Google AI Open Image Classification
Individual Report

Richa Joshi

Introduction:

The dataset for this project is taken from Google Open Image Dataset V4 website provided by google. Our training set contains 28,000 images and test is 7,000 images including person, food stuff, animal tree and many more. The average size for these images is 1024x760. The type of objects in these datasets vary in shape, size and color across the same class. Since the raw dataset is very large, only a portion of it is used for our analysis. We have total 10 classes. In this group project we have total three members, and each member have equally shared the work. The project was tough so met several times and done project together.

Our first step in achieving this goal was to meet as a group. We threw out ideas on what we were thinking and then discuss our project data. Each member has some datasets and we decided to take one which is Open AI image dataset. We selected that dataset because we were interested in doing object detection and image segmentation. We chose to do image classification for same dataset and we decided to use CNN. We used the convolutional neural network for our project because the main idea of the project is to be able to classify the images. We think that it makes sense to use Deep convolutional neural networks. CNN has been used to great effect in applications such as image classification, object detection and other applications.

Framework:

For our project, we have used tensorflow as the framework. Our main motivation for this project is to learn how to set up, train and build CNN's in tensorflow and other frameworks. We also used keras as our framework to compare our results. We have used RELU and softmax activation functions with different frameworks.

Data Problem:

We have following issues with the data when we started the project:

i. We have many images that has no labels.

ii. The format of the image IDs does not match with labels image ID

iii. The dataset has both gray images and color images

iv. Images are not in the same shape

Data Preprocessing:

Since there was no way for us to use the whole 9 million images dataset and 516 classes to pursue this kind of network, the results are inconclusive. The labels were brought down to 10 classes were labeled manually by us. we are only taking 35,000 images as our input data. Also Since our Labels are in categorical form and we have 10 labels so we changed our labels to binary form by using one hot encoder before we split the data into train and test in keras utils function.

Evaluation:

When evaluating a model and tuning the parameters, I realized that accuracy from our small dataset is low and it might be because of the lack of data. It could also be because of an unfit model for this data. Probably, the biggest challenge I faced in this project is the size of the data. It severely limited what we could actually produce and massively slowed down the training and testing. This approach would have been interesting if we have more data for our 10 numbers of classes. Instead, we have each model with less training data, it would also be very interesting to see the comparison of one model trained on the full dataset and another model on the smaller data. Distribution of classes was very unequal in the dataset, some classes had 10 products in one category and some of them had thousands, which is not very good, so for better training on late stages we might have to use some data augmentation. Below are the results we got from our models.

Frameworks		Accuracy	Layers	Kernel size	Epochs
Tensorflow	No Batch Normalization	35%	4	5x5	200
Keras	No Batch Normalization	40%	4	5x5	20
Keras	With Batch Normalization	45%	6	5x5	20
Keras	K-fold cross validation	50%	6	5x5	20

The model is built using CNN with batch normalization and without batch normalization. The highest accuracy is performed by the model which is implemented using batch normalization. The accuracy is 50%. When we run the model without batch normalization our accuracy was very low then we tried including batch normalization in our layers then the accuracy goes up. The optimizer used in the project is Adam and transfer functions are RELU and softmax.

Conclusion:

The accuracy from our small dataset is low and we think it could be because of the lack of data. It could also be because of an unfit model for this data. Probably, the biggest challenge we faced in this project is the size of the data. It severely limited what we could actually produce and massively slowed down the training and testing. This approach would have been interesting if we have more data for our 10 numbers of classes. Instead, we have each model with less training data, it would also be very interesting to see the comparison of one model trained on the full dataset and another model on the smaller data. Distribution of classes was very unequal in the dataset, some classes had 10 products in one category and some of them had thousands, which is not very good, so for better training on late stages we might have to use some data augmentation.

Overall, it was a very interesting challenge with biggest challenge being the amount of data and time needed to make something with this data. The process of learning through project was helpful to me. The group discussion and sharing the ideas was to each other helped me to understand the concepts. I get the idea of how the CNN model works. Overall, I am happy with this project as it allowed me to experience and improve upon my team working skills, but also improve and challenge myself in relation to my work. Our group worked very well together. Each person expressed their ideas and none of them were met with negativity. There was a constant open line of communication via email and meetings which we all utilized.