# Mask-to-mask detection - Final Presentation

Eulalie Boucher, Cédric Javault and Mael Marguet

Ecole Polytechnique

December 18, 2020

# Overview

# Presentation of our project

## Steps

- 1. **Detect people** and assign bounding boxes
- 2. **Assess whether a mask is worn or not** + assign BB to the faces.
- 3. **Associate People and Face** + propagate mask/no mask information to the persons
- 4. **Track the people** across the different frames of the video
- 5. **Estimate depth** in the images, compute 3D positions of the detected people, and **estimate the inter-person distance**.
- 6. **Output the main results in a new video** so that it is easy to understand what the computer computed.

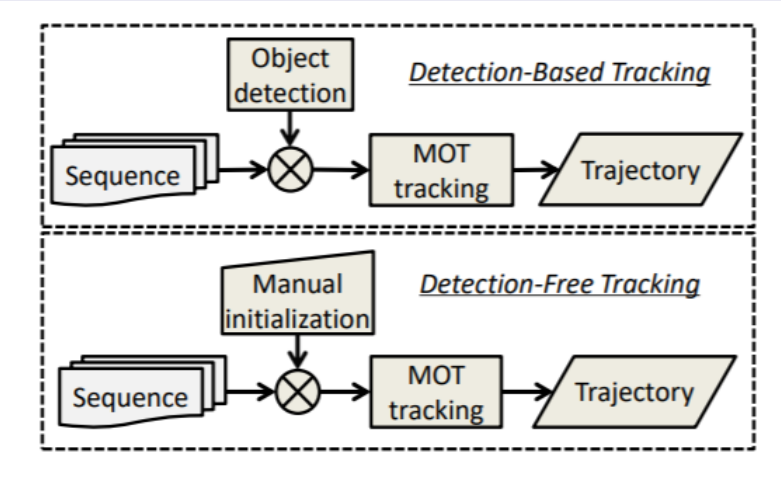
# Methodology: YOLO - You Only Look Once

## Reminder

- Reframes **object detection** as a **regression problem** : from image pixels to bounding box coordinates and class probabilities
- **Single convolutional network** predicts both multiple bounding boxes and class probabilities
- Uses **features** from the **entire image** to predict each bouding box
- Predicts all bounding boxes **at once**

# Methodology: Tracker

## Everything you need to know about Detection-based Tracking




It consist in **associating detected objects** across subsequent frames. What you have at each frame: **BB from detections at frame T** + **target BB generated thanks to previous frames** (ie trackers).

# Methodology: Social Distancing

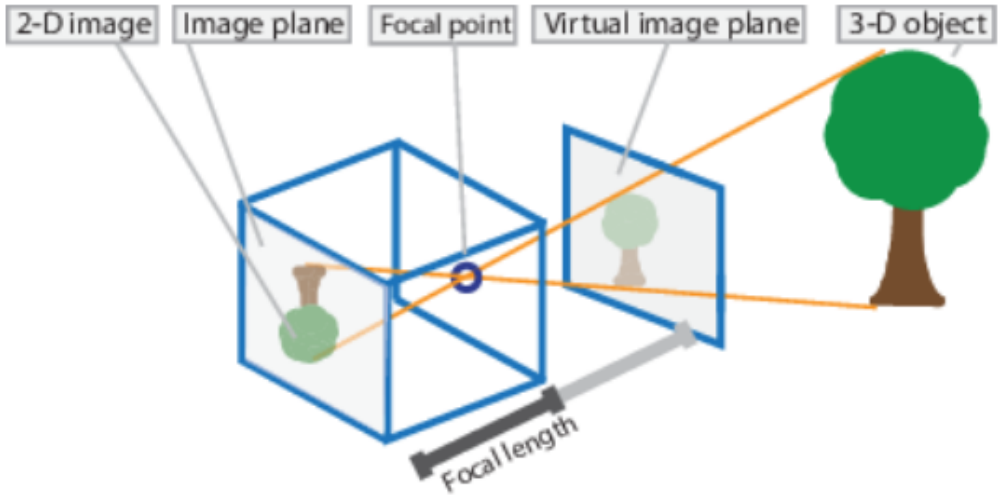
## Social distancing

Our goal is to output a distance between two persons. To do so:

1. **Estimate the distance from the camera to an object**: non AI-based methods exists but we based our approach on a AI-based paper ('Digging into self-supervised monocular depth estimation 2018') which outputs a dense pixel-wise estimation for each image
2. **Estimate the 3D coordinates** of the different objects. You need the camera intrinsic parameters (K matrix) to do it.
3. Then, the distance between the objects is straightforward.

# Methodology: Social Distancing

This is the pinhole camera model:




$$w\,[x\ y\ 1] = [X\ Y\ Z\ 1]\,P$$

Scale factor — Image points — World points

$$P = \begin{bmatrix} R \\ t \end{bmatrix} K$$

Camera matrix — Extrinsics: Rotation and translation — Intrinsic matrix

$$K = \begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

1. Presentation of our project

2. Methodology
   - Object Detection with YOLO
   - SORT Tracker
   - Social Distancing

3. Implementation
   - Implementation: Merge YOLOs
   - Tracker & Post-Processing
   - Social Distancing

4. Results
   - President Sarkozy
   - Streets of Paris

5. Conclusion

# Implementation: Merge YOLO

## YOLOs

Two YOLO algorithms:

1. **People detection**: already trained on COCO
2. **Face/Mask detection**: trained by us on Kaggle dataset

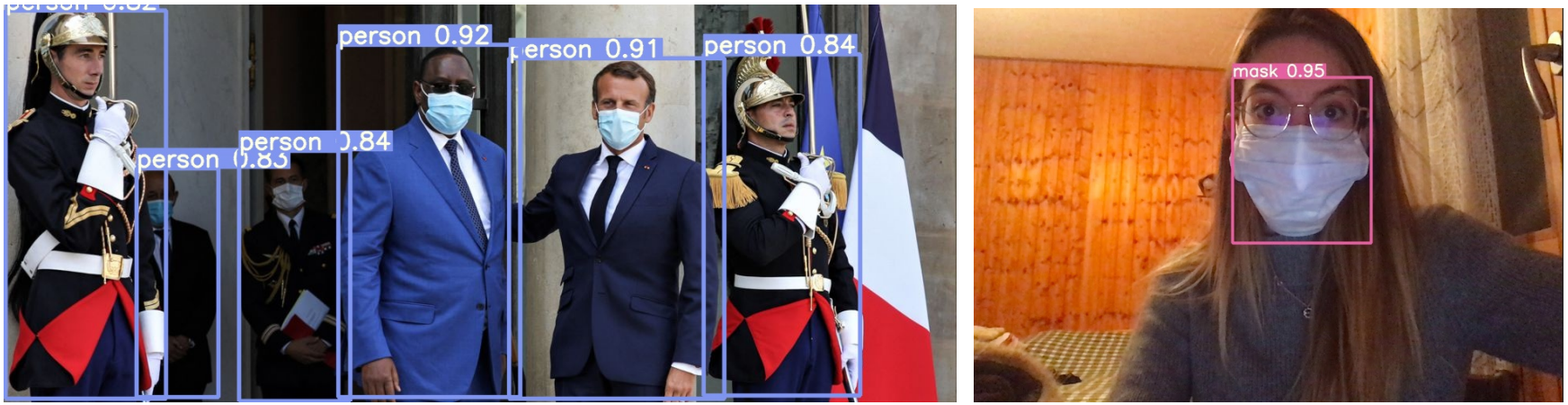



Figure: Example of output of our YOLOs. Left: people detection. Right: Mask detection.

# Implementation: Merge YOLO

## Merge YOLO algorithm

Works as follows **in each image**:

- For each detected person, look at all the bounding boxes of the faces/masks **intersecting** the bounding box of the person
- **Compute score** based on $IoU \times Confidence$
- Keep **best face**

Outputs :

- A class: **Mask, No mask, Unknown** (if no face was found)
- The bounding box of the people
- The bounding box of the face (or again the one of the people for the 'Unknown' class.

# Implementation: Merge YOLO

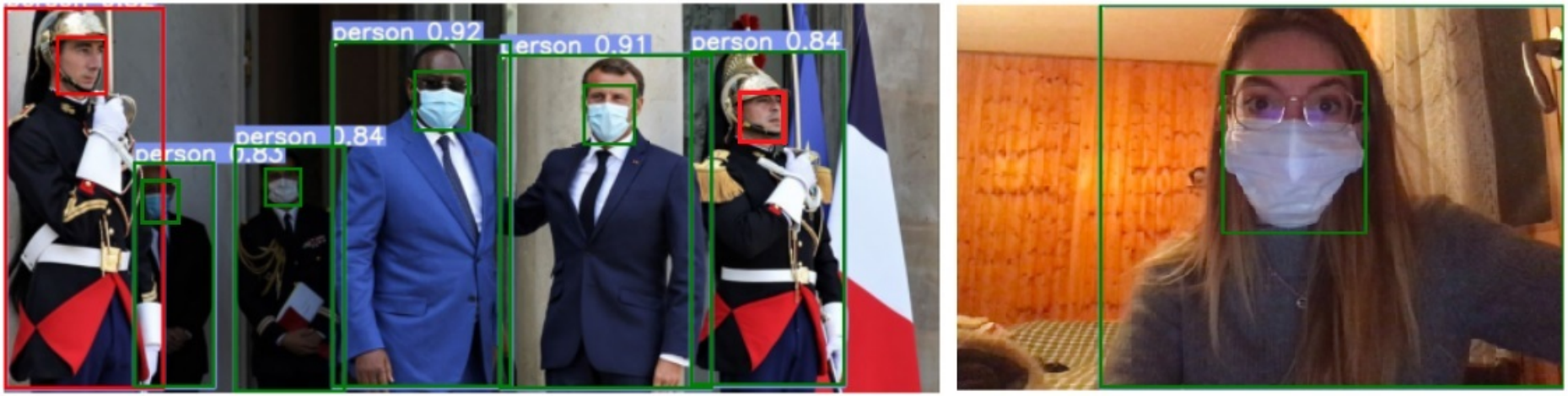



Figure: Output after the Merge Yolo. Here, all the faces are detected and everyone is either red (no mask) or green (mask).

# Implementation: Tracker & Post-Processing

## Tracker

We use:

- **SORT tracker**: Git implementation
- $T_{lost} = 0.7s$ ie 20 frames

## Post Processing

We make **two changes** to the original tracker:

1. Create **another output** (a text file containing the final bounding box coordinates) to serve as an input for the social distancing algorithm.
2. **Back-propagate the majoring class** to all frames in which an object appears

# Implementation: Tracker & Post-Processing


Figure: Frames 860, 886 and 898 of our second video. Above: result of the 'Merge YOLO' before the tracking. Below: same after the tracking: ID are propagated and the children are always from no 'No mask' class (ie red bounding boxes)

# Implementation: Social Distancing

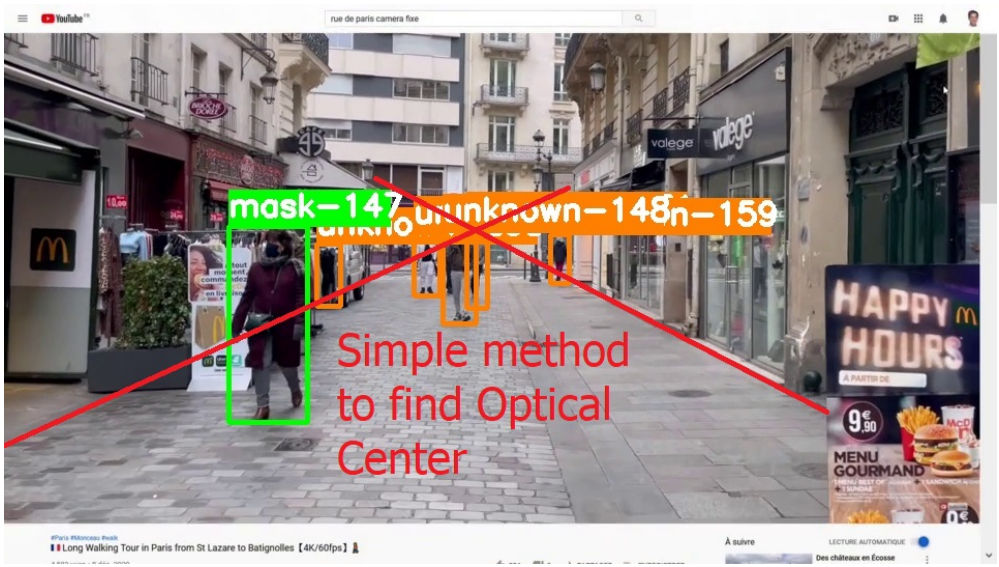
## Problem 1: author's code not well commented

- We had to perform **extensive checks** to be sure that we properly understood the results.
- For example, the values given by the code are not the distance to the camera but its inverse,
- And an unknown scale factor is included. To tune it, we computed the **average speed of the pedestrians** and set the scale factor so that this average speed was 4.5 kilometers per hour.

# Implementation: Social Distancing

## Problem 2: unknown camera parameters

- **Position of the optical center**: can be deduced from the intersection of the vanishing lines
- **Focal length** $f_x = f_y$: we set it so that the width of the street fits our estimation of 5 meters.
- **Skew coefficient**: was put to 0 (image axes are perpendicular).



# Implementation: Social Distancing

## Problem 3: lack of precision

- Distance to the camera was **better estimated by BB of the person** than by BB of the face.
- Predictions were smoothed by **averaging over 20 frames**.

1. Presentation of our project

2. Methodology
   - Object Detection with YOLO
   - SORT Tracker
   - Social Distancing

3. Implementation
   - Implementation: Merge YOLOs
   - Tracker & Post-Processing
   - Social Distancing

4. Results
   - President Sarkozy
   - Streets of Paris

5. Conclusion

# Results

## We worked on 2 videos

- **First Video**: President Sarkozy walking.
  +The tracking is efficient.
  +The processing solves a misclassifications problems, of the 2 body guards at the end.

  -MisClassifications of a tie
  -Tracking losing tracks of a people (due to the processing)
  -Depth estimation is not possible here (no vanishing lines to exploit & defocalization during filming process).

# Result: Depth map issue for video of Mr. Sarkozy

# Results

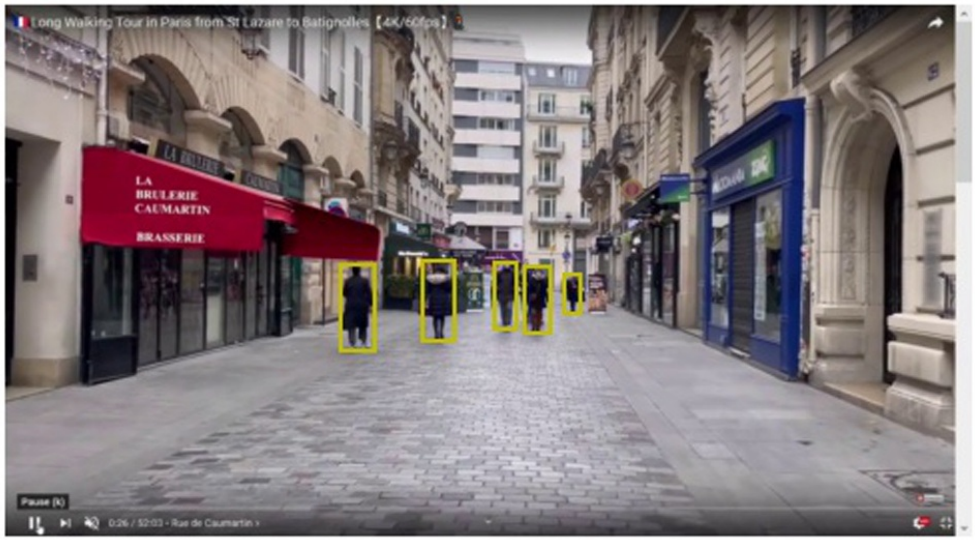
## We worked on 2 videos

- **Second Video**: Paris Walking Tour.
  +The tracking is efficient.
  +The processing solves a mis-classifications problems, of the 2 childrens at the middle of the video.
  +Depth estimation is well applied. Difficult to evaluate the accuracy of the measure, however, when 2 people come close to each other the distance gets really close, and increases when they go away.

  Small issues: Detections of reflected person

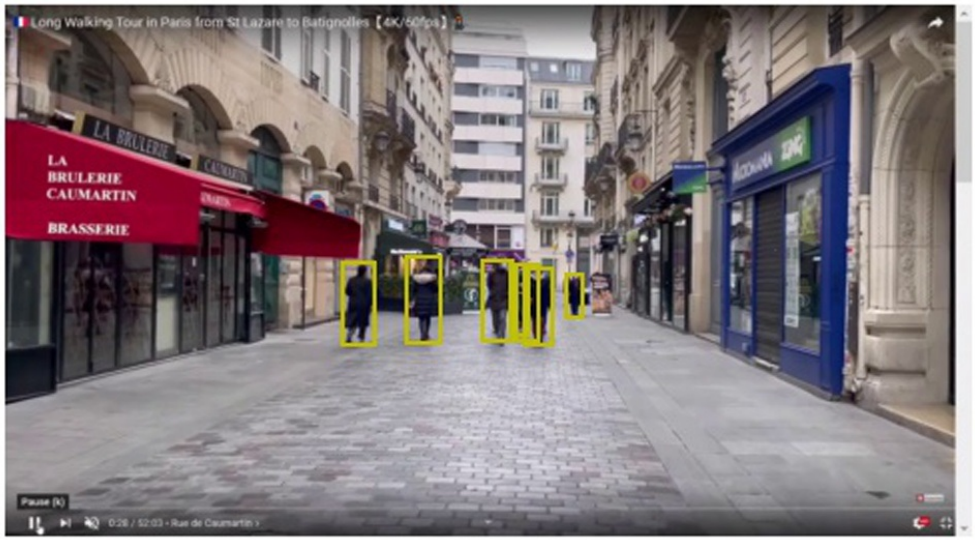
# Result: Streets of Paris - frame 713



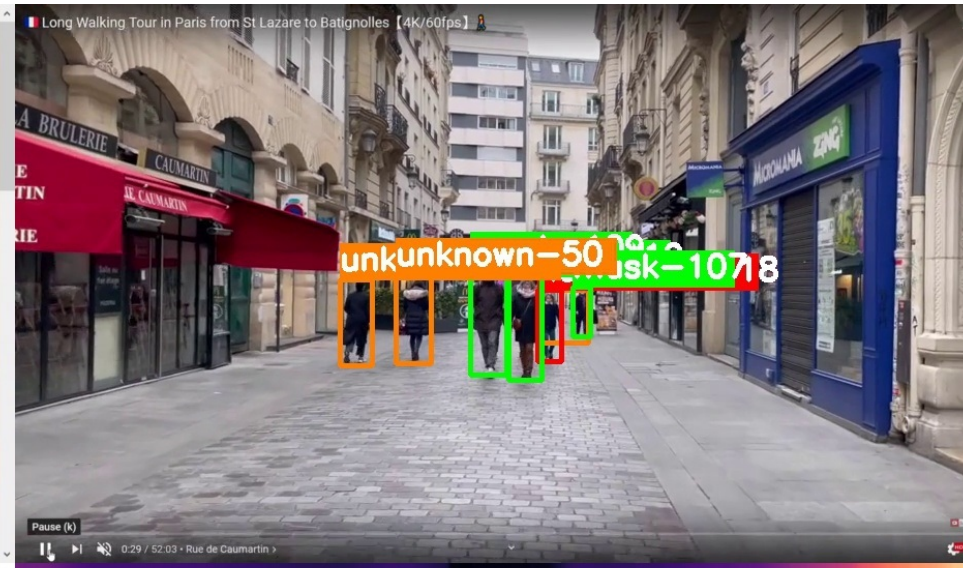


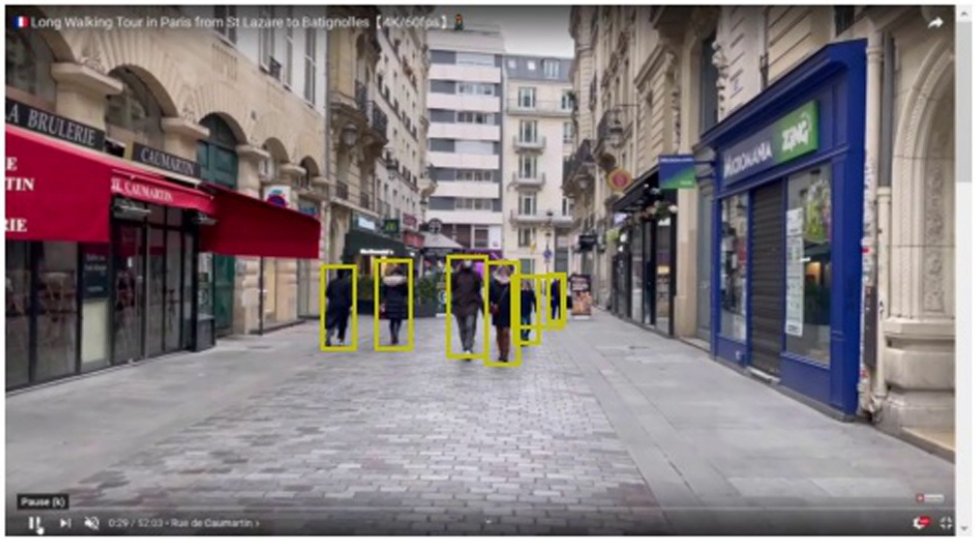

# Result: Streets of Paris - frame 753






# Result: Streets of Paris - frame 783


# Conclusion

-**People/Mask Detector**: **Robust**.

-**Tracker**: Hold its promises (simple and efficient), and **robust**! Lack for long occlusions, but it is a trad off on $T_{Lost}$.

-**Depth estimator**: Works well under **certains asumptions** (with presence of vanishing lines, w.r.t parameters estimations).It could be a source of improvement.

-**Next Step** of our project: Make it **realtime !!**
Tolerate few seconds latency to work with batch-tracking. This would an almost real-time rendering, and permits back propagation of the labels.