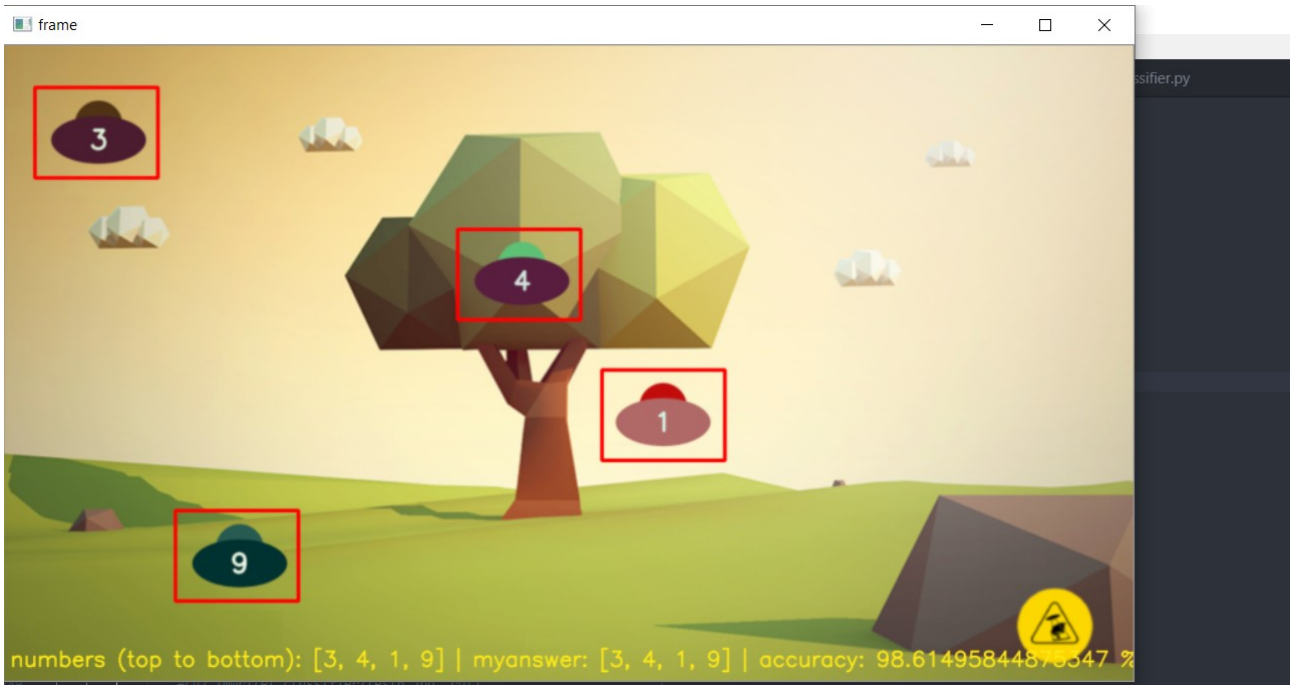
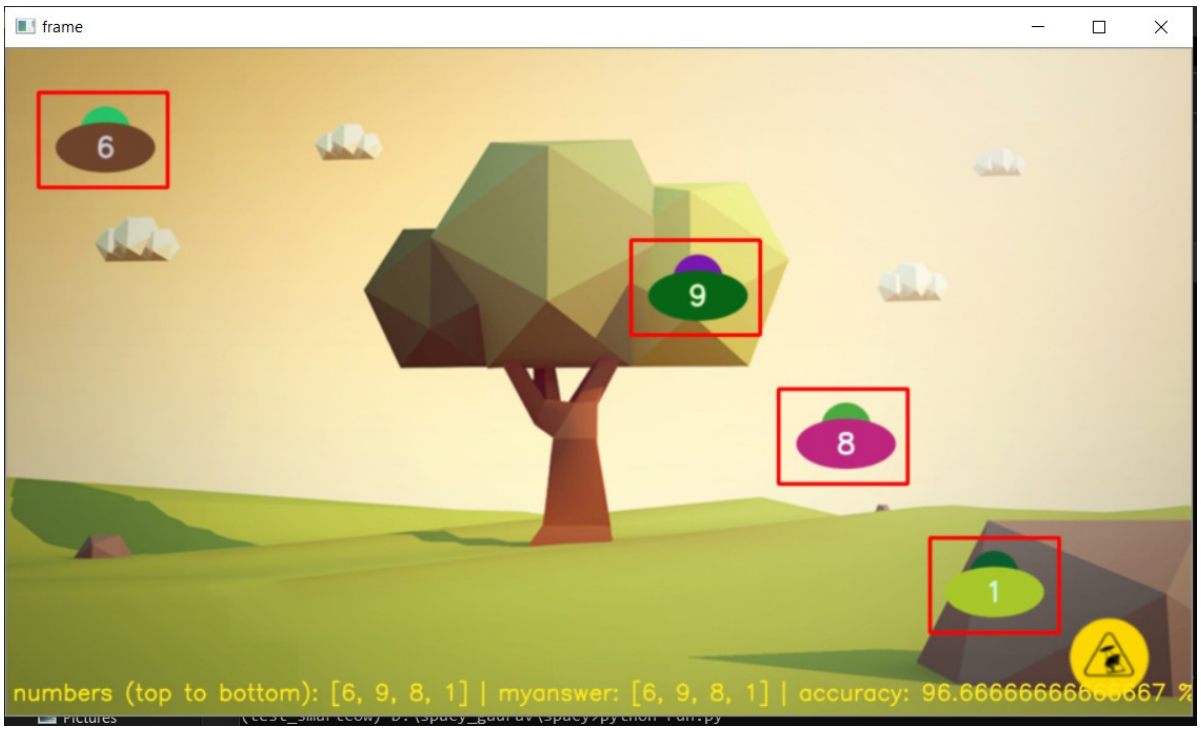
READ ME
Spacy -Gaurav Bhosale

To Run
pip3 install -r requirements.txt
python/python3 run.py

Outputs



Output after running for 5 mins -



Errors -
1) Localization error , due to not find the right contour especially sometimes for number 2.
2) Classification error – Not a robust classifier can be improved with more images . SVM is highly scale variant and translation variant.

I am confident that the accuracy can be brought closer to 100% given more time to train classifier and also localisation with proper tuning of parameters in contours and canny.

Final Approach -

1) Get edge detected Image from Canny detector

2) Get contours and sort them by y axis

3)Use selective filtering on these contours to get only the contours of the numbers inside the UFO.. These was used because the UFO is only the solid object without change ins gradient and thus easier to identify the number inside the UFO.

Use the aspect ratio and area of contours to filter out those who arent numbers.

Since contours of UFO are unreliable due huge amount of randomness thus get blended with background but the numbers and its background are solid.

4)Sometime multiple bounding boxes are generated by number . Used simple co-ordinates to remove nearby BB but NMS(non-max suppression) and hierarchy of contours may also be used.

5) Setup appropriate the ROI to crop the image . (A change of ROI will cause classification to fail)

6)Send the crops to classifier.

7) The classifier is not robust due to been trained on specific 170x115 crops .
Resizing gave an accuracy of 23%

8) I retrained the classier . i kept the same svm classifier although svm is especially scale and translation variant but for this specific task its light and fast to train.

9) Generated images by getting number from the numbers list from spacy.py and matching them with my crops.
Generated around 10k images .

10) Trained on these images to get better accuracy around 97% can be improved,

The Approaches which were followed and failed -

1) Initially after seeing the Image i saw that the gradient of background was changing continously and figured out none of the normal off the shelves algorithms would work without tweaking for accurate detection . Eg blob detection.

2) Tried out blob, ellipse , hough detection , ellipse fit etc.

3) Other approach which might work is adaptive filtering but it requires a longer computation time.

If Images of other approaches or the current approach is required I will provide them.