**CIND860 Advanced Data Analytics Project:**

**Fashion MNIST - Evaluating the Efficacy of CNN against Traditional ML Models**

**Initial Results & Code**

**By: Hasib Chowdhury (500439135)**

**Supervisor: Dr. Tamer Abdou**

**GitHub Link:** [**https://github.com/Hasib147/CIND860-Capstone-Project**](https://github.com/Hasib147/CIND860-Capstone-Project)

**Date of Submission: November 6, 2023**

**Exploratory Data Analyses:**

An embedded technique known as the Random Forest Importance was used to take the top 20 pixels based on the ‘Importance’ value of the Fashion-MNIST train dataset (60,000 rows and 785 columns with column 1 being the dependent variable). The reason top 20 was used was because it is a decent value and the system would not crash (due to limited capabilities) or take a long time to generate a panda profiling report. The following 20 pixels on the train dataset had the highest importance based on the correlation of the dependent variable (label) and other independent variables (the remaining number of pixels). They do contain a high number of zeros, which were taken into account but of a lesser amount than the other pixels.

Rank Pixel # Importance

1 pixel547 0.009806

2 pixel603 0.009392

3 pixel491 0.008809

4 pixel575 0.007279

5 pixel263 0.006986

6 pixel407 0.006045

7 pixel435 0.005642

8 pixel687 0.005324

9 pixel519 0.005292

10 pixel631 0.005287

11 pixel401 0.005284

12 pixel180 0.005119

13 pixel234 0.005010

14 pixel236 0.004944

15 pixel68 0.004934

16 pixel611 0.004891

17 pixel96 0.004734

18 pixel555 0.004687

19 pixel594 0.004502

20 pixel427 0.004482

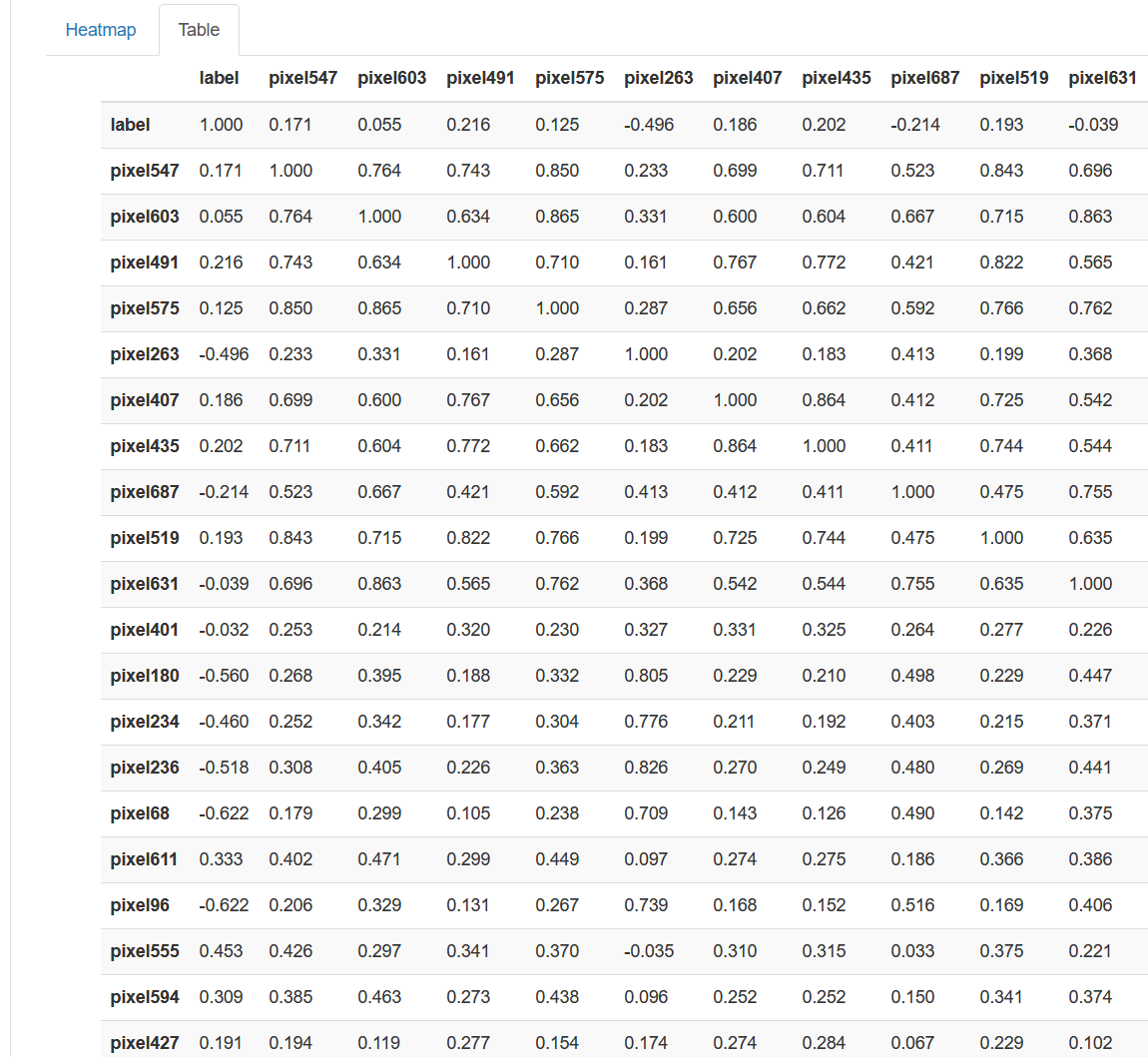
The number of trees used to generate this output was 100 trees based on the Random Forest Importance (RFI). I chose this value is because it fits in nicely with the 60,000 total rows of the train dataset. Also in practice, the number of trees can be tuned using techniques like cross-validation or grid search to find the optimal value for specific problems within deep learning. According to ChatGPT, “Using more trees can lead to a more robust model but might increase computational complexity but using fewer trees might lead to a less complex model but could lead to the model overfitting. The optimal number of trees can depend on factors like the dataset size, complexity, and available computational resources” [1]. I will experiment with the different number of trees and see which value is the most sufficient to use for this specific dataset of the top 20 pixels. In addition, this will be compared to other ML models such as SVM , XGBoost and the traditional simple-CNN model (with the basic parameters done on a simple scaled test set).

**Data Preparation:**

A screenshot of a computer

Description automatically generated

The above graph shows the heatmap and how each pixel correlates to one another and the dependent variable (label). This specific heatmap was generated from the pandaprofiling report using the combined dataset of both training and testing together (70,000 total rows), consisting of the top 20 pixels (ranked according to Importance based on RFI 100 trees). The pixels near the centre of the diagram, are light to dark blue indicating strong correlation between the pixels, whereas the other pixels are somewhat light and sort of faded, indicating a weak correlation.

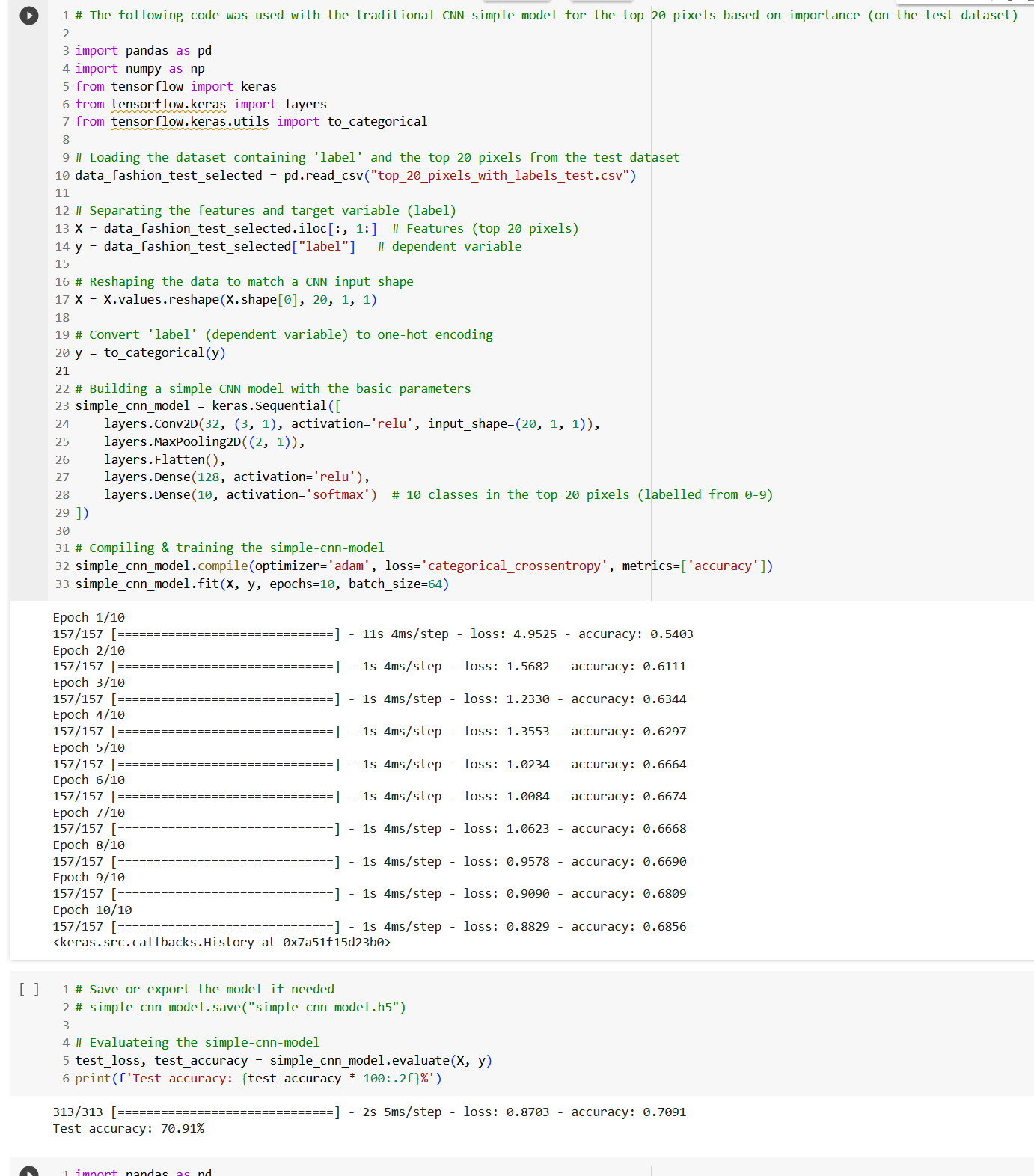


The above table (taken from Pandaprofiling) shows that most of the independent variables (pixels) have a positive correlation with one another and some have negative correlation. They can vary between weak correlation (0.100 to 0.399) to medium (0.400-0.699) to high correlation (0.700-0.999). Each variable has a perfect correlation (1.00) with one another in the horizontal and vertical column.

**Modelling:**

To answer one of the research questions from the modified abstract of this project. I compared the simple-CNN model with the basic parameters and compared it with different machine learning models such as Random Forest, SVM and XGBoost. And to compare their accuracies based off the exact same dataset, top 20 pixels from the test dataset based on RFI 100 trees (with the label column and 10,000 rows).

The following is from the simple-CNN-model:



It gave 70.91% as the test accuracy, which is much lower than expected mainly cause of the small sample size (10,00 rows). In addition, the research paper that I am replicating (on section 5.3 cnn-simple, page 992) indicates that “All those models were modeled based on Keras Sequential model. Convolutional and dense layers used Rectified Linear Unit (ReLU) activation functions, except by the last dense layer on each model (output layer), were Softmax was used. The optimizer used was Adadelta, Batch size was 128 and we trained the models for 12 epochs” [2]. I did this differently from what they did. The reason I chose ‘Adam’ was because I had used it in the past for this exact same dataset in the CIND850 course and in general, it is a popular and effective optimizer that works well generally for many deep learning tasks such as image classification (which is used in this case). The choice of optimizer is problem-specific and I used that but other optimizers like what they used (Adadelta) or other optimizers such as ‘SGD’ or ‘RMSprop’ may also work. As for batch size and the number of epochs, as mentioned earlier I am using only 10,000 rows to test the data so I used lower values than the paper did. I used 10 for number of epochs is cause I’m working with a limited number of rows and columns and I didn’t want it to set it too low, which could cause underfitting or setting it too high which can lead to overfitting. I think 10 is a decent value for number of epochs in this particular situation, however if it was over 100,000 rows I may have done 100 epochs to speed up the process. For batch size, 64 is a decent value and I think it will work well for this particular dataset, reason being too high of a batch size can speed up the training but use too much computer memory (which I am limited to) and too low of a batch size such as 16 or 32 can be decent as well since a smaller batch size allows for more frequent weight updates and can help the model converge faster, but it may also require more epochs. That is why I chose 64 as the batch size value.

One other thing the research paper did differently when they trained the CNN-simple model is they had over 110,000 trainable parameters which would cause my system to crash so I only did 10,000 trainable parameters (for the test set) and 60,000 trainable parameters (for the train set) based on the ‘Importance’ value for the top 20 pixels (using RFI 100 trees). And the top 20 pixels in the test set (pixel 547, 603, 49, …, 427) were replicated the same way as the top 20 pixels on train set.

The following is from the Random Forest method:

A screenshot of a computer program

Description automatically generated

I ran 3 different values for the ‘n\_estimators’ to try out 3 different values for the trees and see which gives the most significant results. They were very close with one another (less than 2 percent difference), 10 trees gave an accuracy of 71.15%, 50 trees gave an accuracy of 72.20%, whereas 100 trees gave a 72.40% test accuracy. These values can differ each time due to the specific trees being chosen to be evaluated and it is very high due to only the top 20 pixels being chosen from test set (784 total pixels in total). If 100-200 pixels were used instead of 20, the test accuracy may have been lower but can’t be used due to the system crashing.

The following code is from the SVM & XGBoost model respectively:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Out of the 4 models compared, the test accuracy were very close to one another in the high 60’s to low 70’s range. The XGBoost and the Random forest with 100 trees were the highest with 73.40% and 72.40% respectively.

References

[1] OpenAI. (2023). ChatGPT (GPT-3.5 model) [Large language model]. <https://chat.openai.com/chat>

[2] LEITHARDT, V. (2021). Classifying garments from fashion-MNIST dataset through CNNs. *Advances in Science, Technology and Engineering Systems Journal*, *6*(1), 989-994.