Monitoring COVID-19 prevention measures on CCTV cameras using Deep Learning

Candidate
Cota Davide Antonio Maria

Supervisors
Paolo Garza
Helio Cortes Vieira Lopes

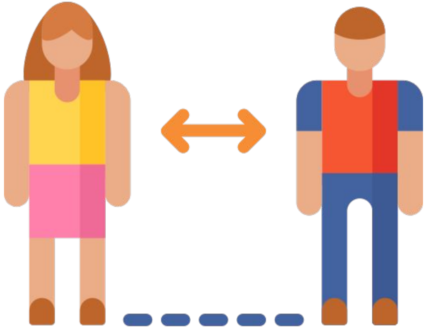
COVID-19 prevention measures

The prevention strategies suggested by WHO consists in **maintaining a correct hygiene**, **wearing face masks** in public and crowded place. **Physical distancing measures** are also recommended to slow the disease transmission. Some examples to be considered as examples of social distancing can be: isolation, closing of public places such as schools, stadiums, cinemas, remote work and **keeping a distance from other people higher than 1 meter**. Most of all closed places limits the maximum number of people to a certain threshold.

NO MORE THAN 10 PEOPLE ARE ALLOWED IN THE STORE AT ONE TIME
IN COMPLIANCE WITH LOCAL LAWS AND IN ORDER TO HELP PROTECT OUR EMPLOYEES AND CUSTOMERS, PLEASE DO NOT STAY WITHIN 6' OF OTHERS FOR AN EXTENDED PERIOD OF TIME
QT
QT
SECURITY
audio/video in use
This door to remain unl
when the buildi

Our project three main goals

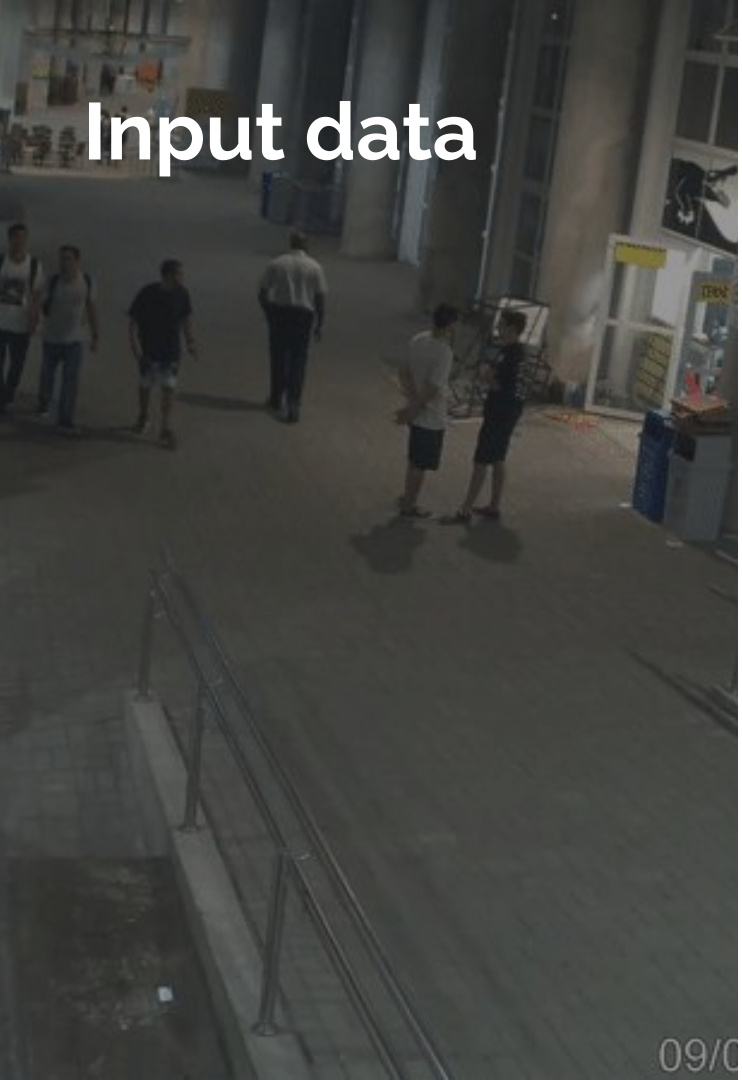
Counting the number of people



Monitoring social distancing

Detecting people wearing face masks

Input data

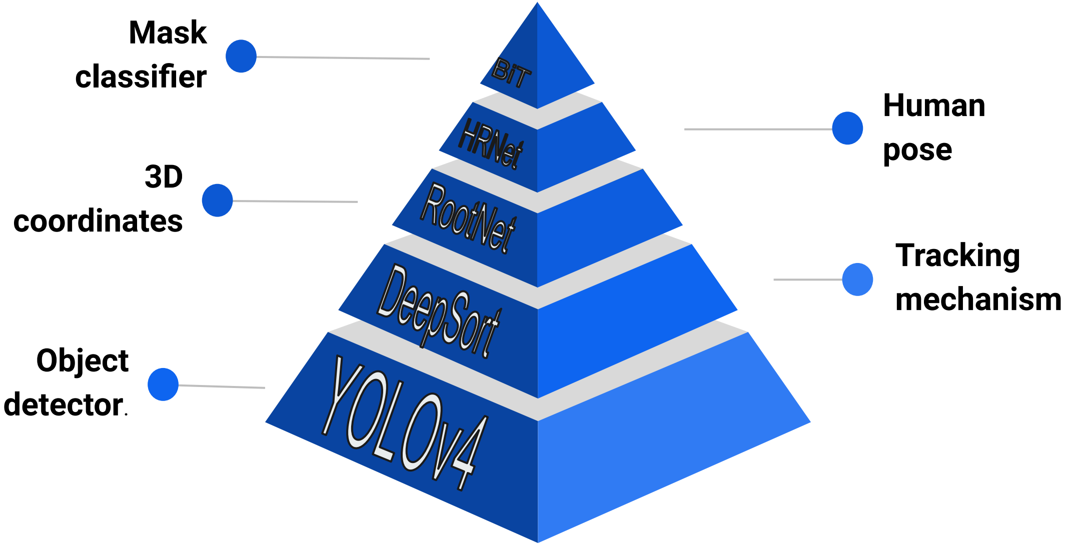


In these days CCTV cameras can be found anywhere, from public places such as airports, hospitals, schools, museums to shops, retail stores, trains, all places that requires an intense monitoring in terms of COVID-19 prevention measures. It makes the perfect instrument to reliably have real time images with no further installations.

For this reason, CCTV videos are the input data for our model. The drawback of the images coming from this source are: **low resolution**, **small objects to detect**, **occlusions**.

Proposed methodology

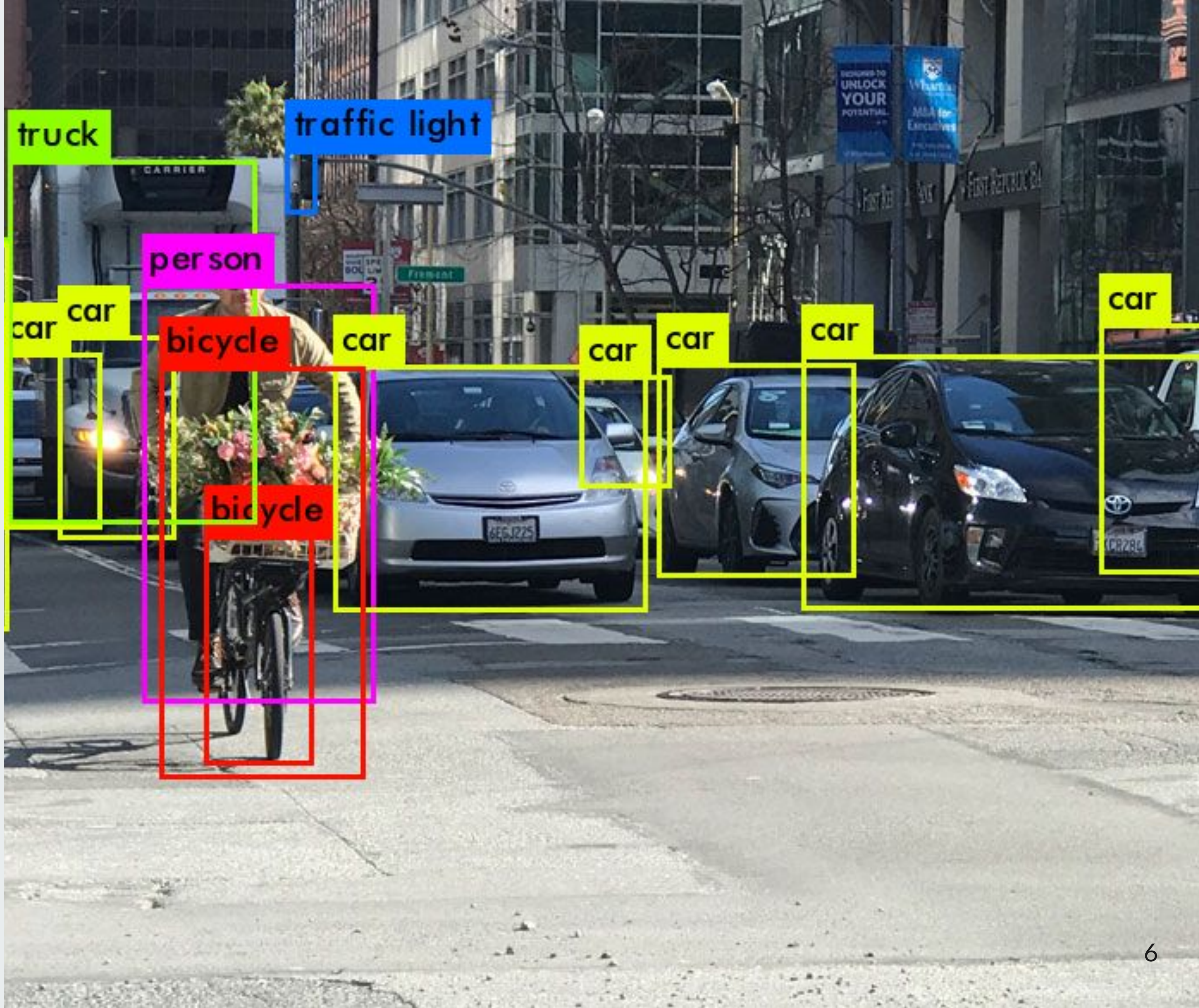
1. **Object detector**: the whole system relies on the YOLOv4 algorithm for the human detection and localization. It addresses also the *people counting* task.
2. **Tracking**: our system needs to store some statistics about each person during the whole time it appears in the video sequence. DeepSort is a fast and reliable tracking algorithm.
3. **Human being 3D coordinates:** RootNet is a neural network that takes in input the YOLOv4 bounding boxes and gives in output the human 3D coordinates.
4. **Human pose estimation:** HRNet estimates the pose of a human being
5. **Mask classifier:** BiT + ResNet



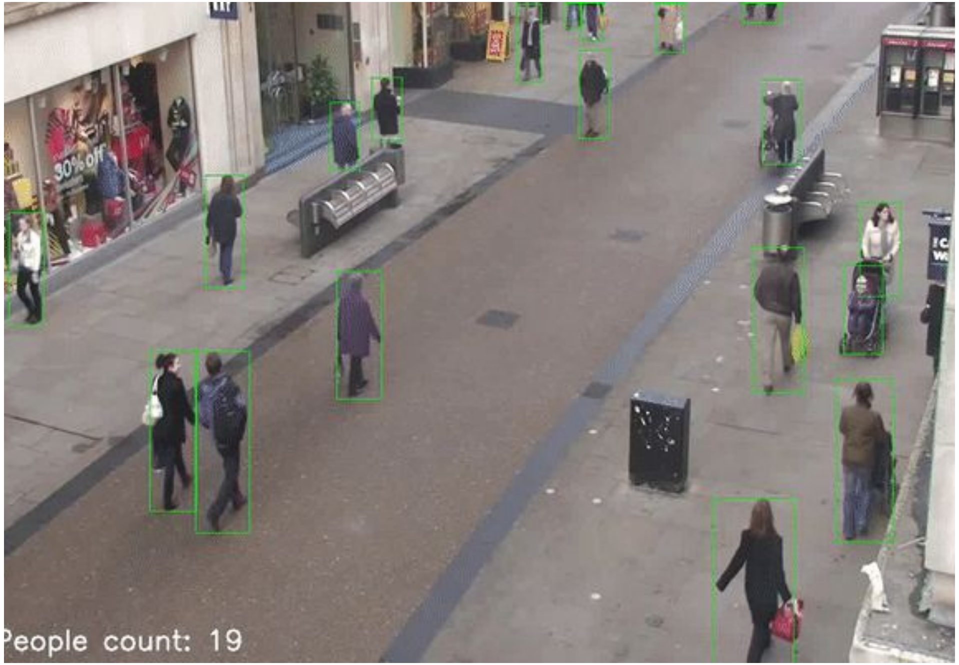
Counting number of people: YOLOv4

YOLOv4 (You Only Look Once) on its 4th version, is the state-of-the-art one stage object detector. It offers **real-time detections** alongside with **high accuracy**.

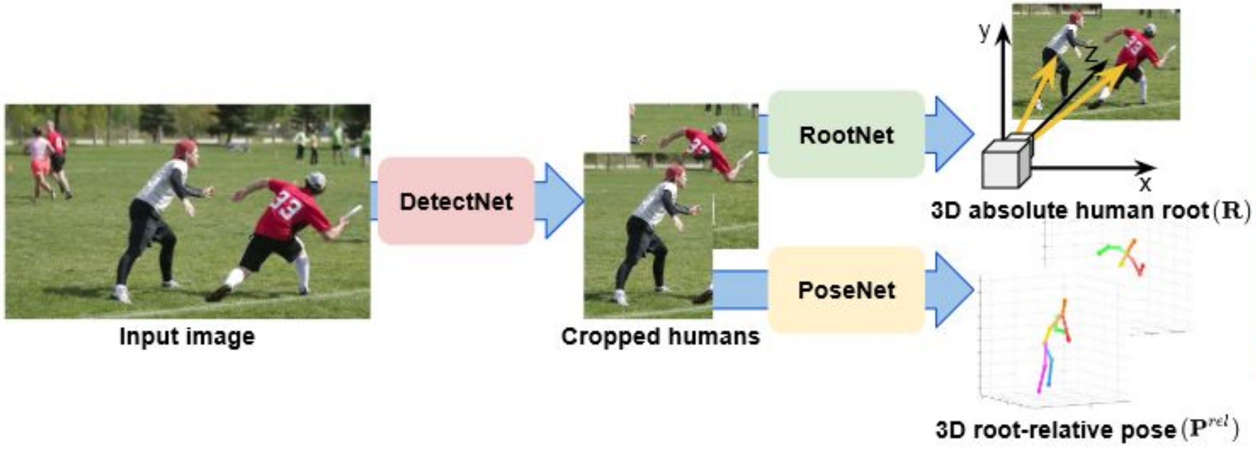
We adopted a YOLO model pre-trained on 117000 images containing objects belonging to 80 classes, including 'human'.

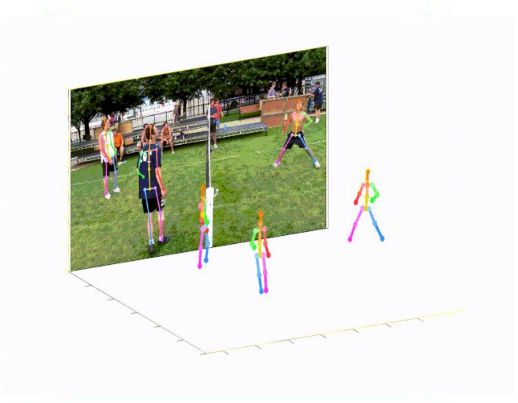


Counting number of people: YOLOv4

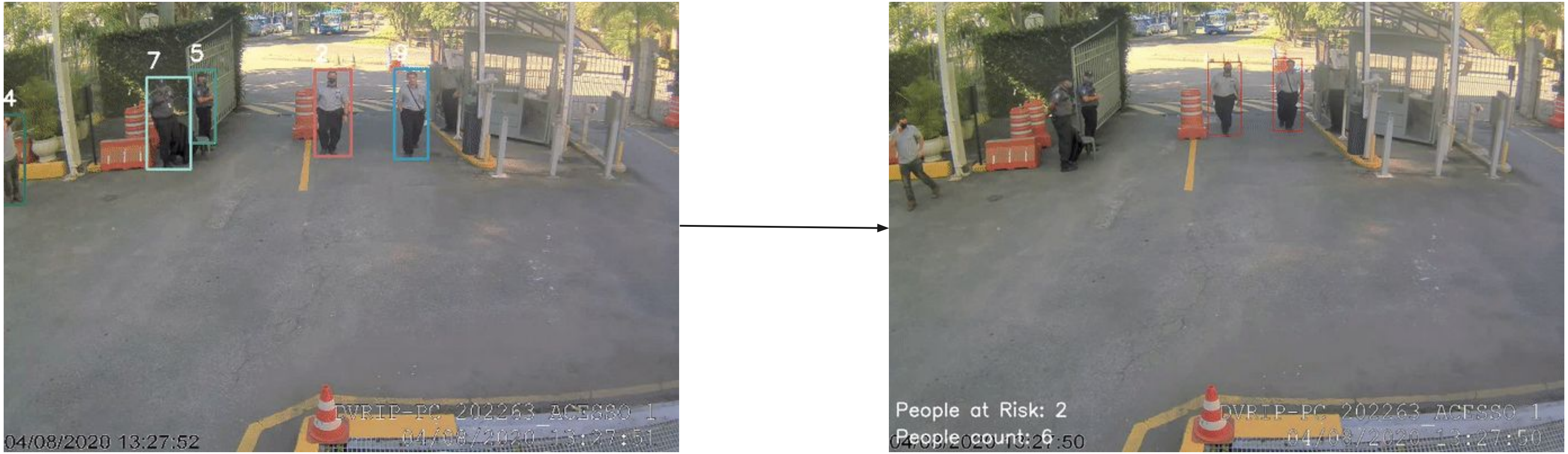


Monitoring social distancing: RootNet


Input image
DetectNet
Cropped humans
RootNet
PoseNet
y
z
x
3D absolute human root $(\mathbf{R})$
3D root-relative pose $(\mathbf{P}^{rel})$



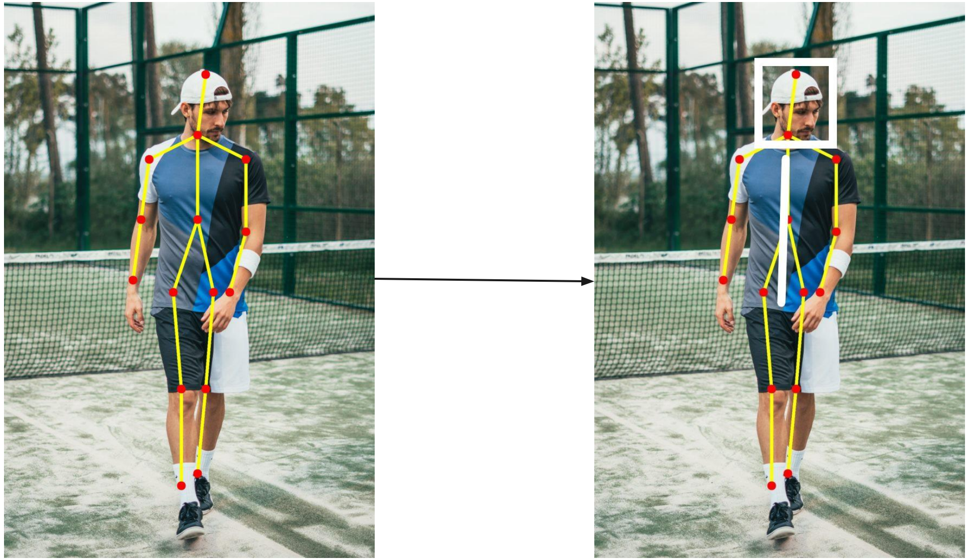
Monitoring social distancing: DeepSort + RootNet



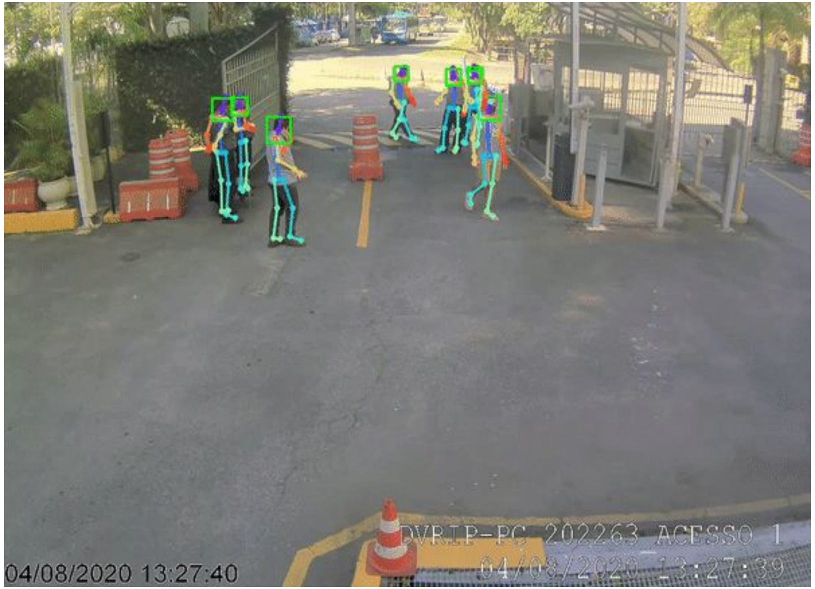
Min. distance: **2 meters** - Num. of frames considered: **3**

Detecting people wearing masks: human pose estimation

The 80% of the line connecting the neck to the hip is the measure of each line of the box including the face, centered on the nose position.



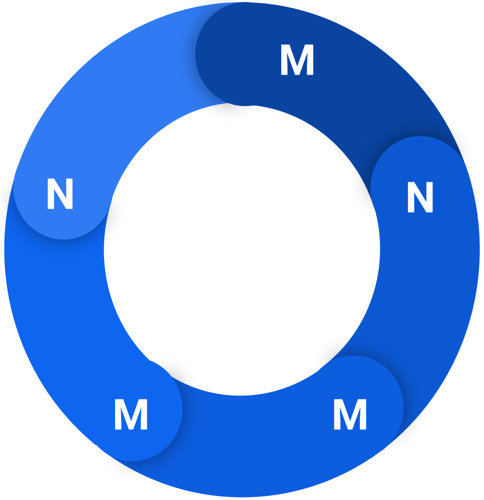
Detecting people wearing masks: HRNet



We consider as valid faces only the ones in which HRNet predicts both eyes with a confidence higher than 80% and the one which dimension is higher or equal than 20x20 pixels.

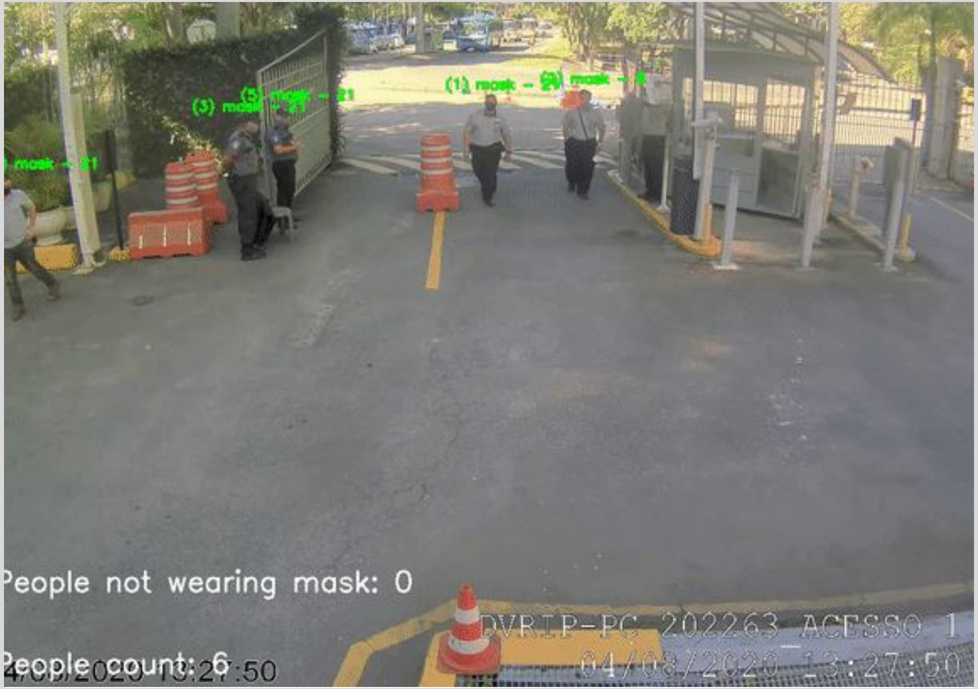
Detecting people wearing masks: DeepSort + HRNet + BiT

1. For each person we create a **circular buffer** of dimension **N**
2. the detected face is given as input to our mask classifier, BiT
3. if the prediction has a **score higher than 80%** (independently by its class), we insert the predicted label in the buffer
4. only when the circular buffer has **more than k votes**, the person is estimated with the most frequent label in the buffer (majority voting mechanism)



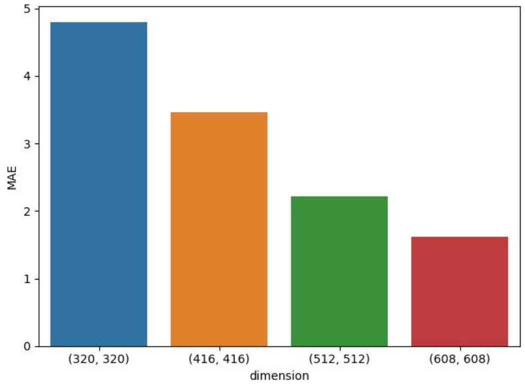
Example of circular of buffer of dimension **5**: the predicted label in this case is *'mask'*, since it is the most frequent label.

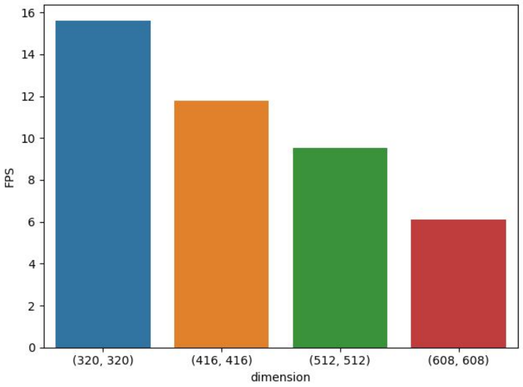
Detecting people wearing masks

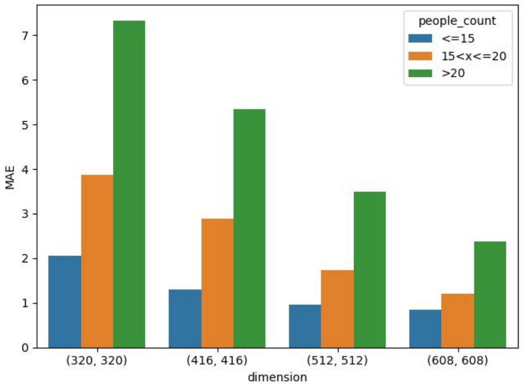


Buffer of dimension N=**21**
k=**3**
eyes threshold=**80%**
prediction confidence=**80%**

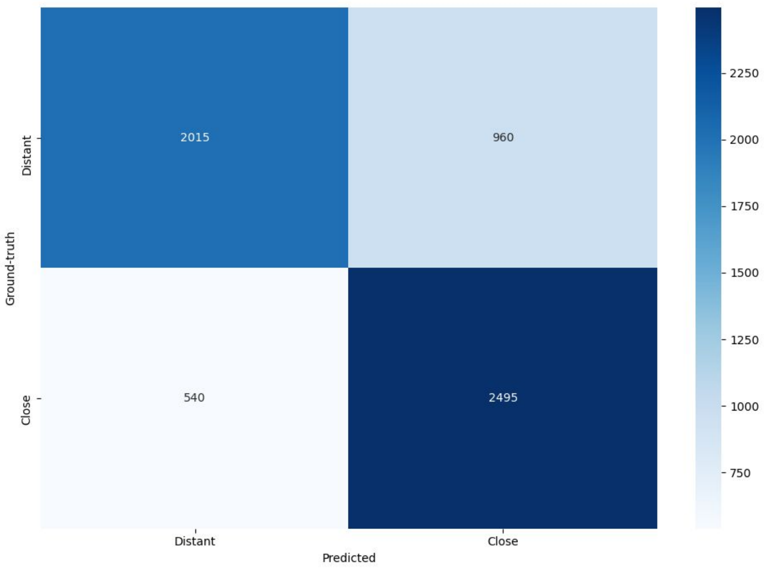
Results: counting people





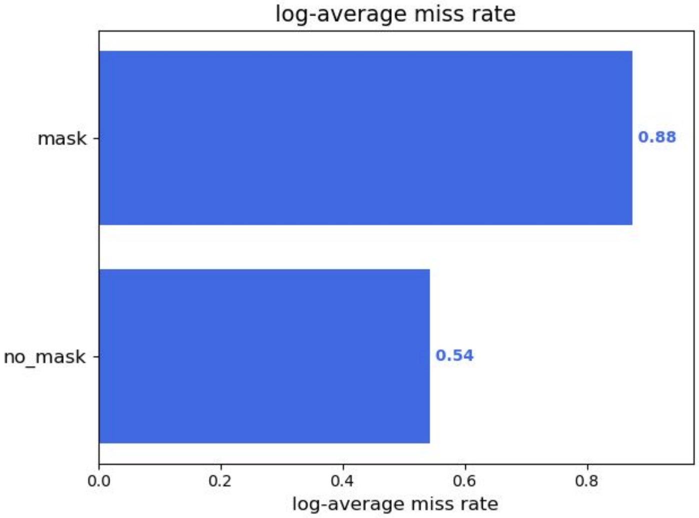


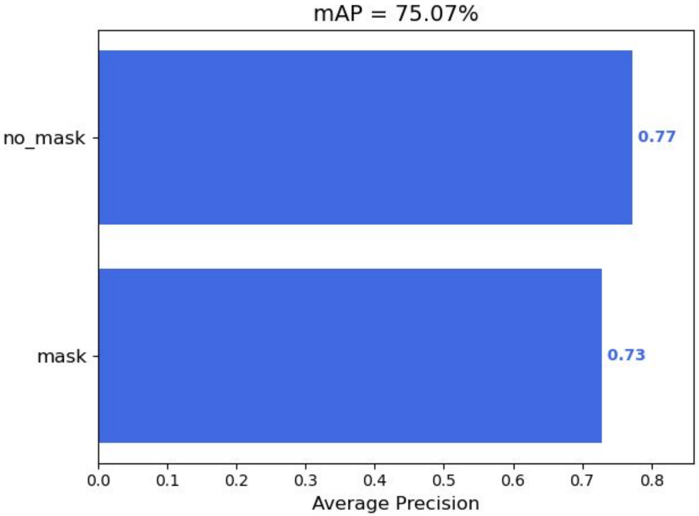
Results: monitoring social distancing



Our model scored an overall accuracy of **75%.** The F1-score for the class *'Distant'* is of **73%** and for the class *'Close'* is of **76%.**

Without using tracking the accuracy scored by the model is of **70%**, while the F1-score got values of **68%** and **71%**, for the class *'Distant'* and *'Close'* respectively.





Results: detecting people wearing masks

Our model scored a *medium Average Precision (mAP)* of **75.07%** on a 5 minutes surveillance camera footage. The Average Precision for the *'no mask'* class is of **77%**, while for the *'mask'* class is of **73%**. Our model outperformed a classical YOLOv4 detector trained on a dataset containing people wearing masks, since it reached a mAP of only **36%** on the same surveillance camera footage.

Future works

1. adopt light-weight approaches in order to deploy the system on embedded machines, with limited computing capacities
2. join our system with some kind of alarms, in order to have an effective way to prevent over crowding

3. use our model as input to calculate some statistics, such as time of day with the highest number of people or percentage over time of people not wearing masks / not respecting distancing

4. implement a classifier capable to detect people wearing mask uncorrectly (not covering the nose for example)

Thanks you

https://drive.google.com/file/d/1nr9jCiH--G8Moj28BMLR1ymm9j823Tme/view?usp=sharing