Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking

Ernesto Antonio Reyes Ramírez
ernesto.reyes@cimat.mx

7 de Junio de 2023

1. Introducción

El problema del seguimiento de objetos consiste en detectar y predecir la posición de un objeto o varios a lo largo de un vídeo (que es una secuencia de imágenes o frames). En este caso, los autores se centran solamente en el seguimiento de un solo objeto. Existen muchas cosas que hacen muy complicada esta tarea tales como las deformaciones, la iluminación, el desenfoque de movimiento, la desinformación del objeto, entre otras.

La mayoría de los modelos de seguimiento de objetos solo utilizan detectores de características con redes neuronales. Esto se limita a solo trabajar con el dominio espacial.

En este trabajo los autores introducen un modelo que logra trabajar en el dominio espacio-temporal utilizando una arquitectura de red neuronal recurrente (RNN). Este modelo procesa la información proporcionada por un extractor de características y por las ubicaciones anteriores del objeto proporcionadas por la RNN. Como extractor de características se utiliza una red convolucional y un modelo pre-entrenado YOLO. Y para la red recurrente se utiliza una de tipo LSTM.

En este trabajo, nuestro objetivo es mostrar y explicar la arquitectura propuesta por los autores en la sección 2. Mientras que en la sección 3 mostraremos los resultados obtenidos por nuestra implementación de dicho modelo, haciendo las debidas comparaciones y conclusiones.

2. Recurrent YOLO

Antes de explicar el modelo principal, vamos a discutir sobre sus componentes principales, el modelo YOLO y la arquitectura de red recurrente LSTM.

2.1. LSTM

El modelo LSTM es un modelo de red recurrente que contiene una celda de memoría c_t que actúa como un acumulador de información. La celda es accesada, escrita y limpiada mediante varias puertas de activación, actualización, etc. La LSTM funciona de la siguiente manera. Cada vez que tenemos una nueva entrada esta será acumulada si la compuerta i_t es activada. El estado pasado c_{t-1} será olvidado en este iteración si la compuerta f_t está activada. Si la última salida de celda c_t se propagará al estado final h_t se controla aún más mediante la puerta de salida o_h. Todo esto se resume en las siguientes ecuaciones:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i),\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f),\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o),\\
g_t &= \sigma(W_{xc}x_t + W_{hc}h_{t-1} + b_c),\\
h_t &= o_t \odot \phi(c_t).
\end{aligned}
$$

Figura 1: Ecuaciones de la red LSTM

donde σ es la función de activación bien conocida y $\phi(x) = (e^x - e^{-x})/(e^x + e^{-x})$.

2.2. YOLO

En un sistema de seguimiento es muy importante la velocidad con la que procesamos en tiempo real los frames ya que es preciso mostrarlas seguidamente en secuencia (como video).

El modelo YOLO small procesa imágene a 155 FPS. YOLO regresa múltiples detecciones. Se emplea una matriz de costos de asignación que se calcula como la intersección sobre la unión (IOU) entre la detección actual y la media de su historial a corto plazo de detecciones validadas. La comparación de la primera imagen se realiza calculando el IOU entre todas las detecciones y la caja real que encierra el objeto solamente.

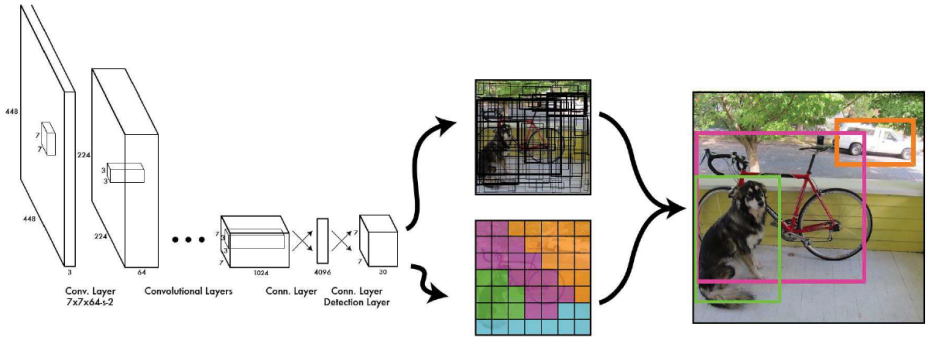

Figura 2: YOLO

2.3. ROLO

El modelo propuesto es llamado Recurrent YOLO (ROLO) y consiste en utilizar el modelo YOLO en una red neuronal recurrente del tipo LSTM.

El modelo YOLO se utiliza como un detector robusto de características visuales, así como de la ubicación preliminar del objeto en cuestión. Y la LSTM se utiliza para combinar las características extraídas junto con la información proporcionada por las ubicaciones anteriores.

Primero tenemos una secuencia de frames (el video), cada uno lo pasamos por una red de capas convolucionales y obtenemos un vector de características de tamaño 4096. Este vector lo pasamos por un lado por una red densa conectada, que es el modelo YOLO, y obtenemos un vector de tamaño 6 de la forma (c, x, y, w, h, p) donde c es la clase, (x, y) es el centro de la caja que encierra al objeto, w y h el ancho y alto de la caja y p la confianza. Este vector se pasa opcionalmente por una capa Heatmap para poder visiualizar mejor lo que está aprendiendo la red. Luego, el resultado se aplana y se concatena con el vector de características de tamaño 4096. Está será una de las entradas de la capa LSTM, la otra vendrá dada por la ubicación del objeto detectada en el paso anterior de la recurrencia. A continuación se presenta la arquitectura del modelo ROLO.

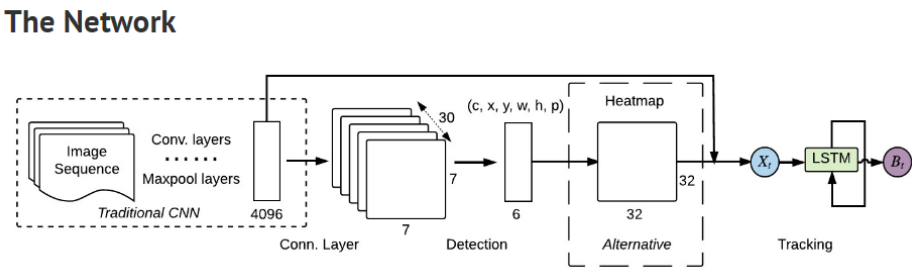

Figura 3: Recurrent YOLO

2.4. Entrenamiento

El entrenamiento consiste en tres fases. Primero se entrena la red con capaz convolucionales que recibe la entrada con los datos de ImageNet para tener un extractor de características robusto.

Para la parte del modelo YOLO se utiliza un modelo pre-entrenado llamado YOLO small, esto con el fin de no aprovecharse de las grandes capacidades de un modelo mucho mejor de YOLO.

Finalmente, la capa recurrente LSTM se entrena con base de datos VOC, con clases de 20 objetos diferentes.

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n} ||B_{target} - B_{pred}||_2^2,$$

Figura 4: Función de pérdida.

3. Resultados

Para evaluar el modelo se utilizó la base de datos OTB-30 que contiene frames de 30 vídeos así como los bordes de las cajas para los objetos detectados. Para medir su error utilizamos la medida Intersection over Union (IOU) la cual mide que tanto se intersecta una caja dada por un modelo sobre una caja real que encierra el objeto. A continuación presentamos una tabla con la IOU promedio con cada uno de los videos de la base de datos para los modelos YOLO y ROLO de nuestra implementación.

Video	YOLO average IOU	ROLO average IOU
Human2	0.6547218275239143	0.5472447390312158
Human9	0.5483551500195223	0.40364509915864233
Suv	0.548955019824414	0.6314443593780622
BlurBody	0.6155735360130521	0.6033018270623427
BlurCar1	0.41656934694246855	0.5916236284534581
Dog	0.24352141154193913	0.44721305753702445
Singer2	0.4594324055544526	0.5873020996893892
Woman	0.39676115521599015	0.6795797044412343
David3	0.2802496761332659	0.6392716033043412
Human7	0.6252457006421871	0.6729941492592387
Bird1	0.09731868783917048	0.38109249249221416
Car4	0.6720950694928947	0.7703013761742545
CarDark	0.31531260658830196	0.672993947619681
Couple	0.4066280908853392	0.5694841106238069
Diving	0.34024618958854475	0.677229059288835
Human3	0.42332055572616417	0.5843294464445226
Human6	0.4503298624877524	0.5504250231990875
Singer1	0.36408388544029285	0.6598766794337381
Walking2	0.3853043027322812	0.5963053232349034
BlurCar3	0.6202744320158	0.6566091673522461
Girl2	0.37185449748469224	0.5162708444751097
Skating1	0.4667009605795149	0.572273289245419
Skater	0.36971198726682597	0.6181941588752446
Skater2	0.563710839336688	0.6647732039535126
Dancer	0.560962144271328	0.7677164175837053
Dancer2	0.2649572324730092	0.6317630493932905
CarScale	0.6646040866845566	0.5690170039845344
Gym	0.4979228265785056	0.6043336596966733
Human8	0.45038072975568705	0.3888087457714761
Jump	0.2910276855206211	0.5481568668799119

Además vamos a mostrar algunas imágenes de las detecciones hechas por el modelo.

Figura 5: Mujer caminando.

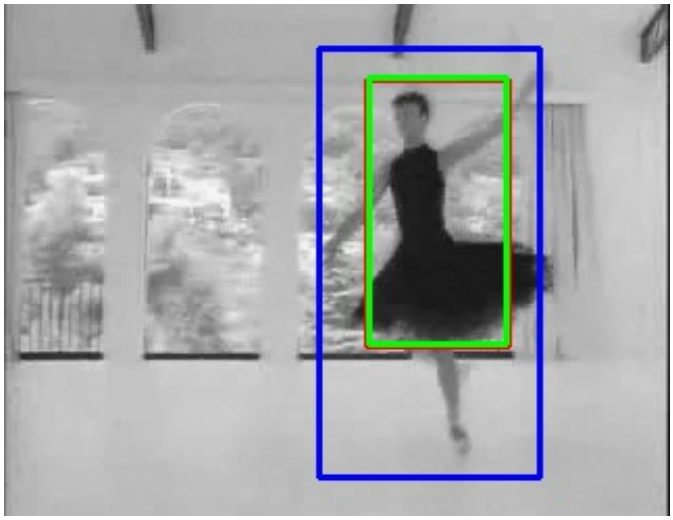

Figura 6: Bailarina.

Figura 7: Hombre caminando.

Figura 8: Carro en movimiento.

4. Conclusiones

El problema del seguimiento de objetos es un proceso bastante complejo y sensible a una gran cantidad de errores. Como se mencionó antes, los autores trataron de mitigar la mayoría de los problemas mediante la inclusión de la predicción temporal y no solo espacial, al tomar en cuenta las posiciones anteriores de los objetos, esto ayudando a predecir donde podría encontrarse la siguiente vez.

Como se mencionó en los resultados, se comparó un modelo simple basado en YOLO, que consiste en solo hacer búsqueda al frame actual, con el modelo propuesto ROLO. De la tabla de resultados podemos ver que el modelo ROLO ganó en 25 de los 30 vídeos, un resultado bastante bueno. Además, de los videos se puede percibir lo siguiente.

ROLO es un modelo muy estable, no se pierde cuando un objeto se le cruza enfrente, mientras que YOLO si se pierde y detecta al otro.

ROLO no es sensible a los cambios bruscos de movimiento de la cámara. Gracias a que también predice sobre el tiempo puede tener cierta estabilidad. Cosa que no hace YOLO solo ya que si es una gran interferencia para él según lo visto.

ROLO no es tan robusto en cuanto a objetos que no conoce, ya que si por ejemplo lo que se quiere detectar es la cara, ROLO y YOLO va a detectar el cuerpo completo.

YOLO le gana a ROLO en los vídeos donde el objeto que se quiere seguir tiene un recorrido lineal, es un auto o con alguna forma definida que no cambia, ya que es más fácil y rápido detectar en un frame esto. Pero si es por ejemplo una bailarina que abre y cierra sus brazos o piernas seguidamente, ROLO funciona muy bien y YOLO no.

En conclusión, el modelo Recurrent YOLO (YOLO) muestra una mejora considerable en comparación al modelo YOLO al tratar con los problemas de cruce objetos, poca luz, movimientos bruscos de la cámara y del objeto, por

lo que es un buen modelo para la tarea de seguimiento de objetos. Además, como comentan los autores, presenta resultados alcanzando el resultado del arte para vídeos nuevos antes vistos por el modelo.

Por la parte de la implementación no entrené el modelo completo por mi cuenta porque no contaba con los recursos tecnológicos necesarios, así que utilicé un modelo pre-entrenado que ellos proporcionan, y que funciona muy bien, fue el que me dio los resultados de la tabla. Esos fueron mis incovenientes y observaciones.

Referencias

[NZH+17] Guanghan Ning, Zhi Zhang, Chen Huang, Xiaobo Ren, Haohong Wang, Canhui Cai, and Zhihai He. Spatially supervised recurrent convolutional neural networks for visual object tracking. In *2017 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–4. IEEE, 2017.