Clothing detection with machine learning

Hunter Nelson

My Goals

3 main goals

Easy model

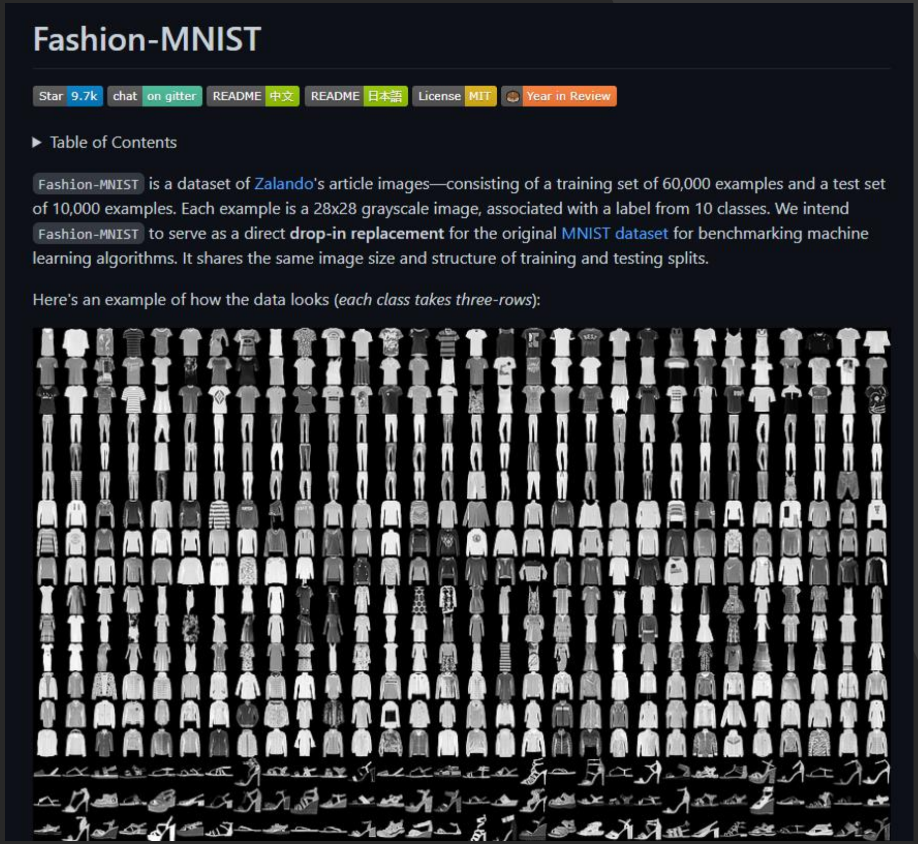
Fast training time

High accuracy

Dataset

Fashion-MNIST

- 10 types of categories
- 60,000 examples in the training set
- 10,000 examples in the test set



Fashion-MNIST

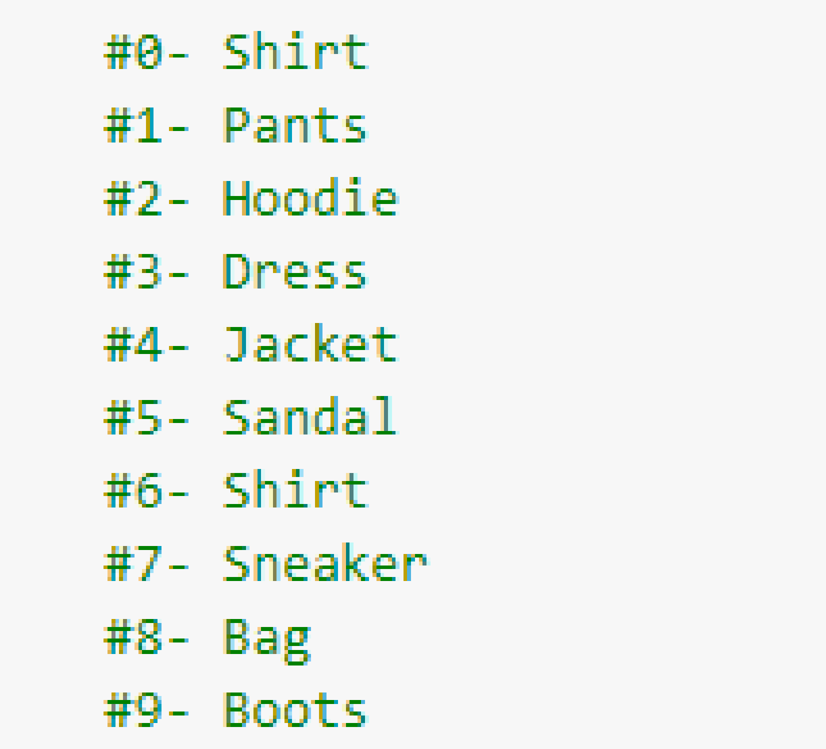
`Fashion-MNIST` is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. We intend `Fashion-MNIST` to serve as a direct **drop-in replacement** for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.

Here's an example of how the data looks (*each class takes three-rows*):

Categories

Categories all come with thousands of images

Provides me with many different examples for each option

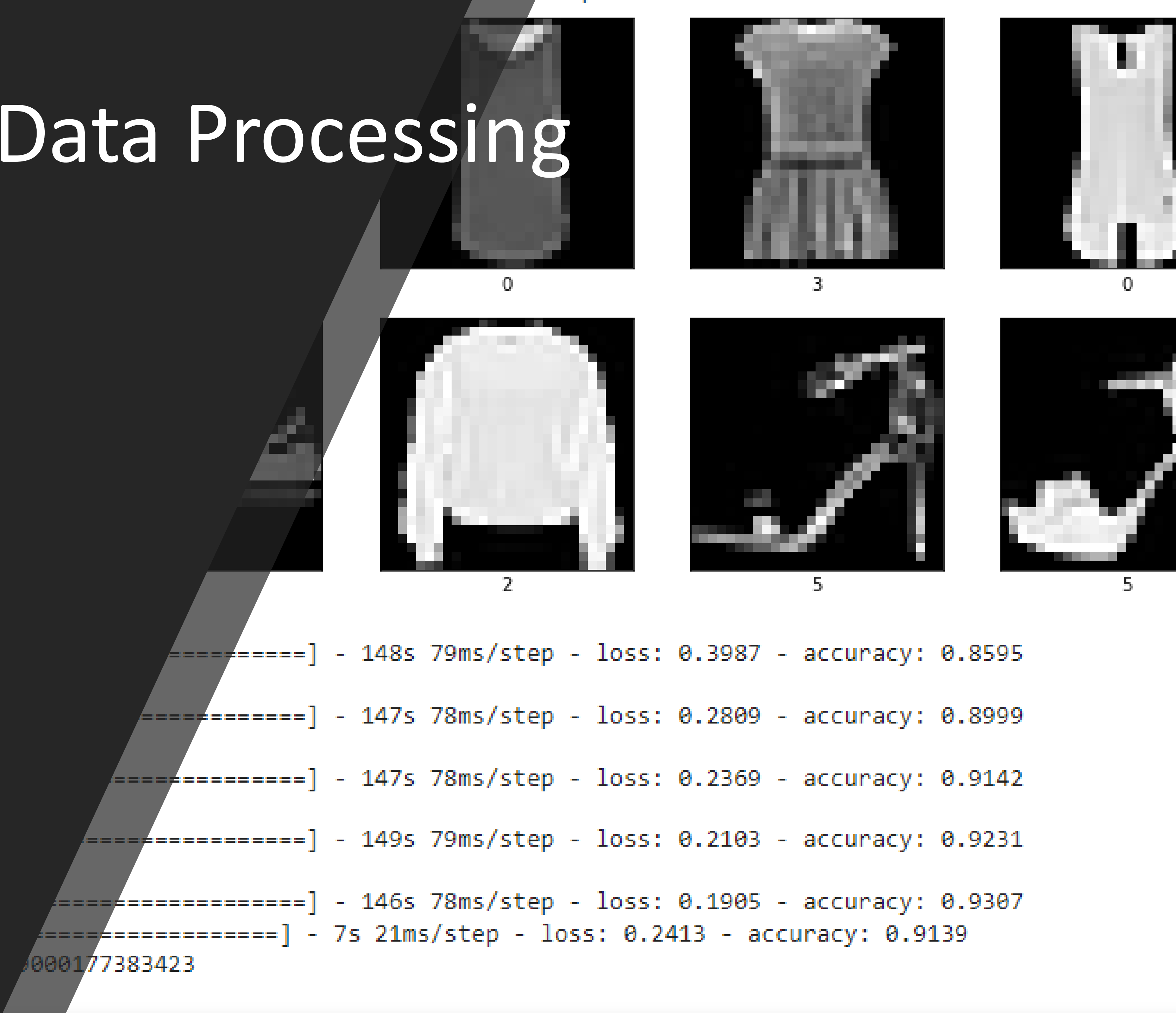


```
#0- Shirt
#1- Pants
#2- Hoodie
#3- Dress
#4- Jacket
#5- Sandal
#6- Shirt
#7- Sneaker
#8- Bag
#9- Boots
```

Data Processing

Dataset already normalized

- Black and white images
- 28 x 28 greyscale image

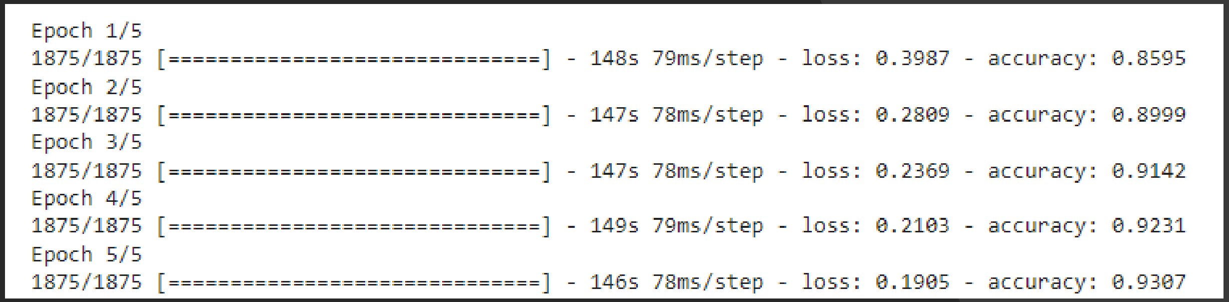


```
==========] - 148s 79ms/step - loss: 0.3987 - accuracy: 0.8595
============] - 147s 78ms/step - loss: 0.2809 - accuracy: 0.8999
===============] - 147s 78ms/step - loss: 0.2369 - accuracy: 0.9142
==================] - 149s 79ms/step - loss: 0.2103 - accuracy: 0.9231
====================] - 146s 78ms/step - loss: 0.1905 - accuracy: 0.9307
==================] - 7s 21ms/step - loss: 0.2413 - accuracy: 0.9139
0000177383423
```

Training

When choosing how to train my program I wanted something fast with an accuracy over 90%, I found that by doing 5 passes of 1875 images.

You can easily increase the images or passes to boost the accuracy, but you will also increase the time it takes to train

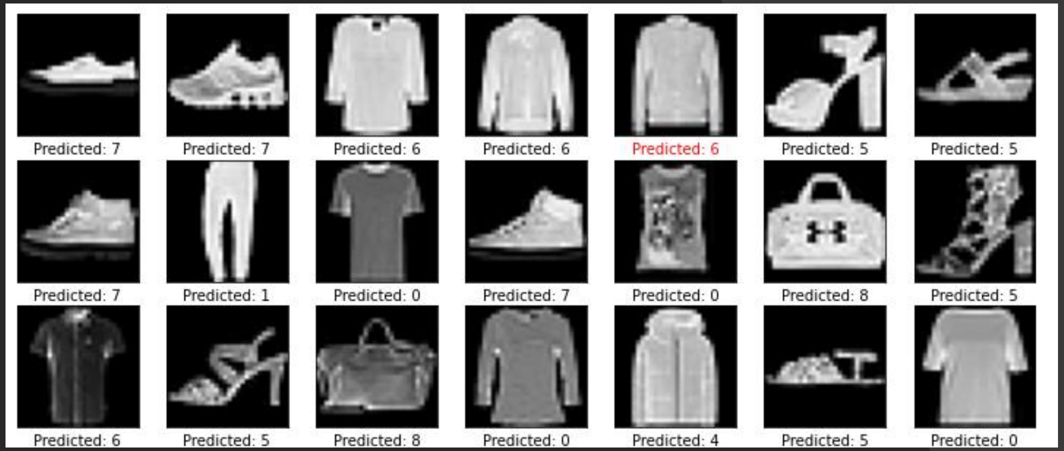


```
Epoch 1/5
1875/1875 [==============================] - 148s 79ms/step - loss: 0.3987 - accuracy: 0.8595
Epoch 2/5
1875/1875 [==============================] - 147s 78ms/step - loss: 0.2809 - accuracy: 0.8999
Epoch 3/5
1875/1875 [==============================] - 147s 78ms/step - loss: 0.2369 - accuracy: 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
```

Display

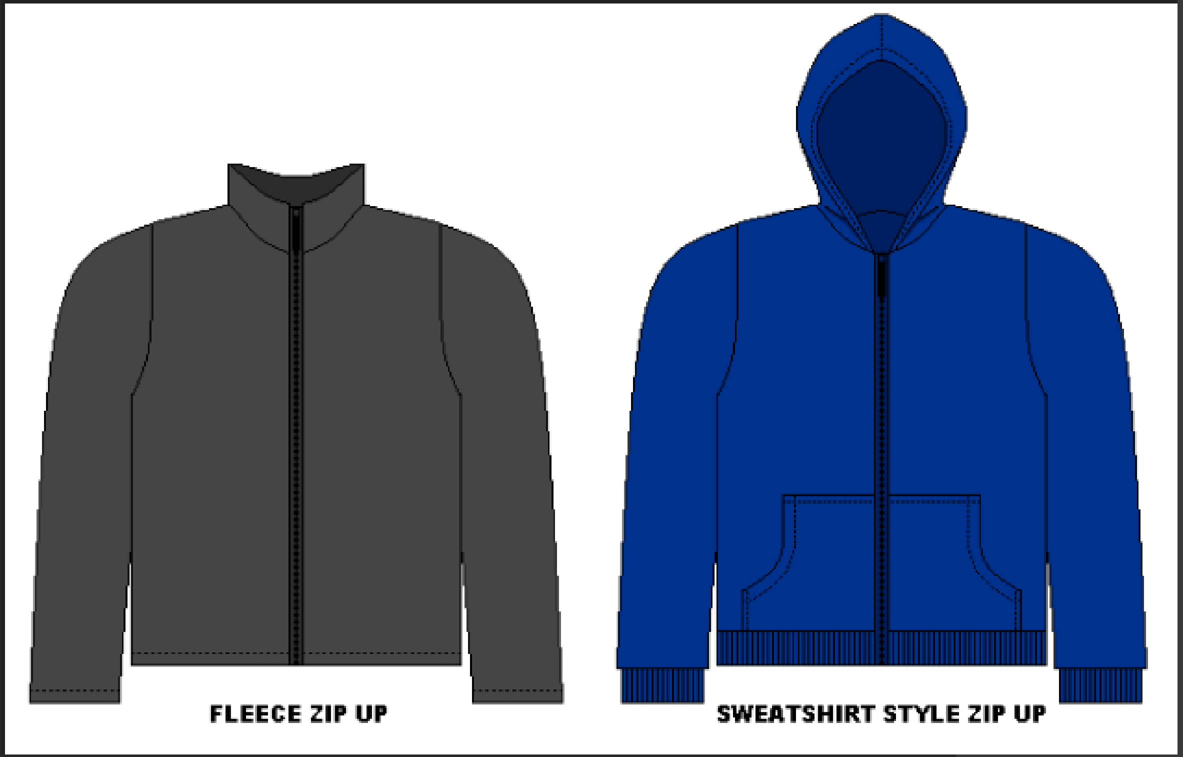
I was able to output the test images and their predictions, the red names are the ones the code predicted wrong

In this example the code thought the sweater was as shirt



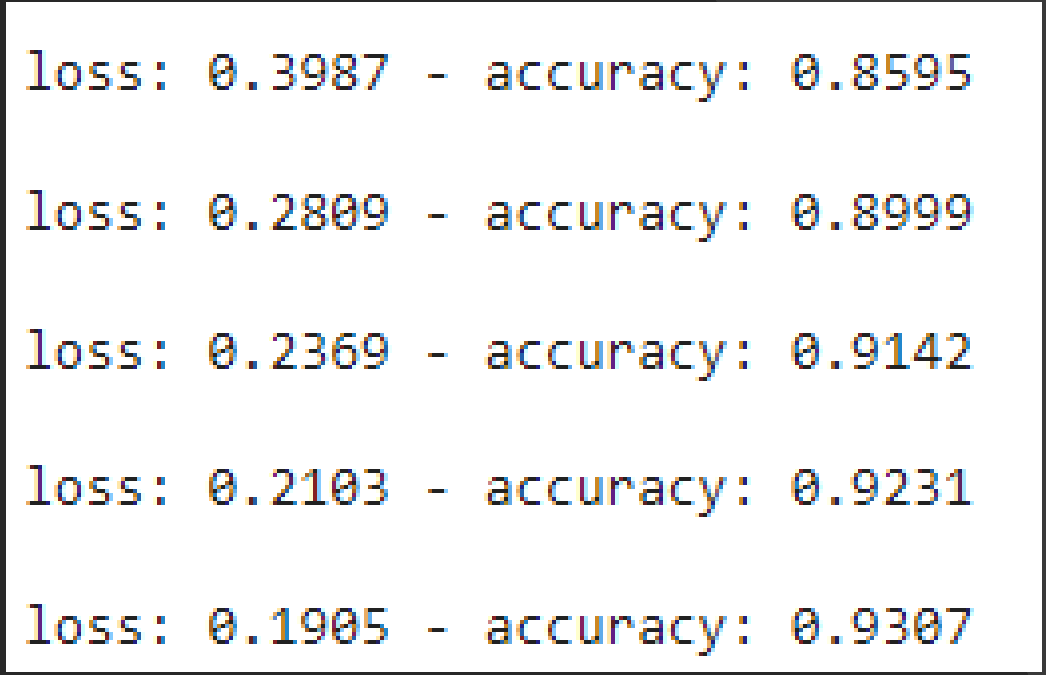
Challenges

- Clothing that looks the same
  - Shirts vs hoodies
  - Boots vs sneakers



Results

I was able to get my project up to 93% accuracy with only 5 passes. Adding more passes we can easily get the program up closer to 99%.

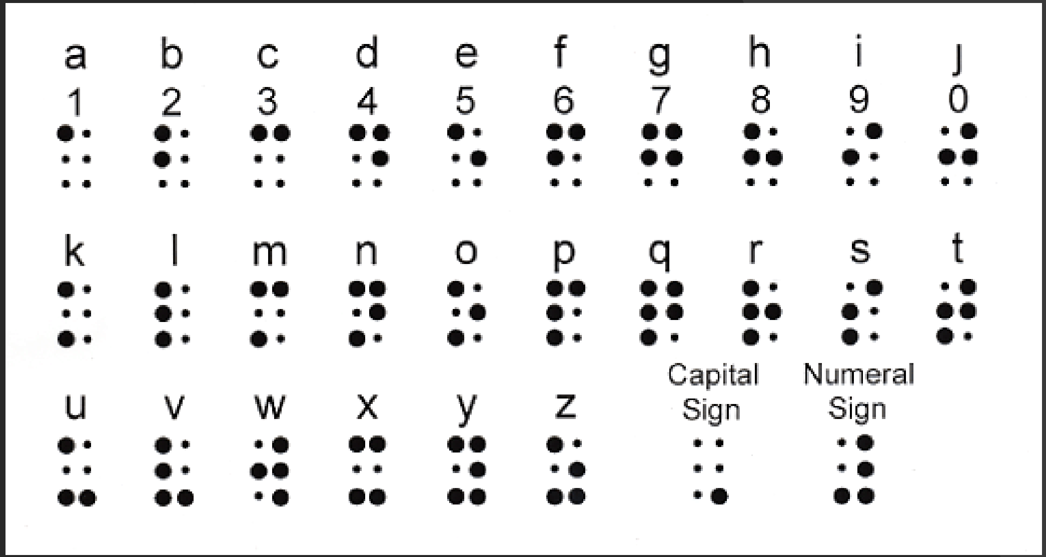


```
loss: 0.3987 - accuracy: 0.8595

loss: 0.2809 - accuracy: 0.8999

loss: 0.2369 - accuracy: 0.9142

loss: 0.2103 - accuracy: 0.9231

loss: 0.1905 - accuracy: 0.9307
```

Implementations

Visual impaired individuals

- Through an app and cameras apply it to the real world
- Add in a color option to tell the individual what color it is



Conclusion

Given more time and resources an individual could create a software that helps the visual impaired with picking out clothes.

References

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

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

Questions?

Thank

You!