Mohamed Elenany

A4 Writeup

Github Link for the code can be found below:

https://github.com/MohamedElenany/MAIS202-Kaggle-Competition

1. Implementation of the model:
   To implement my model, I started first by doing some preprocessing of the images that were inputted. I start by applying morphing and blur into the pictures to try and make the outline of the objects look clearer and decrease the effect of the black dots in the images. Then I applied dilation on the images to expand the shapes in the pictures. I do the same applications on both, the training set and the test set. Now, for the model, I have used a model implementation that was proposed in the following YouTube video:
   https://www.youtube.com/watch?v=GI7yim2NqGo
   The CNN is composed of a 3 Relu activated convolution layers followed by a step of max-pooling to be able to reduce the number of parameters by half. After that, we add a 25% dropout to our data to regularize the data to avoid overfitting. Then, we flatten the results to be able to convert it into a vector. After that, we have a dense layer to create a fully connected graph, followed by dropout again, and finally we end with another dense layer that uses the SoftMax activation function to output a vector of length 10, representing the ten classes of the data. Finally, we train and test the model, then we iterate each output vector to get the class with the highest probability and output it as the predicted result.

2. Results:
   In the beginning, the model I used was the original model that was proposed in the YouTube video. After training and testing, the training accuracy reached a maximum at an epoch of 12 with an accuracy of approx. 85%. Then I got the predictions of the test data from the model and received an accuracy of 86.6% from the model that was proposed. I experimented by adding a third convolution layer to the model, reaching the model that was explained in the previous section. I decided to add this layer to experiment more with how increasing the depth could affect the accuracy of the model, and I found out that the model became more complex and took a lot more time to train. In the end, the results from this model were an 89% accuracy in training data and a test set accuracy of 87.98% received from the Kaggle evaluation; This was the highest accuracy I was able to reach. Even though the model became more complex, I decided to keep my modified model as I recognized more stability of accuracy between the epochs during the training phase, in addition to producing a higher accuracy than that from the original model.

3. Challenges:
   The challenges I faced were more of facing challenges with understanding the models and also facing a lot of issues with errors given off by keras. It took a long time for me to understand a couple of layers like the dense and the flatten layers, but as soon as I understood why and when to use them, things became a lot easier after. Also, a lot of the issues I faced during implementation was not understanding what the errors that keras gave meant, and where they

really came from. The errors where very ambiguous and gave no clue about where exactly the errors may be. To solve those errors, I took a lot of time to search for similar errors on websites like stack overflow, and then understand why they occurred. Most of the errors occurred due to issues with the input dimension of the images and the dimensions of the layers, but after understanding how dimensions work and the transformations that each layer creates, I was able to easily tackle most issues thereafter.

4. Conclusion:
   All in all, this project offered a fruitful learning experience that helped me further understand and grasp the main idea behind CNNs and how they work. After the lecture about CNN, the ideas seemed to be adequately understood inside my head, however I had no clue how to apply them. This project helped me understand the model and architecture of CNNs, in addition to helping me grasp when and where to use certain layer; in other words, when is it time to add a certain layer and what to do if I want to reach a certain output dimension. I was able to understand and experiment with how overfitting and model performance can differentiate based on the data complexity, the depth/complexity of the model, and the number of epochs the model runs on. I was able to understand and interpret the result outputted from different activation functions like Relu and SoftMax. Finally, I was able to gain a better understanding about image pre-processing and transforming images to get better results.

5. Statement of Contribution:
   Not applicable…Working Individually.