M8 Project

Kaggle: Digit Recognizer

Johnathan Wang, Nicholas Charles Vitellaro, Makenna Shae Owens, and Marc Mahanna
COSC523: Artificial Intelligence, November 24, 2024

Abstract - **This project focuses on using machine learning (ML) and computer vision (CV) approaches to implement models classifying images of handwritten numerical digits. The format of this project is that of a Kaggle competition** [1]**. Goals of this project are to 1) implement a baseline score and 2) try to improve on the baseline score.**

DESIGN CHOICES AND IMPLEMENTATION

The approach selected was to implement a baseline and an improved model intended to improve score beyond the baseline. Convolutional Neural Network (CNN) is the default implementation and is provided in the Kaggle competition tutorial [2]. CNN is also the improved model providing the best performance after much team experimentation. For each experiment and subsequent baseline and improved implementation, data was loaded, data was cleaned, features were explored and selected, models were trained, and test results were produced.

The baseline model followed the Kaggle tutorial. Minimal adaptation was required to get the baseline to execute. Data cleaning and preparation was minimal, nulls and normalization. The selected feature are pixel values. Train/test split is 80%/20%. Batch size is 32; Epoch quantity is 2. Resulting performance when submitted to Kaggle was 97.910%.

The improved model was the result of much team experimentation. The first set of team experimentation performed centered around using different layers in lieu of convolutional layers including input, flatten, and dense layers. Epochs was kept at 2 matching the baseline model from the tutorial. Resulting performance when submitted to Kaggle was 96.660%. The second set of team experimentation centered around incorporating convolutional layers into the first set of experimentation. Epochs was also increased to 20. Resulting performance when submitted to Kaggle was 98.775%. The third and best set of team experimentation centered around data augmentation in line with the tutorial. The team experimented with the proscribed image data generator which included generation of training data permutations via rotation, scaling size, width and height lateral shifts. Additionally, epochs were increased to 30 and batch size to 86. Resulting best performance when submitted to Kaggle was 99.292%.

CHALLENGES AND OBSTACLES

The project group members are familiar with machine learning concepts and have some prior experience through coursework and other projects with computer vision.

This project employed a very different type of data set of monochromatic 2-dimensional pixel grids. Employing traditional mechanisms for data cleaning and normalization do not result in the same performance improvements as seen in multi feature data sets. Different approaches are needed for imagery datasets. The team found that rotation, scaling size, width and height lateral shifts provided an improvement in performance where value normalization and horizontal/vertical mirroring did not.

The baseline tutorial Jupyter notebook started with very high performance out of the box. Keras convolutional layers are very capable out of the box and require a lot of experimentation to achieve better than baseline performance. Exploration of the model parameters did not reveal obvious combinations as the baseline had settled at the peak possible performance given baseline configuration. Going beyond this baseline performance required experimentation and interpretation of results. The team found that an increase in performance only comes through use of computer vision techniques supporting model generalization, in this case intentionally adding variations to the training dataset presenting alternative representations of the same data to the model. This approach allows the model to better learn what makes a digit a digit.

DISCUSSION AND FUTURE WORK

The team believes that combining computer vision and machine learning is an incredibly powerful approach affording understanding of remote sensed data in real world applications. Application of CV and models is different than other ML approaches requiring different techniques. Additionally, while the team did not find it true in the case of this project, most real-world CV applications generally require greater CPU (and GPU) resources and runtime versus other ML models.

The team attempted several approaches to application of CV and ML through experimentation and learned a few things that worked and many that did not improve overall performance beyond the already significant baseline. This tracks with the historical level of effort invested into the field of character recognition and handwriting recognition.

The team is interested in future CV and ML experimentation especially in areas where the training set is less controlled with a wider range of inputs such as color or multispectral imagery. Additionally, the team is aware of existing object detection approaches such as YOLO and similar models and is interested in applications beyond the base training set.

SUMMARY

This project provided the opportunity to compare the performance of different machine learning and natural language processing models. A consumable data set and a challenge issued in the form of a competition provided an enriching and rewarding experience for the project group.

REFERENCES

[1] "Digit Recognizer," [Online]. Available: https://www.kaggle.com/competitions/digit-recognizer/overview.

[2] "Introduction to CNN Keras - 0.997 (top 6%)," [Online]. Available: https://www.kaggle.com/code/yassineghouzam/introduction-to-cnn-keras-0-997-top-6.