Radu Madalin-Cristian

Group 407

Computer Vision – Project 4

The task

In this project, we are tasked to create a program that takes images of dartboards and computes the score for the darts thrown.

Task 1

In this task, we are to predict the number of darts and their position in a classic dartboard with 9 concentric circles. To achieve this, I did the following steps:

- Find the concentric ellipses, representing each numbered concentric circle (seen from the side) by:
    - Converting the dartboard image to hue, saturation, and brightness
    - Blurring the image
    - Splitting the image into background and foreground
    - Finding contours with cv2.findContours
    - Filtering contours based on area
    - Fitting an ellipse for each filtered contour
- Add the smallest ellipse (the bullseye) which I found empirically.
- Load a pretrained yolov8 model (source).
- Train the model further with the darts training images, which I annotated with bounding boxes using roboflow (source) (this section is commented out, since there is no value in training the model at every run and since we won't be including training data in the submission.
- The best model is included in the submission files
- Detect the bounding boxes for the darts using the trained model
- Use the left edge of the bounding box and half its height to estimate the point where the dart was thrown

- Go through all ordered ellipses (the numbered concentric circles) and stop when the estimated dart point is located inside the ellipse (to see this, I used cv2.pointPolygonTest)
- Do this for all images and all darts and write the results into Radu_Madalin_407/Task1

Task 2

For this task, we needed to predict the score of the darts when thrown on a dartboard that also has a double and triple ring. These sections are red or green. The bullseye is also separated into the outer bull (worth 25 points) and the inner bull (worth 50 points).

My approach for this part was similar to the one before. I trained the model in the same way (using roboflow for annotation and yolov8 for training). To differentiate between the single, double and triple zones, I extracted the red, green, white, and black pixels. I created a mask for each of these zones and used them as follows:

- I used the black mask to extract ellipses, and, with the proper thresholds for the contour area, I extracted the ellipse that fits the whole dartboard
- I used the red mask to extract the inner bull ellipse and the center point of the dartboard
- I used the green mask to extract the outer bull ellipse

Like the task before, I commented out the cells that contain the model's training, since there is no need for it because I included the already trained model in the submission.

The scoring system for this task is as follows:

- Detect the bounding boxes for the darts in each image
- Like in the task before, the estimation of the tip of the dart is done by taking the left edge of the bounding box for the x value of the point and by taking the half of the height of the bounding box for the y value of the dart tip.
- Typically, the darts whose tips start falling before hitting the dartboard will remain on it in an oblique manner. We can't use half of the height for the y value of the dart tip since this will put the estimation for it too high. This is why I introduced a correction for y which lowers the value when the height of the bounding box is big enough.
- Use the dart tip estimation to calculate the score by:

    - Calculating the angle of the point with respect to the dartboard ellipse
    - Getting the number of the sector using the obtained angle
    - Getting the number of points from a dictionary that stores the number of points respective to the sector of the dartboard ellipse
    - Checking if the point is inside the outer bull ellipse
        - If yes, then also check if the point is inside the inner bull ellipse or if it's red (which would also mean it's inside the inner bull ellipse)
    - Checking if the point is in the red or green areas
        - If yes, if the distance to the center is high enough, then the flag is double, otherwise it's triple
    - Append the flag and the score to the results
- Do this for every dart in each image and write the results to Radu_Madalin_407/Task2

Some of the cells might not run if the notebook is not located in the same folder as the training, validation, and auxiliary data.

Bibliography

- https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html
- https://docs.roboflow.com/
- https://docs.ultralytics.com/tasks/detect/