Fundamentals of ML Final Project

Team Members: Damian Huerta & Nathan Wang

Georgia Tech

Methodology

- 32239 Images
- Classifiers
  - K-Means
  - GMM
  - KNN
  - CNN
- Image Resize
  - Initially used torch.resize()
  - Switched to Auto Encoders

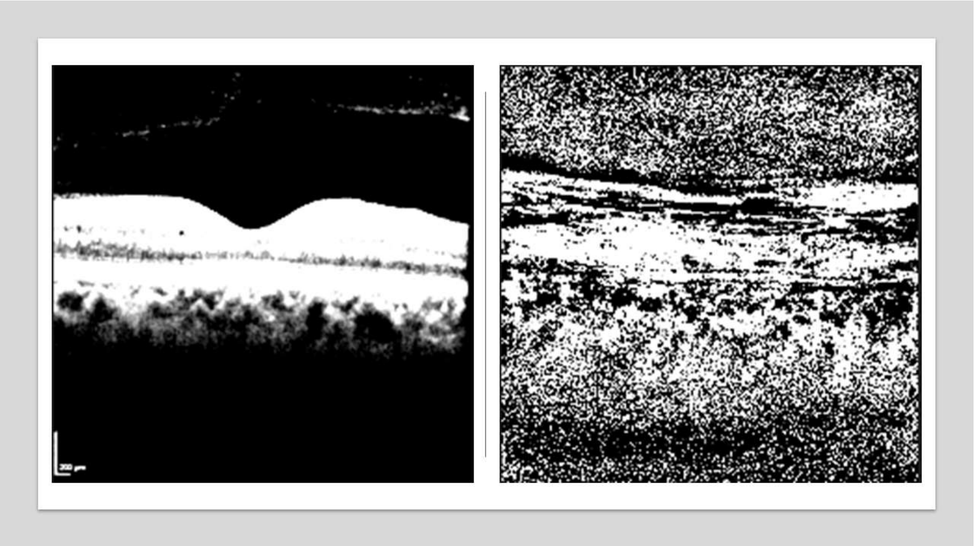

Figure 1. Reconstructed training sample image from trained AE

Methodology Cont'd

K-Means

- Used K-Means class in sk-learn python library
- Tested with multiple classes (ended up with 3)

GMMs

- Used GMM class in sk-learn python library
- Tested with multiple classes (ended up with 3)
- Tested with different covariances (tied, full, diag, and spherical)
    - Ended up using tied covariance

KNN

- Simple apprach to classify based on K neighbors majority vote
- Utilized 50-feature samples
- Tested over various K values to maximize accuracy
- Improved accuracy over default images

CNN

- Initial approach trained ResNet on OLIVE training set
- Pivoted to custom CNN architecture based on AlexNet
- Experimented with Cross Entropy Loss, MSE, and Focal Loss
- Incorporated dropout to combat overfitting

Results: K-Means

- 43.87% Accuracy
- Predicts 77.9% of samples with true class1 as class 1
- Virtually incapable of predicting class 0 and 2
- Potential Reason of Inaccuracy
    - One Big Cluster
    - Poor Image Quality/Orientation

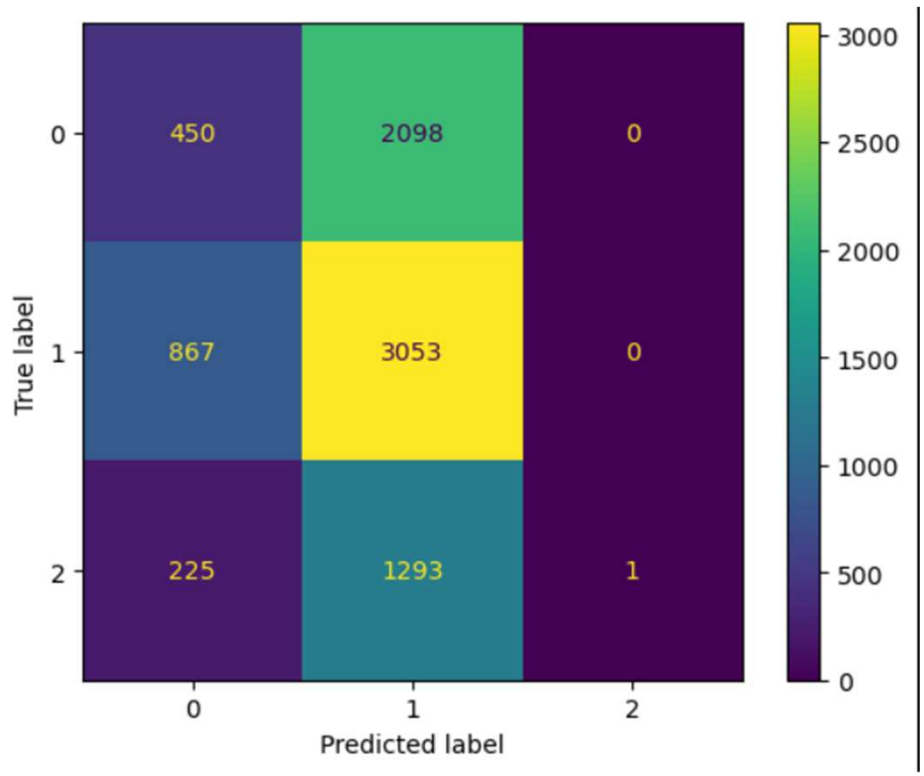

Figure 2. Confusion Matrix for K-Means Classifier on Testing data set

Georgia Tech

Results: GMM

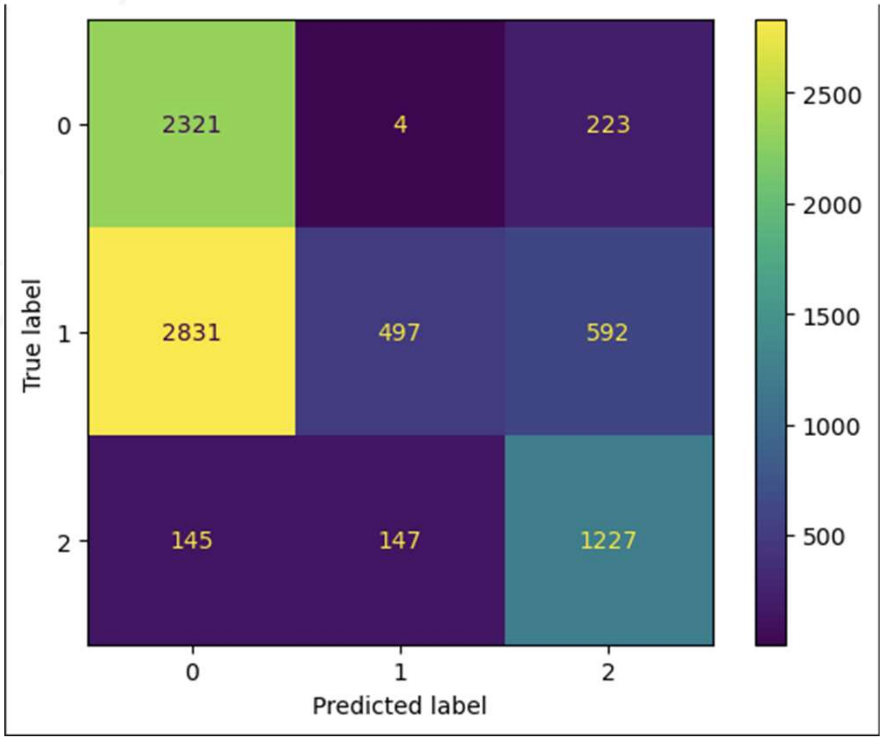

Figure 2. Confusion Matrix for GMM Classifier on Testing data set

- 50.6% Accuracy on Testing Set
- Stability Issues when using flattened images
  - 32x32 resolution
- Predicts True Label 0 accurately
- Struggles to Differentiate True Label1 from 0
- Potential Reason for Inaccuracy:
  - Poor Image Quality

Results: KNN

- Experimentation found that K = 4 performed best with encoded samples
- K>100 resulted in stability issues and decreased accuracy
- Model predicts true label 1 as label 1 with 63.83% accuracy.
- Overall accuracy of 44.71% with encoded samples
  - 38.4% accuracy with flattened images

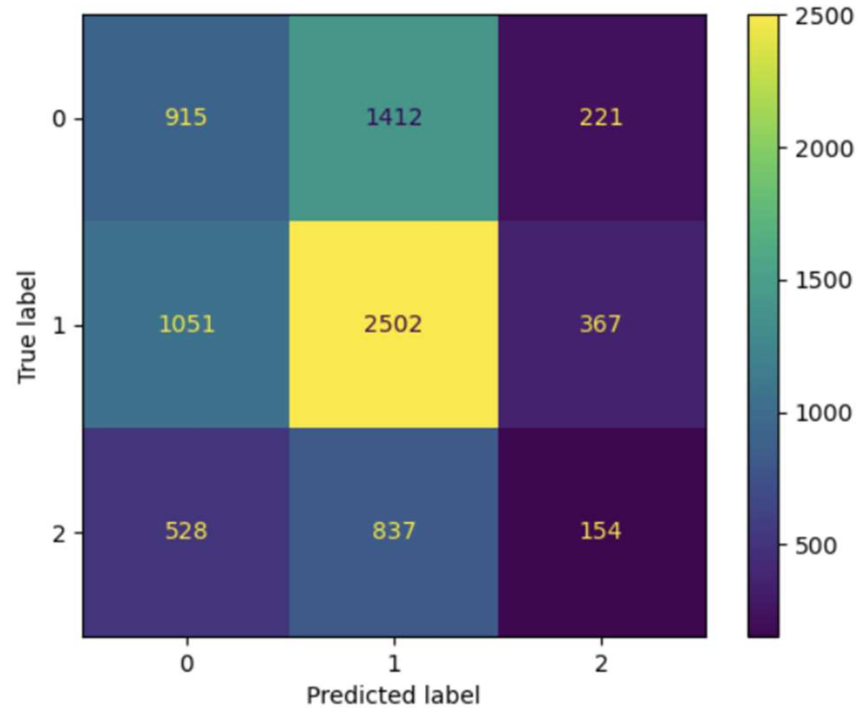

Figure 3. Confusion Matrix for KNN Classifier on Testing data set

Results: CNN

- 44.94% testing accuracy with ResNet
  - Overfitting issue (75.38% training accuracy) -> O-Net (Custom CNN)

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
├─Sequential: 1-1                        [-1, 256, 6, 6]           --
|    └─Conv2d: 2-1                       [-1, 64, 55, 55]          7,808
|    └─ReLU: 2-2                         [-1, 64, 55, 55]          --
|    └─MaxPool2d: 2-3                    [-1, 64, 27, 27]          --
|    └─Conv2d: 2-4                       [-1, 192, 27, 27]         307,392
|    └─ReLU: 2-5                         [-1, 192, 27, 27]         --
|    └─MaxPool2d: 2-6                    [-1, 192, 13, 13]         --
|    └─Conv2d: 2-7                       [-1, 384, 13, 13]         663,936
|    └─ReLU: 2-8                         [-1, 384, 13, 13]         --
|    └─Conv2d: 2-9                       [-1, 256, 13, 13]         884,992
|    └─ReLU: 2-10                        [-1, 256, 13, 13]         --
|    └─Conv2d: 2-11                      [-1, 256, 13, 13]         590,080
|    └─ReLU: 2-12                        [-1, 256, 13, 13]         --
|    └─MaxPool2d: 2-13                   [-1, 256, 6, 6]           --
├─AdaptiveAvgPool2d: 1-2                 [-1, 256, 6, 6]           --
├─Sequential: 1-3                        [-1, 3]                   --
|    └─Linear: 2-14                      [-1, 4096]                37,752,832
|    └─ReLU: 2-15                        [-1, 4096]                --
|    └─Dropout: 2-16                     [-1, 4096]                --
|    └─Linear: 2-17                      [-1, 4096]                16,781,312
|    └─ReLU: 2-18                        [-1, 4096]                --
|    └─Dropout: 2-19                     [-1, 4096]                --
|    └─Linear: 2-20                      [-1, 3]                   12,291
|    └─Softmax: 2-21                     [-1, 3]                   --
==========================================================================================
Total params: 57,000,643
Trainable params: 57,000,643
Non-trainable params: 0
Total mult-adds (M): 720.24
==========================================================================================
Input size (MB): 0.19
Forward/backward pass size (MB): 3.76
Params size (MB): 217.44
Estimated Total Size (MB): 221.39
==========================================================================================
```

Figure 5. O-NET Model summary

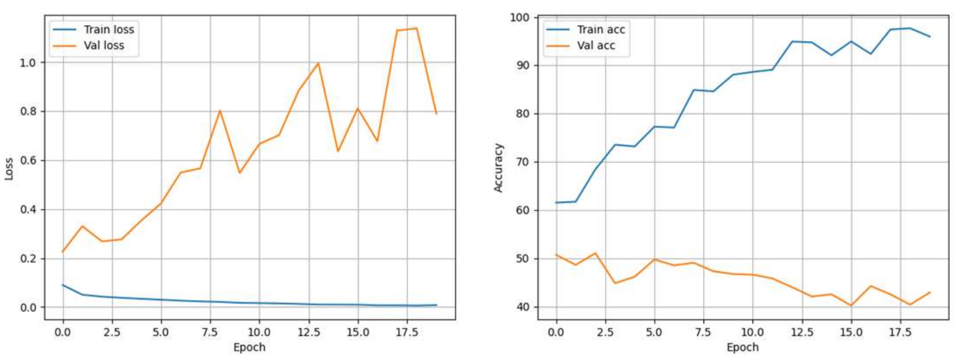

Figure 4. Loss of training and testing dataset during each epoch of training

Results: CNN

- 51.06% testing accuracy on O-NET
  - .2 dropout, Focal Loss, 21 layers, Adam Optimizer used for best performing model
  - Still had issue with overfitting despite dropout

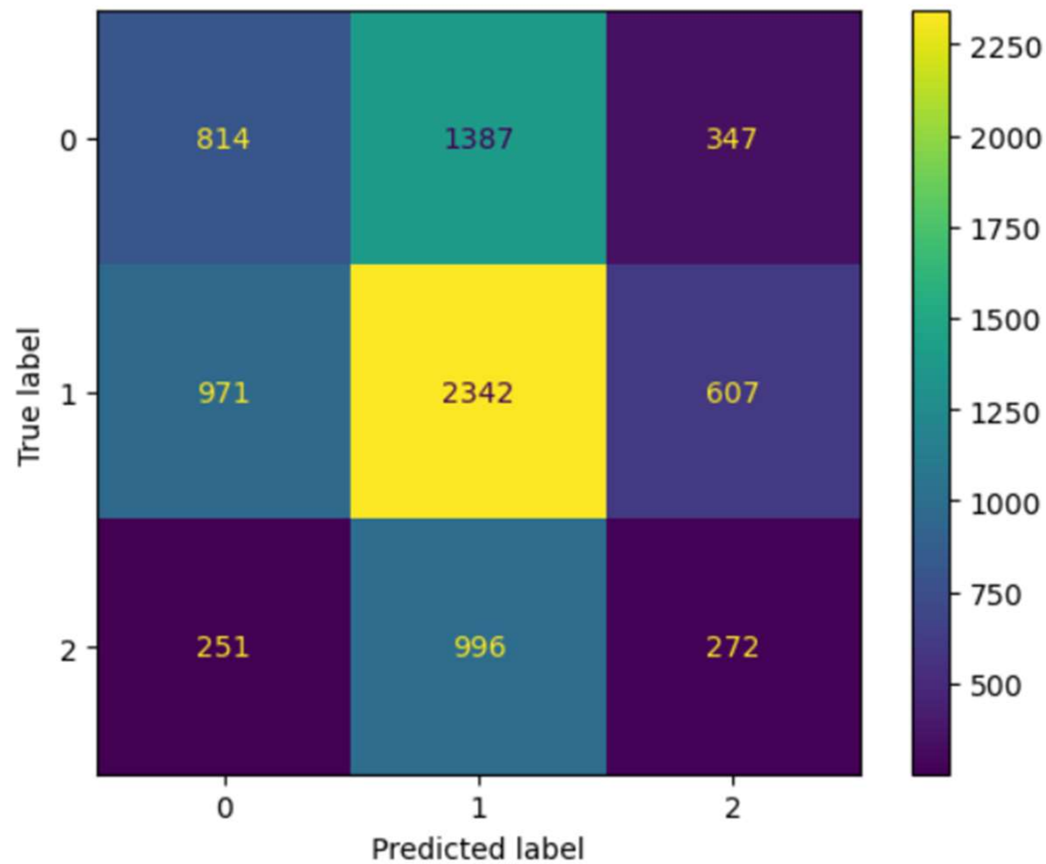

Figure 6. Confusion Matrix for CNN Classifier on Testing data set

Conclusion

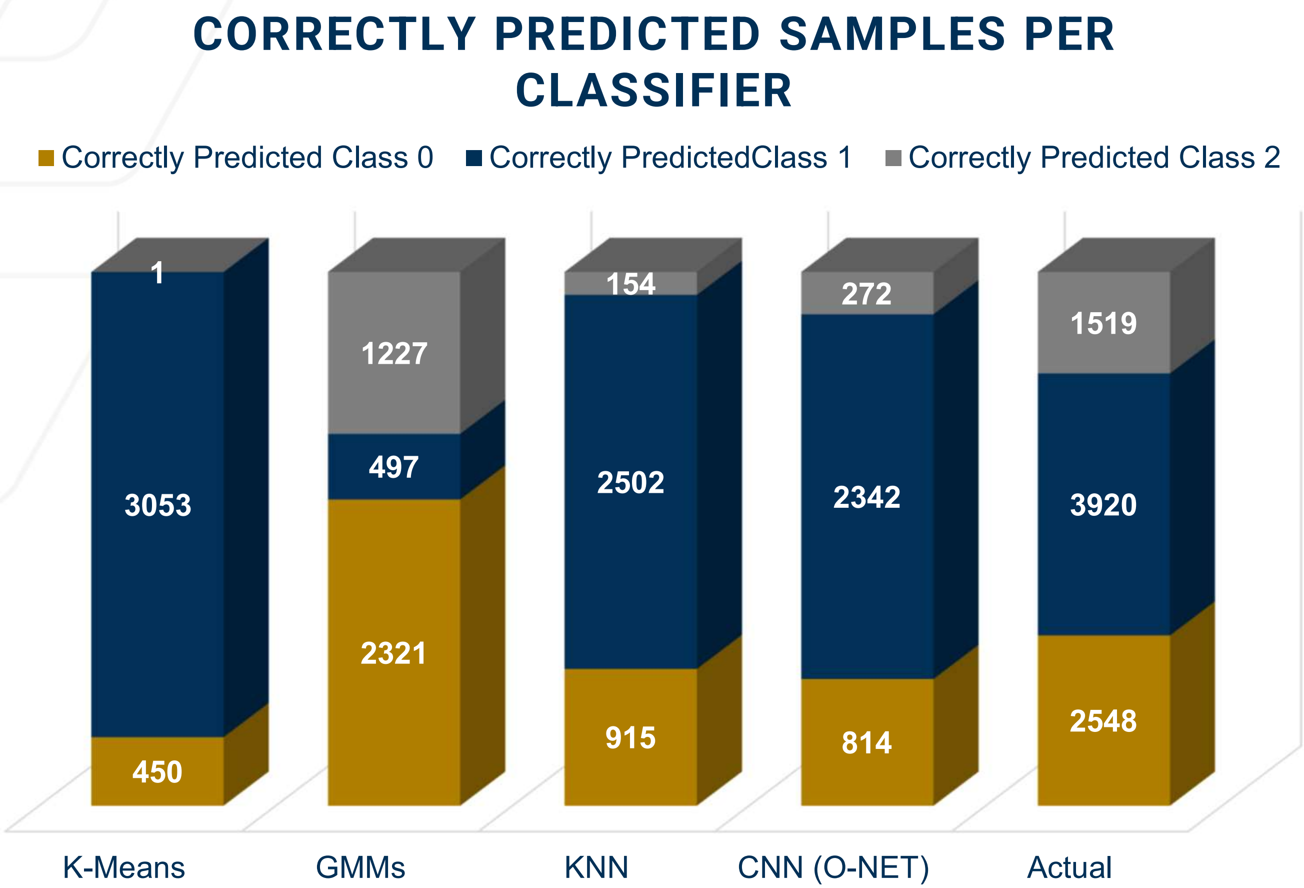

Figure 7. Correctly predicted samples per classifier

- CNN has the best testing/training accuracy overall
- Different models perform well on different classes
- Potential Reason:
  - Model has better adaptability (K-Means, GMM)
  - Feature space representation

Conclusion

- Dataset is heavily staggered
  - Almost half of data is Class 1
  - Could result in bias towards choosing Class 1 over other classes
- GMM catches 80.77% of highest severity class (2) and 91.09% of lowest severity class (0).
- K-Means catches/performs best for class 1

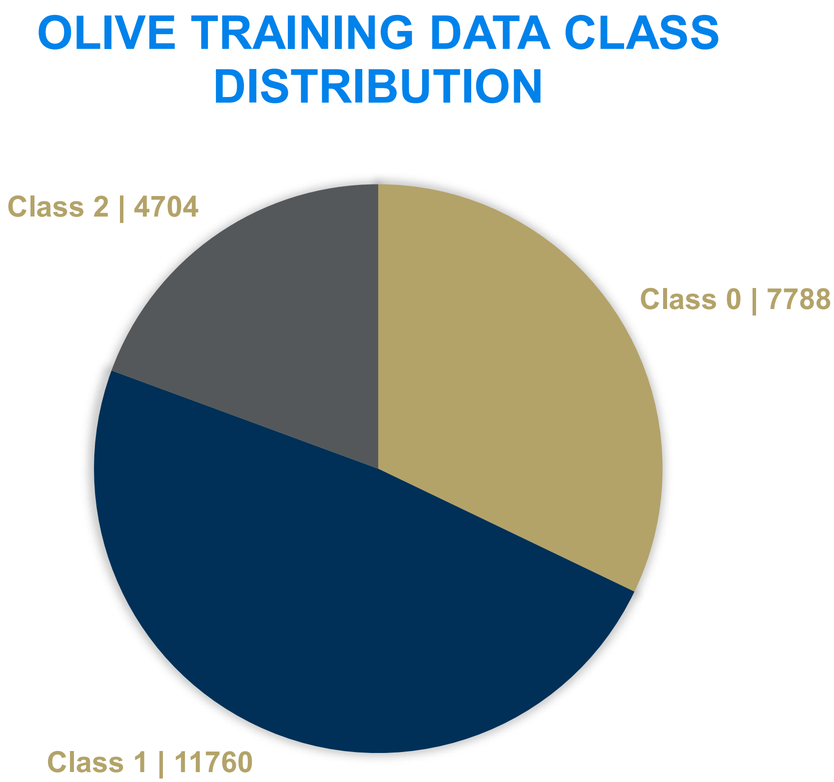

Figure 8. Olive dataset class distribution

Future Works

- Exploration of Feature Selection
  - PCA, CAE, etc.
- Combination of multiple classifiers
  - Exploit classifier's strong classes
- Data Set Distribution
  - Even Class Distribution and larger sample size
- CNN Architecture
  - Layers, Custom Loss, Dropout, etc.

Georgia Tech