Deep Learning With EMNIST

Identifying Handwritten Letters With Basic Computers

Cameron Wood, 10/14/2024

Github Repo

Agenda

- Problem Statement
- Machine Learning Approaches
- Results
- Summary

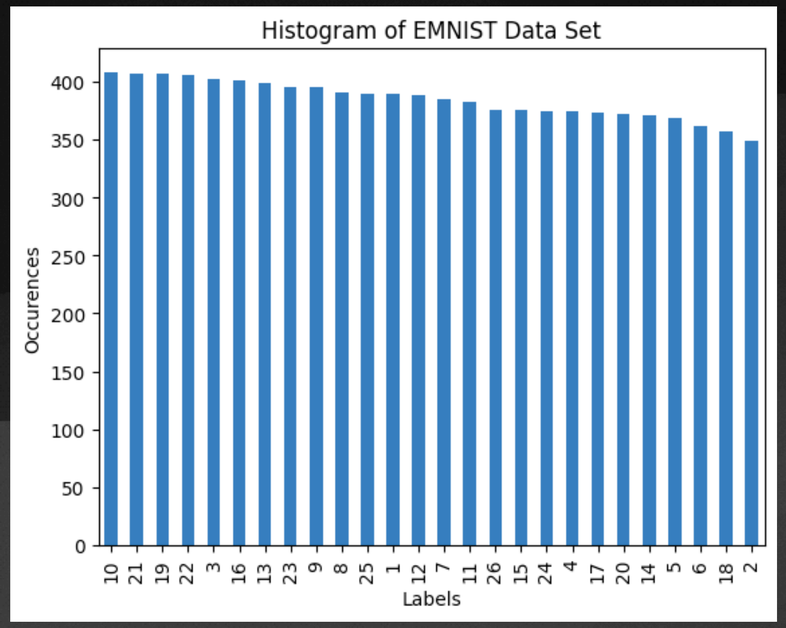
Problem Statement

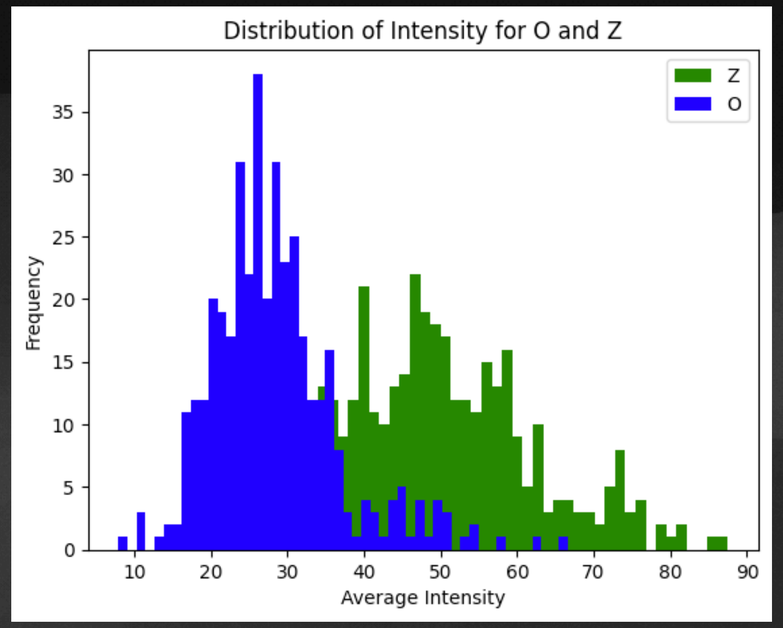
- Image Classification is becoming increasingly important in many industries, including: Medicine, Agriculture, Autonomous Vehicles, Natural Language Processing, and many more. One specific use case this project will be focusing on is the classification of handwritten letters.

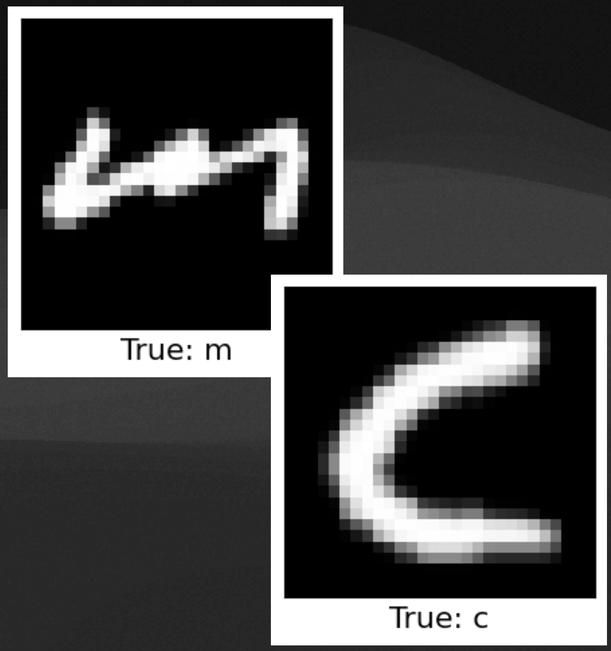
- Often times this type of classification uses computationally intensive Machine Learning models. This project aims to show that it’s possible to use basic Deep Learning and Supervised Learning techniques to train high performance classifiers, which can be ran on everyday computers.

EMNIST

An Overview







- This project uses the EMNIST Letters Dataset, which contains over 88,000 28x28 handwritten letters. This data was compiled from the NIST Special Database 19.
  - Trained on a down-selected, 40,000 image, sample.

Models Used

CNN, LSTM (RNN), and K Nearest Neighbor

- In this project, I chose to use Convolutional Neural Nets, Long Short-Term Memory (or Recurrent Neural Nets) as my Deep Learning Models, along with KNN as a Supervised Model.
- These are perfect for building simple Classifiers that enable quick iterative designs. All of these fare well in datasets where features are diverse and have unique features.
- As we'll see, the Convolutional Neural Net performs the best, with K-Means and LSTM not too far behind.

Convolutional Neural Network

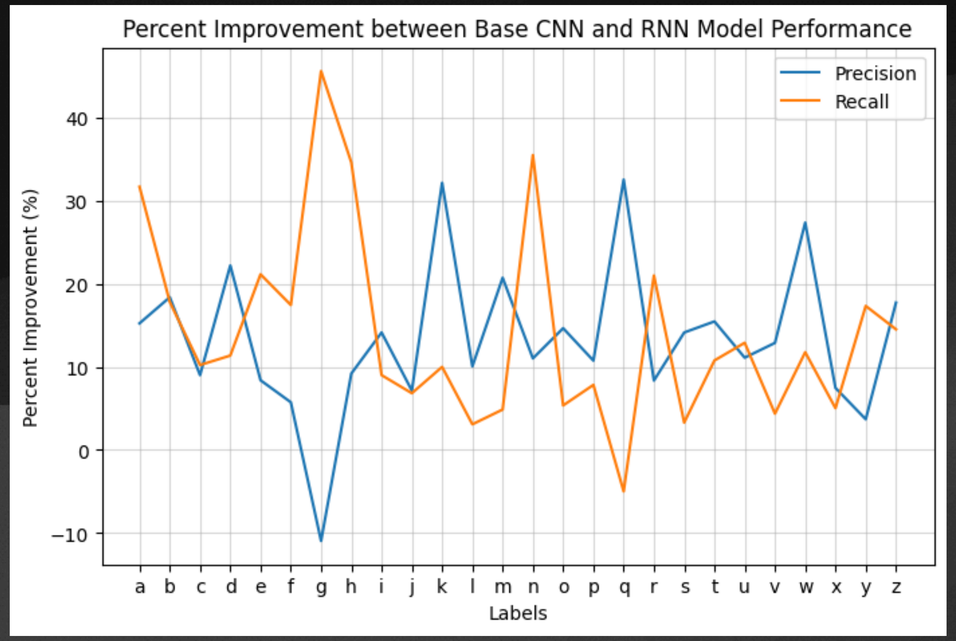
Layer (type)	Output Shape	Param #
conv2d_63 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_42 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_64 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_43 (MaxPooling2D)	(None, 5, 5, 64)	0
conv2d_65 (Conv2D)	(None, 3, 3, 64)	36,928
flatten_21 (Flatten)	(None, 576)	0
dense_64 (Dense)	(None, 128)	73,856
dense_65 (Dense)	(None, 27)	3,483

Long Short-Term Memory (RNN)

Layer (type)	Output Shape	Param #
lstm_19 (LSTM)	(None, 128)	80,384
dense_62 (Dense)	(None, 128)	16,512
dense_63 (Dense)	(None, 27)	3,483

Performance Summary

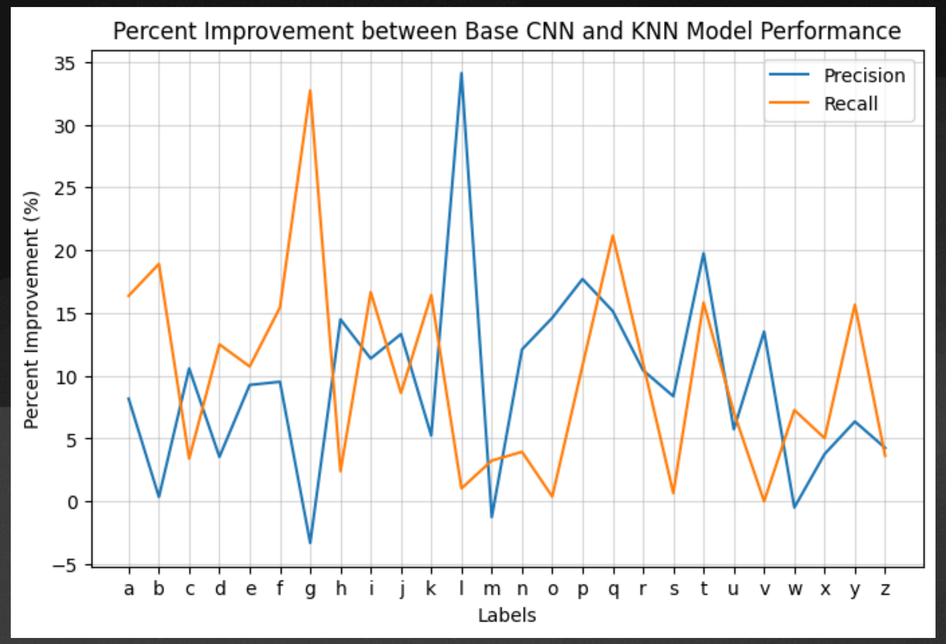
CNN and LSTM

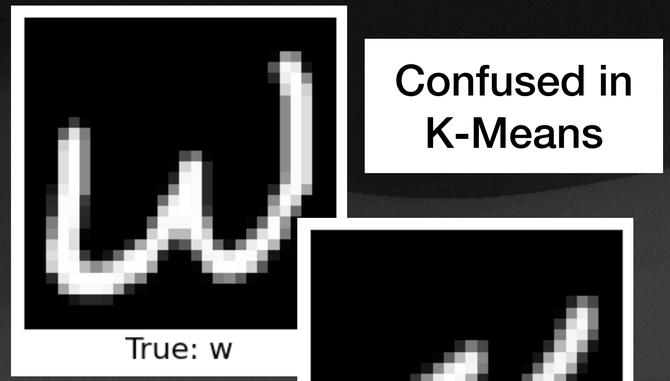


Model	Accuracy	Mean Recall	Mean Precision
CNN	92%	92%	92%
LSTM	79%	79%	80%

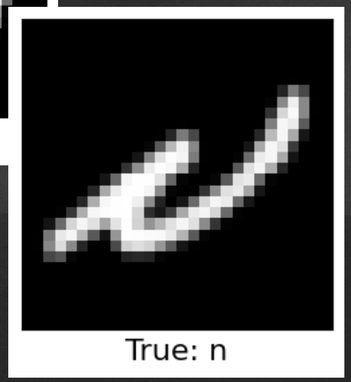
Performance Summary

CNN and K Nearest Neighbor





Confused in K-Means



Model	Accuracy	Mean Recall	Mean Precision
CNN	92%	92%	92%
K-Means	83%	83%	84%

Takeaways

- Able to achieve up to 92% classification accuracy using minimal tuning
- Models are easily developed and iterated upon
- Allows for models to be developed on everyday computers and used on a larger scale.

Thank You!