CS 8395 Assignment 1

Daniel Yan

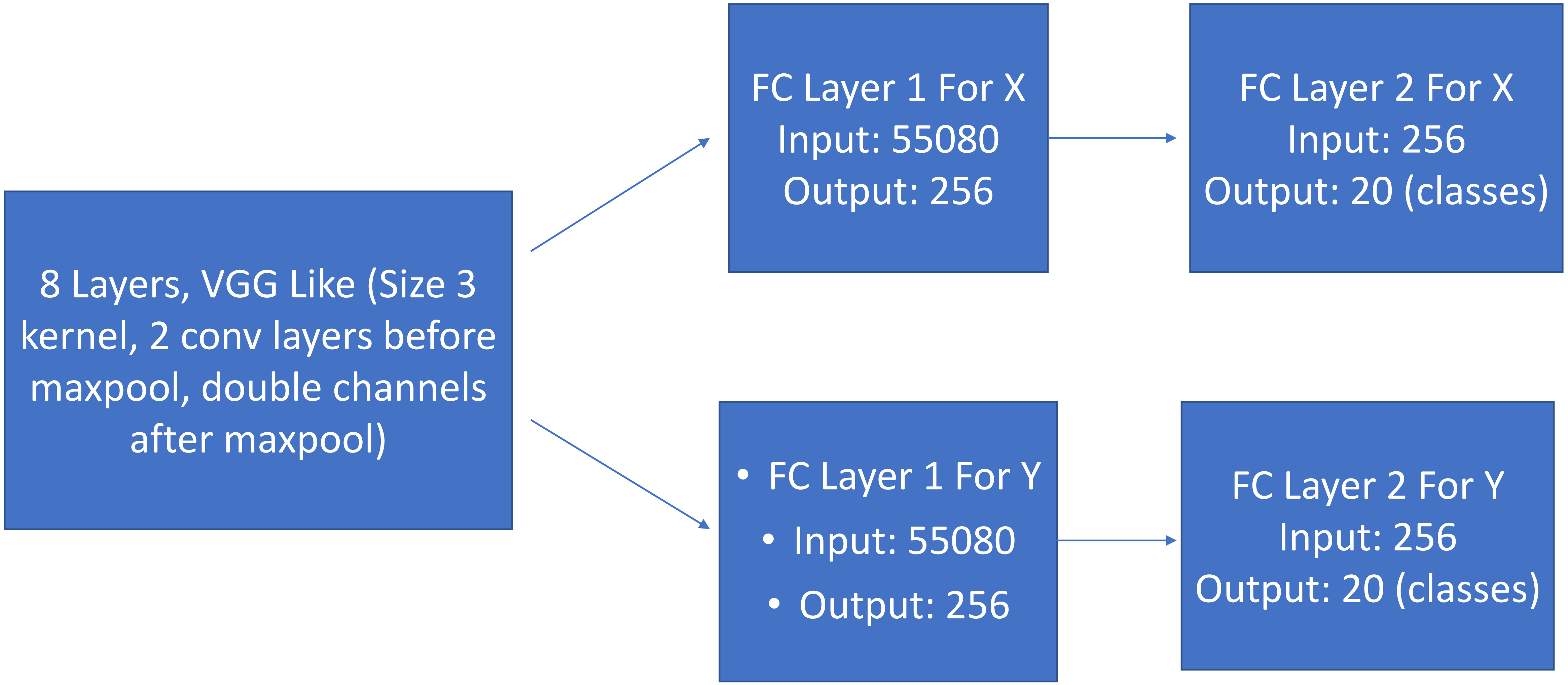
Introduction

- Goal: Predict x and y coordinate for the center of object in image

Rationale

- Originally, I tried regression, but achieved poor results
- My regression model kept predicting things closer to the center to minimize loss
- However, whether my prediction was off by 0.5 or 0.8 didn’t matter to me.
- Instead, I made this a classification problem by dividing the x and y space into 20 equal spaces, and setting the label as whichever window the floating point label fell into
- By changing this to classification, the prediction was only correct if within a small window
- Example: Label 0 was 0.0-0.05, label 1 was 0.05-0.1, etc.
- Two separate labels for x and y classes with 20 classes each: not enough training examples for more classes

Network Architecture



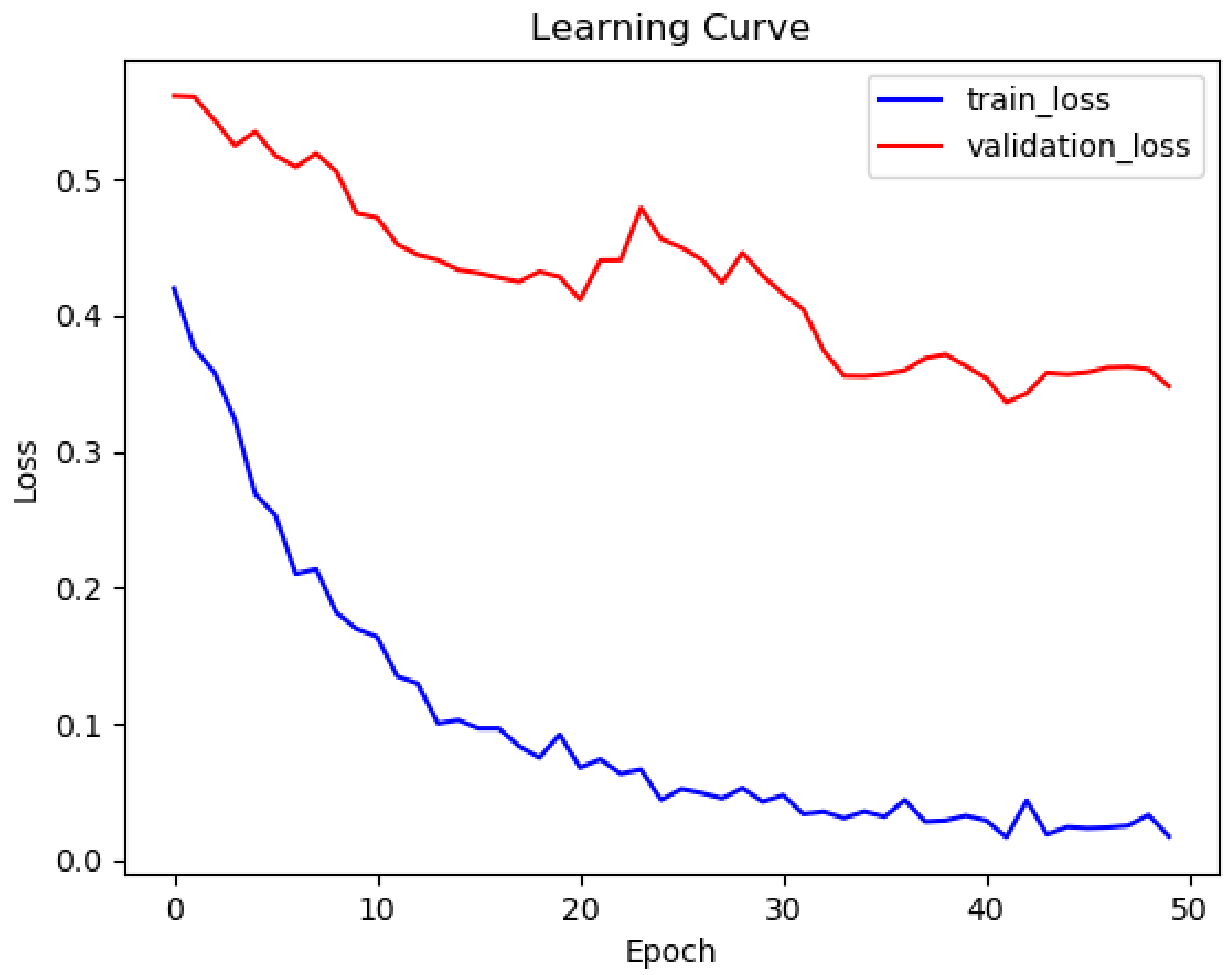
Tricks

- Preprocessing: Convert floating point coordinates to class labels corresponding to which “window” the coordinate fell in
- Postprocessing: Convert labels back to floating point coordinates by choosing the centerpoint of that window
- Early Stopping: Always save the model with best validation loss so far
- Random Search for hyperparameters

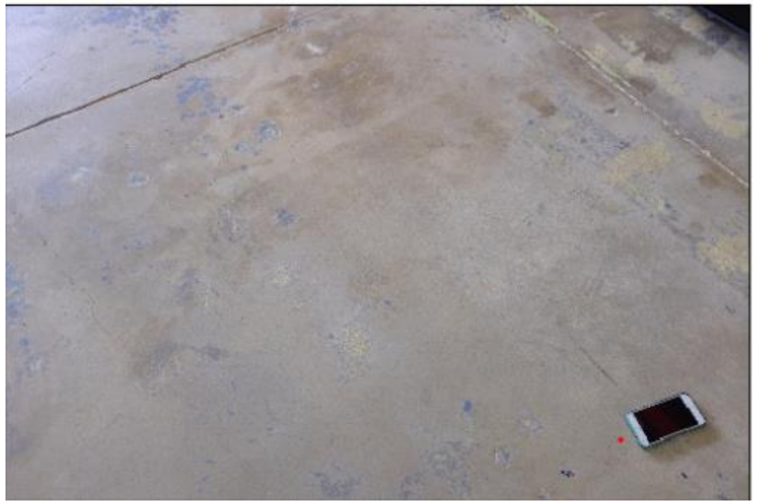
Hyperparameters

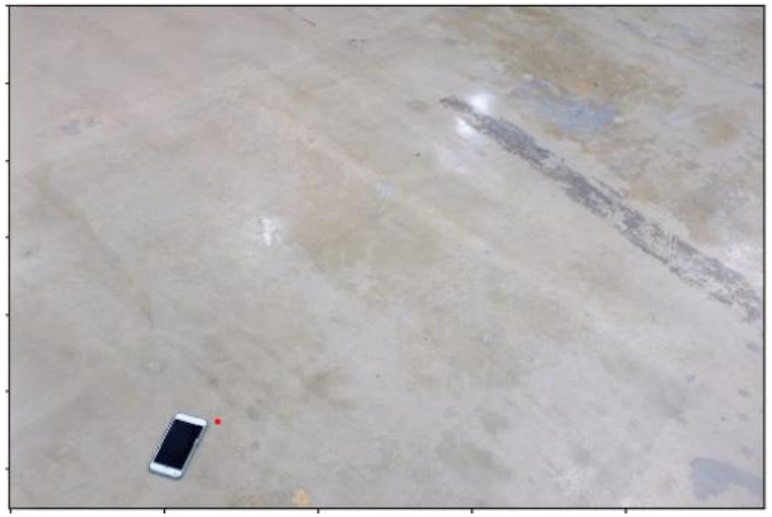
- Random Search over 50 combinations of learning rate (between 0.0008 and 0.002) and gamma (between 0.7 and 1).
- 50 epochs with early stopping: save model with best validation loss
- Loss: Cross-Entropy
- Parameters of Layers:
  - ReLu Activation
  - Kernel Size 3
  - Maxpool of 2 after every 2 conv layers
  - Double number of channels after each maxpool
  - Batch normalization between each layer
  - Dropout of 0.45 after each layer
  - Softmax activation after 20 output channels each for x and y output
- Adam Optimizer
- Batch Size: 12 (GPU memory limitations)
- OS: Windows
- GPU: GTX 970M

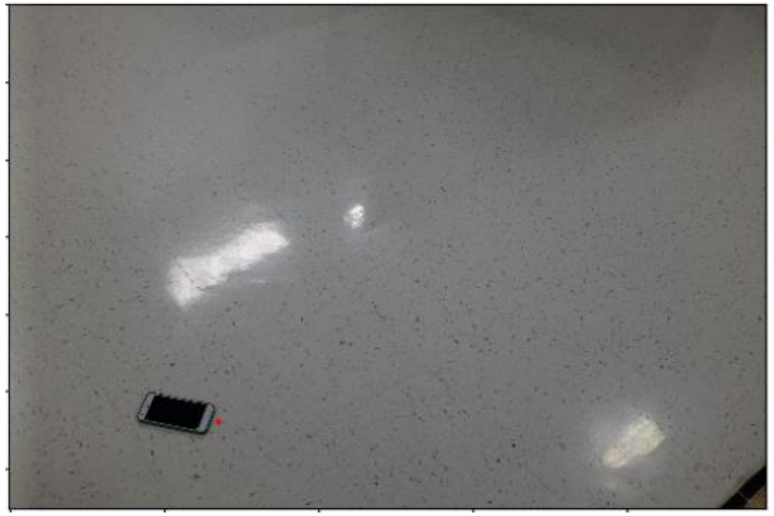
Learning Curve

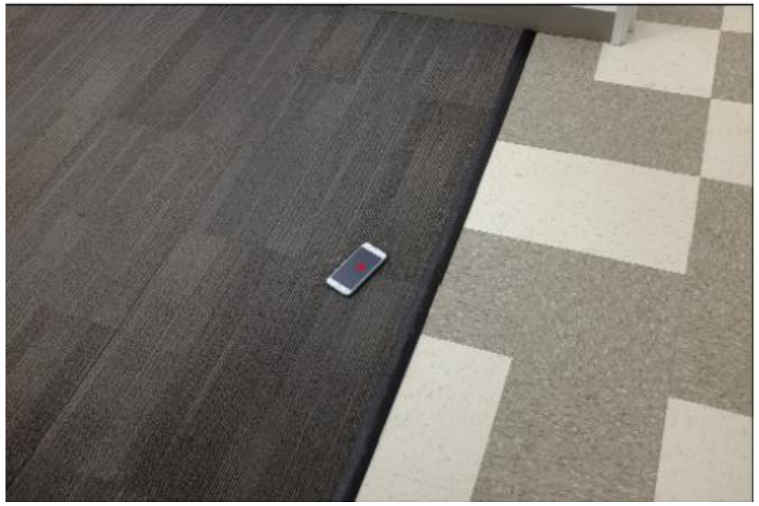


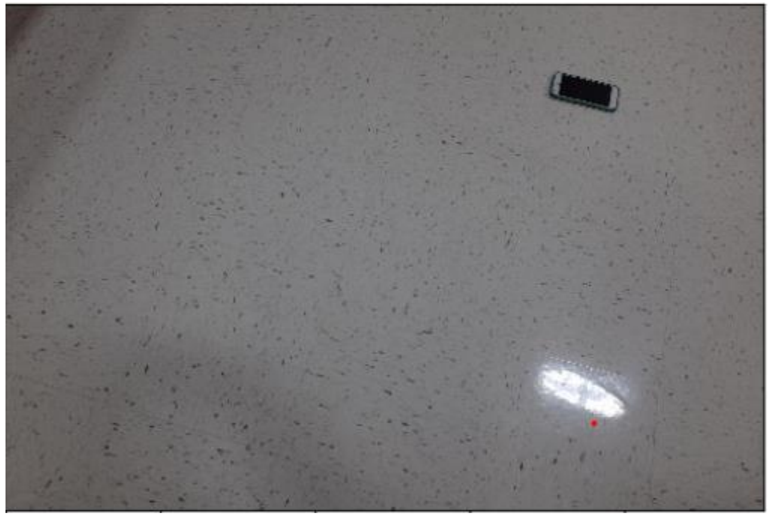
Testing Detection


120.jpg


121.jpg


122.jpg


123.jpg


124.jpg

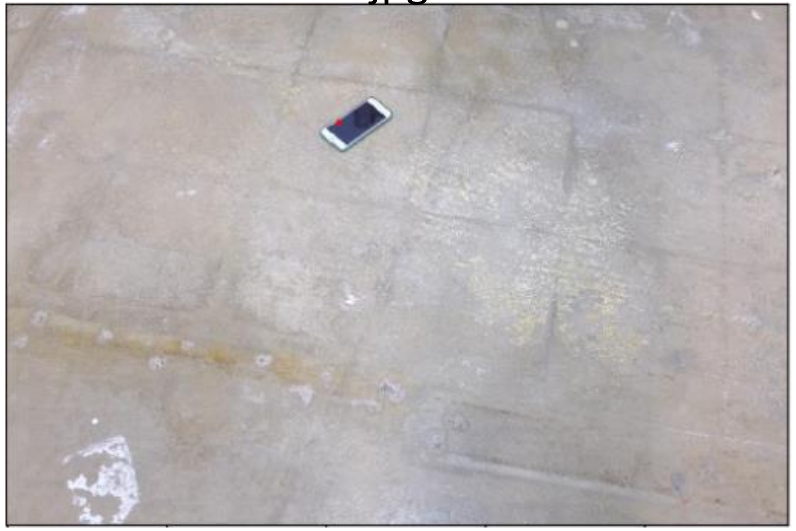

126.jpg

Table of Predictions

Name	Coordinate 1	Coordinate 2
120.jpg	0.825	0.875
121.jpg	0.275	0.825
122.jpg	0.275	0.825
123.jpg	0.475	0.525
124.jpg	0.775	0.825
126.jpg	0.425	0.225

Conclusion

- Difficulties
  - Unreliable convergence during training
  - Small dataset meant some classes were rarely predicted because of low occurrence in original dataset
  - Overfitting
- Limitations
  - Using classification means that even a “correct” prediction isn’t necessarily at the center of the object

Further Investigation

- Data augmentation: more data so classes can be more balanced; also may help with overfitting
- Weighing output classes to get prediction that is average of highest confidence classes rather than single class
- Custom Loss Function
- Larger number of epochs and hyperparameter search combinations (more time required)
- Explore different network architectures (maybe simpler) to reduce overfitting