CS 5600 Final Project

Matthew Backlund
A02242891

Project Log

I went on a week long vacation during this project. My sister in law also had her baby earlier than expected, which took away a significant amount of time from this project as I helped babysit and spent time traveling. Because of this I couldn't frontload the project like I wanted to. Below is the planned schedule for this project:

Nov 18, 2024

- Finish scripts to convert data into YOLO format.
- Save all images and labels in train and validation folders.

Nov 22, 2024

- Get YOLO running on the dataset.
- Obtain a baseline performance reading.

Dec 2, 2024

- Complete hyperparameter tuning and training on the YOLO model.
- Save the trained model.

Dec 6, 2024

- Develop a user interface where users can upload pictures and view results.
- Implement code to overlay images on top of the face image.

Dec 9, 2024

- Finalize containerizing the project.
- Complete project testing.

Dec 10, 2024

- Write the project report.
- Submit the finalized project.

And here is how the project actually ended up:

Nov 18, 2024

- Finished scripts that converted data from csv format with dots for facial features into bounding boxes.

December 6, 2024

- **Git Setup**: Added a gitignore file to manage repository tracking more effectively.
- **Data Transformation**: Completed transformations to format data for the YOLO model.
- **Data Preparation**: Created a compressed and updated dataset of face images that was under the 100mb limit set by GitHub**.**

December 7, 2024

- **YOLO Integration**: Added code for YOLO model integration and tested the PyTorch setup.

December 11, 2024

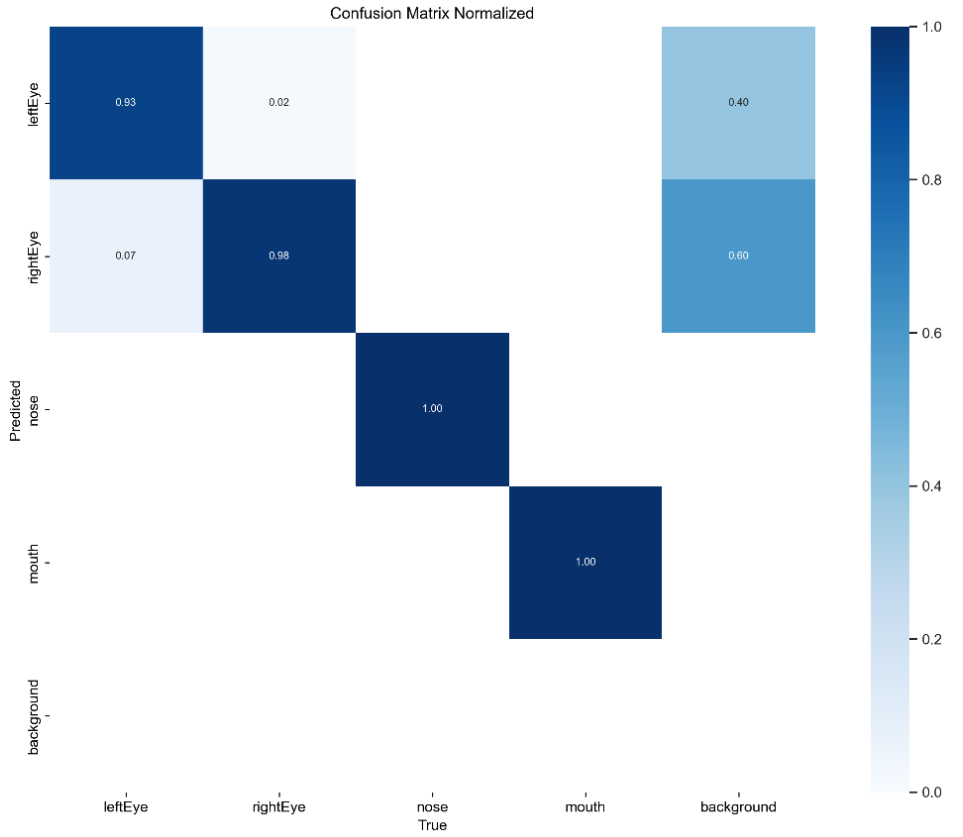
- **YOLO Model Training**: Completed training of the YOLO model.
- **Image Generation Work**: Began incorporating image generation capabilities.

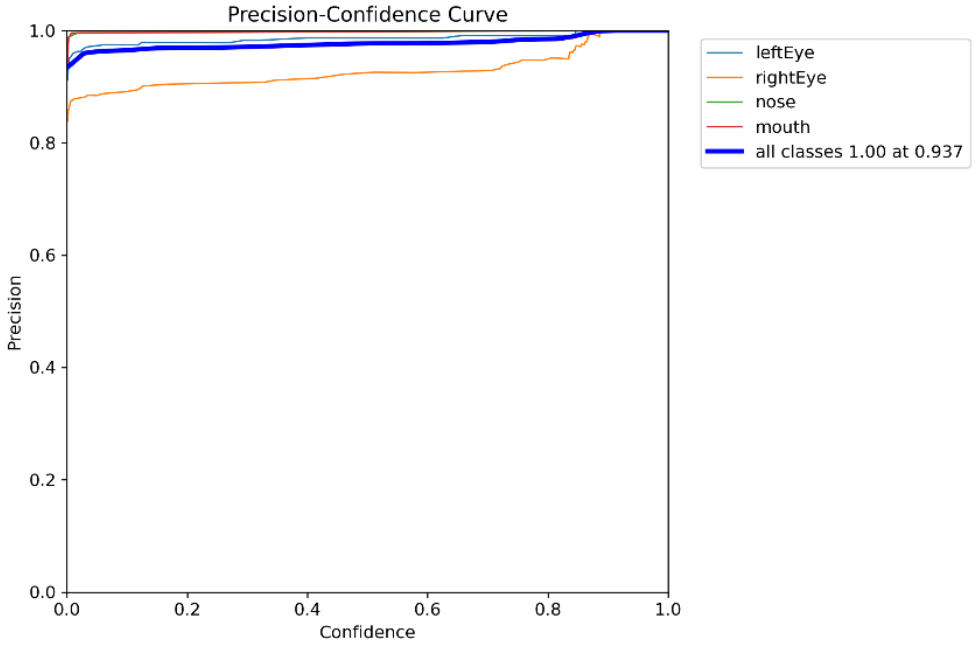
December 13, 2024

- **Flask API Development**: Implemented an API to handle image processing and return modified images.
- **Front-end and Back-end Containerized**: Created Docker files for both the front end and back end to streamline deployment.
- **Application Completed**: Integrated UI and API to ensure full functionality.
- **README Updated**: Included project instructions and finalized the documentation.
- **Final Report**: Wrote final report with information on the process of training the neural network.

Image Recognition scores

The above image shows the performance of the DNN on a sample of 16 validation images. I only trained the model for 100 epochs with the small YOLOv8 model. The training set consists of 980 images. Because there are so many images in the training set, the neural network is able to be very accurate without training for too many epochs. My tests have yielded very positive results, with the model being able to predict eyes with about 90% accuracy and other facial features nearly 100% of the time. The following items shows the performance of the model on the validation set.


Confusion Matrix Normalized
leftEye
rightEye
nose
mouth
background
Predicted
0.93
0.02
0.40
0.07
0.98
0.60
1.00
1.00
leftEye
rightEye
nose
mouth
background
True
1.0
0.8
0.6
0.4
0.2
0.0


Precision-Confidence Curve
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
leftEye
rightEye
nose
mouth
all classes 1.00 at 0.937