[3 points] Effort to curate/create your dataset (at least a thousand records/images)

I created the data set by using my phone to take videos of a 10-sided die, by moving the phone around the die to get different angles. I then parsed the video by scrapping the audio and getting the images from each frame. I did that for 3 different 10-sided die, creating folders of around 2000 images. I then looked at each image to see if the image was of good enough quality to be able to use for the project and deleted the low-quality images.

[3 points] Effort to visualize input data

I used matplotlib imshow to output what the resized images look like in various parts of the project.

[3 points] Effort correctly split data into 3 sets

The data has been split up into 3 different folders aTrain, aTest, and aValidation. Though in the project to try and get it working I used K-fold to split the training data into training and validation data.

[3 points] Effort to design and test various neural network architectures

I used CNNs architectures. I would look into more RESNet classification but thought CNN would get at least a 40% before I moved onto RESNET.

[3 points] Effort to evaluate your results

[3 points] Effort to benchmark your method / results

There is a good benchmark by Michael Sugimura. He built a first stage object detector that specializes in just detecting the dice faces and then a second stage CNN which would use the outputs from the first model to determine the numbers. He got around 97% accuracy with ResNet 101. Mine achieved around 14.5% at best.

[3 points] Documentation efforts (report preparation)

Documentation is listed in the comments section of the project and this report.

[3 points] Effort to document the training time

Overall the training took around 2 – 5 seconds an epoch for 100 epochs. The bach size was 50 for the majority of time I trained the model.

[3 points] Effort to study learning curves

[3 points] Effort to prepare a "reproducible" Python Notebook (.ipynb) file

It is under my GitHub account in the following link. <https://github.com/ColeAlves/DiceDetector/blob/master/DiceDector.ipynb>