CS436/5310/EE513 Computer Vision Fundamentals

Instructor's Name: Murtaza Taj
29 October 2020

Multi-camera System

Mini-Project 1

Requirements

1. 3 or more smartphones or IP webcams
2. Internet
3. Github account: You are required to maintain your code using Github repository

The overall idea of the project is to combine visual recognition and multi-camera geometry and build a useful application. You can chose a scenario of your choices it can be a home,, shop or outdoor. Irrespective of the scenario area you choice you have to perform the following 6 tasks and deliver them at the mentioned deadline.

Project Proposal

You can complete the project in a group of 2 students
You must form the group within this week and submit your proposal by mid-night Tues 2nd Nov 2021 via following Google Form link:
https://forms.gle/aomviV9Vrw19Bm6o6

Task 1: Multi-camera Setup & Recording

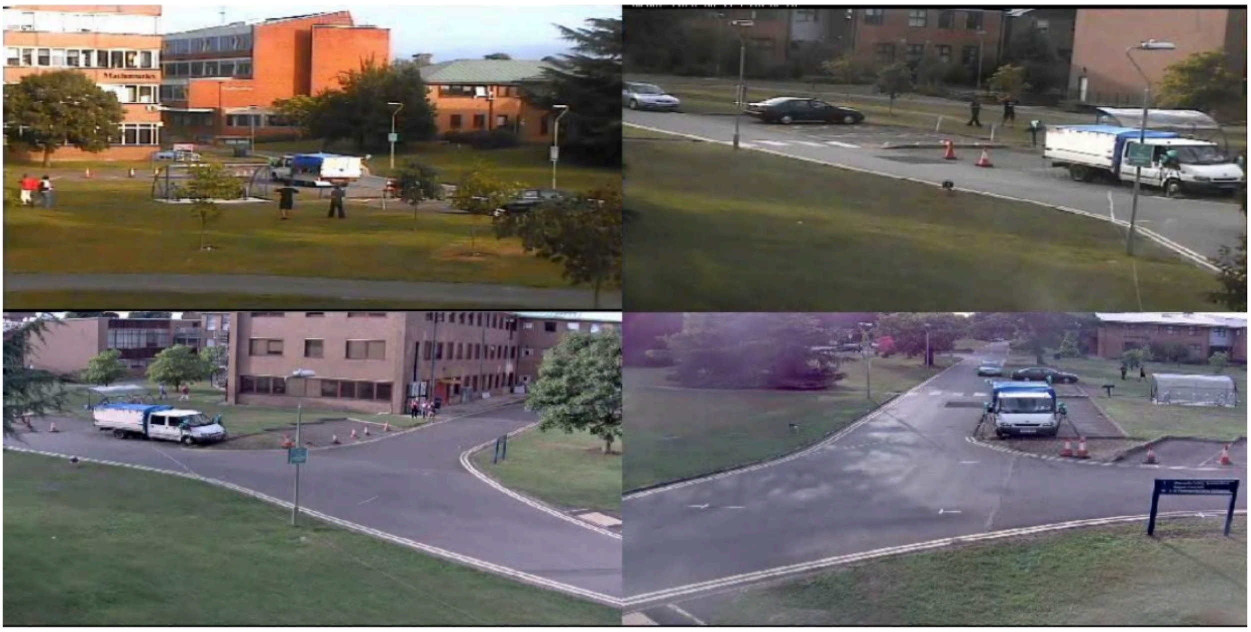
1.1 Collect three cameras within your home (smart phones, laptop camera etc) or from your friends, project partners etc. In the worse case scenario you can search for a multi-camera dataset on the web and use it for the project but that would result in deduction in marks.

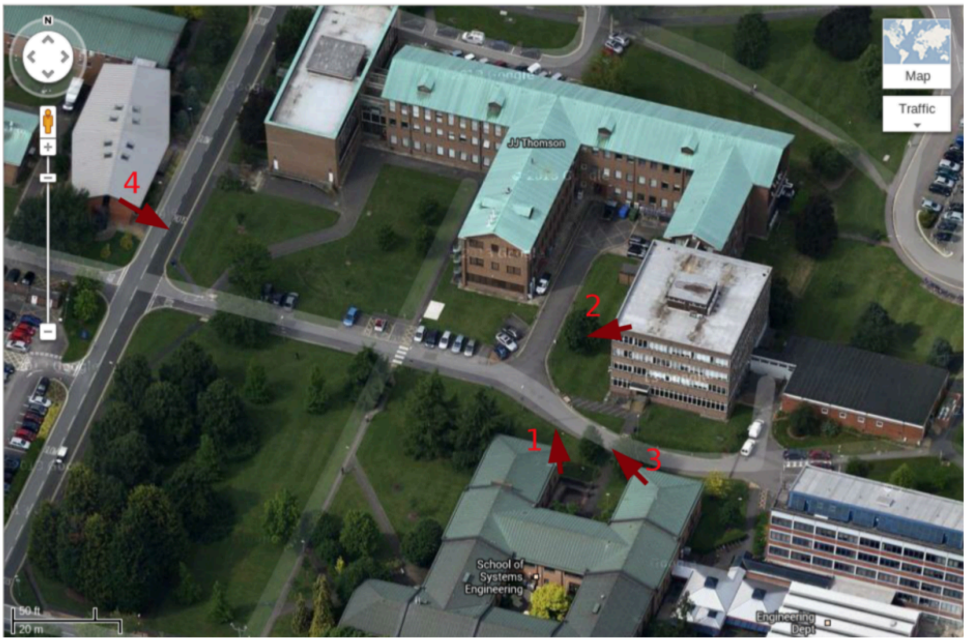
1.2 Install IP Webcam app (or other similar app) on each of the smart phone

1.3 Open camera on smart phones and test the video feed on your laptop using VLC player

1.4 Use the given python and OpenCV code to record the feed from all these cameras on your hard drive. The program should also optionally display the live feed from all three cameras.

1.5 Once the recording setup is ready and tested mount the cameras at three different locations (for e.g. see image below)





Task 2: Mask/Non-Mask Person Detection

In this task you are required to detect people and also classify them as those wearing masks and those who are not wearing mask. To classify mask/non-mask people you can create a dataset of faces only followed by binary classifier or you can update YOLO to directly give localisation and classification.

2.1 Setup and run YOLO object detector

2.2 From the recorded videos annotate few people or faces from each of the camera view using LabelMe software

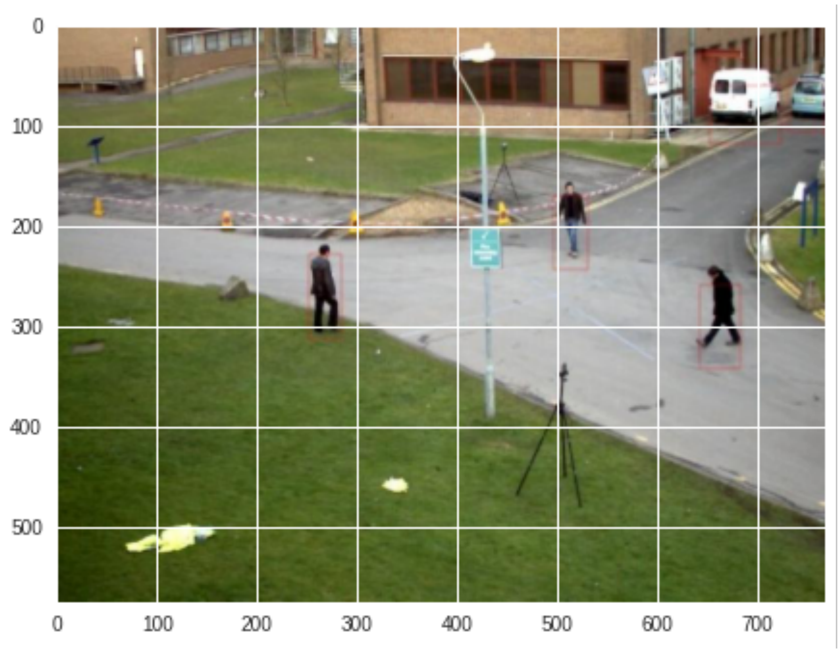
2.3 Either fine tune the YOLO object detector using the generated annotations to detect separate bounding boxes with separate class information for mask/non-mask persons. Depending upon your data you may fine tune *n* different versions of YOLO one for each camera view

OR

Train a classifier to distinguish between faces with mask and those without mask.

2.4 Write a program that takes perform "Mask/Non-Mask Person Detection" on the video feed obtained in Task 1. The program should have two modes online and offline. In online mode the program should read a frame from the live feed from each camera and perform detection and classification and display the output from all the cameras simultaneously on the screen as well as record the bounding box position on the hard drive

2.5 Offline mode: In this task your are required to repeat the previous task for pre-recorded videos



Task 3: Orthographics Top-View Generation

3.1 Either using Google maps or otherwise obtain a top-view of the site

3.2 Mark (either manually or using GPS coordinates) the position of each camera on the top- view.

3.3 Find the corresponding points between top-view image and your camera image from first camera

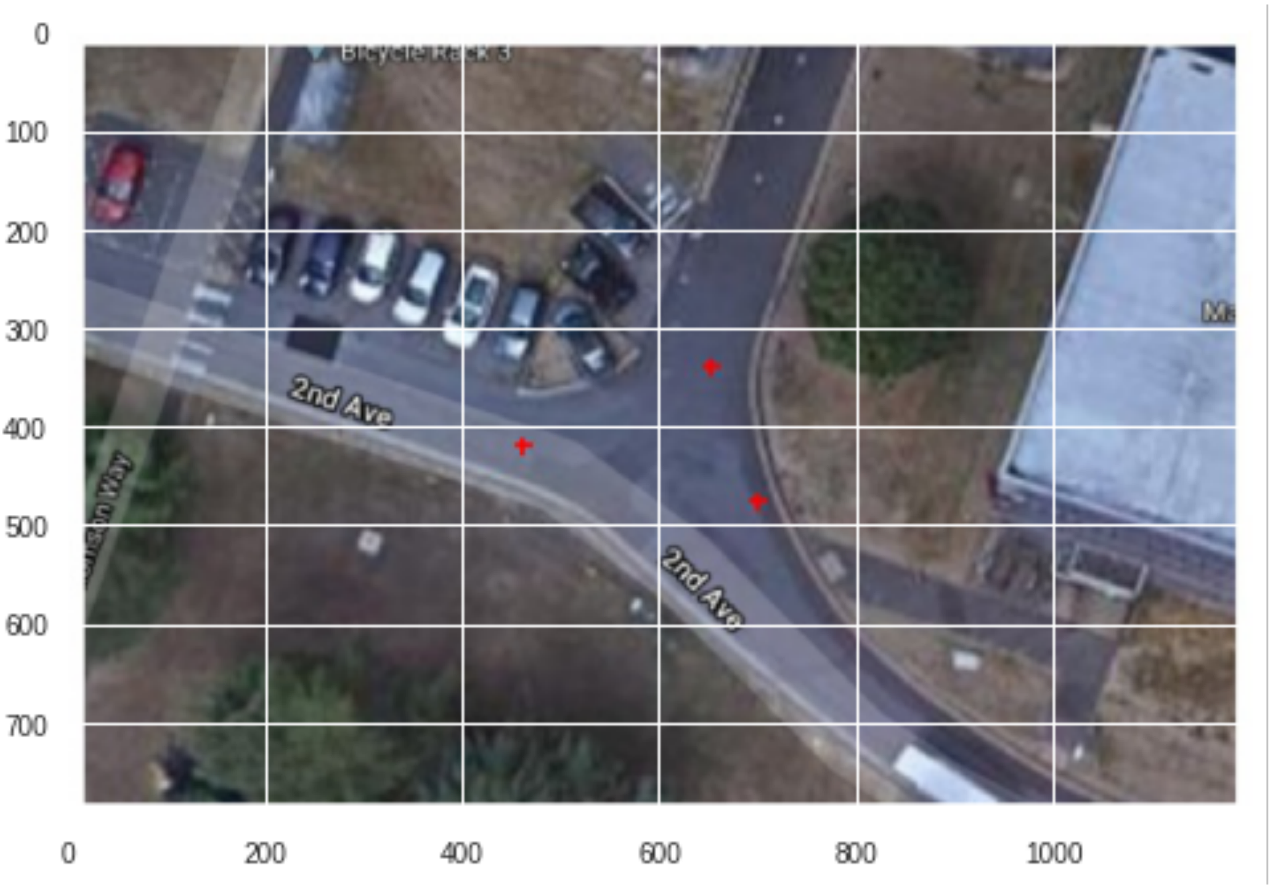
3.4 Compute the homography between the corresponding points

3.5 Project the camera view on the top-view using the computed homography and produce a top-view video of the camera view (this option should work for both online and offline modes discussed earlier i.e. live feed and pre-recorded videos respectively)

3.6 Repeat the above process for all the cameras

3.7 Combine all the top-view videos into a single top-view video of the site. In this task a frame from each of the camera will be obtained and projected on the common top-view and then you go to the next frame and repeat the process in this way you will result in a top-view video feed. If there are speed limitations in your own implementation of warping you can use the OpenCV implementation instead. Your final visualisation should show live feed from all the cameras as well as the generate top-view video

Task 4: Visualise Object Detection on Orthographic Top-View



4.1 Repeat task 2.4

4.2 Object detector will give you a list of bounding boxes, note the mid-point of the base of bounding box and project it on to the top view/satellite view using the computed homography

4.3 Save all the top view locations in a separate file. You need to perform all the above mentioned steps for each view separately for both online mode as well as offline mode. Your final visualisation should show live feed from all the cameras with overland object detection results as well as the visualisation on the top-view

Task 5: COVID-19 SOP Violation Detection

In this task you are required to identify group of people who are violation the social distancing policy of COVID-19. Using the real-world position of the detected people identify which set of people are within 2m distance of each other. This can either be done on top-view by manually finding a relationship between real world and pixel level distances.

Task 6: Heatmap Visualization Top-View

6.1 **Static Heatmap:** Using all the recorded mid-point location on the top-view from all the camera generate a heat map. Heat map can be generated by placing a Gaussian with certain mean and sigma at each top-view location. All the Gaussian for all the locations for all the frames should be summed up and normalised by the number of time steps to generate a single heat map.

6.2 Animated Heatmap: In this task you are required to repeat task 6.1 but only for last *k* time steps i.e. from time step *t-k-1* to time step *t*. At time step *t* the heat map should exclude the data from time step *t-k* and instead include the data for time step *t*. This will result in updating the heat map for every time frame thus result in an animated heat map.

6.3 SOP Violation Heatmap: In this task you are required to repeat task 6.2 but only for set of people who are violating the SOP.

#	Description	Deadline on LMS	Submission Marks	Marks	Intermediate Evaluation Meetings
0	Group Formation	02 Nov 2021	0		
1	Setup & Recording	08 Nov 2021	2	8	
2	Person Detection	15 Nov 2021	3	12	1st Evaluation
3	Orthographics Top-View Generation	22 Nov 2021	3	12	
4	Visualise Object Detection on Orthographic Top-View	29 Nov 2021	3	10	2nd Evaluation
5	COVID-19 SOP Violation	06 Dec 2021	3	10	
6	Heatmap Visualization Top-View	13 Dec 2021	3	12	
7	Final Submission	20 Dec 2021	3	16	3rd Evaluation
		Total	20	80	

Each submission on LMS is 2-3% weightage, evaluation meeting are weighted 20%, 22% and 38% respectively. In case you failed to complete the task by deadline on LMS your efforts will still be evaluated in the meetings. 2nd evaluation is for the progress on all the task till that date, similarly final evaluation is for the entire project and not just for heat map visualisation.