Aaron Lo
atl2374
14 November 2023

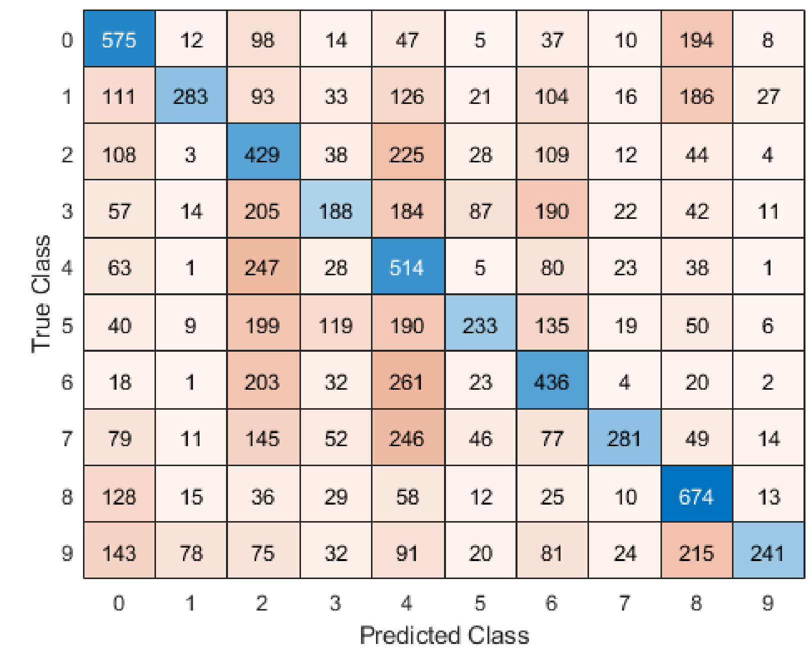
CS376 Assignment 4

I. Short answer problems
   A. What is the relation between image classification and object detection? Please give two concrete examples on how they are related (e.g., one is applied to solve another)?
      1. One is the use of a system to identify the number of cars in a given image for autonomous driving. Image classification can give information about whether or not a car is in the scene while object detection can provide bounding boxes around the cars. While object detection in this scenario can do both, the image classification can act as a faster filter to not have to run the more expensive objection detection on every image.
      2. Another is you can have object detection identify various location of objects in a scene but have image classification categorize the objects and detail them more specifically. For instance, the object detection can group animals while image classification will tell you which animals as a post processing step.
   B. Please compare semantic segmentation, instance segmentation and object detection. Describe the similarities and differences
      1. Semantic segmentation assigns a class label to the pixels in an image, diving the image into different regions of semantic classes. If there are two people in the image, they would be classified into the same class of pixels.
      2. Image segmentation groups individual instances of objects in an image and gives them a class label. If there are two people in the image, they would be given the same class label but be part of different groups
      3. Object detection aims to provide a bounding box around detected objects and give it a label. This is less granular but in between their complexity.
   C. Describe at least two plausible neural network designs for the task of semantic segmentations.
      1. The first has a structure similar to a U. It will convolve the image down with convolutions and pooling to allow for a reduction in dimensions. Then at that level, it becomes easy for pixel spatial relations. After that is set, deconvolutions can be used to return to the overall image size, applying a pixel classification output layer to map pixels to classes. To help preserve information, skip connections between the downsampling and upsampling can be preserved.
      2. Another neural network design is the use of different filters to generate different feature maps for each pixel. Training weights to determine how

much to weight to given each feature map, these maps can be grouped into different classes using a simple convolution network. Then, reversing hte map, we can apply the different segments to the pixels, creating a semantic segmentation group.

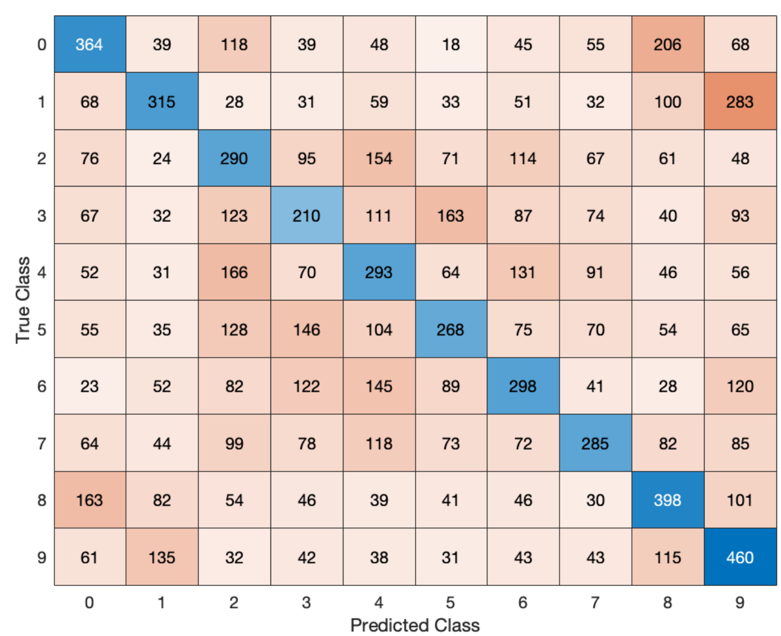
D. What are the applications of generative models in solving core computer vision tasks?
   1. Application of generative models in solving core computer vision I think best fit with the lack of information. Image synthesis, instead of having to stitch together textures, can not generate new images from scratch that are similar to the given dataset. Another application is for resolution enhancement. Generative models allow for enhancements of low resolution images to high resolution and reduce the need to just rely on filters.
E. Discuss the differences between convolution neural networks and deconvolution neural networks.
   1. CNNs are generally used in feature extraction where it uses convolutional layers to apply filters to the kernel. Then, it will have pooling layers that reduce and downsample its features to reduce its spatial dimensions. This is used in image classification, object detection, and image recognition.
   2. DNNs are generally used for upsampling and reconstructing data. They use transposed convolutional layers to increase spatial resolution of the input and upsample the feature maps. This is useful in image synthesis, image generation, and semantic segmentation.

II. Image Classification
   A. KNN Classifier:
      1. Accuracy: 32.65%



B. Adaboosting Classifier

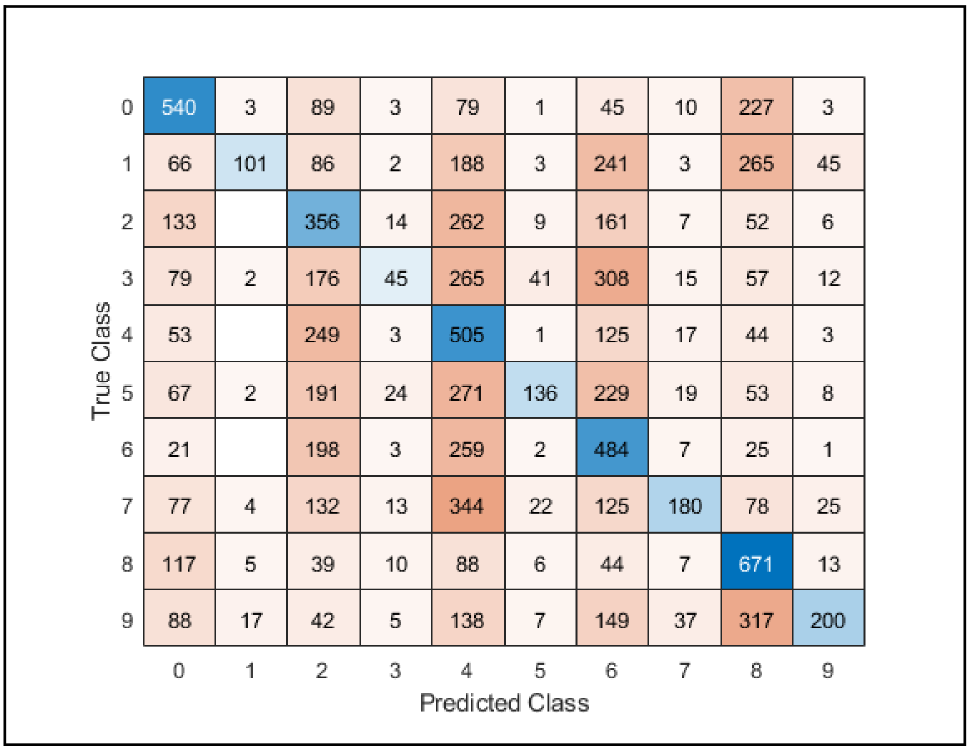
1. Individual accuracy between 55-% to 70%. Accuracy: 33.96%



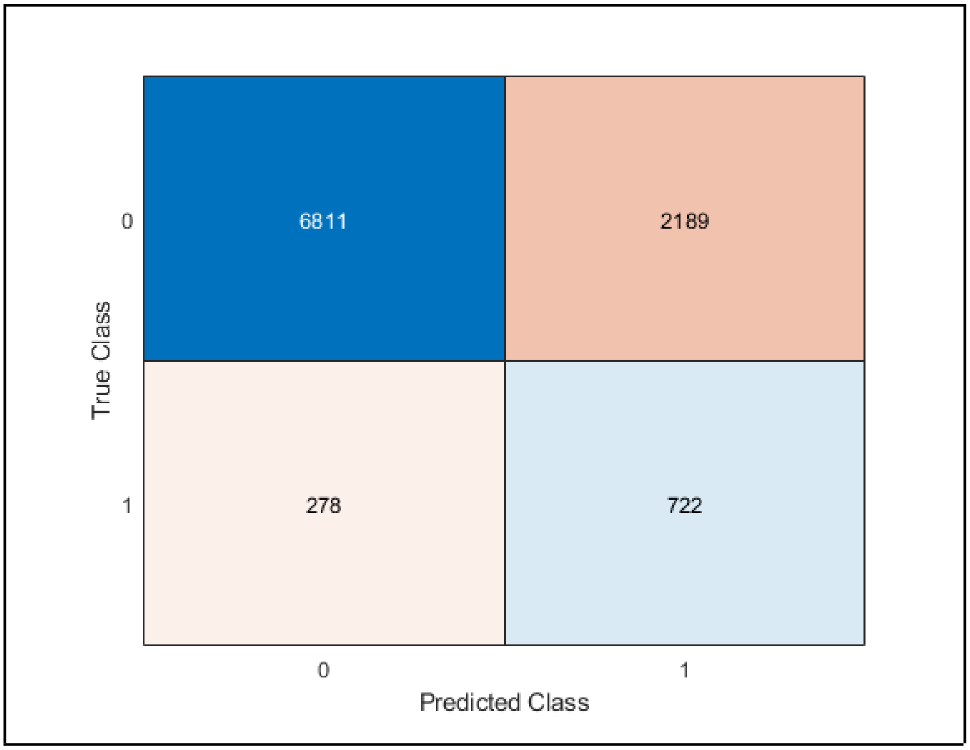
C. Comparison of KNN classifier and Adaboosting classifier

1. KNN and ADAboosting perform at relatively the same levels for identifying the different classes. While the Adaboosting Classifier training took significantly longer to train (KNN had no training time), the time running the Adaboosting Classifier is faster than the KNN classifier. The KNN classifier appears to do better for groups 0, 2, 4, 6, and 8. Why this is because it guesses those classes more and therefore hits it more. However, when judging by a false positive case standard, KNN does better than the other groups. This stems from the groups 0, 2, 4, 6, and 8 being more homogenous, thus leading to more classes looking like them compared to the others. As such, the KNN guesses those classes more. But, for the odd groups, when there is a match, KNN will not guess it wrong. Adaboosting classifier generally has a more spread among its guesses.

D. Cross-validation KNN Classifier
   1. K = 95



E. Cross-validation AdaBoosting Classifier
   1. 1000 weak classifiers for 75.33%



2. This was only tested on group 1 due to training multiple adaboosters would take too long of a time. I tested KFolds on the different amount of weak classifiers used to determine output accuracy from 10 to 1000 in intervals of 10. I found that 1000 worked better, which makes sense as from the slides, it details that adaboosting does not overfit and the more iterations you use, the more accurate it is. If I tried even more weak classifiers, the test accuracy would go up even more.