



Ciencia y Tecnología

Secretaría de Ciencia, Humanidades, Tecnología e Innovación



CIMAT
UNIDAD MÉRIDA

SEGUIMIENTO DE OBJETOS BASADO EN DEEP LEARNING

Dr. Francisco J. Hernández López

SECIHTI – CIMAT-Mérida

fcoj23@cimat.mx, www.cimat.mx/~fcoj23



SIMPLE ONLINE AND REAL-TIME TRACKING (SORT)

- Realiza el seguimiento de objetos con información de algún detector de objetos y consta de cuatro módulos:
 - **Detección.** Por ej. YOLO
 - **Estimación.** Predecir la posición del objeto en el siguiente *frame* f_t , usando por ej. el filtro de Kalman
 - **Asociación.** Comparar la estim. del filtro de Kalman con la detección en f_t
 - **Gestión.** Las detecciones no asignadas a un seguidor existente se toman para la creación de un nuevo seguidor



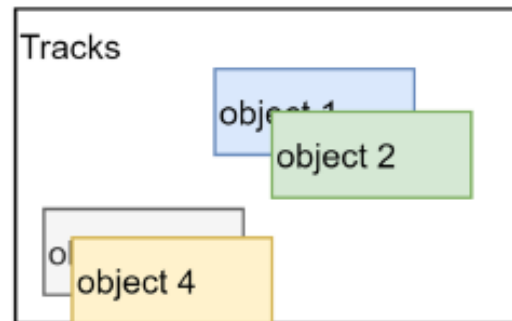
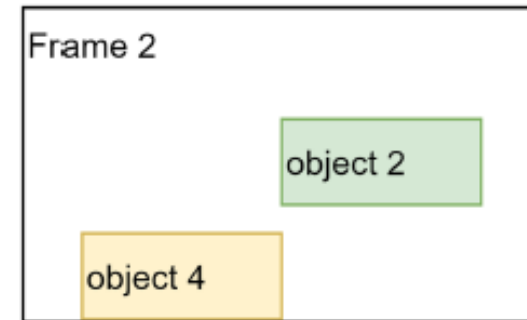
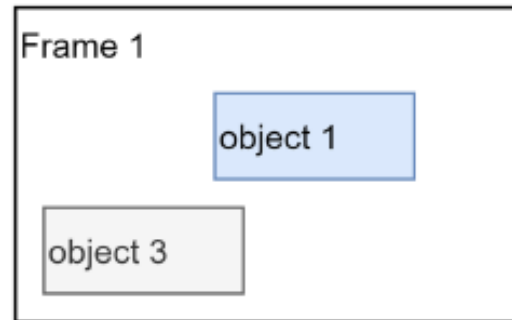
<https://github.com/abewley/sort>

MODELO DE ESTIMACIÓN

- El estado de cada objeto (*target*) se modela como
$$\vec{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$$
 - (u, v) es la posición central del objeto
 - s es la escala
 - r es la razón de aspecto del BBox
- Cuando una detección se asocia a un objeto, el BBox se utiliza para actualizar el estado del objeto usando el filtro de Kalman
- Si no hay una BBox asociada al objeto, entonces se predice el estado sin corrección, usando el modelo de velocidad lineal

ASOCIACIÓN DE LOS DATOS

- La matriz de costos de asignación se calcula usando el IoU entre cada detección y todas las BBox predichas
- El problema se resuelve usando el algoritmo Húngaro
- Se fija un umbral IoU_{min} para descartar aquellas asignaciones en donde el traslape es menor que dicho umbral



Assignment

	Obj1	Obj3
Obj2	0.7	0.0
Obj4	0.0	0.8

Youssef, Y., & Elshenawy, M. (2021). Automatic vehicle counting and tracking in aerial video feeds using cascade region-based convolutional neural networks and feature pyramid networks. *Transportation Research Record*, 2675(8), 304-317.

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3464-3468). IEEE.

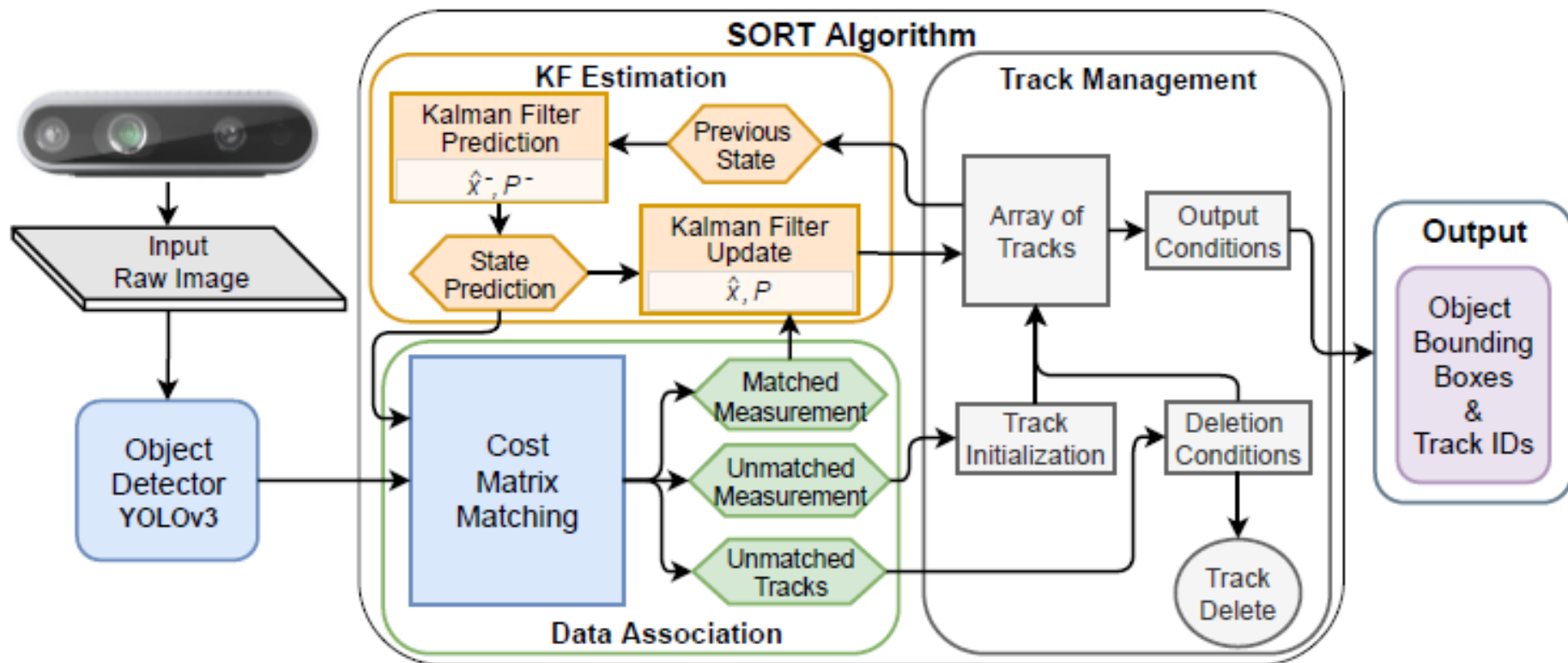
Seguimiento de Objetos - Deep Learning. Francisco J. Hernández-López

Ene-Jun 2025

GESTIÓN DE LOS SEGUIDORES

- Cuando un objeto es detectado, a este se le asigna un identificador (ID)
- Si el objeto sale de la escena, entonces el ID se elimina
- Si existe una detección con un traslape menor que el IoU_{min} , entonces se crea un nuevo seguidor
- El seguidor se inicializa con el BBox, velocidad en cero y covarianza grande (ya que no hay observación de velocidad)
- Los seguidores se eliminan cuando no hay detecciones en los siguientes T_{Lost} frames. En todos los experimentos $T_{Lost} = 1$

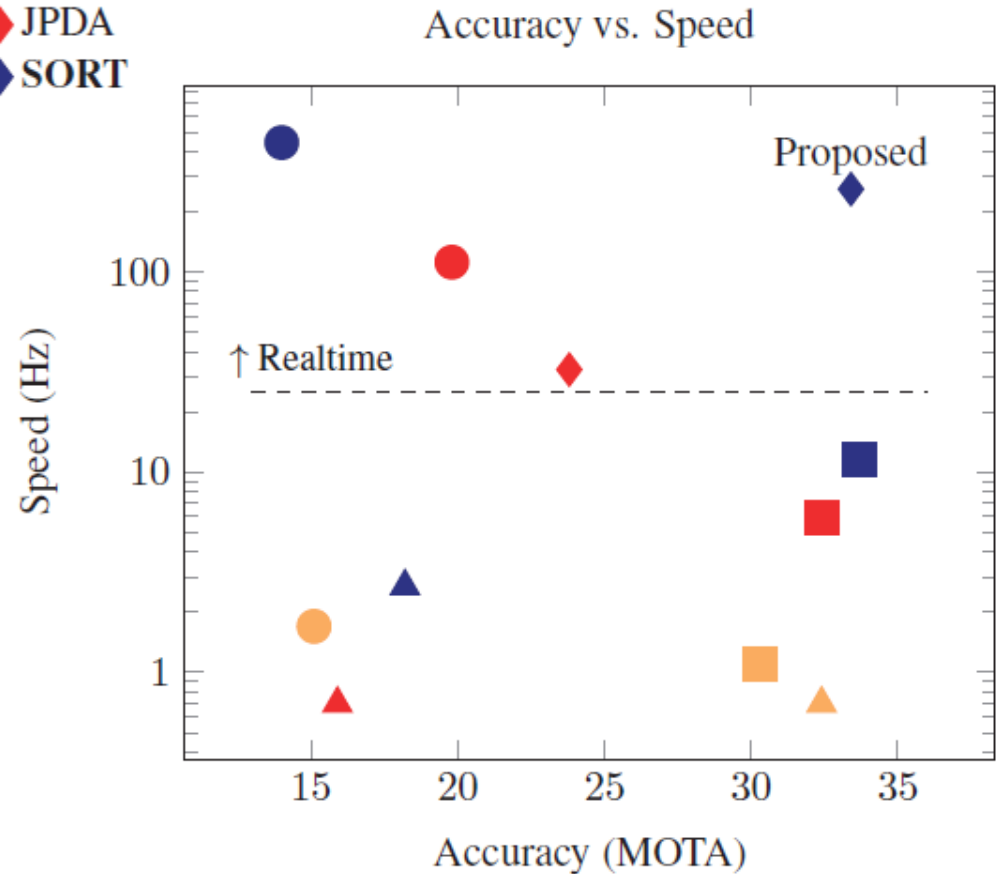
ALGORITMO SORT



Pereira, R., Carvalho, G., Garrote, L., & Nunes, U. J. (2022). Sort and deep-SORT based multi-object tracking for mobile robotics: Evaluation with new data association metrics. *Applied Sciences*, 12(3), 1319.

EVALUACIÓN DE SORT C. R. A OTROS

■ TDAM ● LP2D ▲ TBD ◆ JPDA
■ NOMT ● DP_NMS ▲ SMOT ◆ SORT
■ MDP ● TC_ODAL ▲ MHT_DAM



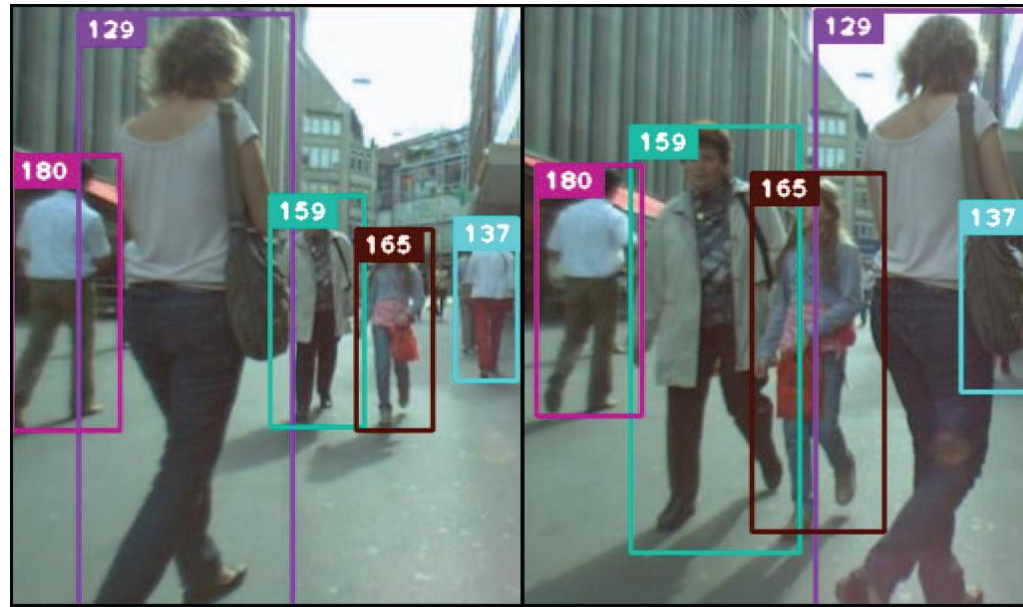
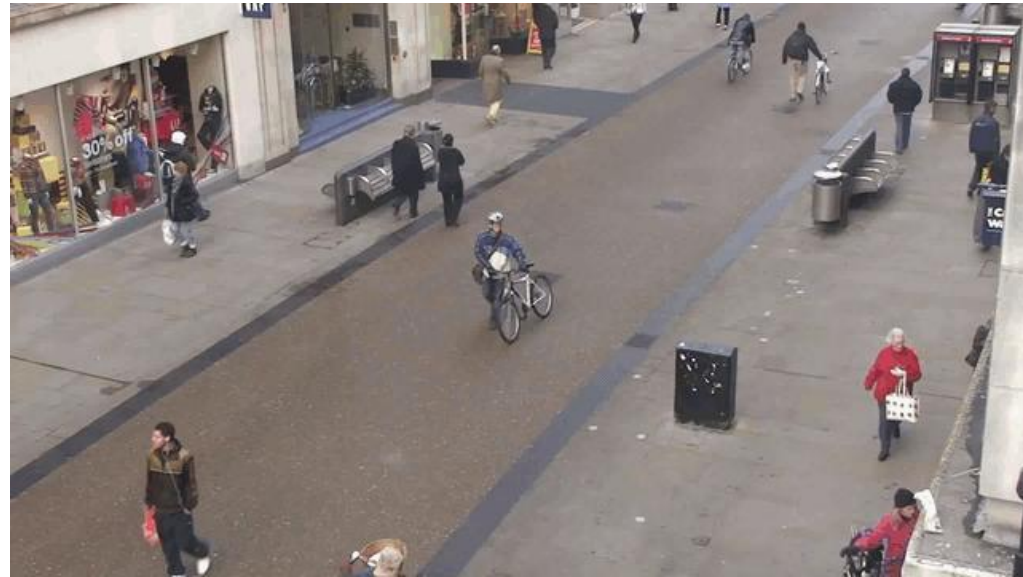
Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3464-3468). IEEE.

Seguimiento de Objetos - Deep Learning. Francisco J. Hernández-López

Ene-Jun 2025

DEEP-SORT

- Incorporan información de apariencia por medio de una métrica de asociación profunda, para mejorar el algoritmo SORT.
- Deep-SORT puede seguir objetos a través de largos periodos de oclusión, reduciendo el número de intercambios de ID.



Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3645-3649). IEEE.

Seguimiento de Objetos - Deep Learning. Francisco J. Hernández-López

MODELO DE ESTIMACIÓN

- El estado de cada objeto (*target*) se modela como

$$\vec{x} = [u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h}]^T$$

- (u, v) es la posición central del objeto
 - γ es la razón de aspecto del BBox
 - h es la altura
- Se utiliza el filtro de Kalman con velocidad constante y modelo de observación lineal
- Para cada seguidor se cuenta el número de *frames* a partir de la última asociación de medición exitosa. Este contador se incrementa durante la etapa de predicción del filtro de Kalman. Se pone en cero cuando el seguidor se asocia con una medición
- Los seguidores que exceden una edad máxima A_{max} son eliminados, pues se considera que han dejado la escena
- Los seguidores que no se han asociado con alguna medición dentro de los primeros tres frames, son eliminados

PROBLEMA DE ASIGNACIÓN

- Se integran la información de movimiento y apariencia a través de una combinación de dos métricas:

- Distancia de Mahalanobis (Estado del seguidor y el BBox detect.)

$$d^{(1)}(i, j) = (\mathbf{d}_j - \mathbf{y}_i)^T \mathbf{S}_i^{-1} (\mathbf{d}_j - \mathbf{y}_i),$$

con $(\mathbf{y}_i, \mathbf{S}_i)$ el espacio de mediciones y $\mathbf{d}_j$ el j-ésimo BBox detectado.

Además, se aplica un umbral

$$b_{i,j}^{(1)} = \mathbb{I}[d^{(1)}(i, j) \leq t^{(1)}], \text{ con } t^{(1)} = 9.4877.$$

- Distancia del coseno (Vectores de caract. de apariencia)

$$d^{(2)}(i, j) = \min \{1 - \mathbf{r}_j^T \mathbf{r}_k^{(i)} \mid \mathbf{r}_k^{(i)} \in \mathcal{R}_i\}, \quad b_{i,j}^{(2)} = \mathbb{I}[d^{(2)}(i, j) \leq t^{(2)}]$$

Para cada $\mathbf{d}_j$ se calcula un descriptor de apariencia $\mathbf{r}_j$ con $\|\mathbf{r}_j\| = 1$.

Para cada seguidor k se mantiene una galería $\mathcal{R}_k = \{\mathbf{r}_k^{(i)}\}_{i=1}^{L_k}$ de los últimos $L_k = 100$ descriptores de apariencia asociados

COMBINACIÓN DE LAS MÉTRICAS

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1 - \lambda) d^{(2)}(i,j)$$

- Es una asociación admisible si está dentro de la región de ambas métricas

$$b_{i,j} = \prod_{m=1}^2 b_{i,j}^{(m)}$$

- En los experimentos probaron $\lambda = 0$, cuando hay movimientos significativos de la cámara

DESCRIPTOR DE APARIENCIA

- Se utiliza una CNN entrenada en el conjunto de datos MARS, para identificación de personas
- MARS contiene 1,100,000 de imágenes de 1,261 peatones
- La arquitectura de la CNN contiene dos capas conv., seis bloques residuals y una capa densa que devuelve un mapa de caract. de dimension 128

Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and ℓ_2 normalization		128

Arquitectura de la CNN



Caract.

HOG3D

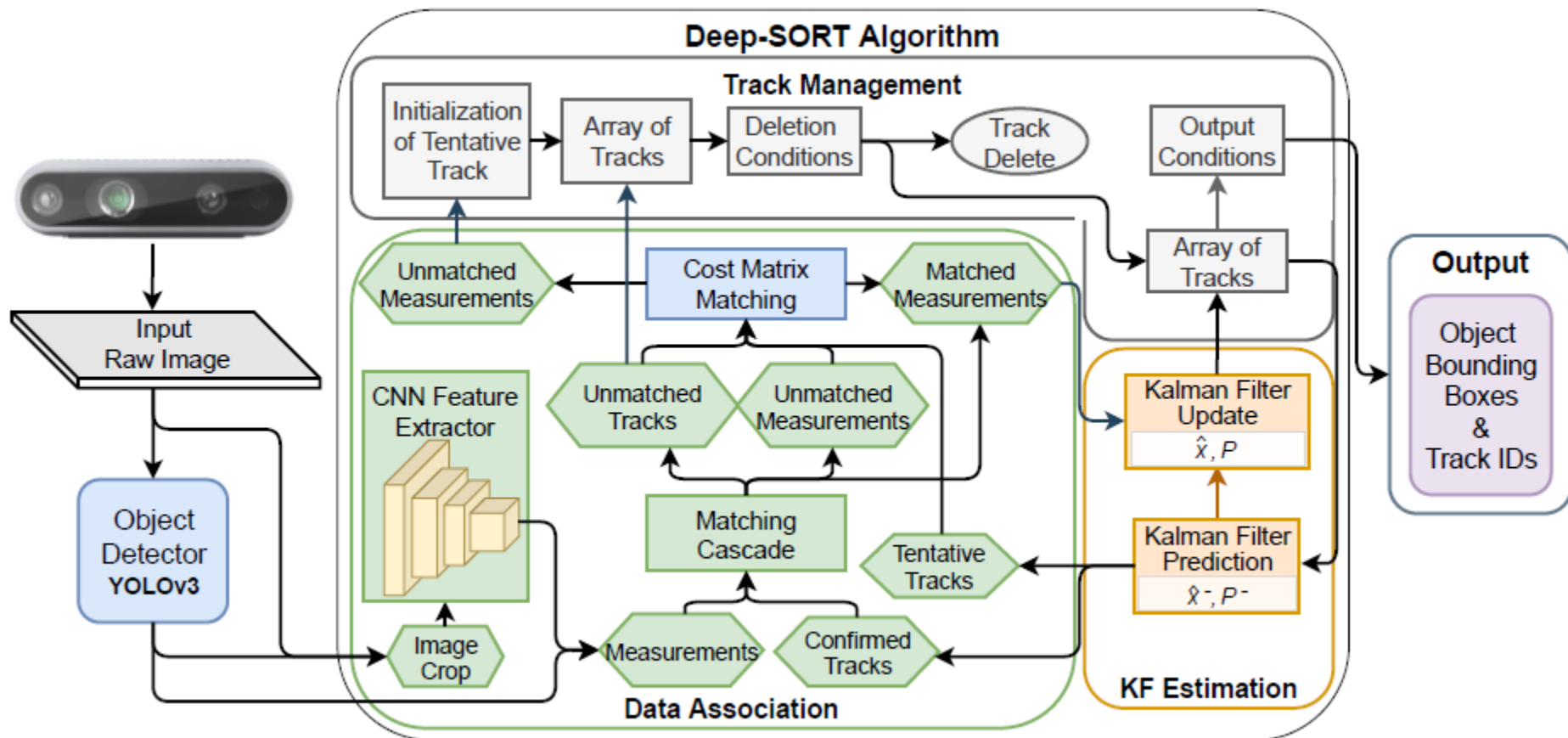
CNN

Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., & Tian, Q. (2016). Mars: A video benchmark for large-scale person re-identification. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14* (pp. 868-884). Springer International Publishing.

Seguimiento de Objetos - Deep Learning. Francisco J. Hernández-López

Ene-Jun 2025

ALGORITMO DEEP-SORT



Pereira, R., Carvalho, G., Garrote, L., & Nunes, U. J. (2022). Sort and deep-SORT based multi-object tracking for mobile robotics: Evaluation with new data association metrics. *Applied Sciences*, 12(3), 1319.

PROBANDO CÓDIGO DEEP-SORT

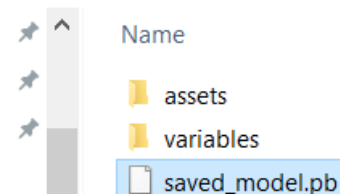
- <https://github.com/theAIGuysCode/yolov4-deepsort>
- `conda create -n env_py37 python=3.7`
- `conda activate env_py37`
- `conda install -c conda-forge cudatoolkit=10.1 cudnn=7.6`
- `pip install opencv-contrib-python==4.1.1.26`
- `pip install tensorflow-gpu==2.3.0`
- `conda install conda-forge::matplotlib`
- `conda install conda-forge::lxml`
- `conda install conda-forge::tqdm`
- `conda install conda-forge::absl-py`
- `conda install conda-forge::easydict`
- `conda install conda-forge::pillow`

CONVERTIR EL MODELO DARKNET A TENSORFLOW

Podemos descargar los pesos del YOLOv4:
<https://github.com/AlexeyAB/darknet>
Agregarlo en: yolov4-deepsort-master\data

■ python `save_model.py` --model yolov4

yolov4-deepsort-master > checkpoints > yolov4-416 >



```
tf_op_layer_concat_17 (TensorFl [(None, None, 4)] 0
```

```
tf_op_layer_Reshape_14 (TensorF [(None, None, None)] 0
```

```
tf_op_layer_GatherV2_1[0][0]  
tf_op_layer_Reshape_14/shape[0][0]
```

```
tf_op_layer_concat_18 (TensorFl [(None, None, None)] 0
```

```
tf_op_layer_concat_17[0][0]  
tf_op_layer_Reshape_14[0][0]
```

```
=====  
Total params: 64,429,405  
Trainable params: 64,363,101  
Non-trainable params: 66,304
```

```
2024-11-25 20:07:11.591841: W tensorflow/python/util/util.cc:348] Sets are not currently considered  
ding using them.
```

```
INFO:tensorflow:Assets written to: ./checkpoints/yolov4-416/assets
```

```
I1125 20:07:58.406810 24308 builder_impl.py:775] Assets written to: ./checkpoints/yolov4-416/assets
```

PROBANDO DEEP-SORT PARA VIDEO

- python [object_tracker.py](#) --video ./data/video/test.mp4 --output ./outputs/test.mp4 --model yolov4



PROBANDO DEEP-SORT CON LA WEBCAM

- `python object\_tracker.py --video 0 --output ./outputs/webcam.mp4 --model yolov4`



GRACIAS POR SU ATENCIÓN

Francisco J. Hernández López

fcoj23@cimat.mx

WebPage:

www.cimat.mx/~fcoj23

