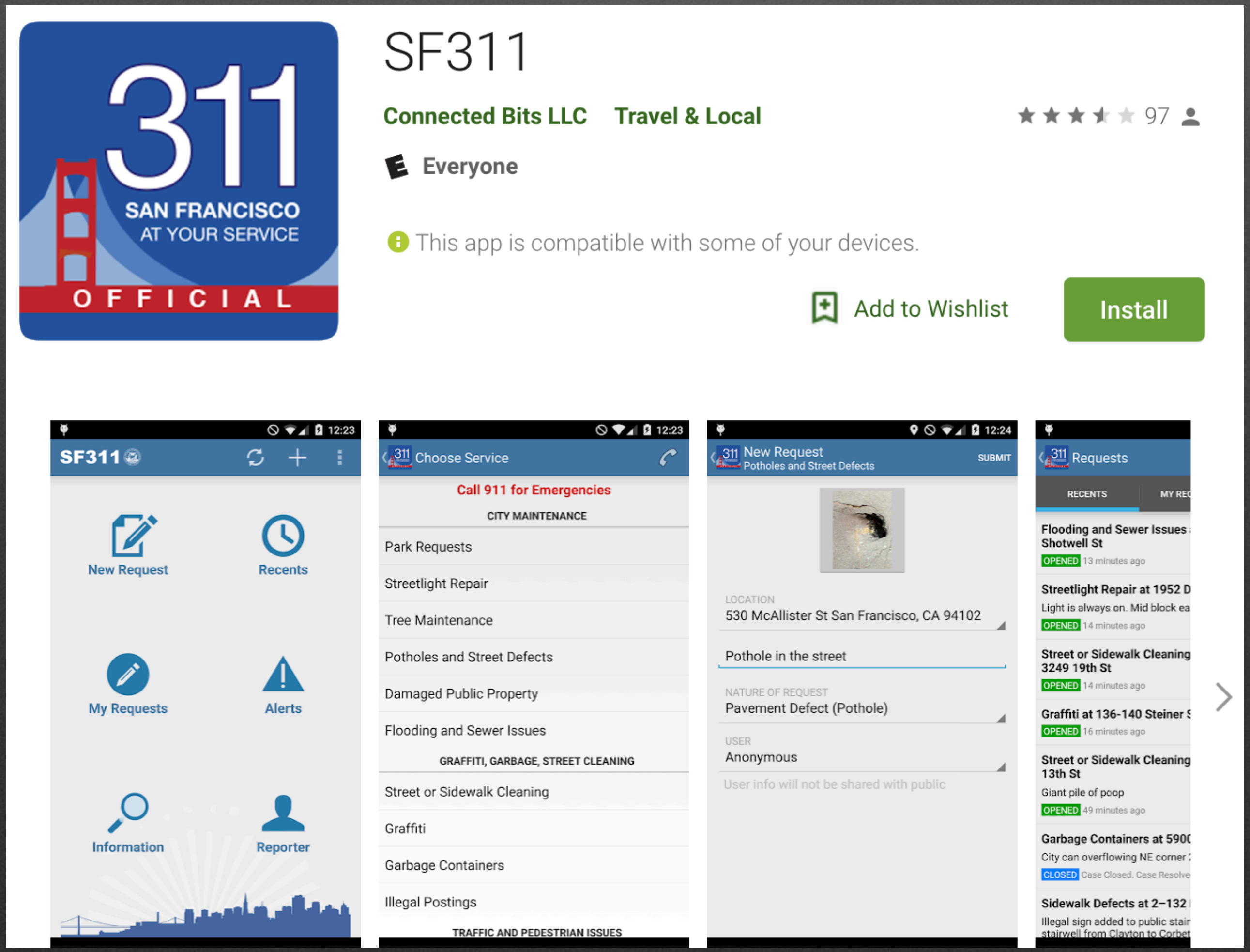
San Francisco 311 Reports Neural Network Image Classification

Crystal Bevis

San Francisco 311Cases

- 311 Reports - Non-emergency reports requesting city service
  - Example: Trash, graffiti, damaged roads
- Primarily created by mobile app
- Open data - images and categorical data about cases available



Objective

- Determine the type of case by using the images, location, and date data.
- Motivation:
    - Make it easier to submit cases
    - Audit cases - can predict which forms may need human review
    - Accessibility - all languages, vision impaired
    - Better allocate city resources

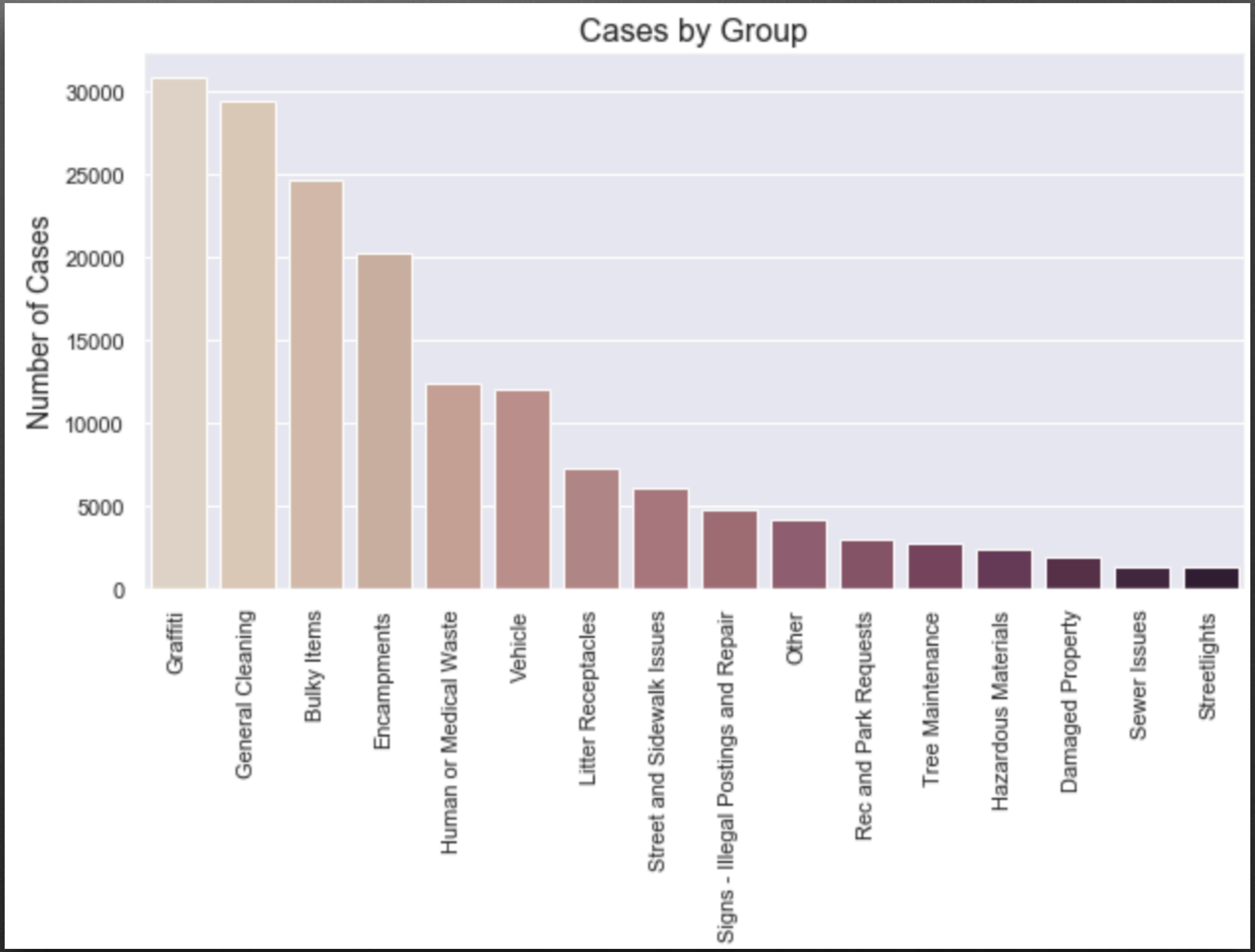
311 - Data

SF Open Data: https://data.sfgov.org/City-Infrastructure/311-Cases/vw6y-z8j6

- Large dataset
  - 3.48M samples
  - 20 features
- Selected a subset
  - Samples with valid image URLs
  - 2017-2018
  - Approximately 160,000 images
- Cropped edges, resize, grayscale, and save as 128x128 images on download

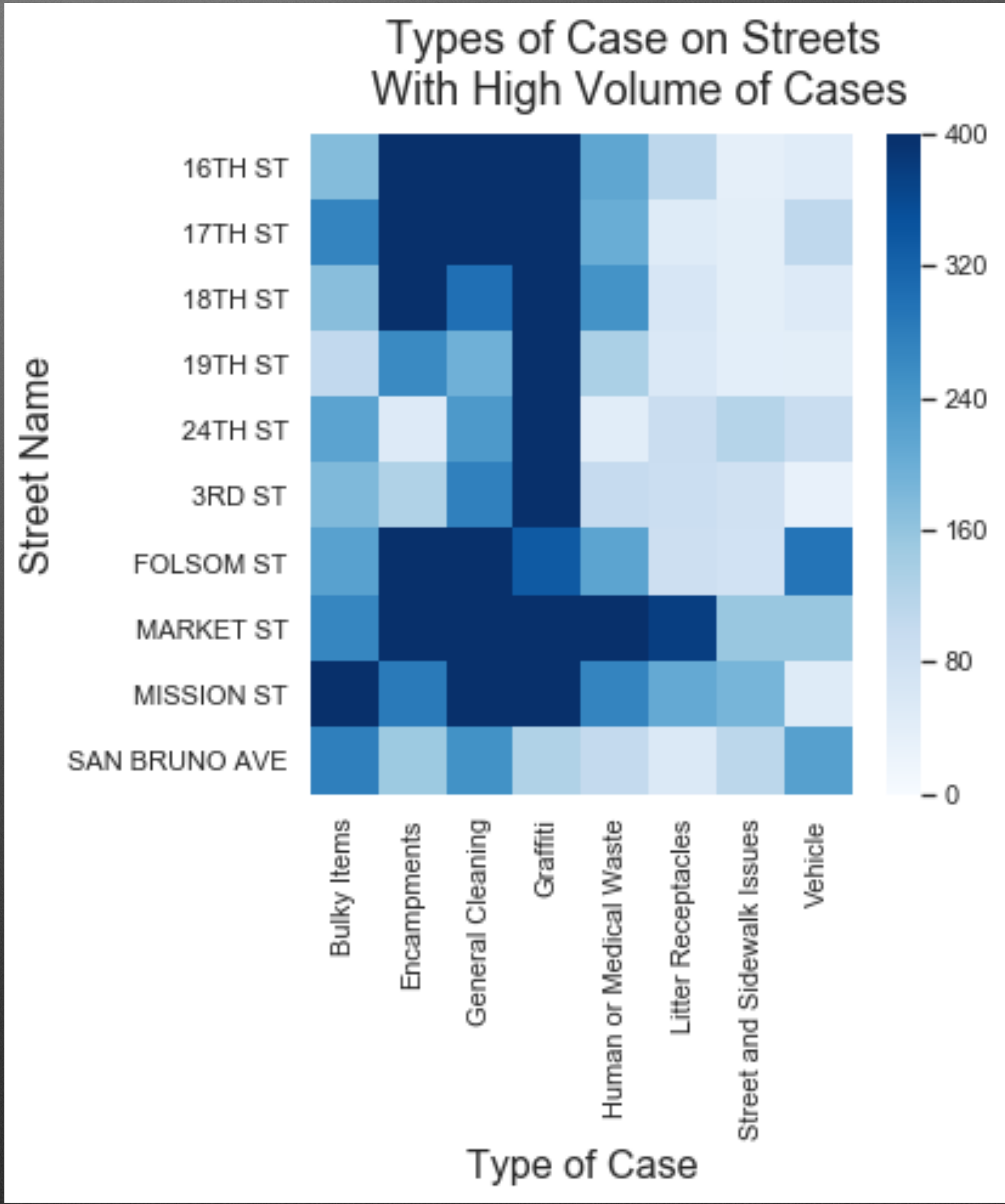
311 Data - Labels

- Which labels?
  - Category - 31 unique labels, ~50% in one class
  - Request Type - 119 unique labels
- Combined small categories
- Split large category
- Example:
  - Parking enforcement, abandoned vehicles -> Vehicles
- 16 Groups



Exploratory Data Analysis

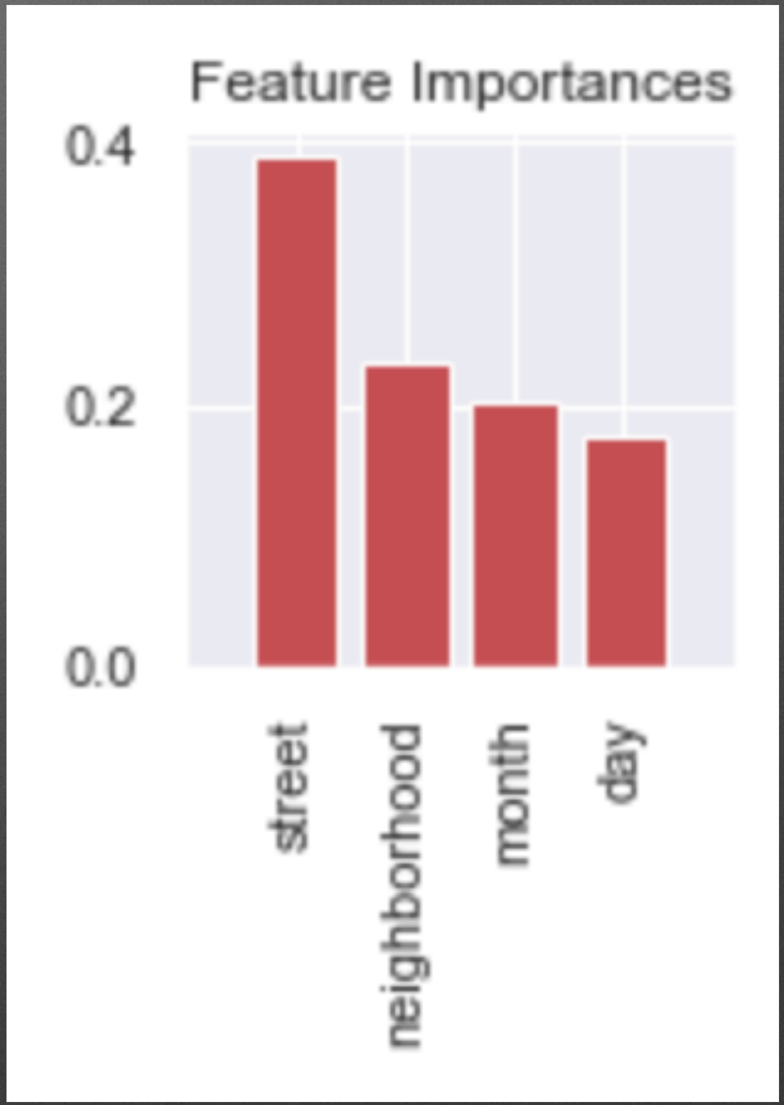
- Categorical features
  - Street
  - Neighborhood
  - Date
- Split date
  - Month
  - Day of the week
- Scale on heat map shown set to maximum of 400 to better see the distribution



Categorical Data Modeling

Random Forest

- Feature importances
- Handle categorical data
- Determine a baseline for MLP model
- Train accuracy: 38%
- Test accuracy: 32%



Multilayer Perceptron

- One hot encoded categorical features
- Applied variance threshold
  - ~ 400 features remaining
- Dense 128, activation Relu
- Dropout
- Dense 64, activation Relu
- Dropout
- Dense, activation Softmax

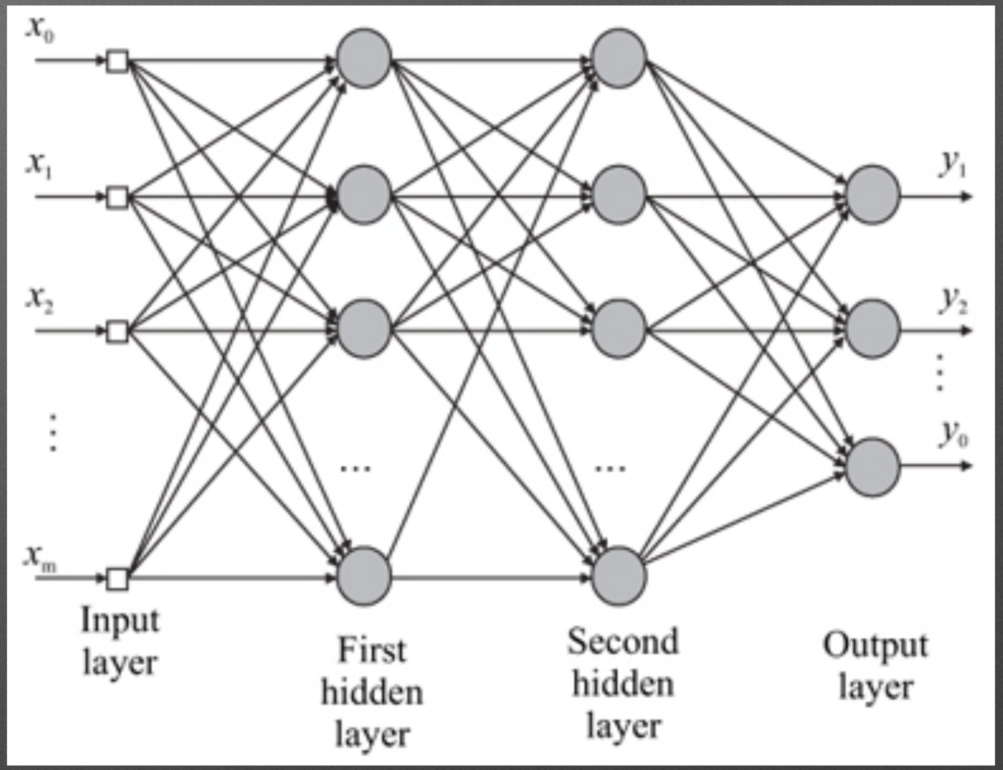


Image Source: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-77602013000200012

Multilayer Perceptron

- Training Accuracy: 0.34
- Test Accuracy: 0.32

- Top AUC:
  - 9: Parks and Recreation (0.95)
  - 15: Vehicles (0.81)
  - 2: Encampments (0.80)
- Bottom AUC:
  - 5: Hazardous Materials (0.61)

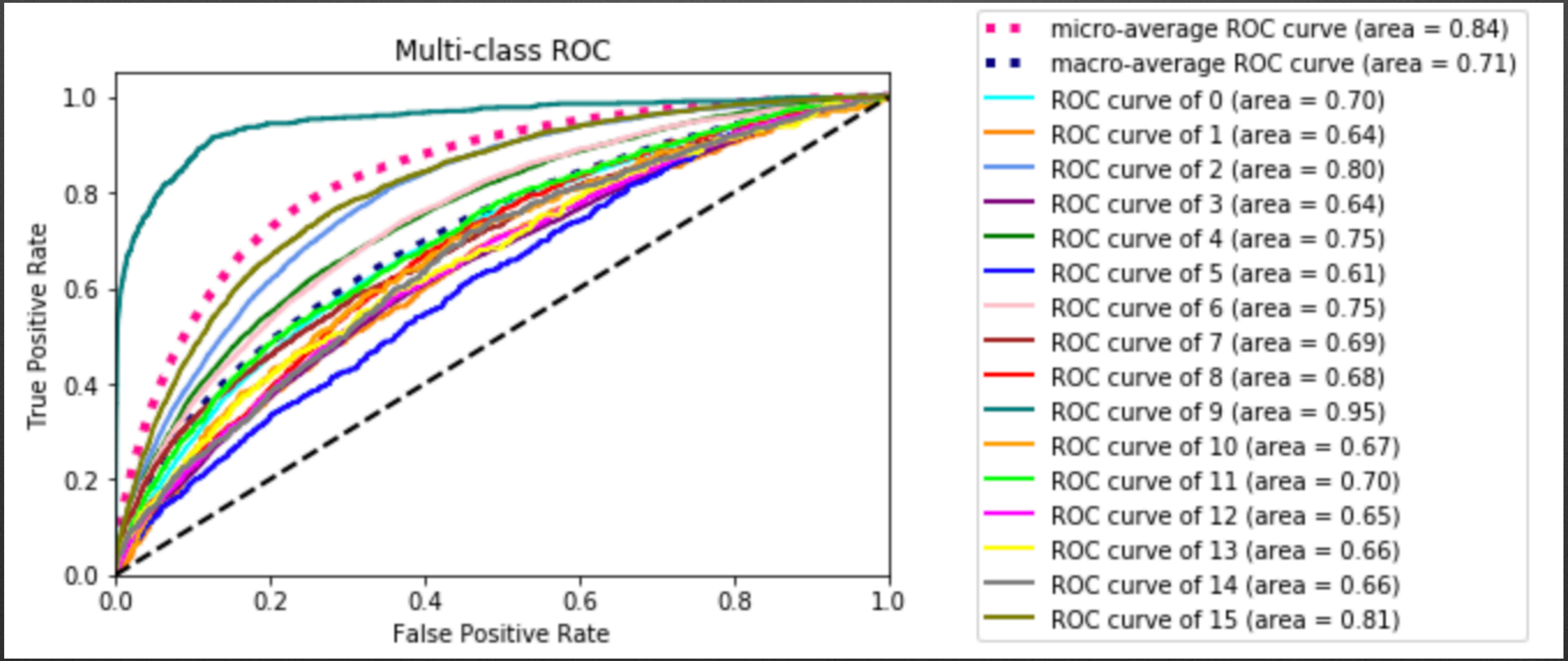


Image Data Modeling

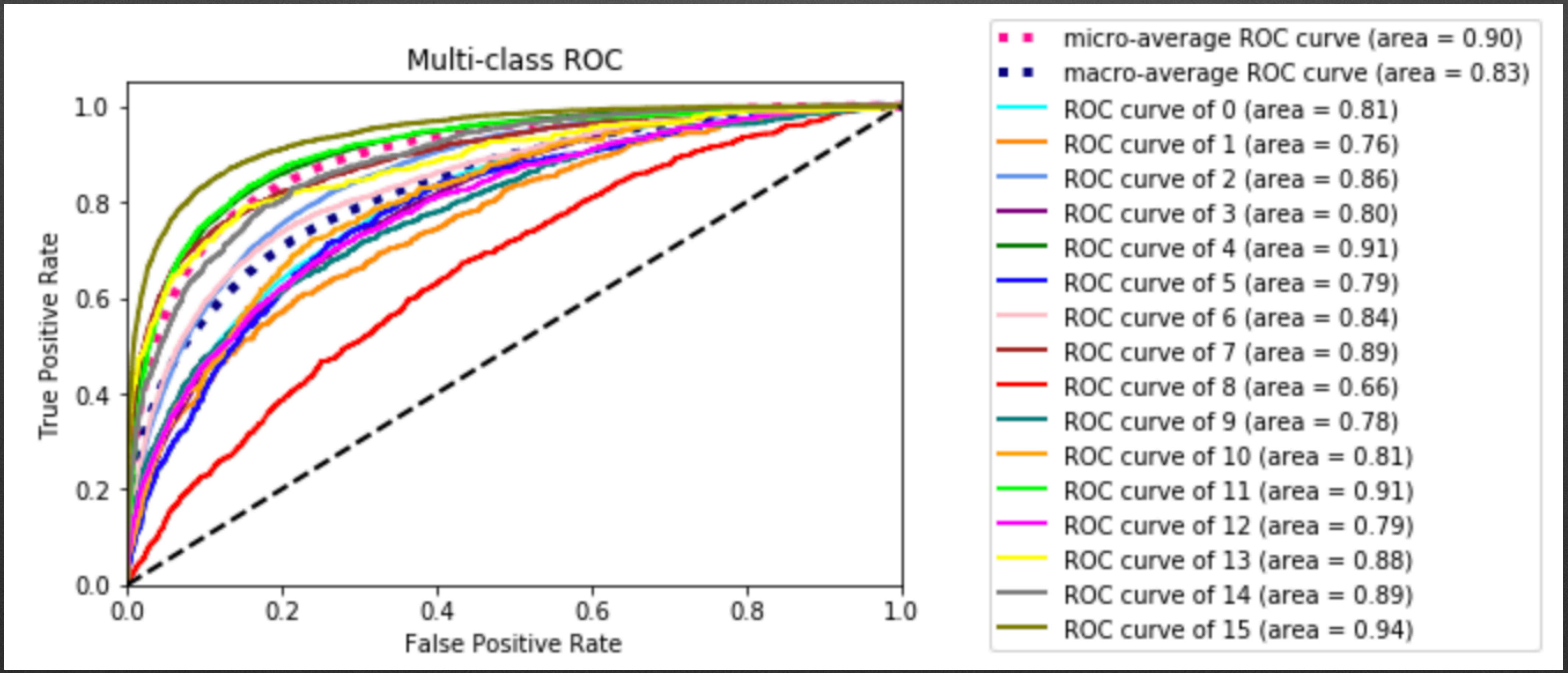
Convolutional Neural Network

- Moved from local machine to Google Colab (14 hrs -> 15 min)
  - 40,000 data points, basic version of model
- 6 Convolutional 2D layers
  - Kernel sizes (3 x 3)
  - ReLu activation
- Model overfits - used several techniques to reduce this
  - Batch normalization
  - Drop out
  - Max Pooling

Convolutional Neural Network

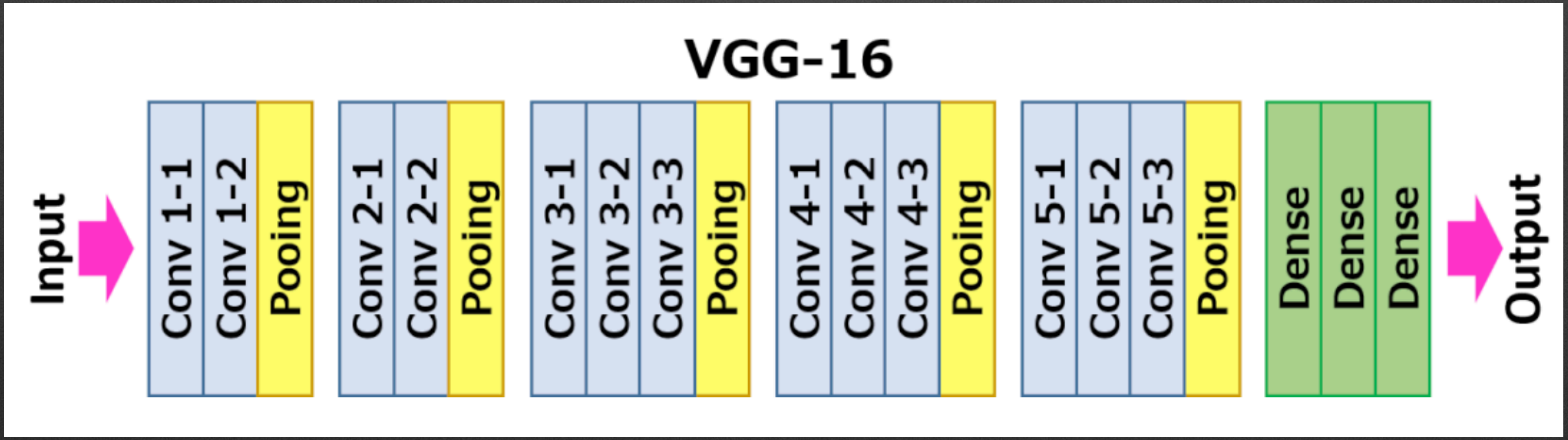
- Training Accuracy: 0.54
- Test Accuracy: 0.48

- Top 3 AUC:
  - 15: Vehicles (0.94)
  - 11: Signs - Illegal Postings and Repairs (0.91)
  - 4: Graffiti (0.91)
- Bottom AUC:
  - 8: Other (0.66)



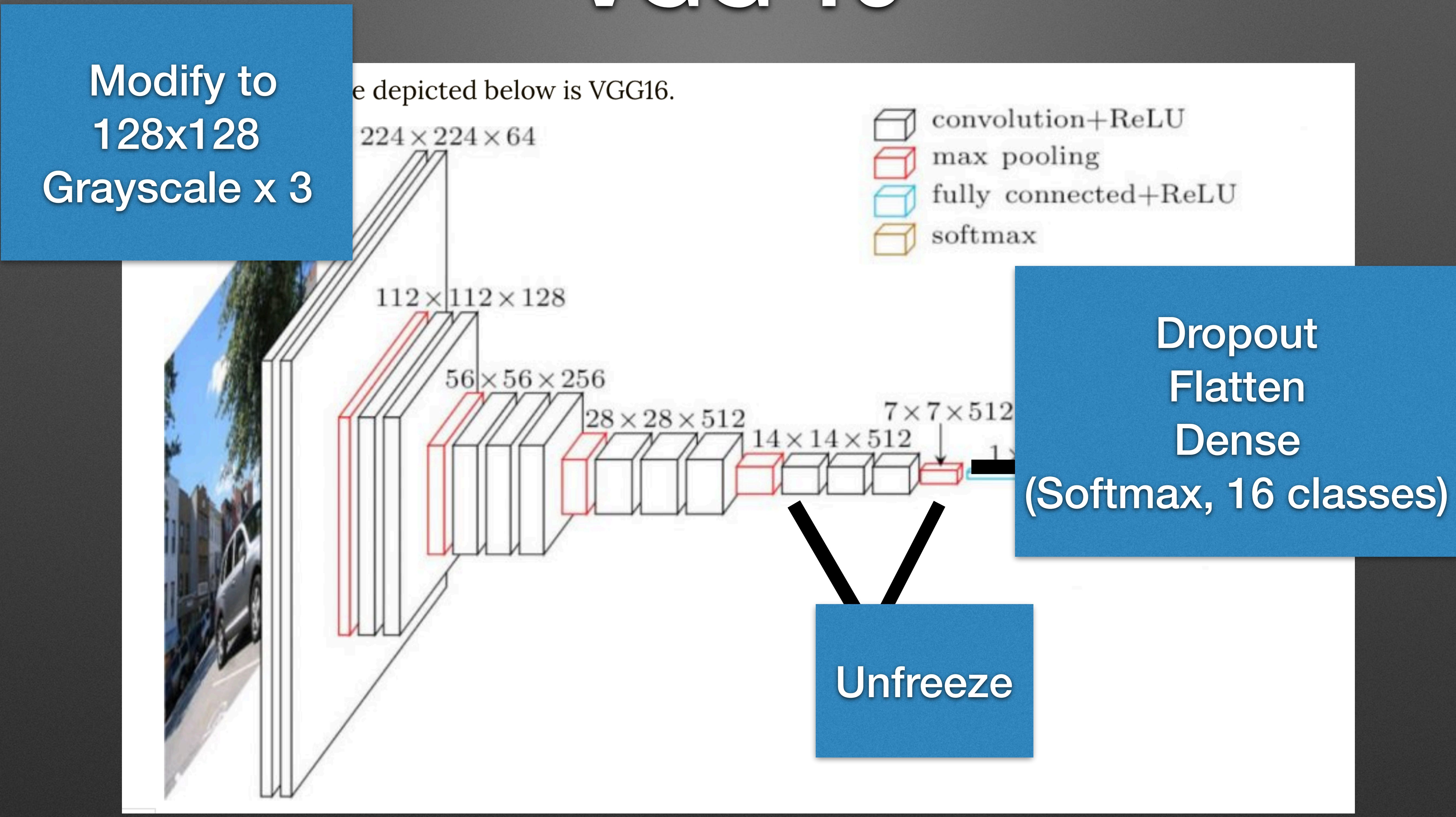
VGG 16

- Developed and trained by the Visual Geometry Group (VGG) at Oxford
- Very Deep Convolutional Network for Large-Scale Image Classification
- Pre-trained on ImageNet
  - Recognize 1,000 different object categories
- Easy to implement - included in Keras
- Large - over 533MB



https://neurohive.io/en/popular-networks/vgg16/

VGG 16

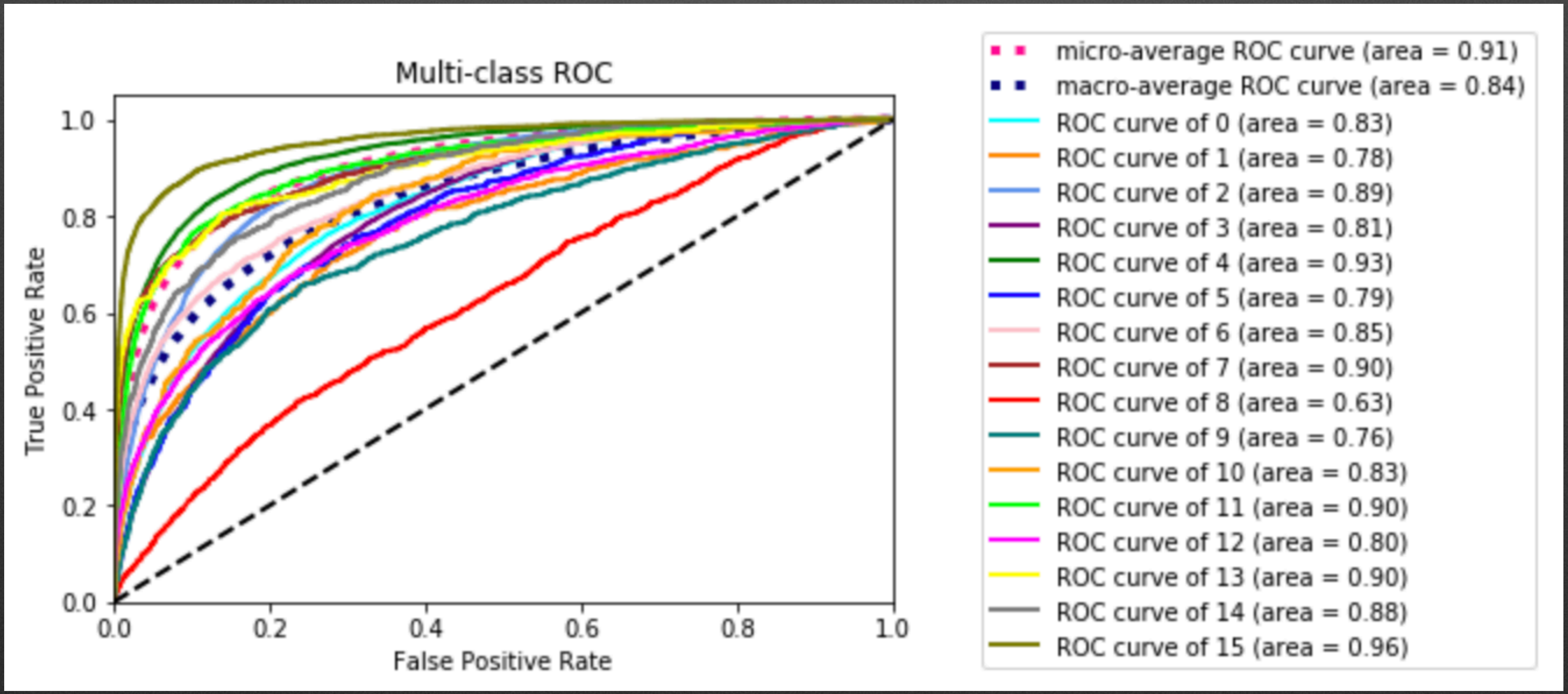


e depicted below is VGG16.
convolution+ReLU
max pooling
fully connected+ReLU
softmax
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
Modify to
128x128
Grayscale x 3
Dropout
Flatten
Dense
(Softmax, 16 classes)
Unfreeze

VGG 16 Results

- Training Accuracy: 0.78
- Testing Accuracy: 0.53

- Top 3 AUC:
  - 15: Vehicles (0.96)
  - 4: Graffiti (0.93)
  - 7: Litter Receptacles, 11: Signs - Illegal Postings and Repairs, 13: Streetlights (0.90)
- Bottom AUC:
  - 8: Other (0.63)



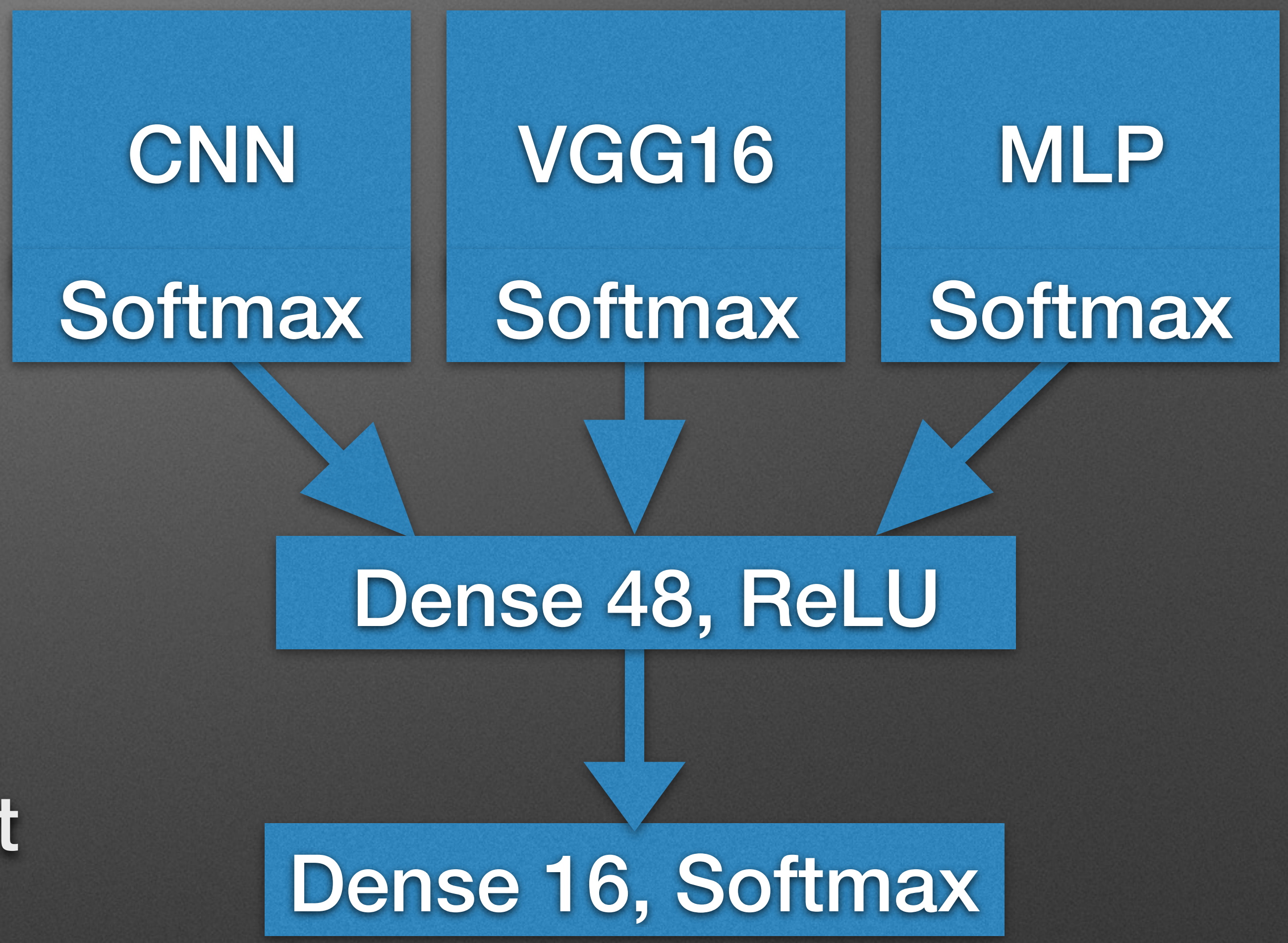
Combining Mixed Data Types

Multiple Data Types Model

- Train accuracy: 62%
- Test accuracy: 54%

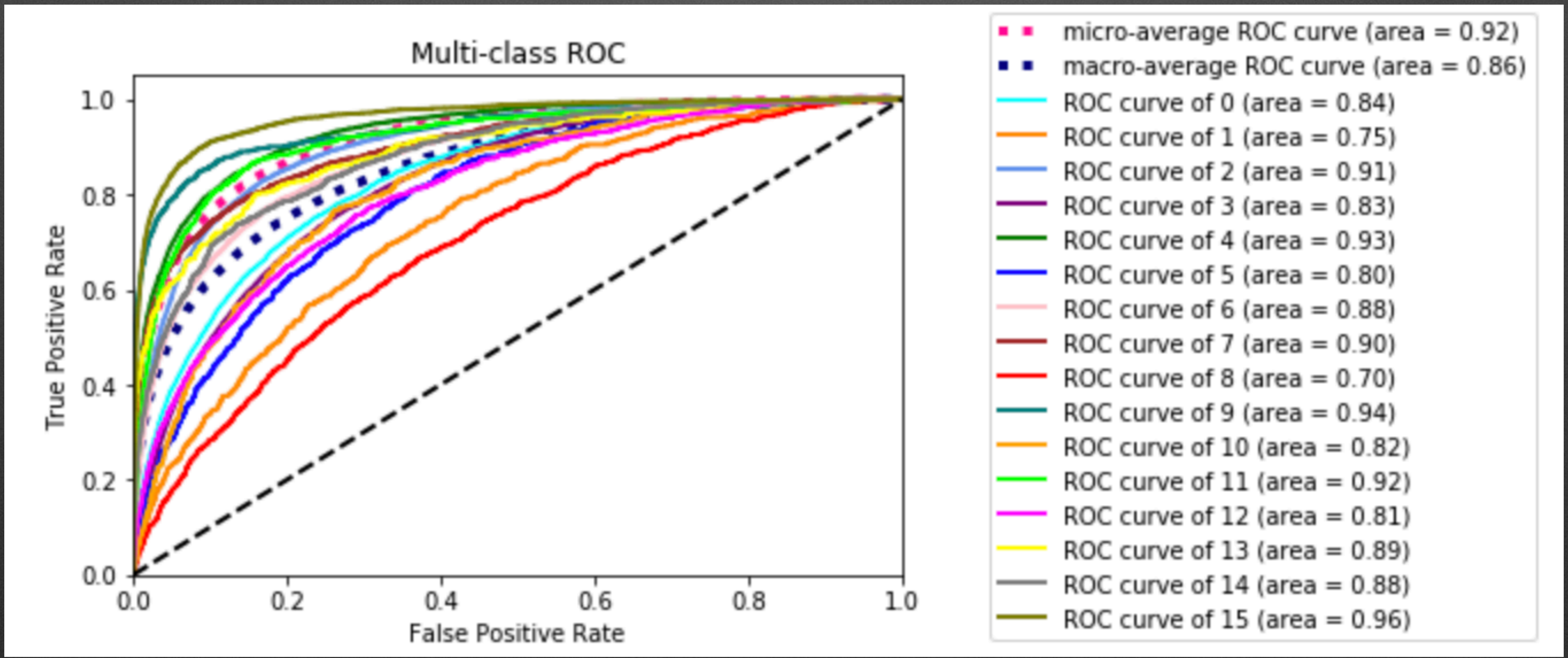
- Using 2 models would max at 32%

- Overtraining, but adding dropout hurt the test performance



Multiple Data Types Model

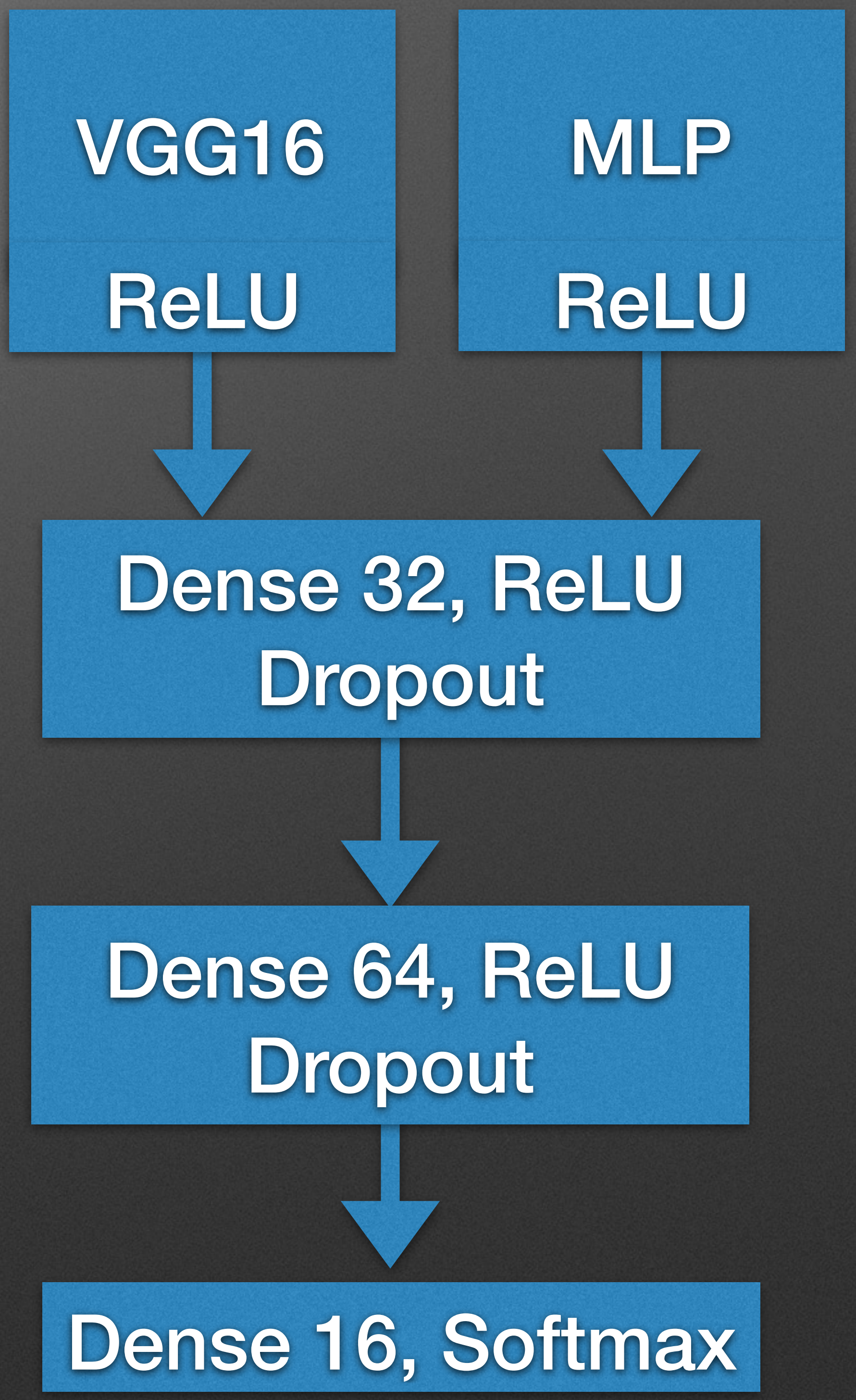
- Train accuracy: 62%
- Test accuracy: 54%
- Best AUC:
  - 15: Vehicles (0.96)
  - 9: Parks and Recreation (0.94)
  - 4: Graffiti (0.93)
- Worst AUC:
  - 8: Other (0.70)
  - 1: Property Damage (0.75)
- Overall accuracy not much higher than VGG16 alone, but it correctly classifies parks and recreation



Multiple Data Types
VGG 16 and MLP

- Reinitialize models before training
- Changed outputs of individual models to ReLU
- No Softmax until the end

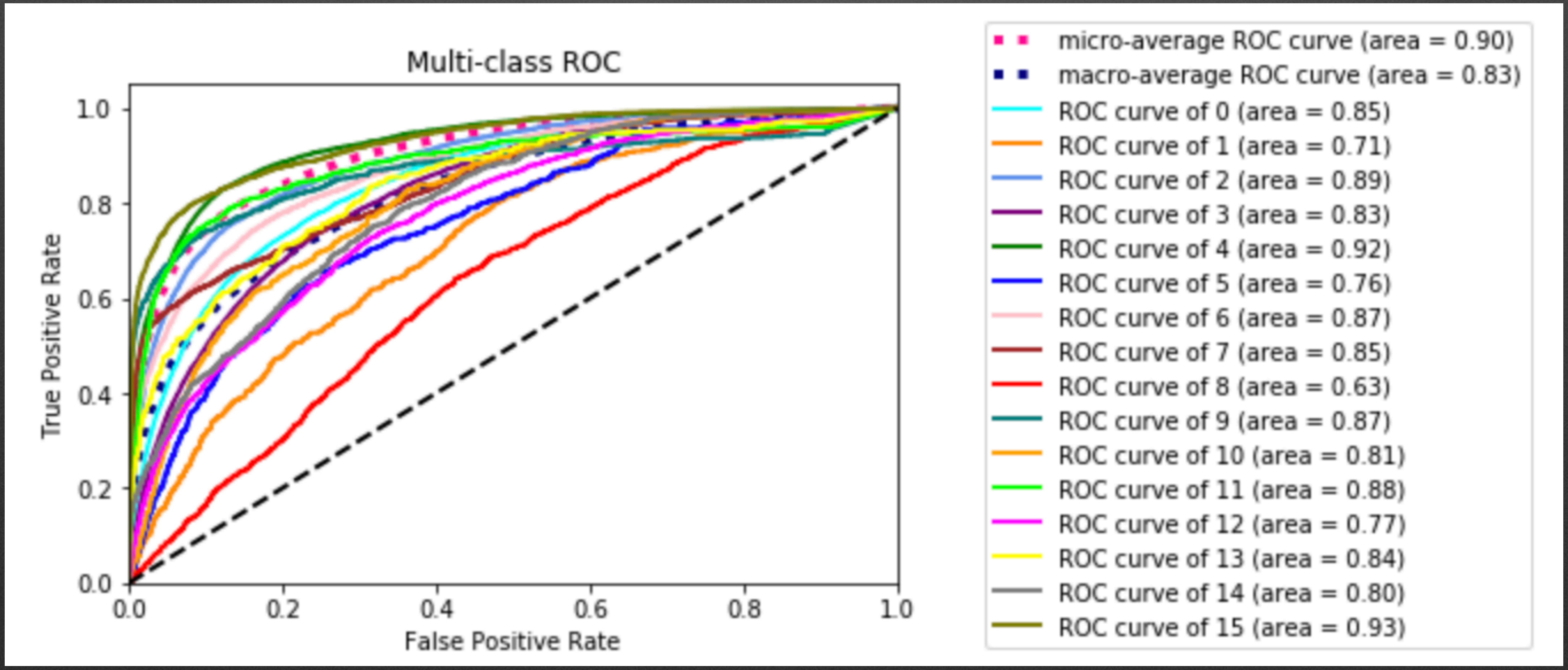
- Train Accuracy: 0.54
- Test Accuracy: 0.54



Multiple Data Types VGG 16 and MLP

- Best AUC:
  - 15: Vehicles (0.93)
  - 4: Graffiti (0.92)
  - 2: Encampments (0.89)
- Bottom:
  - 8: Other (0.63)

- Less overtraining
- Less complex - less memory involved per batch
- Slightly lower AUC scores than 3 model combo



Example 1

Graffiti (2%)

Predictions

- General Cleaning (33.4%)
- Bulky Item (32.9%)

Example 2

Vehicle

Prediction

- Vehicle (99%)

Example 3

Bulky Item

Predictions

- Bulky Item (58%)
- General Cleaning (17%)

Example 4

Signs - Illegal Postings and Repair (5.9%)

Predictions

- Graffiti (93.6%)
- Signs - Illegal Postings and Repair (5.9%)

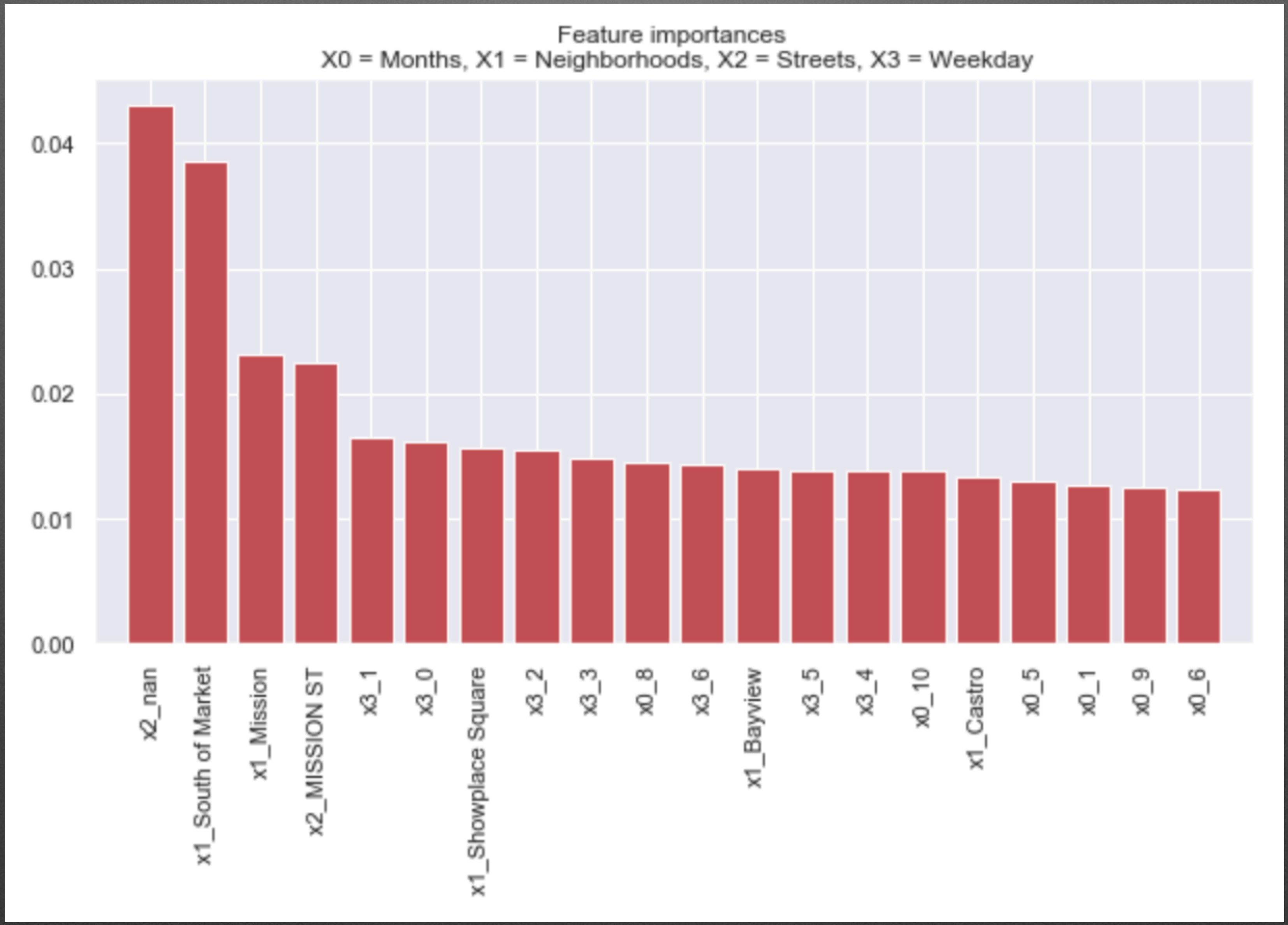
Future Work

- Expand to include time series information across years: 2013-2019
  - Are types of cases increasing or decreasing over time?
- Include more images or more pixels
- Use RGB images
- Include image rotation
- Refine groupings

Conclusion

- Small accuracy gain using mixed data types over images alone (1% gain)
- Biggest increase in AUC in a couple of smaller groups
- Preferred model
    - VGG 16 and MLP combined
        - Small drop in AUC scores for less complexity and less memory intensive

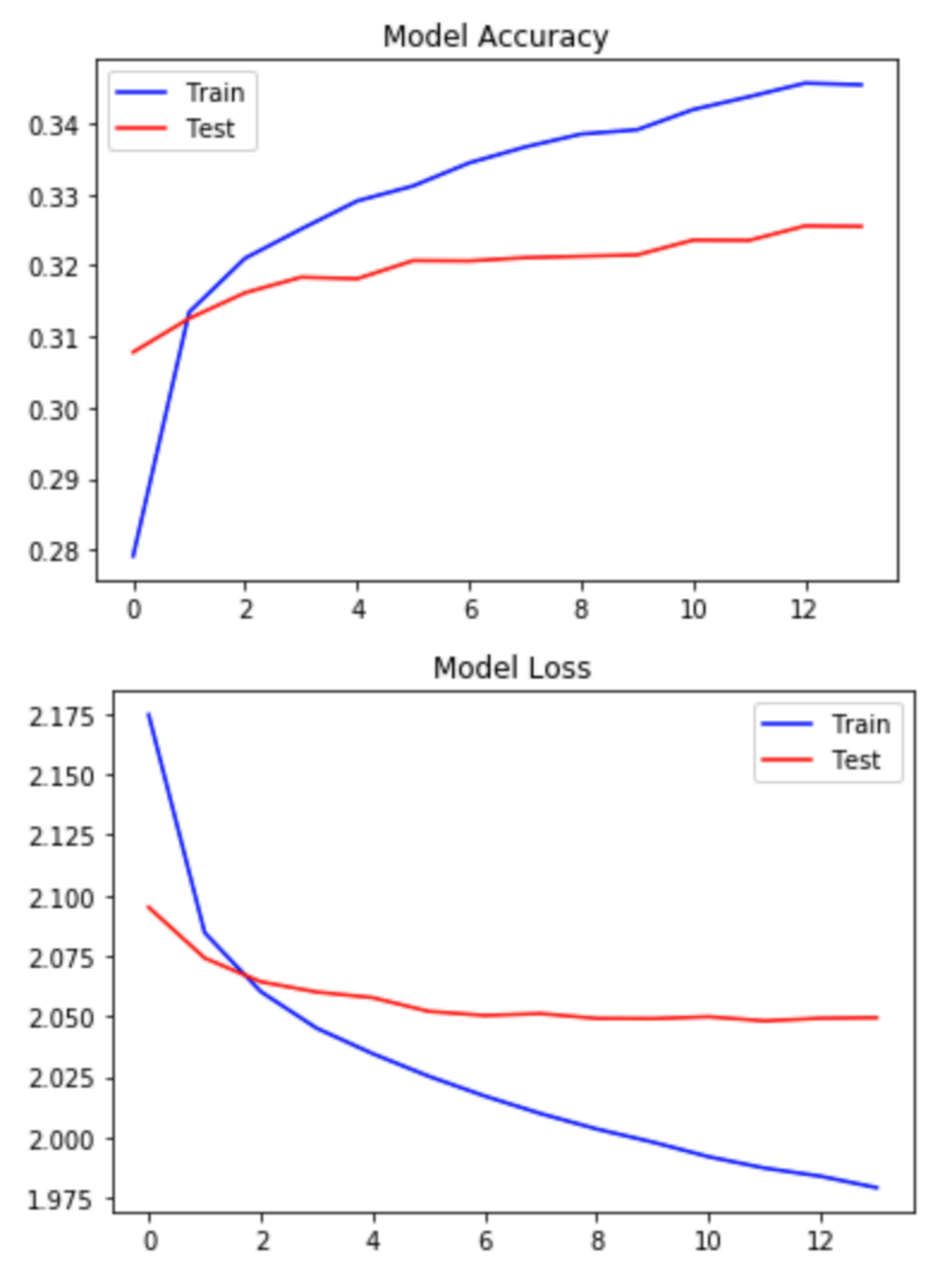
Thank you!



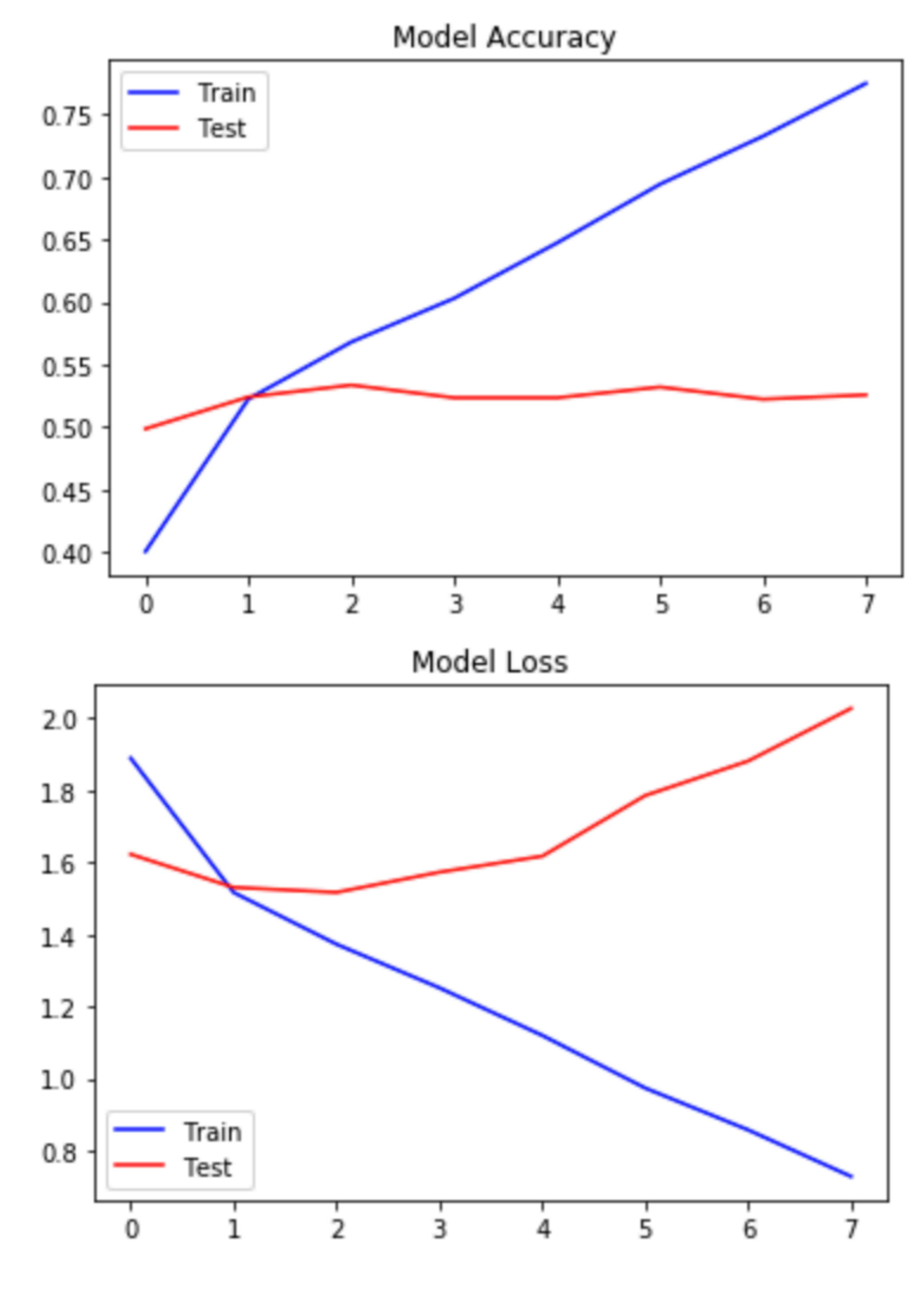
Feature importances
X0 = Months, X1 = Neighborhoods, X2 = Streets, X3 = Weekday
0.04
0.03
0.02
0.01
0.00
x2_nan
x1_South of Market
x1_Mission
x2_MISSION ST
x3_1
x3_0
x1_Showplace Square
x3_2
x3_3
x0_8
x3_6
x1_Bayview
x3_5
x3_4
x0_10
x1_Castro
x0_5
x0_1
x0_9
x0_6

MLP

VGG 16

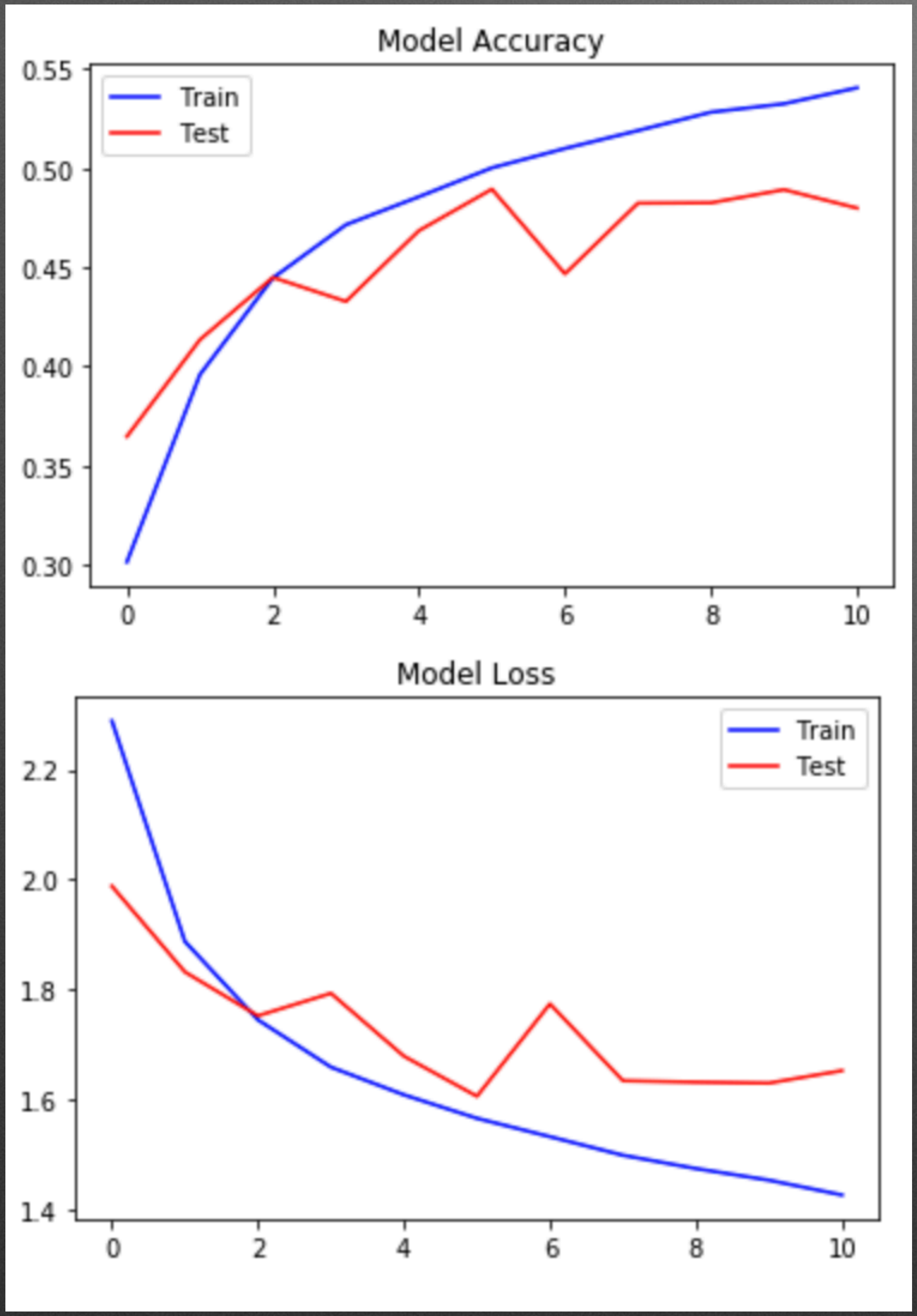


Model Accuracy
Train
Test
0.34
0.33
0.32
0.31
0.30
0.29
0.28
0
2
4
6
8
10
12
Model Loss
Train
Test
2.175
2.150
2.125
2.100
2.075
2.050
2.025
2.000
1.975
0
2
4
6
8
10
12



Model Accuracy
Train
Test
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0
1
2
3
4
5
6
7
Model Loss
2.0
1.8
1.6
1.4
1.2
1.0
0.8
Train
Test
0
1
2
3
4
5
6
7

CNN


Model Accuracy
Train
Test
0.55
0.50
0.45
0.40
0.35
0.30
0
2
4
6
8
10
Model Loss
Train
Test
2.2
2.0
1.8
1.6
1.4
0
2
4
6
8
10