Student: Mihai-Cristian Popa
Group: 407

Computer Vision
Video Analysis of bowling Project

Task1

The approach for this task can be split into the following points:

1. Fine-tuning of the model YOLOV8 for Object Detection on a Dataset containing bowling pins;
2. All the pictures that pass throught the system are cropped on the sides to become squares, while the important content is maintained. Then the square pictures are resized to 640x640, the dimension of the training data from the Pin Detection Dataset.
3. Extracting the bounding boxes using the fine-tuned model of the pins from the lane images which will then be used as template. The maximum detections are set to 10, to not get overlaps and also the confidence was set to 0.55, higher than the default. One last thing is that we only keep one class from the model, representing the pin class, as the data was trained with two extra classes.
4. Extra handling was needed for the sorting (descending on the bottom point and ascending on the left point) of the bounding boxes of the pins from the lane images to be properly ordered.
   - There were two template pictures were the middle pins were being obstructed so their bounding boxes generally had the bottom point higher then the rest of the pins. So their “unobstructered” bounding boxes were created based on the ones of the side pins. For these we kept both the “obstructered” and “unobstructured” bounding boxes to check agains the processed images.
   - There were the last two templates, for which the bounding boxes were a bit smaller on the right side, probably because of the camera angle. Here we took the average of their base and set that for the entire line and the sorted again with the same rule as earlier.
5. Then the bounding boxes of the pins from the processed images are extracted using the same model. Here the parameters stay the same, besides of the confidence needed for returning a bounding box, as it was generally noticed that is better to have some not so great bounding boxes, then to have non for the entry you are looking for. So confidence was set to 0.07, probably could be set even lower. Those were also sorted, but here no extra processing was done, and probably there were many more edge cases to handle. So this was kept in mind for the next steps.
6. For each of the processed images, we compute the correlation with each of the lane images, by template matching. Then each of the images will be compared agains the lane that was rated as the most similar.
7. Next step is to pass through the queries. For each of the queries we take the pins in reversed order with the purpose of limiting the risk of having bounding boxes from the template’s front line confused for pins in the last line.
8. We take the key from the query, and look it up in the list of bounding boxes’ of the matching template.
9. With the bounding box (could be two in case of the middle pins from the “obstructed” templates) of that specific key from that template we compute the IoU against each of the identified bounding boxes from the processed image.

Student: Mihai-Cristian Popa
Group: 407

10. The IoU method has some caveats which are based on the edge cases found when running agains the train images. Initially I had a logic for which if there was no overlap between the rectangles then that meant that the IoU will be returned as 0. Then I added a check, that if the bottom and top point are very close between the two rectangles, and the sides are less then a distance from between two pins aside, we can still consider this rectangle for IoU computation, with the left and right points being the averages of the two rectangles.
11. There is also a check in the IoU function for the difference between bottom points of the two compared bounding boxes. Generally by checking the templates situations, the variations were smaller than 10 pixels. The limit was set to 30.
12. The IoU threshold was also amended as there were a few cases were the threshold of 0.5 was too large, so we went with 0.34.
13. Then for a key if a match is found then 1 is set, otherwise 0. Also if one of the entries from the processed image bounding box is matched, then it is removed from the list of bounding boxes.
14. The final step of the process is to build the txt files from the mapping build at the last step.
15. One other important step in the building of the model has been the visualization of the lane images and the processed images with their correspondent bounding boxes.

Task2

The approach for this task can be split into the following points:
1.Fine-tuning of the model YOLOV8 for Object Detection on a Dataset containing bowling balls;
2.Initializing the list of bounding boxes with the coordinates from the input file
3.From the second frame onwards use the track method of the model with max detections set to 1, confidence threshold set to 0.1.
4.An average bounding box is maintained for each of the video, each true match being averaged over with the previous average. In case of a non-detection, this average bounding box is used.

Task3

The approach for this task can be split into the following points:
1.Fine-tuning of the model YOLOV8 for Object Detection on a Dataset containing bowling pins;
2.Reading the first frame from the video, and cropping from the sides(to remove the pins from the other lines) and from the bottom
3.Using the predict method of the model, using the pin class, maximum detections 10 and confidence threshold 0.31. From the result of the predict method, we count the number of bounding boxes that are found.
4.We pass through the video, saving all the remaining frames. After the video is done, we take the last frame from the list of frames and we use the same approach as in step 3 for obtaining the number of standing pins.
5. In the end a file with the results is created for each of the videos.

Datasets used:
Task2: https://universe.roboflow.com/bsm-ecg3e/bowling-model/dataset/10

Task 1 and 3 : https://universe.roboflow.com/lsc-kik8c/bowling-pin-detection