**Lab 5 – Tracking**

***T0.*** *Using your UPF email account, create a Codalab account at* [*https://codalab.org/*](https://codalab.org/) *and register in our* [*challenge*](https://codalab.lisn.upsaclay.fr/competitions/11141)*. Your team and tracker name must be stated in your report.*

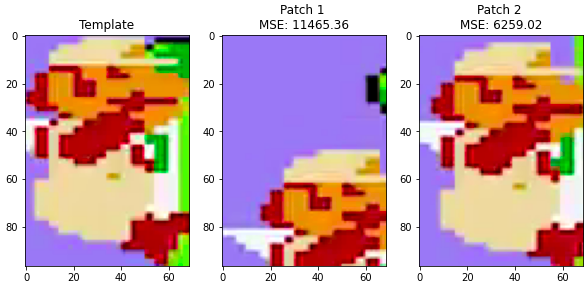
Nom Codalab = mariagomez

Team name = M\_A\_P

***T1. Comparing the appearance using the MSE metric****. Use the mean\_squared\_error function defined in skimage.metrics to compare 2 patches at frame i=4 with respect to our RGB template. The bounding boxes are defined by [590, 92, 659, 189] and [601, 137, 670, 234].*

* *Determine the two resulting MSE values and state which patch is more similar to the template.*

Dibuixem dos patches del frame i els comparem amb el template usant el MSE. Com podem veure a la següent imatge, el patch 2 s’assembla molt més al template que el patch 1. Això també ho veiem reflexat al número de MSE ja que aquest valor és major si hi ha més diferència entre les imatges, per tant, el patch 1 té un valor més gran de MSE ja que s’assembla menys al template.



* *What values are expected for very similar and not similar patches using the MSE?*

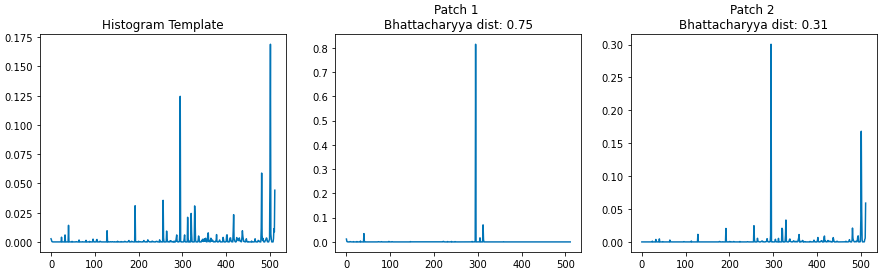
Com hem comentat anteriorment, si el patch s’assembla molt al template, el valor de MSE disminueix, per tant, si tenim un patch molt semblant al template el MSE serà petit, en canvi, si tenim un patch no molt similar al template, tindrem un MSE força gran.

***T2. Comparing histograms using the Bhattacharyya distance****. Compare the histograms of 2 patches at frame i=30 with respect to the histogram of our RGB template. The bounding boxes of the two patches are defined by [606,102,675,199] and [601, 507, 670, 604]. First, compute the histograms using the calculate\_patch\_hist function defined in the file regions.py. Then, calculate the Bhattacharyya distance using the bhattacharyya\_dist wrapper function defined in the file regions.py*

* *Determine the two resulting Bhattacharyya distance values and state which patch is more similar to the template.*

A continuació podem veure els histograms del dos patches calculats anteriorment juntament amb la distància de Bhattacharyya corresponent i l’histograma del template.

Si mirem els histogrames podem fer-nos una idea de quin s’assembla més a l’histograma del template, en aquest cas, veiem que l’histograma del patch 2 s’assembla més al template que el del patch 1. Per acabar de confirmar la nostra hipòtesi, calculem la distància de Bhattacharyya entre els histogrames del patches amb el del template i veiem que la distància obtinguda al patch 2 és menor que la que obtenim al patch 1, per tant, podem concloure que el patch 2 és més similar al template.



* *What values are expected for very similar and not similar patches using the Bhattacharyya distance?*

Per dos patches molt similars esperarem una distancia de Bhattacharyya molt petita, en canvi, per patches diferents esperem una distància gran.

***T6. Similarity matrix.*** *In the next cell, the values of the non-square distance matrix are plot.*

* *What does it mean that an element of the matrix has a small value?*

Si té un valor petit significa que tenim dos centroides que estan a prop

* *What does it mean that an element of the matrix has a large value?*

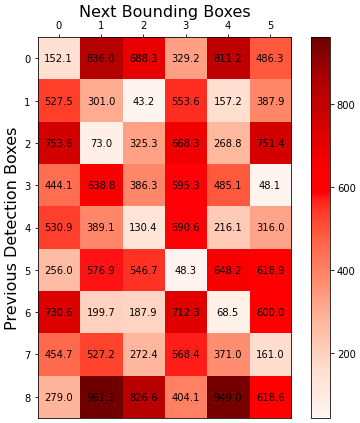
Si té un valor gran significa que el centroide entre dos objectes a detectar estan a més distància

* *State the possible causes of having less bounding boxes in the next frame.*

Possibles causes de tenir menys bounding boxes en el següent frame seria pel fet de tenir oclusions, que l’objecte hagi sortit d’escena i per tant no el podem detectar, o també que el frame tingui soroll i no pugui detectar bé a l’objecte

* *How does it affects the multiple-object tracking?*

Això afecta de manera que canviï d’assignació als objectes, i un mateix objecte pugui passar per diferents Bounding Boxes al llarg de tots els frames, depenent si està més a prop o més lluny de la primera que se li havia assignat.



***T7. Matching bounding boxes and IDs assignation.*** *Complete the calculate\_matches function in the following cell. You can use a greedy approach or the Hungarian algorithm*

Per assignar IDs a les bounding boxes fem servir el *Hungarian algorithm*. Aquest es basa en comparar la distància de les previous bounding box amb les next boxes i assignar un match a la que tingui menys distància.

***T8. Build a distance MOT tracker.*** *The following code forms the skeleton of a distance MOT tracker. Use the calculate\_distance\_matrix and calculate\_matches functions to complete the track\_multiple\_object function.*

Primer assignem la el primer frame amb bounding boxes a previous detection. Un cop ja la tenim inicialitzada recorrem tots els frames:

* inicialitzem les bounding boxes d’aquest frame
* calculem la distance matrix
* trobem el match de les bounding boxes
* posem els IDs

Quan la funció ha recorregut tots els frames ja tenim el array detections amb totes les bounding boxes enumerades de cada frame.

***T9. Create a Single-Object Tracker for megaman, race, and motorcycle videos****. You are not expected to build one tracker for each individual video and it is highly recommended to use a single function for all of them that receives different parameters. You can reuse the skeleton created in T4. In your report clearly state:*

* *For each video clearly describe:*
  + *The algorithm design used on it.*
  + *The influence of the parameters.*
  + *An explanation of the features used by your algorithm.*
  + *An analysis describing its strong points and pitfalls.*
* *Provide a comparison between the results of the three videos.*
* *An interpretation of the obtained challenge results metrics (HOTA) for all videos.*

Per construir aquest Single-Object Tracker hem fet servir la funció definida anteriorment *track\_single\_object(frames, target, search\_area\_shape)*. Aquesta funció utilitza els frames del video, el target, que és la posició de la bounding box inicial i l'àrea que recorrerà per trobar un match.

Un cop tenim la primera detecció fent servir el primer frame i la bounding box target, recorrem tots els altres comparant l’anterior amb el present mitjançant la funció search neighborhood que utilitza una sliding window, de l'àrea que li hem demanat, per trobar quina de les boxes té una Bhattacharyya distance més petita amb la anterior.

Un dels errors que hem tingut durant la implementació és que al posar una search area massa petita no es genera cap bounding box inicial i per tant el codi dona error. En canvi si l'àrea és massa gran el codi tarda molt en executar-se.

**RACE:**

Si observem el resultat obtingut al video race veiem que detecta perfectament l’objecte en tot moment, això és degut a que el personatge no canvia estèticament en tot el video, per tant, la BoundingBox inicial ens serveis per detectar l’objecte d’inici a final.

**MEGAMAN:**

En aquest video no observem tanta exactitud com al video de race. Quan arribem al frame 466 aproximadament, perdem la detecció de l’objecte. Això és degut a que en un moment determinant un efecte tapa al personatge i per tant la box perd l’objecte. Aquest problema l’hem intentat solucionar creant una nova BoundingBox, al frame 467, per que a partir d’aquest frame es faci la detecció del personatge a partir d’un nou target. El problema que hem tingut ha sigut que tardaba molt en executar-se i per tant, no hem pogut obtenir el resultat final i no ho hem implementat.

**MOTORCYCLE:**

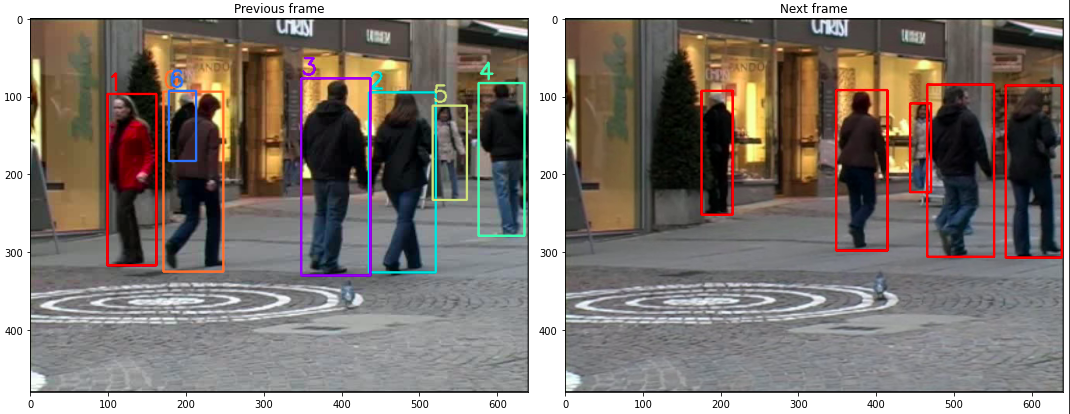
Per últim, al video de la moto veiem que des del principi no detecta massa bé l’objecte, això creiem que és degut a que la BoundingBox és massa petita i per tant està detectant només la roda de la moto. A meitat del vídeo més o menys veiem que perdem la detecció que volem i passa a detectar la roda d’un altre vehicle. Per tant, creiem que si la BoundingBox hagués sigut més gran hauria detectat millor l’objecte. De nou, no ho hem pogut implementar ja que al intentar-ho no acabava mai d’executar-se ja que requeria molt temps.

En conclusió, creiem que per fer millors deteccions necessitaríem molt més temps ja que és un programa molt dens que requereix molt temps d'execució i és molt difícil fer proves si per veure un resultat ens fan falta 30 mins o més.

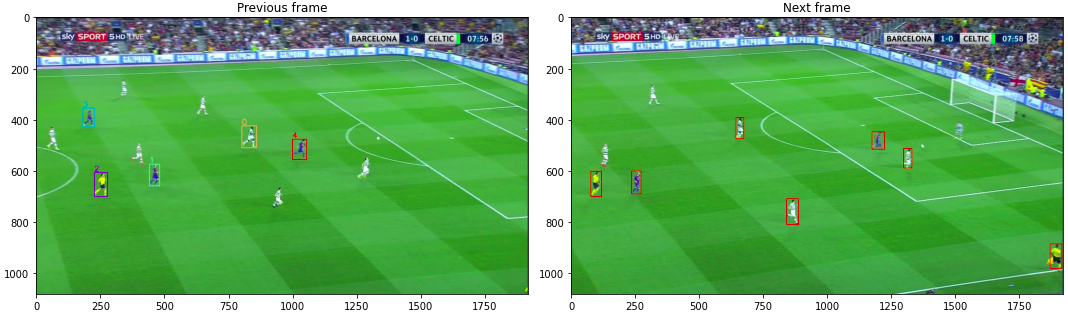
***T10. Create a Multiple-Object Tracker for football and pedestrian videos.*** *You are not expected to build one tracker for each individual video and it is highly recommended to use a single function for all of them that receives different parameters. In your report clearly state*

* *For each video clearly describe:*
  + *The algorithm design used on it.*
  + *The influence of the parameters.*
  + *An explanation of the features used by your algorithm.*
  + *An analysis describing its strong points and pitfalls.*
* *Provide a comparison between the results of both videos.*
* *An interpretation of the obtained challenge results metrics (HOTA) for all videos.*

Per construir un Multiple-Object Tracker utilitzem el *same approach* durant tot el document. Ens basem en una segmentació semàntica inicial per trobar les bounding boxes del primer frame i a partir d’aquí per trobar les següents apliquem el **distance MOT tracker,** definit en l’apartat T8. Aquest procés l’hem aplicat per els dos videos, pedestrians i football.

**PEDESTRIANS:**

Podem veure que un punt fort de pedestrians és que detecta totes les persones molt bé. Notem un canvi de IDs a una mateixa persona això és degut a que l’objecte s’allunya de la bounding box que se li había assignat i passa a una altre, per tant, un mateix peató passa per varies BoundingBox.

**FOOTBALL:**

En aquest cas, contrari a pedestrians, el ID es manté durant més temps. El problema també és el contrari a pedestrians, no consegueix detectar totes les persones del frame.

Un cop analitzats els dos videos veiem que semantic segmentation és una eina que va molt més ràpid que seguir cada objecte un per un, però a vegades no acaba de detectar-los tots. També s’haurien d’implementar d’alguna manera millor els IDs ja que quan es creuen no acaba de funcionar molt bé .

És important mencionar que en el SOT la bounding box és sempre del mateix tamany i per tant per algunes aplicacions pot servir però no és una eina molt general. Semantic segmentation ens permet que quan un objecte es fa més gran o més petit la bounding box també, fent el programa més ràpid i fiable.