Clothing Detection with Machine Learning

Hunter Nelson

Computer Science

Hood College

Fredrick, MD, USA

Hn5@hood.edu

Abstract -

Machine learning allows the classification of many different things better than a human could do. In this project I worked through the thought process and challenges of trying to deploy a classification program made for clothing detection. I use machine learning, keras, and CNN to work through these problems and create the program.

Concepts -

- Machine Learning
- CNN networks
- Keras Classification

1 Introduction

Classification using machine learning isn't something new, however it has recently become the new focus of computer science. Machine learning classification can provide details and small features of something scientists are not able to see with their own eyes. In this project I used machine learning to classify common clothing objects such as shirts, hoodies, sneakers, bags and many more. This software can be used to help visually impaired individuals if worked on and furthered more.

2 Dataset

Fashion Mnist is a dataset created by Zalando, made to replace the common mnist dataset. This dataset is made up of 60,000 training images and 10,000 test images. All images have been put into greyscale and in 28 x 28 format. This dataset is one of Zalando's article images. It was originally made to be a replacement for the normal mnist dataset. The main goal of the mnist dataset is to make a dataset that's easy to work with so in the end you can focus on your models and training instead of focusing on normalizing the data. The dataset has 10 categories: T-

shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot. Here are some example images from the dataset.

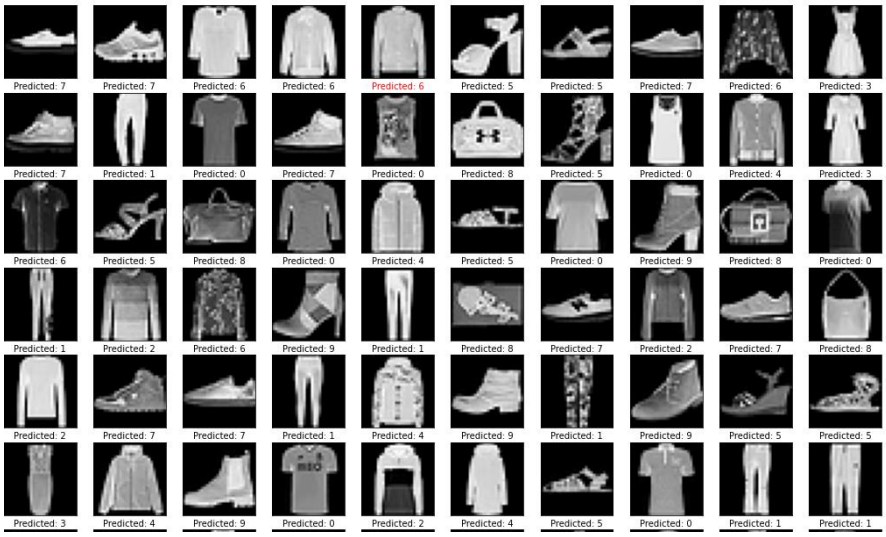

Figure 1: Fashion Mnist

2.1 Challenges in the dataset

As I was attempting to complete this program, I ran into some problems whether they were big or small. The most common problem I ran into was the dataset having some images that looked very close to other types of things. For example, in the figure below the software thought the sweater was a t-shirt causing it to give a bad prediction, this problem can be worked out by having clearer pictures or training the model more and more. This also brings up another issue though, what if the pictures are miss labeled for example these my dataset contains over 70,000 images that are prelabeled for training, but who labels all those images and does anyone check these labels?

Figure 2: Wrong Prediction

3 Model

For my model I decided to use keras layers because of many different reasons. The main reasons I chose keras layers were because they are user friendly and fast to deploy/learn. Keras layers are very user friendly, they allow the programmer to simply declare their model, input, reshape, and other parameters then deploy them. During the project I tried other ways of creating my model and keras seemed to be the easiest to get into and deploy within the time frame I was working

with, this does not mean they are the best, but they worked for my situation. I was hoping to continue my work by trying different other models and finding which model provided the best accuracy.

3.1 Training

In my model I used 5 epochs with 1875 images in each one. This provided the training model with around 9,400 images to train with, this allowed the program to get a training accuracy of around 93%. I could have allowed my program to run more epochs, however I felt that an accuracy of 93% was very good, I later realized that adding the extra epochs to boost the accuracy would have been more than worth it in this work.

```
loss: 0.3987 - accuracy: 0.8595

loss: 0.2809 - accuracy: 0.8999

loss: 0.2369 - accuracy: 0.9142

loss: 0.2103 - accuracy: 0.9231

loss: 0.1905 - accuracy: 0.9307
```

Figure 3: Training

4 Results

Like I stated above I ended up with an accuracy of 93%, this is great for the implementation of my program. The program was able to achieve this with 5 epochs, comparing the training of the model to others, you can see that having 5 epochs is a small amount. This provides me with a large amount of room to work with, I could change the training with more epochs or more images in the epochs and continue to make this model even better.

```
Epoch 1/5
1875/1875 [==============================] - 148s 79ms/step - loss: 0.3987 - accuracy: 0.8595
Epoch 2/5
1875/1875 [==============================] - 147s 78ms/step - loss: 0.2809 - accuracy: 0.8999
Epoch 3/5
1875/1875 [==============================] - 147s 78ms/step - loss: 0.2369 - accuracy: 0.9142
Epoch 4/5
1875/1875 [==============================] - 149s 79ms/step - loss: 0.2103 - accuracy: 0.9231
Epoch 5/5
1875/1875 [==============================] - 146s 78ms/step - loss: 0.1905 - accuracy: 0.9307
```

Figure 4: Epochs

The program was also able to pick up on all the categories and output the image selected with the prediction, I added an option to make the color of the prediction red if it got that prediction wrong, this didn't help or hurt my results, but it provided me with an easy way to look back at the results and read them.

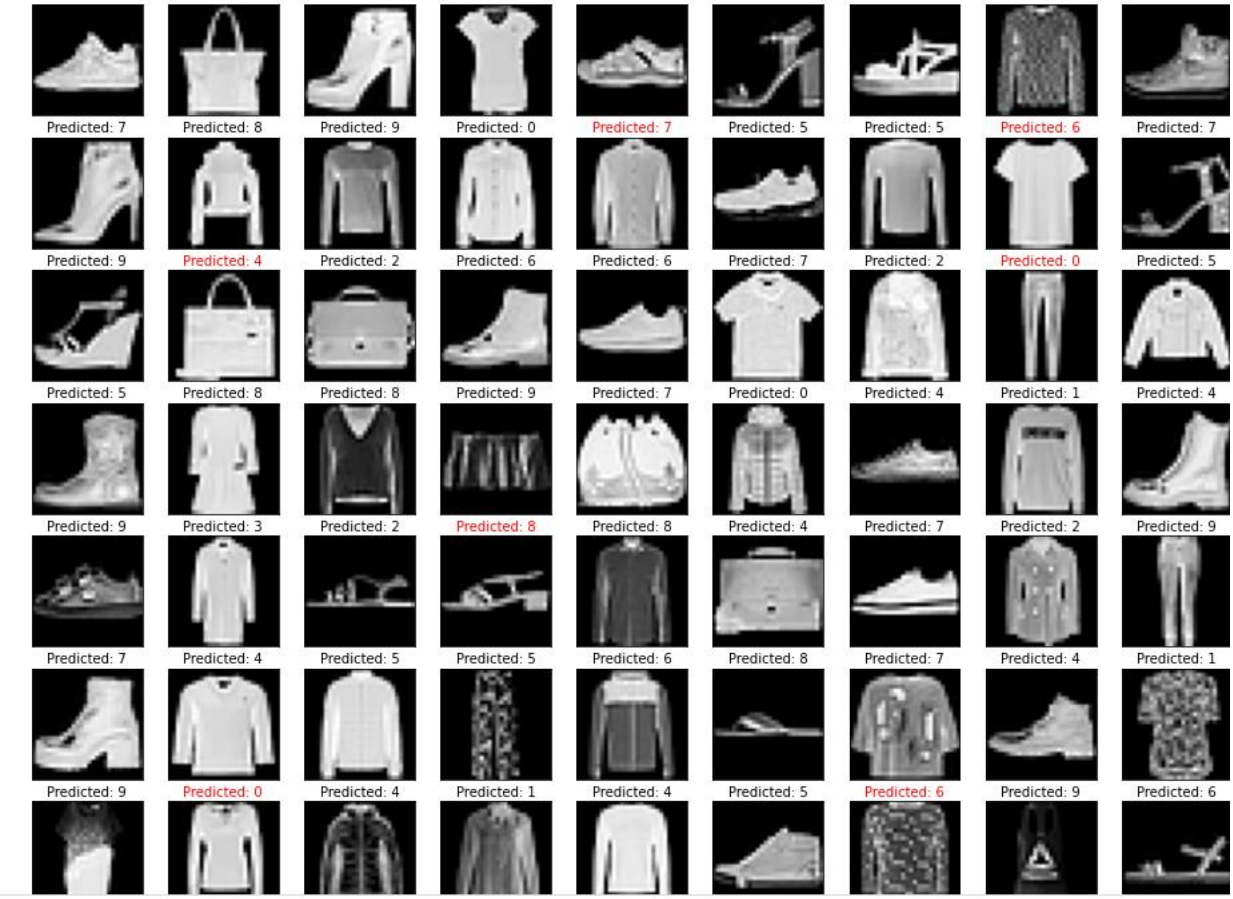

Figure 5: Prediction

5 Conclusion

Considering all aspects of my program, I can conclude that fashion classification using keras provides us a friendly and effective way of giving the visual impaired a way to pick out clothes. This can also provide future computer scientists with an entry into machine learning and keras. Machine learning seems very hard to many entry level computer scientists who have never had any experience with it and tends to scare them off but with this being an entry level project they may be able to find a passion in it, not to mention that this whole project can be very useful to visual impaired individuals.

References

[1] Fashion Mnist - https://github.com/zalandoresearch/fashion-mnist

[2] Basic Classification - https://www.tensorflow.org/tutorials/keras/classification