

Object detection

This pdf shows summary snapshots from a jupyter collab notebook created to explore modifications to a pre-trained model (Resnet 50 and the COCO dataset) to use it for single stage object detection (SSD).

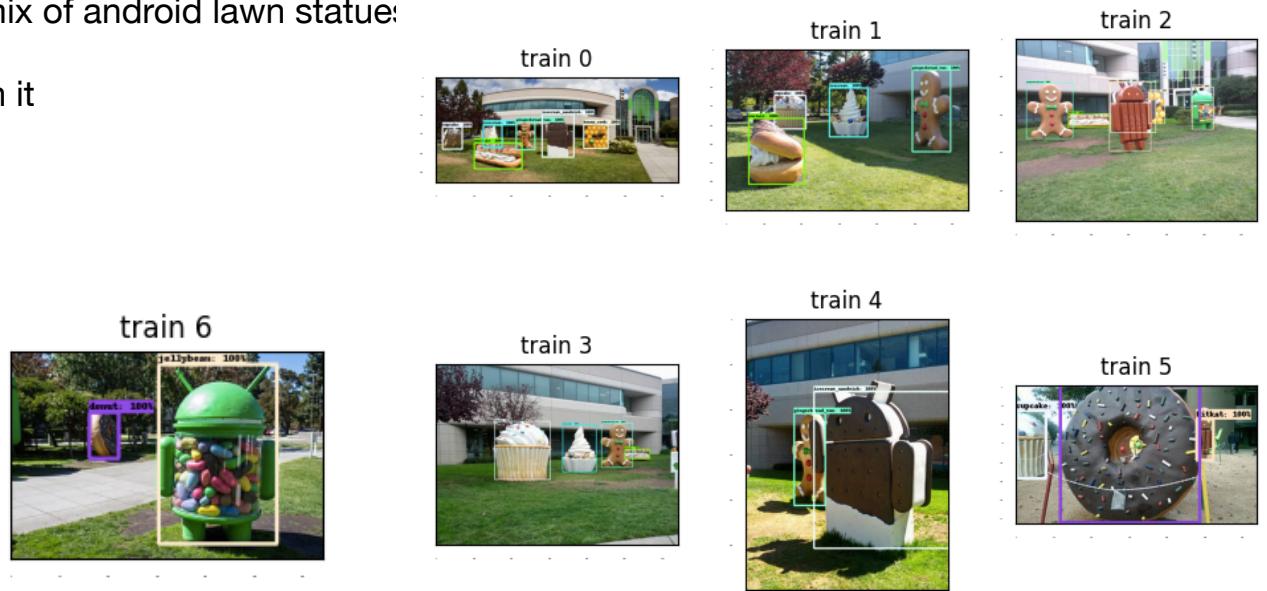
I use the feature extractor and bounding box detector from the pre-trained model and re-train the new model for multi-class classification (which uses a feature extractor) with the training set images of Android statues to find the statues in a test video.

see notebook at:

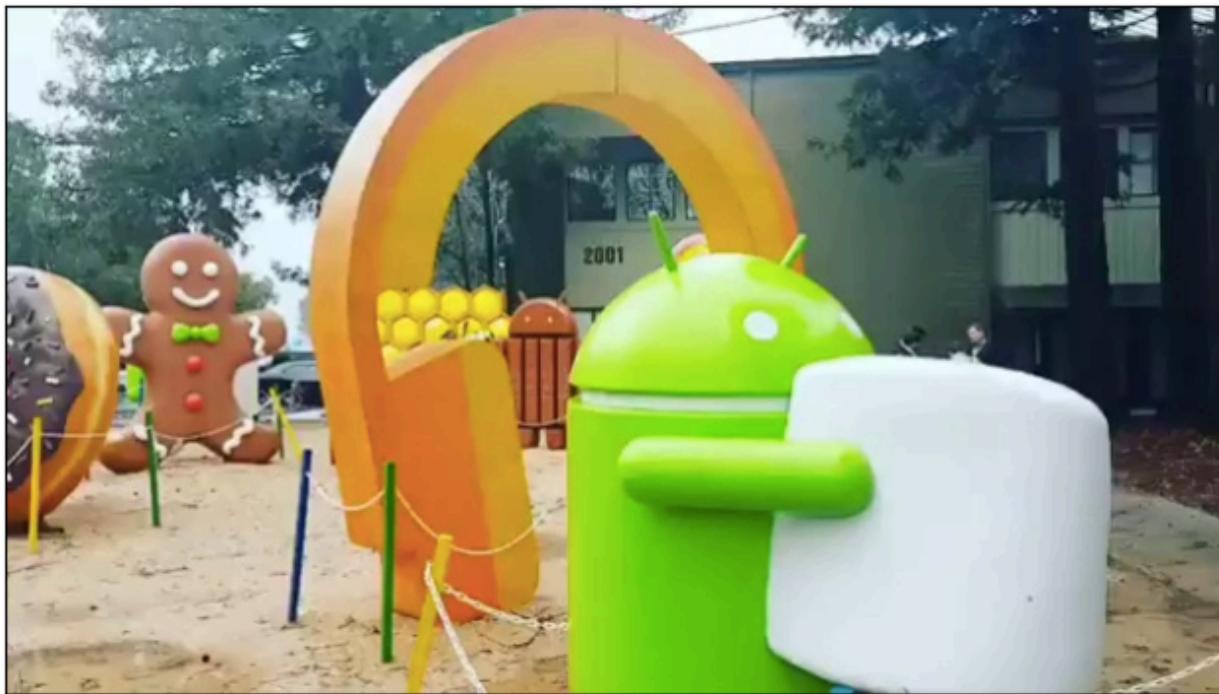
https://github.com/nking/curvature-scale-space-corners-and-transformations/blob/master/src/duckies_and_statues%2C_interactive_eager_few_shot_obj_det_training_colab.ipynb

There are 7 training set images containing a mix of android lawn statues:

class id	class name	#of images w/ class in it
0	cupcake :	4
1	euclair:	4
2	icecream:	3
3	gingerbread_man:	5
4	icecream_sandwich:	3
5	honeycomb:	2
6	kitkat:	2
7	jellybean:	2
8	donut:	2



These are the object detections of statues in frames of the test video



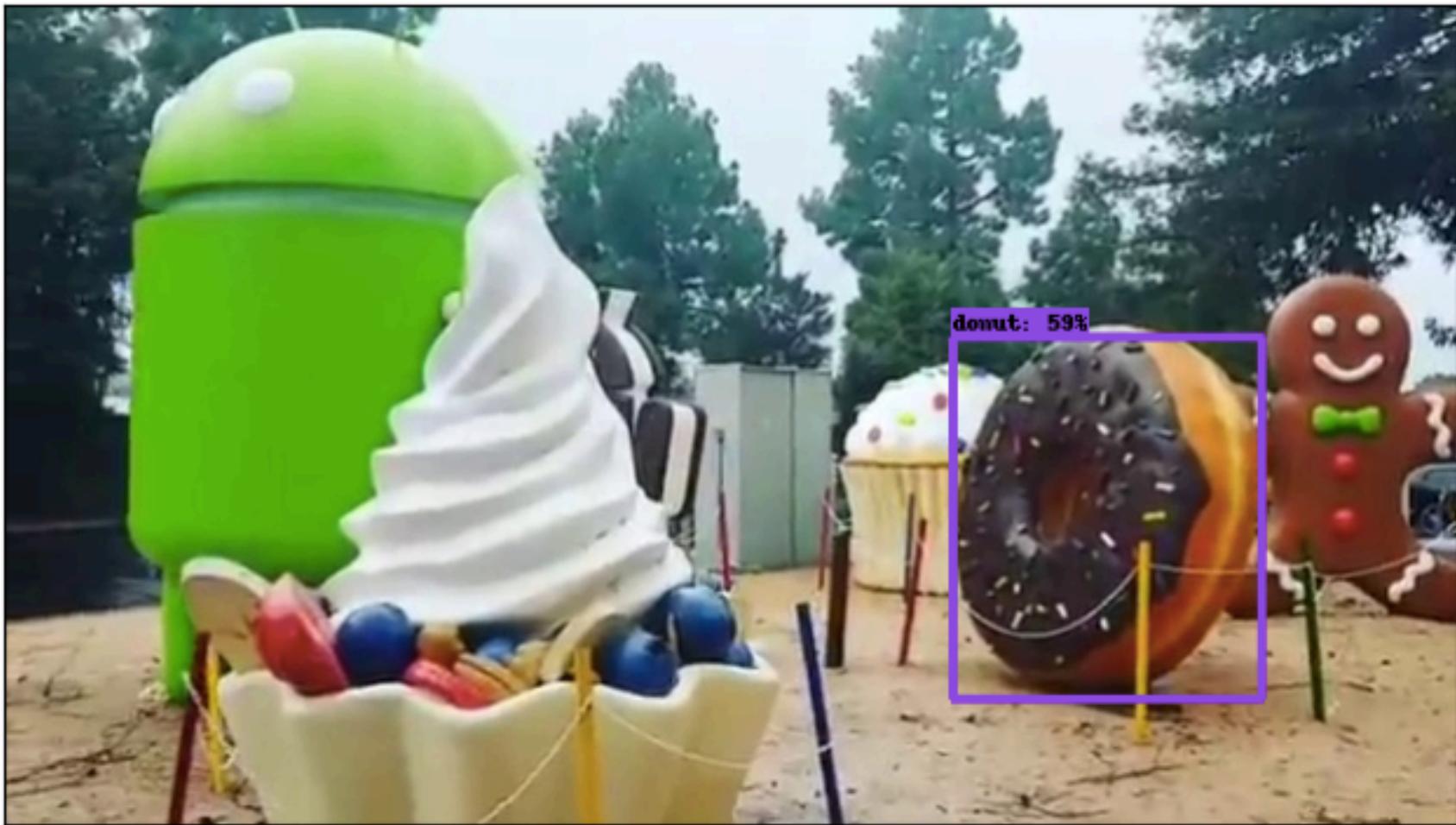
```
pred_1=[]
score_1= []
```



pred_2=[0]
score_2= ['0.56']



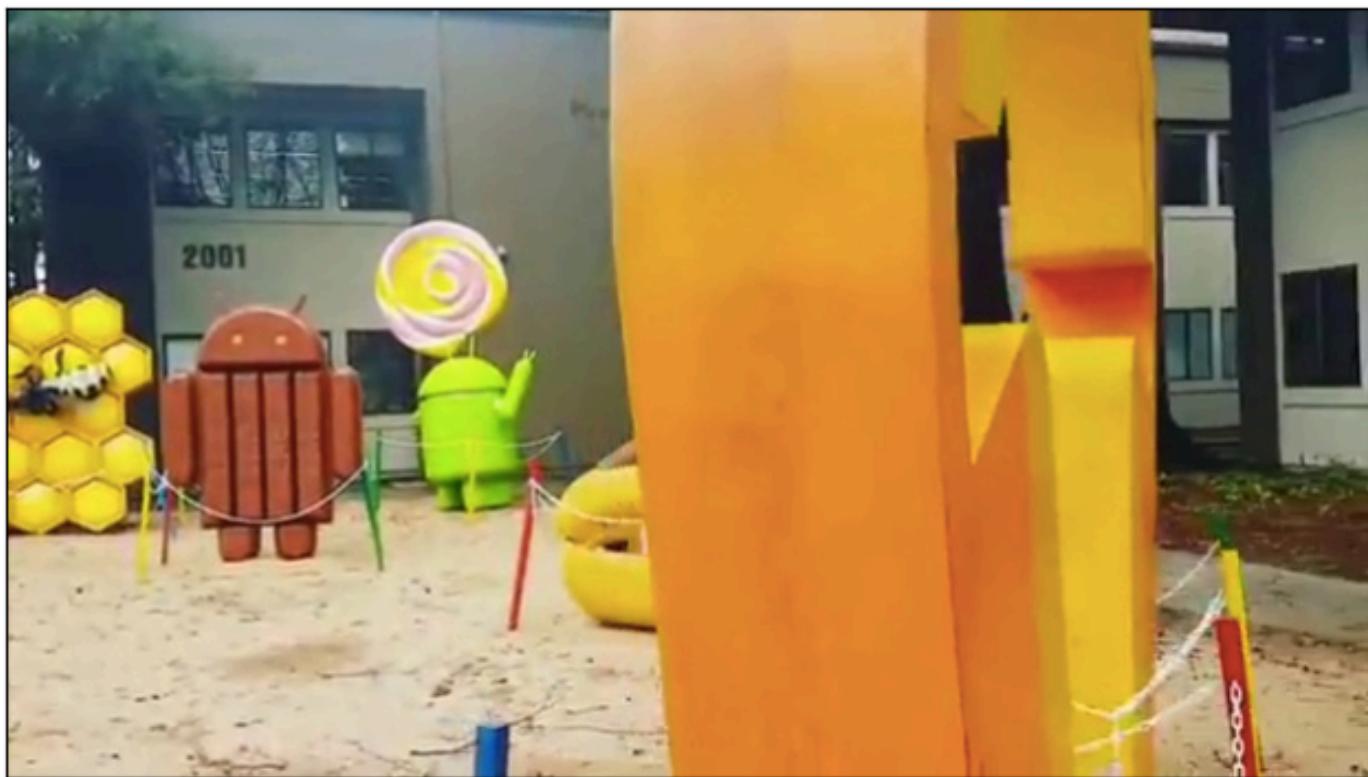
```
pred_3=[3]
score_3= ['0.64']
```



pred_4=[8]
score_4= ['0.59']



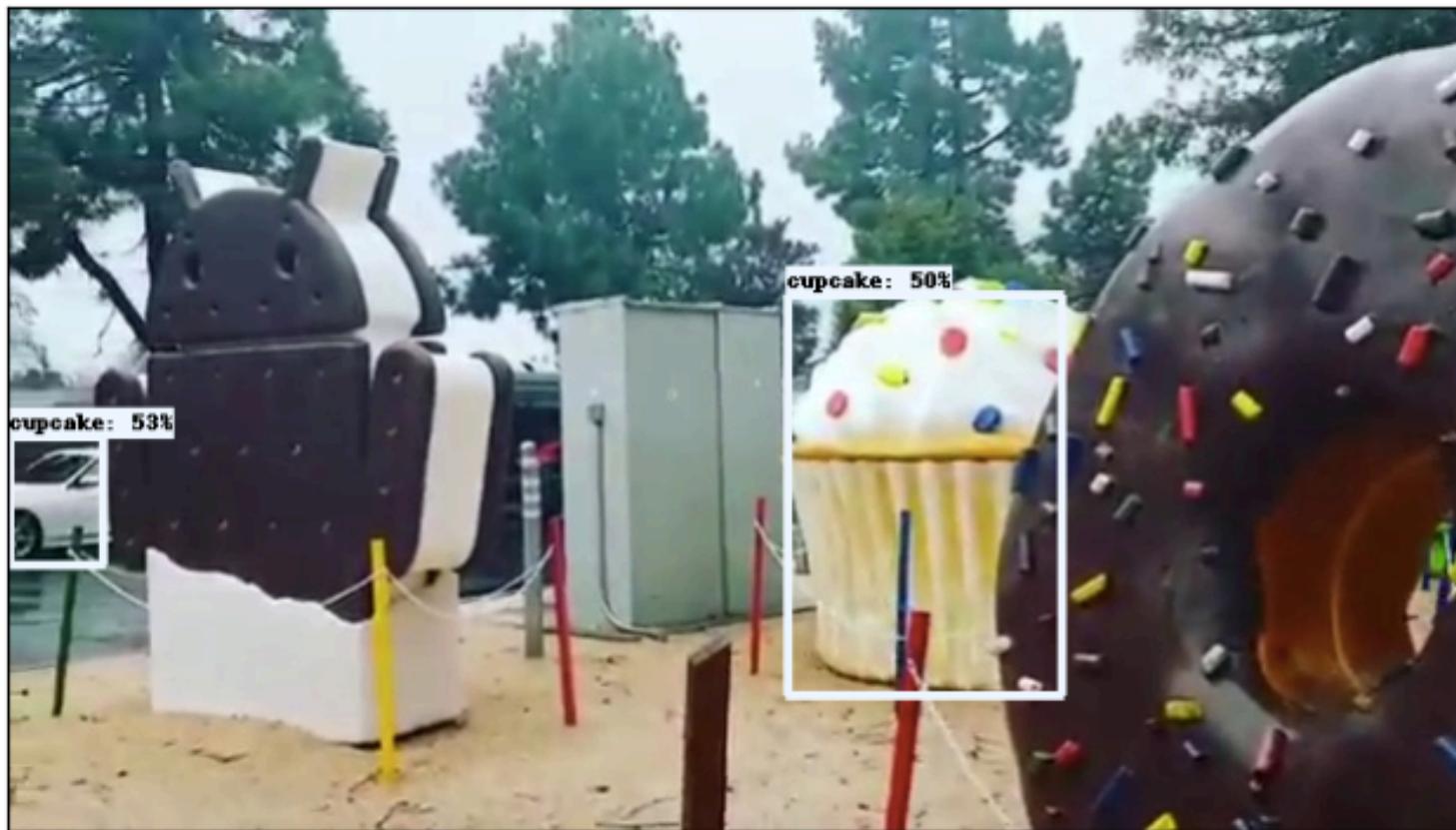
```
pred_5=[8, 3]  
score_5= ['0.67', '0.51']
```



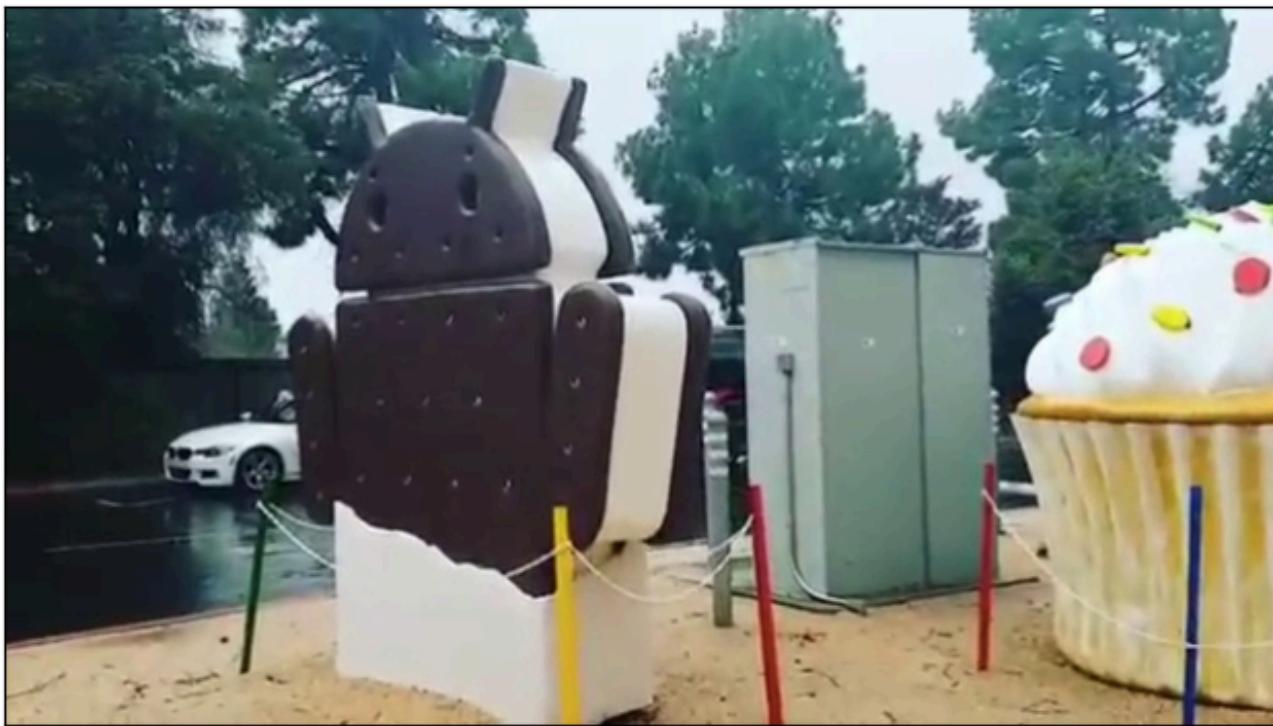
```
pred_6=[]
score 6= []
```



```
pred_7=[3, 5]
score_7= ['0.77', '0.52']
```



pred_8=[0, 0]
score_8= ['0.53', '0.50']



```
pred_9=[]
score_9= []
```

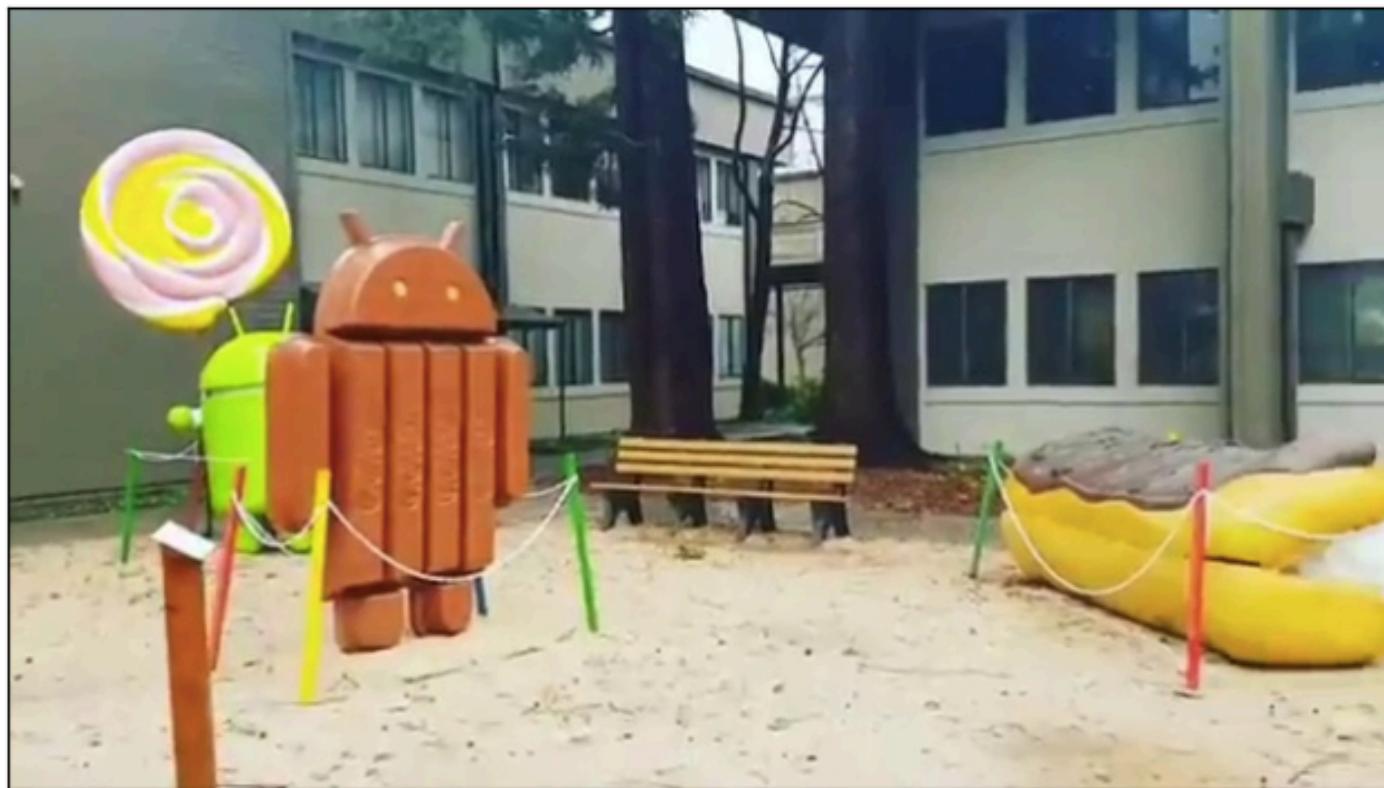


```
pred_10=[]
score_10= []
```

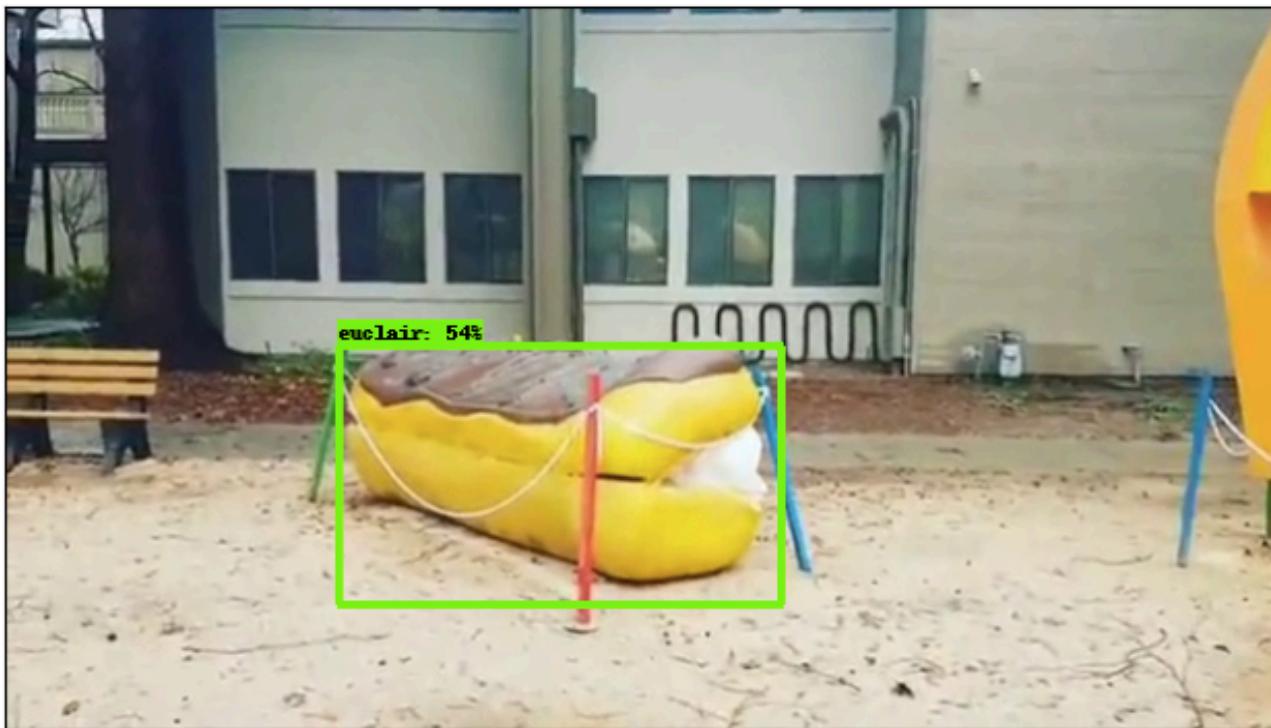


gingerbread_man: 69%

```
pred_11=[3]  
score_11= ['0.69']
```



pred_12=[]
score_12= []



```
pred_13=[1]
score_13= ['0.54']
```



```
pred_14=[]
score_14= []
```

```
ground_truth_labels_all=[3, 9, 2, 2, 3, 9, 5, 2, 2, 2, 2, 2, 3, 2, 8, 8, 2, 8, 8, 3, 8, 9, 9, 3, 6, 3, 5, 8, 9, 8, 0, 0, 9, 4, 9, 9, 9, 9, 8, 0, 0, 4, 0, 4, 4, 9, 9, 0, 9, 0, 9, 0, 3, 0, 3, 7, 1, 6, 9, 1, 9, 6]
```

```
predicted_labels_all= [3, 7, 0, 0, 3, 7, 5, 1, 0, 2, 0, 2, 3, 1, 8, 8, 2, 1, 8, 3, 1, 4, 7, 5,
```

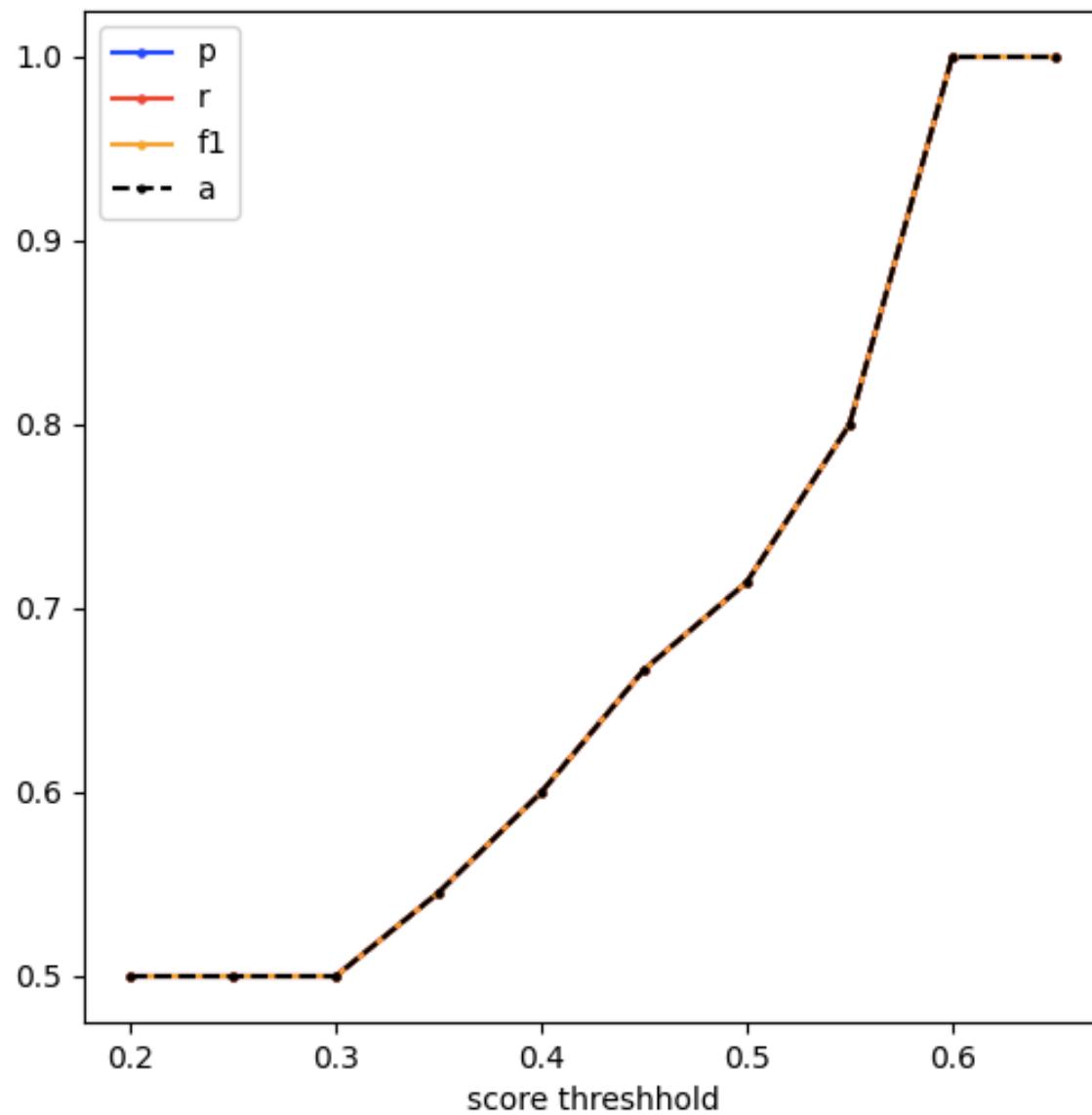
```
scores_all= ['0.29', '0.25', '0.55', '0.45', '0.39', '0.33', '0.32', '0.32', '0.30', '0.29', '0.25', '0.25', '0.53', '0.38', '0.29', '0.62', '0.32', '0.31', '0.66', '0.45', '0.43', '0.29', '0.28', '0.27', '0.32', '0.63', '0.47', '0.44', '0.42', '0.32', '0.29', '0.28', '0.26', '0.52', '0.48', '0.45', '0.37', '0.33', '0.30', '0.28', '0.28', '0.44', '0.34', '0.39', '0.36', '0.35', '0.33', '0.31', '0.31', '0.28', '0.27', '0.27', '0.61', '0.48', '0.35', '0.28', '0.35', '0.28', '0.50', '0.47', '0.38', '0.35']
```

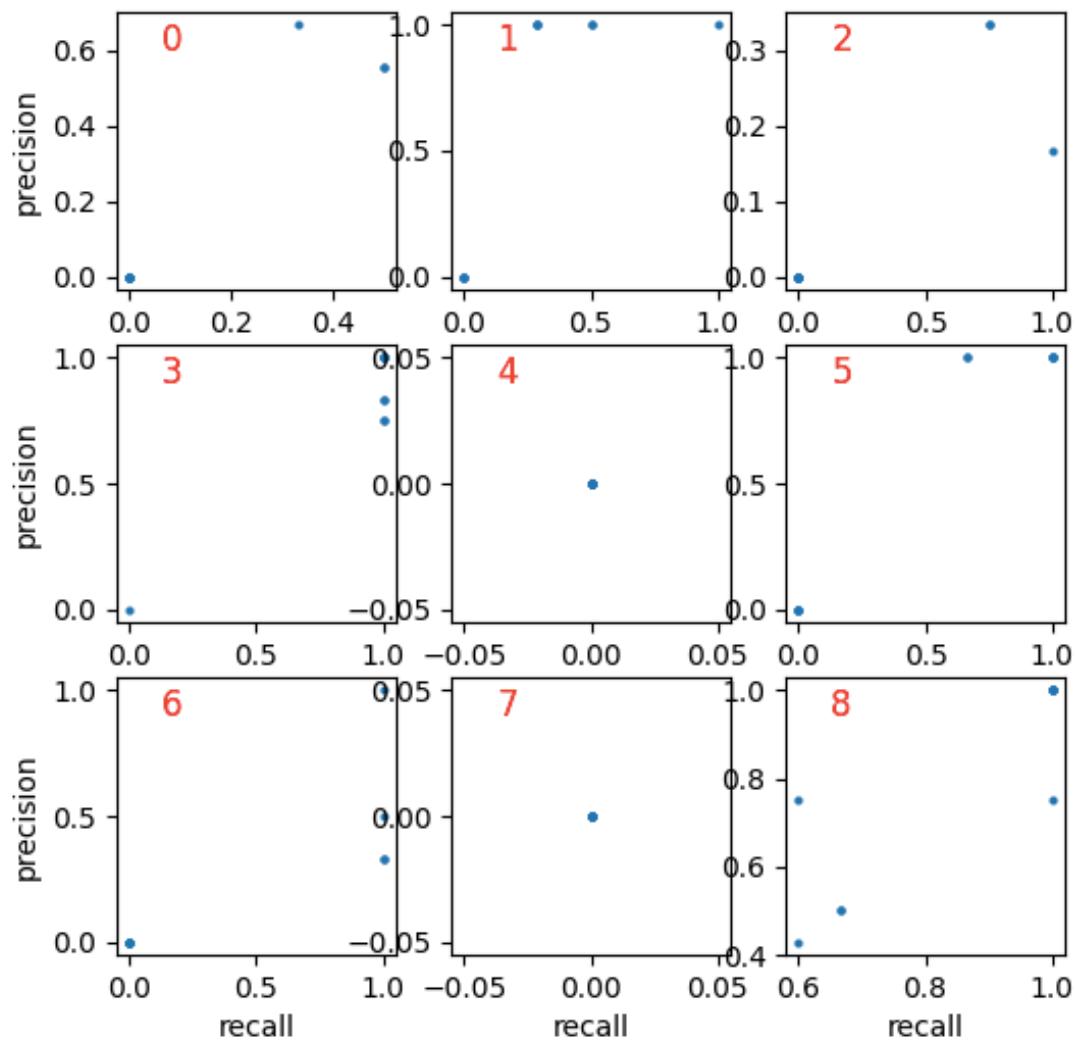
precision, recall, f1 score and accuracy were calculated as a function of score threshold and a threshold of 0.5 was found to result in a macro averaged precision, recall, f1 score and accuracy > 0.72. For that reason, THRESH above has been set to 0.5 now. Here are the statistics derived from THRESH 0.2 to 1.0, which can also be found [here](#)

class id	class name	#of images w/ class in it
0	cupcake :	4
1	euclair:	4
2	icecream:	3
3	gingerbread_man:	5
4	icecream_sandwich:	3
5	honeycomb:	2
6	kitkat:	2
7	jellybean:	2
8	donut:	2

14 test frames were taken from a YouTube video.

these are the
same for
multiclass classification





In Conclusion next steps could be:

try other pre-trained models followed by training for classification of android statues.

try other variants of transfer-learning with object detection models.

add more training data

for the test frames from the video, could consider applying HMM inference to the best scoring identifications for “smoothing” inference where missing object detections. can use “filtering” queries for the next observations... can use particle filtering for similar reasons. can build a better training set by adding the inferred missing identifications (bounding boxes and new ground truth labels) to the existing training dataset and then get a new test dataset.

consider whether using optical flow for a small number of frames after good identifications would be a fast addition to the boot-strapping of bounding boxes between frames instead of inference with HMM or smoothing.

consider multi-object tracking models.