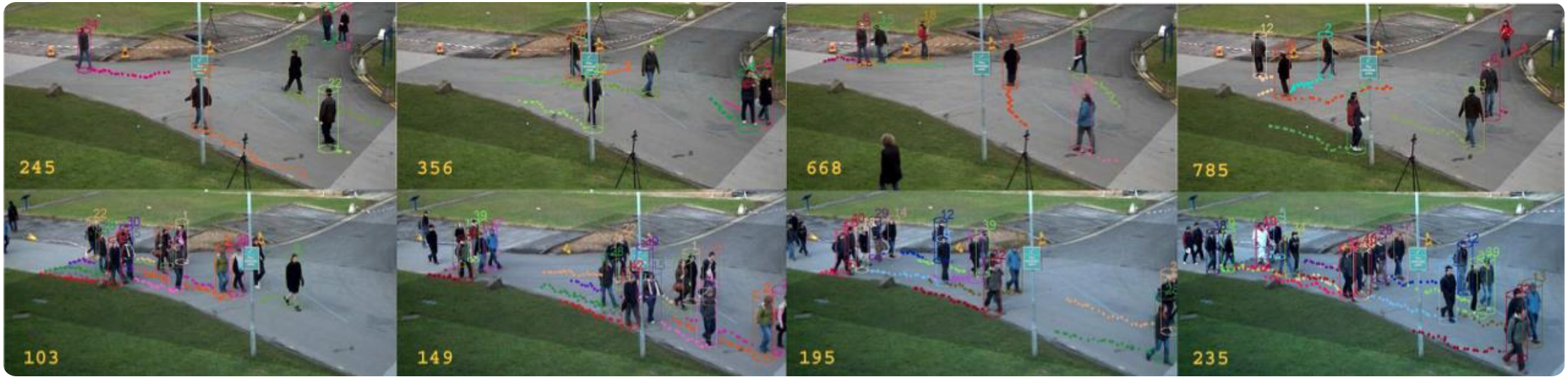
Multiple Object Tracking (MOT)

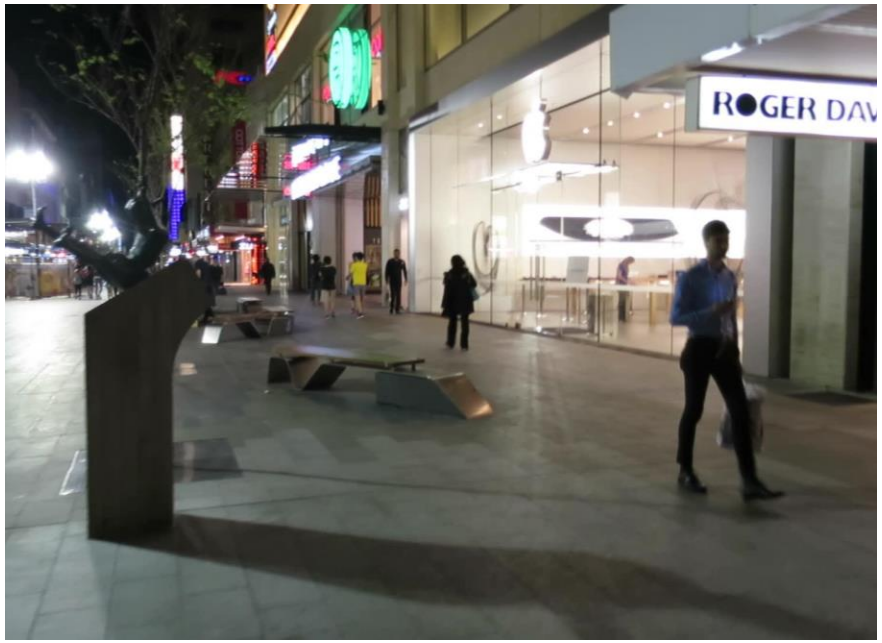
- Team Name: BlindPC
- Team Members:
  - Siddhant Bansal (2019900091)
  - Madhav Agarwal (2020900022)
  - Piyush Singh (2020701031)
  - Himanshu Kumar (2019201094)

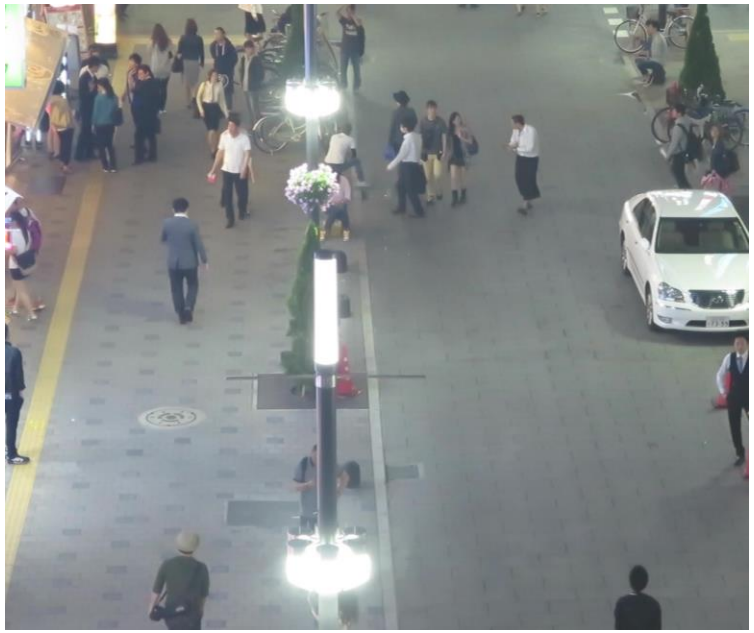


Objective

- Determining the trajectories of multiple object instances in a video;
- Use appearance information with Kalman Filter and a deep association metric.

- Figure credits: Z. Wu, A. Thangali, S. Sclaroff and M. Betke, "Coupling detection and data association for multiple object tracking," CVPR 2012.





Overview of MOT Dataset

Steps for Multiple Objects Tracking

Object Detection
(Bounding Box)

Appearance Descriptor

Tracking

Method Overview

- Tracking using appearance information;
- Inputs:
  - Set of object detections from MOT.
  - Appearance descriptor of each bounding box using a CNN.
- Output:
  - Assign unique id to each person and track their movement.
- The figure here shows example input frames for tracking pedestrians.

- Figure credits: Brasó, Guillem, and Laura Leal-Taixé. "Learning a neural solver for multiple object tracking." CVPR 2020

Method Overview

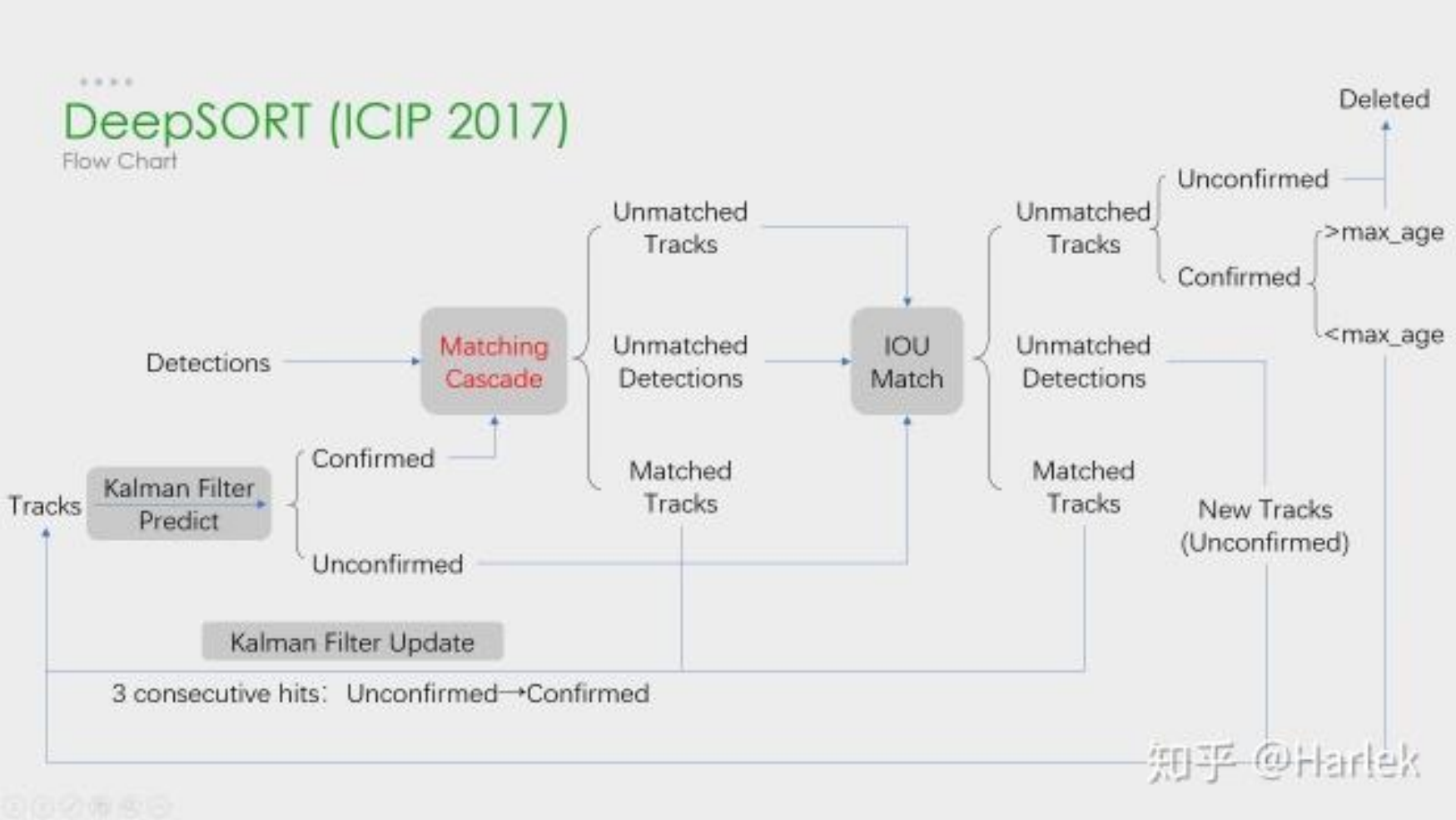
- Kalman Filter is used to estimate the state of the tracker given by an 8-dimensional vector (x, y, a, h, vx, vy, va, vh)
- Mahalanobis Distance is used for predicted Kalman state and newly arrived measurements
- Hungarian Algorithm is used to match trackers in consecutive frames.
- Offline pre-training is performed to learn a deep association metric on a large-scale person re-id dataset.

Method Overview

Deep Association Metric is trained using three models each of which are obtained by a training a Person Re-Id task (Mars dataset)

- Cosine Metric Learning (proposed in the paper): This model uses cosine similarity for metric learning.
- Triplet (Ours): This model samples triplets from dataset and the training uses triplet loss.
- Magnet (Ours): This model also uses triplet sampling. But the loss formulation is bit different from triplet loss.
- Each of these is based on (Person + Tracklet) combination as class label and trains a Siamese kind of network.

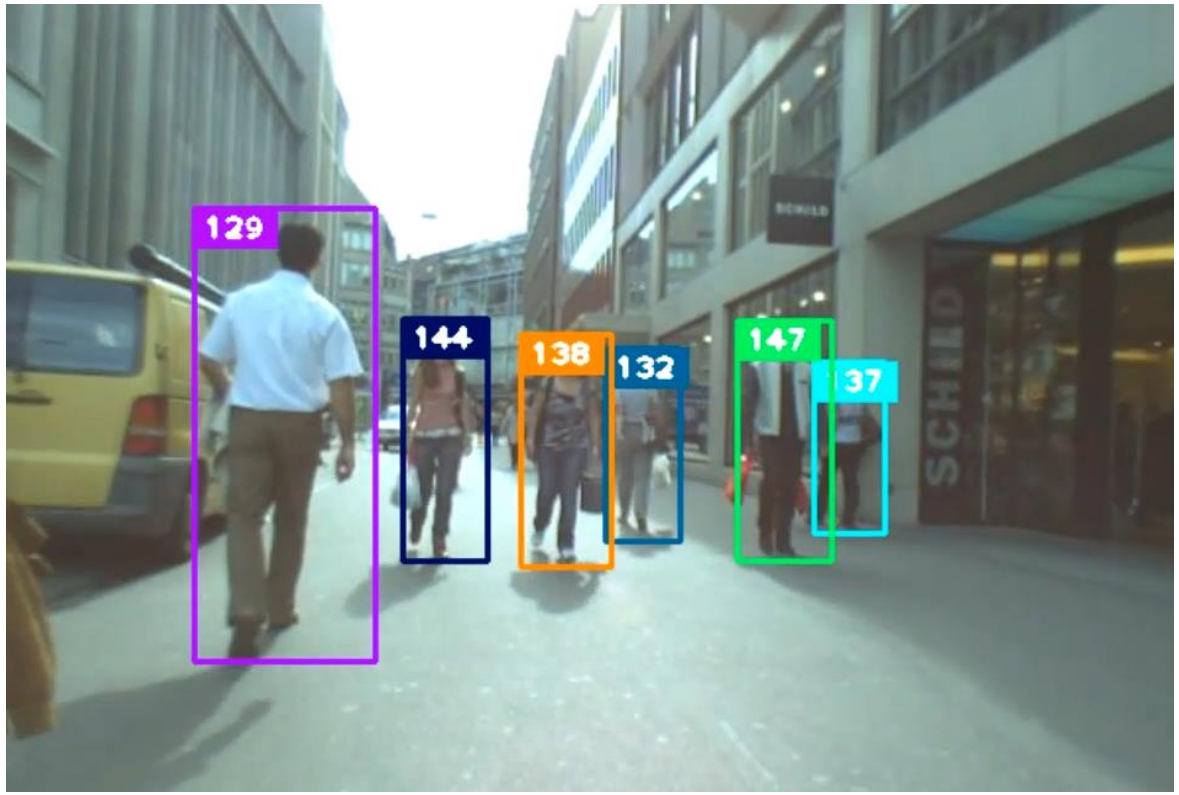
DeepSort: Flow Chart

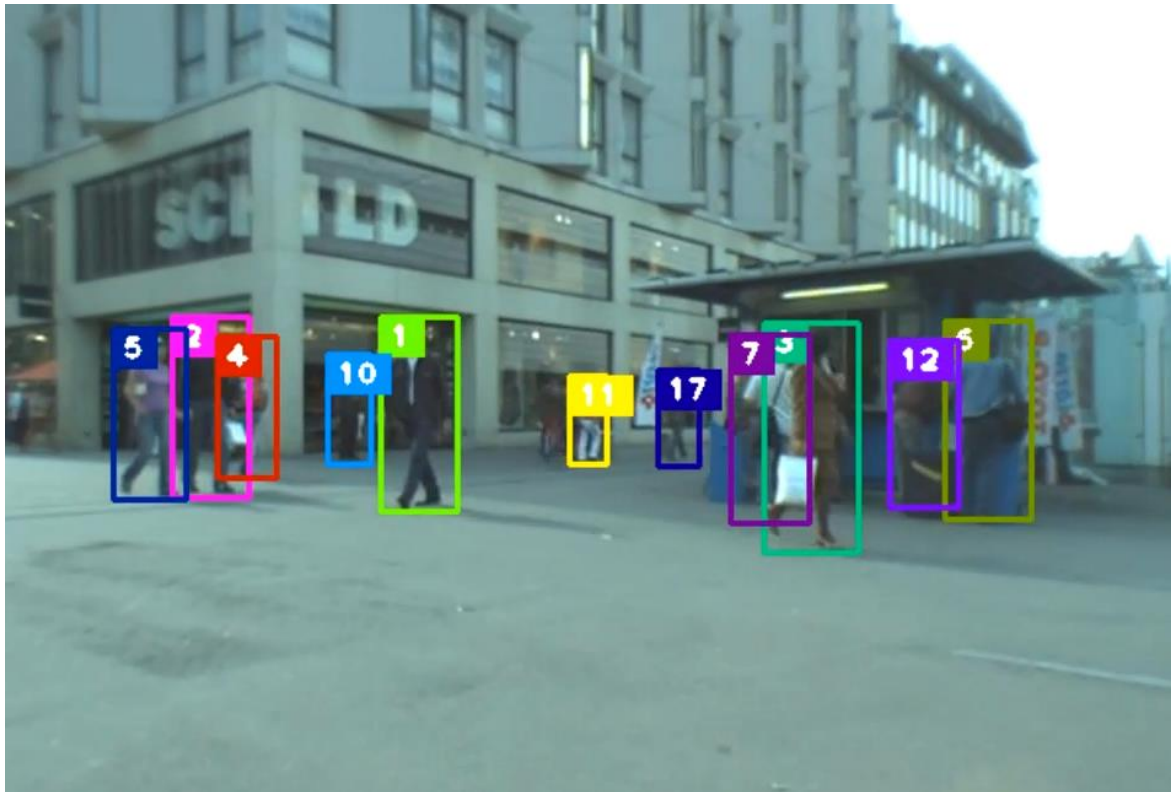


- Figure credits: https://www.programmersought.com/article/17005126187/

Real Time Tracking

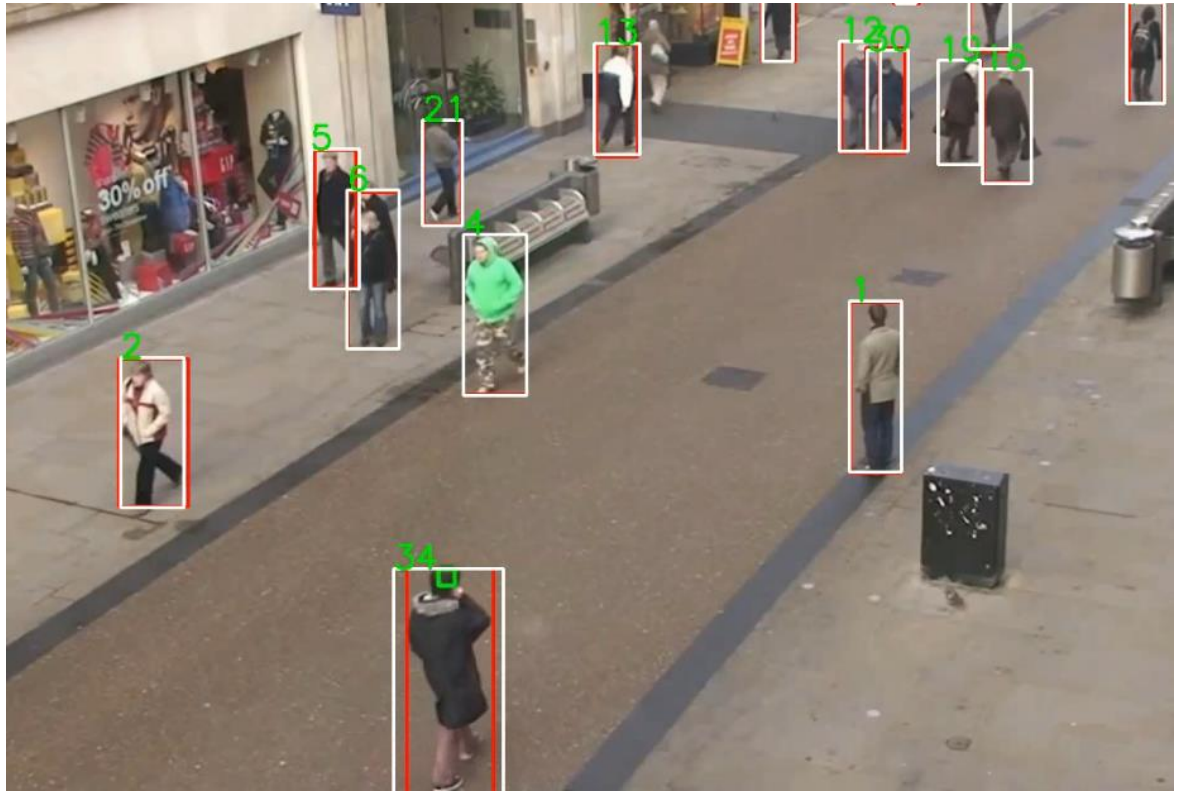
- We integrated the YOLO algorithm to the pipeline for generating real-time and accurate person detections.
- Yolo implementation by AlexeyAB in Darknet was used for generating the detections.
- The detections generated were then processed and passed to DeepSort algorithm.
- Finally, the DeepSort algorithm assigns tracking ID to each detected person.

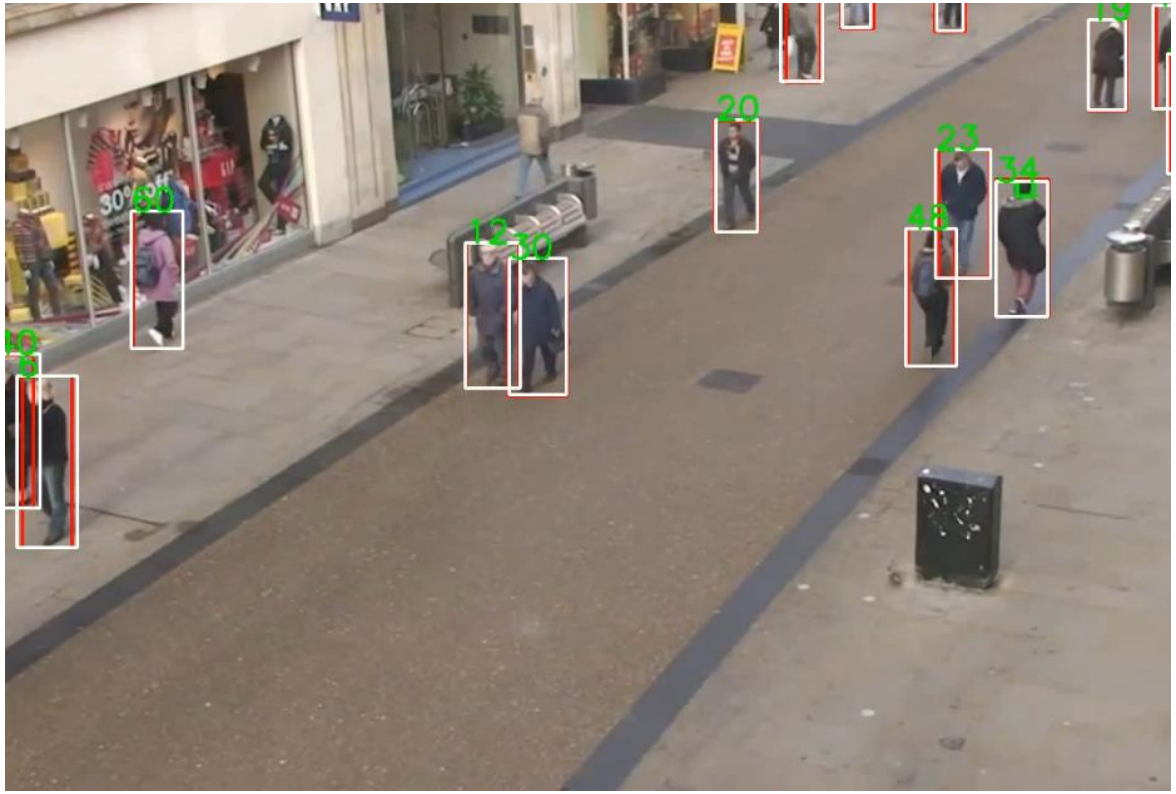




Results

Visualization of DeepSort on MOT-16

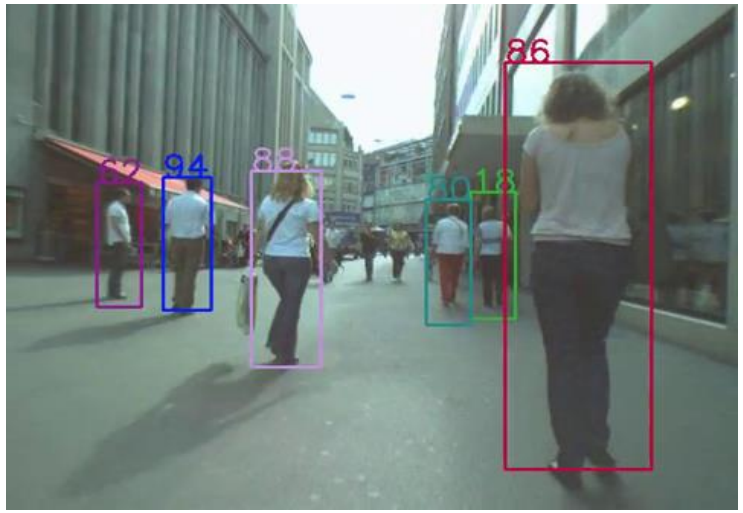




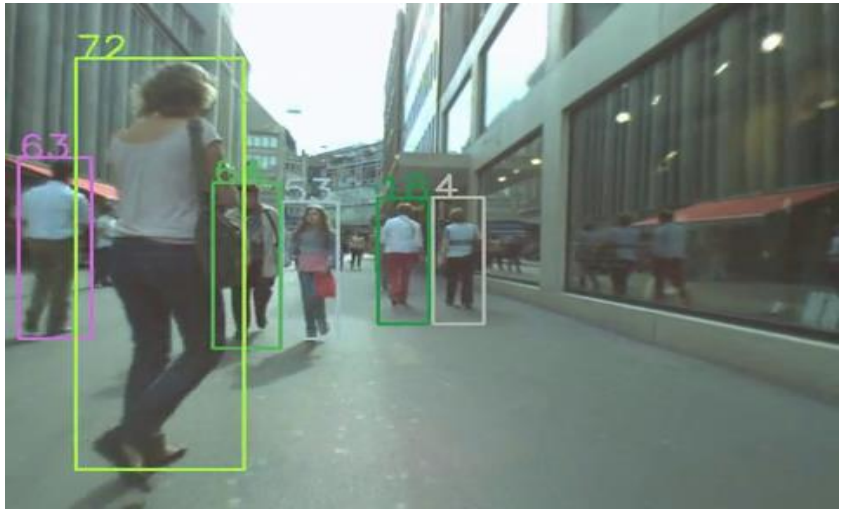
Results

Real-Time tracking using Yolo and DeepSort

DeepSort on MOT-16



Triplet



Magnet

Cosine Metric

Triplet: https://web.microsoftstream.com/video/c55824f4-4ec9-43ce-861e-3b8f488a295f

Magnet: https://web.microsoftstream.com/video/8372f62b-d489-4254-a313-cb54e5eb34c0

Cosine Metric: https://web.microsoftstream.com/video/9732b28c-713d-491a-bc6d-5524a52e4504

Real-Time tracking using Yolo and DeepSort

Original Video

Tracking using Yolo and DeepSort

Original Video: https://web.microsoftstream.com/video/fdfe3bfc-e67b-49dd-a1a1-c8ad4679114d

Tracking using Yolo and DeepSort: https://web.microsoftstream.com/video/b562302e-852c-43ca-b58c-00e97091c402

Individual Contribution

Name	Tasks
Madhav Agarwal (2020900022)	Implemented DeepSort tracker: tracker.py, Kalman Filter, Nearest Neighbour Distribution, Darknet setup (for Yolo): darknet_demo.ipynb, yolo_with_deepsort.py, Demo Videos, Benchmark Results on MOT16 using cosine_metric, result_to_video.py, end-to-end integration and bug-fix of entire code, Documentation (ReadMe), Presentation (ppts)
Siddhant Bansal (2019900091)	Deepsort app (app.py), visualize.py , NonMaxSupression (nms), Presentation (ppts)
Himanshu Kumar (2019201094)	freeze_model1.py, detection_process.py, Presentation (ppts)
Piyush Singh (2020701031)	Triplet model results, magnet model results, bug fix (detection_process.py), Presentation (ppts)

Thank You