificando potenciales objetos peligrosos como armas durante los escaneos a equipajes en los aeropuertos. El aprendizaje en el paso de detección puede mejorar considerablemente el rendimiento de un algoritmo de rastreo. Las CNN son esenciales en la extracción de características: el uso de características de apariencia también es fundamental para un buen rastreador y las CNN son particularmente eficaces en su extracción. Además, los rastreadores fuertes tienden a usarlos junto con características de movimiento, que pueden ser computadas usando LSTM, filtro Kalman.

Los algoritmos offline funcionan mejor que los algoritmos en línea; usan toda la secuencia de video, aunque en aplicaciones de videovigilancia es más factible usar los segundos, especialmente si el sistema funciona en tiempo real, dejando los algoritmos offline para tareas de preprocesamiento, como rastrear el comportamiento de un individuo en una secuencia de video previamente obtenida. Los algoritmos Sort y Deep Sort son los más eficientes en el rastreo de objetos, teniendo en cuenta el balance entre la precisión y la velocidad, por ejemplo, en el seguimiento de peatones y autos.

Se recomienda para el uso de un sistema de videovigilancia en tiempo real utilizar los algoritmos SSD con Mobilenet de base para la tarea de detección, y Deep Sort como rastreador; de esta forma se consigue un sistema balanceado entre la precisión y la velocidad con una adaptabilidad a distintos escenarios sin disponer de muchos recursos computacionales.

Referencias

AIDOUNI, E. y MANAL, 2019. Understanding YOLO and YOLOv2. [en línea]. [Consulta: 16 noviembre 2019]. Disponible en: https://manalelaidouni.github.io/manalelaidouni.github.io.

BEWLEY ALEX, GE ZONGYUAN, OTT LIONEL, RAMOS FABIO, AND UPCROFT BEN, 2016. Simple online and realtime tracking. International Conference on Image Processing (ICIP). S.l.: s.n., pp. 3464–3468.

ANKIT, S., 2017. A Quick Guide to Object Tracking: MDNET, GOTURN, ROLO. [en línea]. Disponible en: from: https://cv-tricks.com/object-tracking/quick-guidemdnet- goturn-rolo/.

ARCOS-GARCIA, A., AALVAREZ-GARCIA, J.A. y SSORIA-MORILLO, L., 2018. Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing, vol. 316, pp. 332–344. DOI 10.1016/j.neucom.2018.08.009.

BATHIJA, A. y SHARMA, P., 2019. Visual Object Detection and Tracking using YOLO and SORT. Visual Object Detection and Tracking using YOLO and SORT, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY [en línea], vol. 8, no. 11. Disponible en: https://www.ijert.org/visualobject- detection-and-tracking-using-yolo-and-sort.

BOCHKOVSKIY, A., WANG, C.-Y. y LIAO, H.-Y.M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection.

CIAPARRONE, G., LUQUE SANCHEZ, F., TABIK, S., TROIANO, L., TAGLIAFERRI, R. y HERRERA, F., 2019. Deep learning in video multi-object tracking: A survey. Neurocomputing. Elsevier BV, vol. 381, pp. 61–88. DOI 10.1016/j.neucom.2019.11.023.

DAI, Jifeng, LI, Y., HE, K. y SUN, J., 2016. R-FCN: Object detection via region-based fully convolutional networks. En Advances in neural information processing systems. 2016. p. 379-387.

DARWISHALZUGHAIBI, A., AHMED HAKAMI, H. y CHACZKO, Z., 2015. Review of Human Motion Detection based on Background Subtraction Techniques. International Journal of Computer Applications. Foundation of Computer Science, vol. 122, no. 13, pp. 1–5. DOI 10.5120/21757-4988.

DING, X. y LI, X., 2013. Target tracking algorithm of information detection for wireless sensor network. Journal of Computer Applications. vol. 33, no. 4, pp. 939–942. DOI 10.3724/sp.j.1087.2013.00939.

DUFFY y FLYNN, 2017. Year in Computer Vision. The M Tank [en línea]. Disponible en: https://www.themtank.org/a-year-in-computer-vision.

DURÁN SUÁREZ, J., 2017. Redes neuronales convolucionales en R: Reconocimiento de caracteres escritos a mano. S.l.: Universidad de Sevilla.

E QING, D., JUN, F. y ZHANG, Y.M., 2013. Human Motion Analysis Based On Silhouette and Centroid Displacement. Applied Mechanics and Materials. Trans Tech Publications, Ltd, vol. 335, pp. 1139–1144. DOI 10.4028/www.scientific.net/amm.333-335.1139.

FERNANDEZ-CARROBLES, M.M., DENIZ, O. y MAROTO, F., 2019. Gun and Knife Detection Based on Faster R-CNN for Video Surveillance. Pattern Recognition and Image Analysis. Springer International Publishing, pp. 441–452. DOI 10.1007/978-3-030-31321-0 38.

FOMIN, I.S. y BAKHSHIEV, A.V., 2019. Springer International Publishing,. Research on Convolutional Neural Network for Object Classification in Outdoor Video Surveillance System, en Cham. S.l.: s.n., pp. 221–229. ISBN 978-3-030-30425-6.

FRIEDMAN, J., HASTIE, T. y TIBSHIRANI, R., 2001. The elements of statistical learning. S.l.: Springer series in statistics New York.

GALARZA, B., MICHELLE, A. y FLORES, M., 2018. Deteccion de peatones en la noche usando Faster R-CNN e imágenes infrarrojas. Ingenius, pp. 48–57. DOI 10.17163/ings.n20.2018.05.

GALLEHUILLOS, C. y BELONGIE, S., 2010. Context based object categorization: A critical survey. Computer Vision and Image Understanding.Elsevier BV, vol. 114, no. 6, pp. 712–722. DOI 10.1016/j.cviu.2010.02.004.

GIRSHICK, R., 2015. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448.

HAROLDWKUHN, 1955. The hungarian method for the assignment problem. Naval research logistics quarterly, vol. 2, no. 1, pp. 83–97.

HE, K., GKIOXARI, G., DOLLÁR, P. y GIRSHICK, R., 2017. Mask RCNN. 2017 IEEE International Conference on Computer Vision, pp. 1440–1448.

HE, K., ZHANG, X., REN, S. y SUN, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. ECCV, pp. 346–361.

HENRIQUES, J., CASEIRO, R., MARTINS, P. y BATISTA, J., 2015. High-Speed Tracking with Kernelized Correlation Filters. Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583–596. DOI 10.1109/tpami.2014.2345390.

JIA, H.-X. y ZHAN, G.-J., 2009. Multiple Kernels Based Object Tracking Using Histograms of Oriented Gradients. Acta Automatica Sinica. China Science Publishing & Media Ltd, vol. 35, no. 10. DOI 10.3724/sp.j.1004.2009.01283.

KALMAN, R.E., 1960. A new approach to linear filtering and prediction problems. Journal of basic. Engineering, vol. 82, no. 1, pp. 35–45.

LAW, H. y DENG, J., 2018. CornerNet: Detecting objects as paired keypoints. ECCV. Meta learning for semisupervised few shot classification. In ICLR.

LEAL-TAIXE, L., MILAN, A., REID, I., ROTH, S. y KONRAD, S., 2015. Towards a benchmark for multi-target tracking.

LIU, Li, et al. Deep learning for generic object detection: A survey. International journal of computer vision, 2020, vol. 128, no 2, p. 261-318.

LIU, Wei, ANGUELOV, D., EEHAN, D., SZEGEDY, C., REED, S., FU, C.-Y. y BERG, A.C., 2016. SSD: Single Shot MultiBox Detector. Computer Vision ECCV, pp. 21–37. DOI 10.1007/978-3-319-46448-0 2.

LIU, W., LUO, Y.-N. y SUN, N., 2009. Mean Shift tracking algorithm based on background optimization. Journal of Computer Applications, vol. 29, no. 4, pp. 1015–1017. DOI 10.3724/sp.j.1087.2009.01015.

LOWE y DAVID, G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. Springer Science and Business Media LLC, vol. 60, no. 2, pp. 91–110. DOI 10.1023/b:visi.0000029664.99615.94.

NAM, H. y HAN, B., 2013. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. , DOI 10.1109/CVPR.2016.465.

NICOLAI WOJKE, D.P., Alex Bewley, 2017. SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC. 2017 IEEE International Conference on Image Processing, pp. 3645-3649.

NING, G., ZHANG, Z., HUANG, C., REN, X., WANG, H., CAI, C. y HE, Z., 2017. Spatially supervised recurrent convolutional neural networks for visual object tracking. 2017 IEEE International Symposium on Circuits and Systems (ISCAS). S.l.: IEEE, pp. 1–4.

PAL, S.K. y SHIU, S.C.K., 2004. Foundations of Case-Based Reasoning. JohnWiley & Sons, Inc,

PETROSINO, A. y MADDALENA, L., 2012. Neural Networks in Video Surveillance: A Perspective View. , DOI 10.1201/b11631-4.

PIOTR, D., RON, A., SERGE, B. y PIETRO, P., 2014. Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 8, pp. 1532–1545.

REDMON, J., DIVVALA, S., GIRSHICK, R. y FARHADI, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recpgnition (CVPR) [en línea]. S.l.: s.n., Disponible en: https://arxiv.org/abs/1506.02640v5.

REDMON, J. y FARHADI, A., 2017. YOLO9000: Better, Faster, Stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [en línea], Disponible en: https://arxiv.org/abs/1612.08242.

REDMON, J. y FARHADI, A., 2018. YOLOv3:An Incremental Improvement. arXiv.org [en línea], Disponible en: https://arxiv.org/abs/1804.02767.

REN, S., HE, K., GIRSHICK, R. y SUN, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems. S.l.: s.n., pp. 91–99.

REN, S., HE, K., GIRSHICK, R. y SUN, J., 2017. Faster R-CNN: Towards Real Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. Institute of Electrical and Electronics Engineers, vol. 39, no. 6, pp. 1137–1149. DOI 10.1109/tpami.2016.2577031.

RUSSAKOVSKY, O., DENG, JIA, SU, HAO, KRAUSE, JONATHAN, SATHEESH, SANJEEV, MA, SEAN, HUANG, ZUIHENG, KARPAATHY, ANDREJ, KHOSLA, ADITYA, BERNSTEIN, MICHAEL, ALEXANDER, B. y LI, F.-F., 2015. ImageNet Large Scale Visual Recognition Challenge. I. nternational Journal of Computer Vision.Springer Science and Business Media LLC, vol. 115, no. 3, pp. 211–252. DOI DOI 10.1007/s11263-015-0816-y.

SALAS, R., 2004. Redes neuronales artificiales. Universidad de Valparaıso. Departamento de Computación, vol.1

SANTOS, L., 2013. GoogleNet Artificial Inteligence. [en línea]. Disponible en: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/M. Lin, Q. Chen, and S. Yan. Network in network.2013. CoRR, abs/1312.4400

SREENU, G. y SALEEM DUEAI, A., M.A., 2019. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. Journal of Big Data. Springer Science and Business Media LLC, vol. 6, no. 1. DOI 10.1186/s40537-019-0212-5.

SUN, J., 2012. A Fast MEANSHIFT Algorithm-Based Target Tracking System. Sensors, vol. 12, no. 6, pp. 8218–8235. DOI 10.3390/s120608218.

VASSILIOS, T. y TASOS, D., 2018. Video surveillance systems-current status and future trends. Computers&Electrical Engineering. Elsevier BV, pp. 736–756. DOI DOI 10.1016/j.compeleceng.2017.11.0.

WANG, X., 2013. Intelligent multi-camera video surveillance: A review. Pattern Recognition Letters, vol. 34, no. 1, pp. 3–19.

WEI, H. y KEHTARNAVAZ, N., 2019. Semi-Supervised Faster RCNN-Based Person Detection and Load Classification for Far Field Video Surveillance. Machine Learning and Knowledge Extraction, vol. 1, no. 3, pp. 756–767. DOI 10.3390/make1030044.

WITTEN, I.H. y FRANK, E., 2005. Data Mining: Practical machine learning tools and techniques. S.l.: Morgan Kaufmann.

ZHANG, X., YANG, Y., HAN, Z., WANG, H. y GAO, C., 2013. Object class detection: A survey. ACM Computing Surveys, vol. 46, no. 1.

ZHANG, Y., WANG, J. y YANG, X., 2017. Real-time vehicle detection and tracking in video based on faster R-CNN. Journal of Physics: Conference Series 2017, DOI 10.1088/1742- 6596/887/1/012068.

ZHANSHENG, X., 2013. Application of Moving Object Tracking Based on Kalman Filter Algorithm. Journal of Applied Sciences. Science Alert, vol. 13, no. 19, pp. 4096–4098. DOI 10.3923/jas.2013.4096.4098.

ZHAO, Z.-Q., ZHENG, P., XU, S. y WU, X., 2019. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232.

Conflicto de interés

No existe

Contribuciones de los autores

David Ameijeiras Sánchez: Su contribución es asociada al desarrollo de la investigación.

Héctor R. González Diez: Su contribución es asociada al desarrollo y supervisión de la investigación.

Yanio Hernández Heredia: Su contribución es asociada al desarrollo y supervisión de la investigación.

Financiación

No aplica