Handwriting Company Logos Recognition

Chihhao Tsai, Dr. Suren Jayasuriya
Arizona State University, Tempe, AZ, USA

Ira A. Fulton Schools of Engineering
Arizona State University

INTRODUCTION

- This project focuses on developing an image recognition system using vision classification techniques to recognize hand drawing image from a predefined dataset.
- A few different structures of deep learning networks would be built as a to compare existing models, such as AlexNet, ResNet50, Swin Transformer, etc.
- ***Logo-2K+:A Large-Scale Logo Dataset****:* This dataset includes over 160,000 images (256 × 256) and over 2300 different brand's logos.

METHODS

Create Graphical User Interface (GUI) → Create My Own Model → Model Training and Testing → Similarity Search

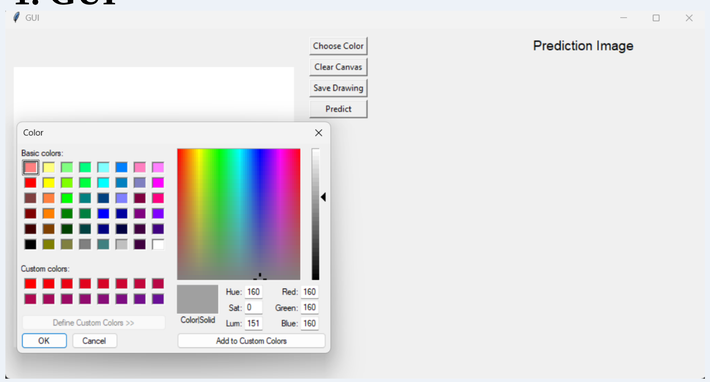
- **Create GUI:**
  This GUI allows users to draw or write intended shapes, text, etc.
- **Create my model**:
  First Model: I created a model (Autoencoder Image Classifier) that combines original CNN with autoencoder structure.
- **Model training and testing:**
  Find the optimal model for similarity search.
- **Similarity Search:**
  1. Get the best trained model
  2. Create a feature map: Extract all features (tensor) in the dataset with corresponding index.
  3. Get the feature from the input image.
  4. Implement mapping algorithm.

CONCLUSION

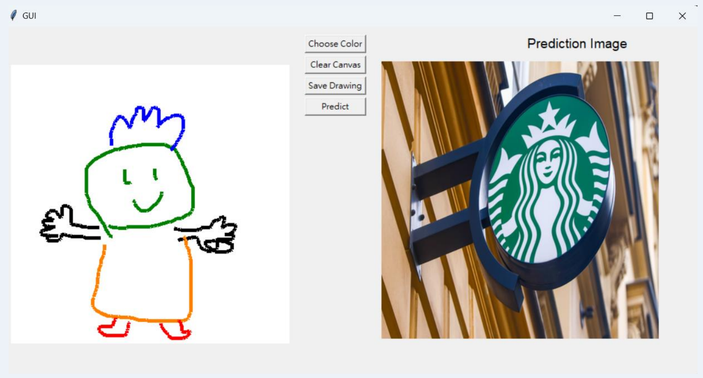
- Create a Vision Transformer with my own configuration (in progress)
- Train the models with more optimal parameters to get better model.

RESULTS

1. GUI



The GUI has a color palette for users choose colors.



It shows the image with highest similarity if 'predict' button is pressed.

2. Models' performance

- **First Training:**

All models were trained with *learning rate=0.000001, batch size=32, Adam as optimizer, Cross Entropy Loss function as loss function* for 5 epochs. However, the testing accuracy is bad.

- **Find the optimal parameters:**

I adjusted some parameters based on the structures and properties for each model and set the number of epochs to 100.

Model Name	Best Testing Accuracy for the 1^{st} training	Best accuracy with adjusted parameters
AlexNet	38.01%	60.5%
ResNet-50	1.1%	35.7%
My Model	34.69%	Still training
EfficientNet-B0	56.3%	Still training
Swin Transformer-Tiny	25.57%	Still training