PROJECT TRANSCRIPT SPEED DETECTION

hey everyone so uh today i'm going to talk about how to implement a traffic speed radar in python using opencv and this is part of my image processing project at kalinga institute of technology so without further ado i'm going to get started so road safety is an integral part of modern roads and in modern cities we see more and more speed cameras being used so even though some you know drivers would not like speed cameras because it catches them and they end up paying a high fee it is important so as to maintain safety on roads and make sure that people stay within the driving limit and we can try to reduce accidents using this way so initially speed cameras used

to use something called the doppler effect and by measuring the change in frequency we're able to measure how fast a vehicle goes but now speed cameras have a lot more than just measuring frequency they have a lot more sensors and they can find different things such as the speed whether a vehicle is going in the wrong lane or maybe even if someone is not wearing a seat belt or is eating food inside their vehicles so as time passes by speed cameras are going to get more and more advanced so moving on to the literature survey i basically looked into five papers and to summarize these papers uh the main ingredients that i'm going to use for my project is the speed measurement tool so how this is going to be done is

we're going to start the timer at a certain segment of row and end the timer at another segment and using that we'd be able to find the time that's elapsed and based on the distance we can estimate how much speed uh the vehicle is going

in also to find out the vehicle uh we're not going to be using cnn or any kind of artificial intelligence what we're going to use is an image processing concept known as image subtraction

so some of you may already know this image subtraction is used to find the difference between two frames so in a speed camera the background is usually the same the only thing that changes is the vehicles that come in front of the background
so to capture only these vehicles subtraction is able to find the difference between the foreground and background solely based on the movement so moving to the problem statement of the project uh

there are mainly three stages uh main stages and that is the vehicle detection the speed estimation and then capturing the vehicle image so for the vehicle detection

as i said we'll be using uh image subtraction for speed estimation we'd be using distance elapsed which if you didn't understand you'll understand with an image later and capturing vehicle image is basically cropping the image and saving the image of the vehicle now the

reason i'm not getting number plates of the vehicle is because the clarity of the video wasn't that great so we can't see the numbers also in the number plates so because of that i thought that we just get the image of the vehicle and associate it with the speed that it's gone so this explains more about what i said so

here's the diagram for uh here's the diagram for my project and this gives a clear idea of what's going on basically in the video equalization part we define a region of interest this is a small portion of road where we just want to measure movement and the vehicles that go in image recognition and region of interest monitored what

we have is first step is image subtraction which helps us find where the vehicles are masking is used to make the image binary and by that i mean it will make it black and white or ones or zeros so the vehicles would be uh white and the background would be black and with that we will be able to pinpoint where the vehicle is

erosion i used basically to distinguish the foreground uh in the background sometimes what happened is when two vehicles are going close by uh they kind of get detected together so with erosion we're able to find the difference between the objects contour detection is a way of finding the shape of the vehicle that's coming in and it saves the details of where it is and draws the bounding box accordingly

now object tracking is something that's really important in this project that's used and the difference between object recognition and object tracking is that tracking specifies an id to a particular vehicle and it follows that vehicle around whereas in object recognition we just find out that object and we don't see where the movement is so object tracking is basically used for videos more often uh speed calculation is where we start the timer and end the timer uh when a car crosses a particular segment of road and based on that we calculate the speed and then finally

we save the vehicle image to a file which i'll show later so this is what i was saying this is the image that we have and after masking this is what it looks like where the white places are vehicles and the black is the background now there is some small white areas but we're not going to consider those because we have a threshold of area which avoids other things which are not vehicles

finally we detect contours we eliminate based on the area of the contour and we have an object tracker to give an id to the vehicle and track the object so this is how it works this is where we start the timer this is where we end the timer the original video will have the line so you know and as soon as we end the timer and calculate the speed at a particular part we capture the cropped image of the vehicle and we save it to a particular file so yeah not only

do we actually save the image to a file what we do is we also make a summary so this is

basically creating a text document and entering the id of the vehicle and the speed so in this case the speed limit is 80 kilometers per hour so if a vehicle exceeds the speed limit what we do is we pinpoint it with exceeded and also there's a summary of the vehicles that's crossed the total amount of vehicles and the vehicles that has crossed the speed limit so yeah these are all the screenshots from the document this is the speed estimation and this card has crossed the speed limit over here this is the saving of the images of all the vehicles that is crossed and over here

we have the summary and the id and speeds of the vehicles so there were some challenges that i did come up with while creating this project and progressing with it and here's uh brief pointers on the

challenges that i faced so first was objects that weren't vehicles were being detected so basically what happened was in this image a huge segment of the road would suddenly light up the rectangle and let's say that was a vehicle where it wasn't a vehicle so the main issue with that was the masking and i changed the masking method after referring to the net and with a better masking i was able to get this kind of an image over here and with this i was able to identify vehicles better uh number plates were unreadable so the

initial project had the idea of reading the number plates as well but uh since the video wasn't that clear it we couldn't read the number plates at all that's why we uh stuck to just taking the picture of the vehicle next is multiple vehicle speeds uh were unable uh to be detected so what happened is when there was one vehicle that was crossing the segment of the road the code worked fine but when there were multiple vehicles uh the code malfunctioned and it was showing a wrong output so i fixed this later and i'll explain how i fixed it later in the code vehicles close together were clubbed as a single object so the way i fixed this issue was to erode as i had mentioned earlier so by eroding the mask image we were able to differentiate between two objects that were apart and also i moved the detection lines a bit further back so that we'd have a clearer image of the vehicle and i moved it as far back as i could so yeah i'm going to move on to

the project implementation now so for project implementation we basically have we have two files speed radar 2 and tracker so yeah so in the speed uh radar what the tracker is basically used for object tracking and to find how the vehicle moves so i'll come to that after explaining the speed radar so what this does is here we have the background subtraction and this is what finds the difference between the foreground and background these kernels these are basically used for the masking technique that i used so initial masking technique that i used did not work that's why i'm using the second masking technique and while true this is basically if you've learned opencv you know that to make a video we'll basically have to loop it and use a range of images so this is a looping function for that and here we're basically we're just reading the image here the region of interest is extracted by taking the frame and finding the space based on the coordinates so this is uh yeah this is to find a better idea of which region we're interested in and we don't want the whole frame in the screen we just want a part of it masking method one was which did not work initially so this could be discarded and this is not really needed in the code different method of masking and this is the one which works later you can see that this was the road was added later because uh it was showing the issue of detecting multiple vehicles as one so contra detection mainly what it does is it finds the shape of the objects that are detected and it saves the details of where it is and how big the image is so if you see we're going through each of the contours and this is the threshold of finding vehicles so uh using uh trial and error i found that if it's above 1000 that means it's more likely to be a vehicle and if it's less then we can ignore any of those cases so this is the main code for object tracking and this this calls the other python function to find out if an object belongs to the same id or not so in this part what we normally do is that there's an if else case so we have something called a get sp which is a function of tracker part 2 so if the speed is lesser than the limit which in this case is 80 kilometers per hour it would put a green rectangle and in case the speed goes above that it would put an orange rectangle so this is basically for the aesthetics and making the looks better and we can have a clear view of which vehicle cross the speed limit and now tracker dot capture this is a function that's called so as to save the image of the vehicle so all these functions are there in tracker 2 and i shall be showing those quick now cv2 dot lines these lines are basically uh used to draw it on the video itself and it gives a better idea of when the timer starts and when the timer ends so the reason there are four lines and not two lines as well the pbt is because sometimes uh when the the vehicle crosses a particular line it did not sense it so i needed to use a small segment of the road so as to start the timer and end the timer finally we display the event and tracker.end is something that's used to create the text files so this is needed only when we're closing the implementation so i'll just run the implementation quick before showing track so you can see the id is specified to the vehicle and it keeps following it so zero is the id that's shown and the speed is 47 right now this is the second vehicle with an id1 and when it crosses this red segment of the road speed is

estimated so it's going at 57 kilometers per hour as per our program so you can see that car has actually crossed the speed limit slightly by 87 so it came in orange square and it will be saved later saying that it crossed the speed limit so initially there used to be an issue with multiple vehicles coming in together but right now that's fixed and multiple vehicles are shown without any issues yeah towards here they converge but that's not really much of an issue okay okay fine i thought that we could cross the speed limit okay so now i'm going to go and explain uh tracker so in tracker what we have is we have defined the limit clearly 80 kilometers per hour and we open a file so this is basically to put in the summary and the speeds of vehicle and i'll show the file later how it's created okay so over here what we have is basically the function so as to track the object that's there and update is the main function that we use so as to see if the object is the same or not so what i mean by this is when an object slightly changes in its frame we need to see if it's the same object or not so how we do that is finding the distance between an object in frame 1 and frame 2. so when a car is slightly let's say in the 8th pixel and it goes up maybe to the 7th pixel so then we can say that it's the same object itself so this mainly finds the distance between two of them and it's like a threshold so if we increase this then it will detect multiple objects for the same object and it will show multiple ids and if we decrease this a lot then vehicles multiple vehicles will get the same id and would be considered as one so if you're using a different video it's important that you [Music] check out for yourselves and try to get the correct distance threshold so as to correctly track vehicles also if you're using a different vehicle that a bunch of numbers are going to change such as these coordinates over here when to start the timer and bend to end the timer also where to draw the lines you'll have to like change those accordingly based on the video that you're taking so this is used to start the timer when the vehicle crosses the first line and over here this is used to stop the time so as you can see i've actually used an array so as to start time and stop time and not use a single integer so my earlier mistake was that i used a single integer itself to calculate the time difference and the main problem with that is uh it will not consider if a seg if there are three vehicles in the same segment and when the first vehicle crosses and the second safe and vehicle comes in uh it creates a mess generally and it's not able to see which car starts the timer and which car stops the time that's why i created an array based on ids so uh id0 card would have its own start and stop timer and id2ca would have its own start and stop now this is basically a flag that i just generated so as to capture the image of the vehicle so the moment this crosses uh the last line and the speed is estimated it creates the flag to one and what this does is once this flag is one it captures the image and once the image is captured and saved the flag would be zero so the fla the capturing image will not be called again and again when speed radar is running so in case a distance is not lesser than 70 a new object would be detected and a new id would be given to it so this segment of code here is used for assigning a new id to the object that's given finally we have the speed function and this function is used to find out the speed of vehicle it's get sp so this number over here i got it based on a distance calculation of how much it is and based on the frame rates so it will depend if you use it will depend on you if you're using a different video but another easier way to find this is to find uh the correct

speed of a vehicle say like the first vehicle that's going 60 and you backtrack in trying to find the correct number and based on that it will relatively find the speeds of other vehicles and you'd be able to know the speed that's going on this one is uh saving vehicle data and what this does is it basically uh just crops the image and it saves so here it it looks like a bit of a mess because i'm simultaneously saving the image to a file i'm also writing into a text so file t is basically used to write into the text file and file over here is used to put the image into a specific file which i should show later limit just returns the speed limit that was mentioned earlier in uh speed radar 2 so in this case it's 80 kilometers per hour and this is def end and uh as the name suggests this actually uh runs in the end of the program when we close everything and it creates the summary of total vehicles and exceeded so these variables are the ones that i've used and updated in capture so if a vehicle crosses count plus one is done and if a vehicle crosses the speed limit that's this part over here exceeded is put one and it's mentioned in the file saying speed has been exceeded and the picture of the vehicle is stored in a separate uh folder called exceeded so now i shall show the saved pictures of this so here we have the pictures of the saved vehicle so we have this specific id and we have the speed also there uh also what we have is the vehicles that cross the speed limit and over here you can see we have a list of vehicles that have crossed and there's one vehicle that has crossed the speed limit so right now there's multiple pictures and that's because i'd executed this before and forgot to delete the pictures so just implement it one more time and i'll show you how it works so as you can see vehicles are crossing and when a vehicle crosses the image of the vehicle is saved over here also this text text file is also updating in the background also i'll be up updating uh uploading this code uh in the description below in github so if you guys want the code you can go check it and hopefully it will be useful so that vehicle crossed the speed limit so i guess that would be enough okay so if i go here see here we can see all the pictures of vehicles are across and these are cropped images of the vehicle that's there this vehicle exceeded the speed limit so that's there as you can see the number plate isn't visible so i haven't uh put that there and here you can see the number of vehicles i call the six vehicles that's crossed and there's one that's exceeded of course the ids might not be in the correct order but that doesn't matter like that it skipped two and it skipped six but uh we have a clear idea of the vehicles that passed and which vehicle exceeded the speed limit so yeah uh that will be all and that's the complete uh project also i'd like to you know thank the institution for giving me the opportunity to look further into this topic and i also like to thank my image processing teacher swathi mam for guiding me throughout the project and helping us improve our concepts in image processing both theoretically and practically as well and uh that's it uh i'll be putting the code in github in the description below if you guys want you can check it out and i really hope this helps so uh thanks just stop the zoom recording