Classification on Fashion MNIST

Mengqin Gong
Sep 2019

Problem

Problem

- A image classification problem
- Data: Fashion MNIST
- Goal: build a model to classify image of size 28*28 pixels into 10 classes

Potential Clients

- Fashion website which is looking for a classification model to automatically classify fashion images
- Anyone who wants to build a classification model on similar dataset with appropriate modification on data and model
- Deep learning beginner who wants to get started with image classification.

Data

Data Overview

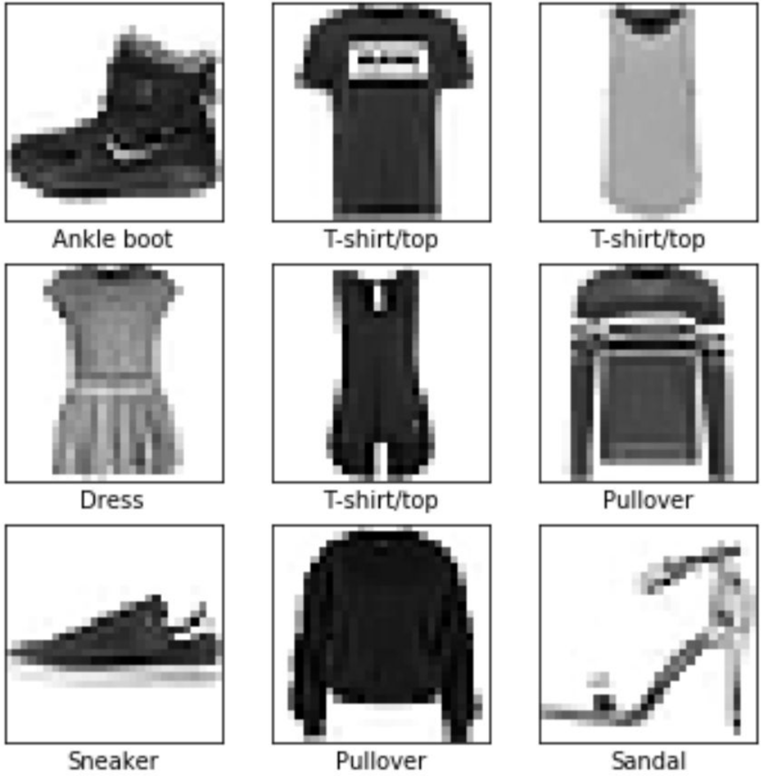
- Training set: 60,000 gray-scaled images
- Test set: 10,000 gray-scaled images
- Image size: 28x28 pixels (784 features)
- Each feature: a value from 0 to 255 (the pixel intensity, 0 for white and 255 for black)

Labels

- Each image is associated with a label from:

0	T-shirt/top	5	Sandal
1	Trouser	6	Shirt
2	Pullover	7	Sneaker
3	Dress	8	Bag
4	Coat	9	Ankle boot

Example


Ankle boot
T-shirt/top
T-shirt/top
Dress
T-shirt/top
Pullover
Sneaker
Pullover
Sandal

Modeling

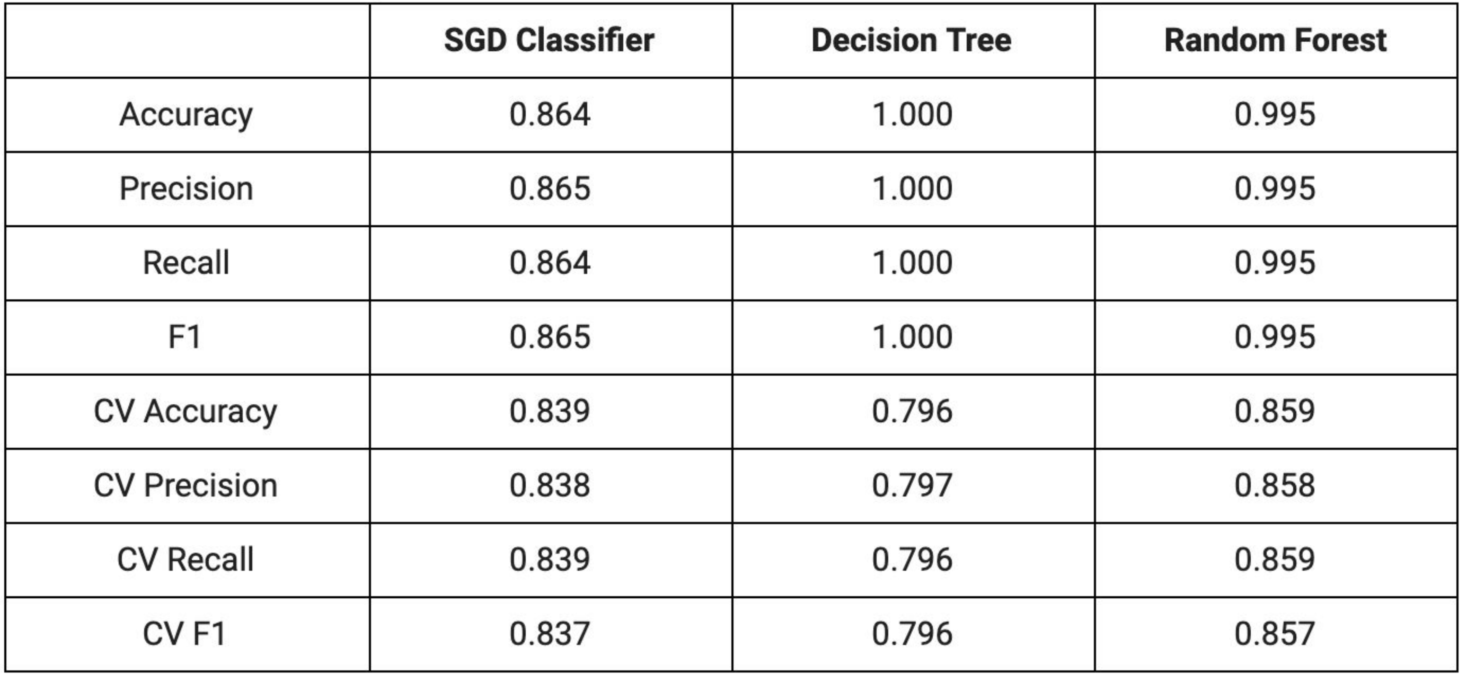
Feature Scaling

- Each feature has a value in the range of 0 to 255
- Normalization is necessary to scale the data dimensions to make them approximately the same scale
- Scale these values to a range of 0 to 1 by dividing them by 255

Classification with Machine Learning

- Models to be used:
  - SGD Classifier
  - Decision Tree Classifier
  - Random Forest Classifier

Classification with Machine Learning



	SGD Classifier	Decision Tree	Random Forest
Accuracy	0.864	1.000	0.995
Precision	0.865	1.000	0.995
Recall	0.864	1.000	0.995
F1	0.865	1.000	0.995
CV Accuracy	0.839	0.796	0.859
CV Precision	0.838	0.797	0.858
CV Recall	0.839	0.796	0.859
CV F1	0.837	0.796	0.857

Classification with Machine Learning

- Random Forest Classifier did a better work on cross-validation score
- Continue to hyperparameter tuning on Random Forest Classifier: use Grid Search technique
- Before fine tuning, dimension reduction is necessary

Principal Component Analysis (PCA)

What is Principal Component Analysis (PCA)?

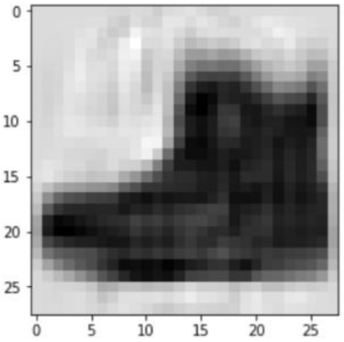
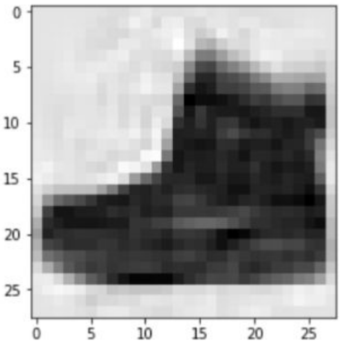
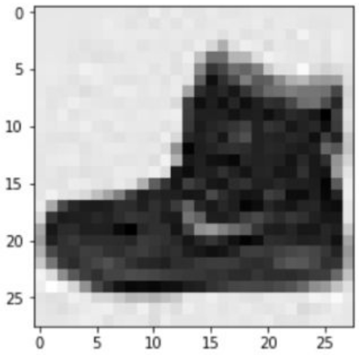
- A dimension reduction technique
- Goal: Reduce dimension of data without losing too much info

Why PCA?

- Each Fashion MNIST image has 784 features
- Grid Search is very time consuming with 784 features

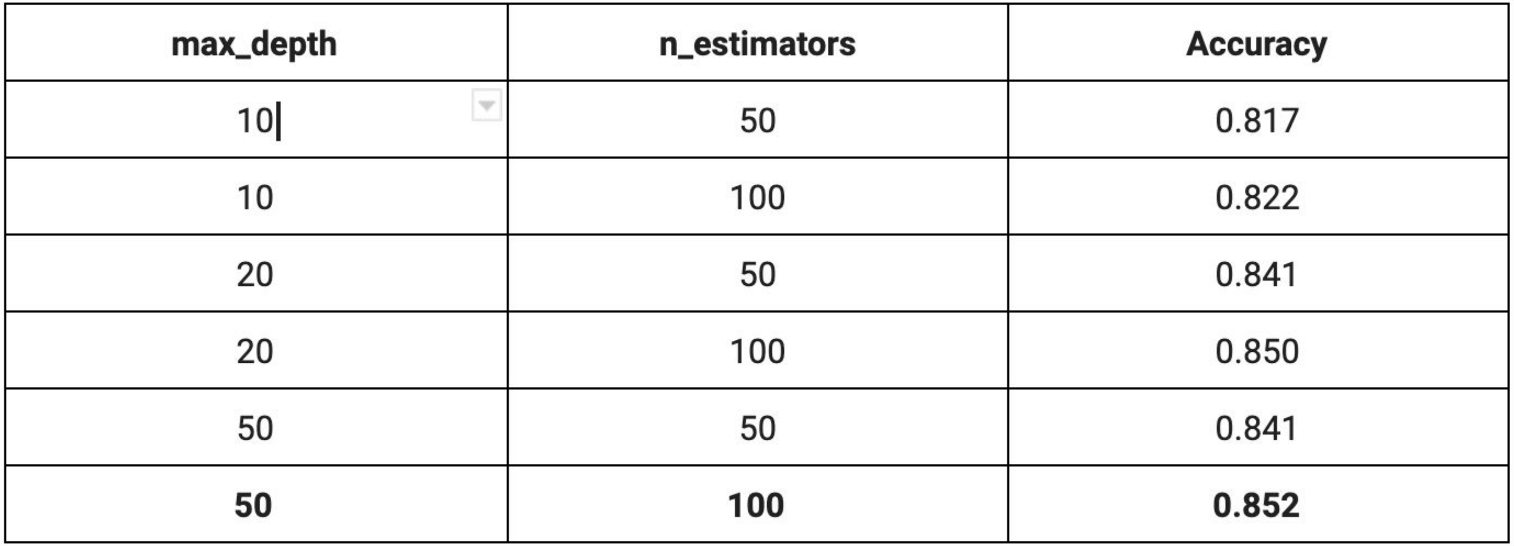
Principal Component Analysis (PCA)

Under n_components=0.9 and 0.95 we lose ~90% and ~75% features, the quality of image is poor, so 0.99 is a good value for n_components

N_components =	0.9	0.95	0.99
# of features left	84 (11.0%)	187 (24.0%)	459 (59.0%)
Sample of image recovered			

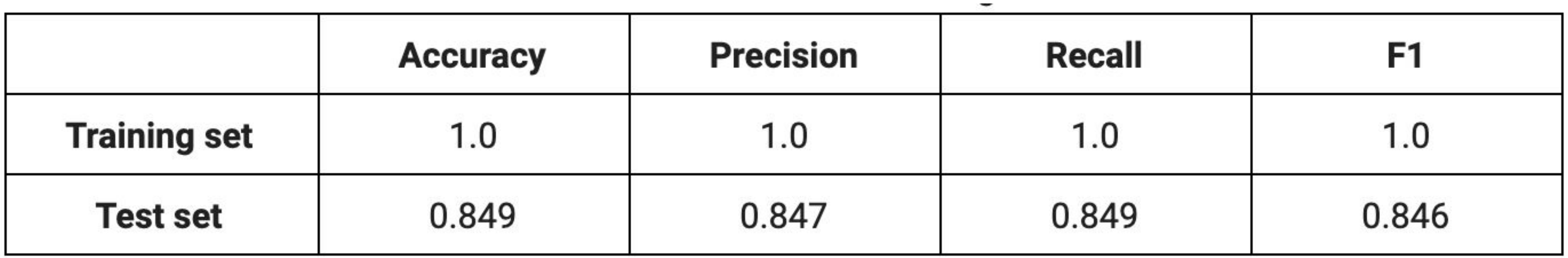
Grid Search

- Parameters to be tuned:
  - n_estimators: number of estimators in the model, values [50, 100]
  - max_depth: maximum depth of trees, values [10, 20, 50]



max_depth	**n_estimators**	**Accuracy**
10	50	0.817
10	100	0.822
20	50	0.841
20	100	0.850
50	50	0.841
50	**100**	**0.852**

Best Random Forest Model Performance



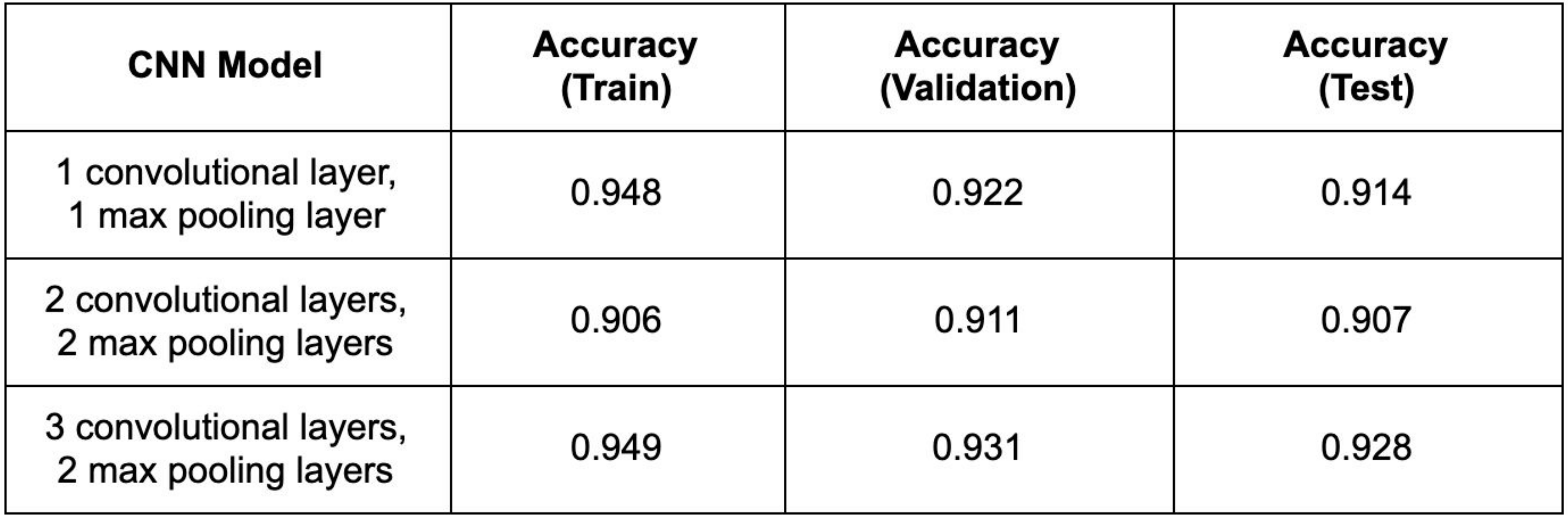
	Accuracy	Precision	Recall	F1
Training set	1.0	1.0	1.0	1.0
Test set	0.849	0.847	0.849	0.846

Classification with Deep Learning

Convolutional Neural Networks(CNNs)

- CNN with 1 Convolutional Layer
- CNN with 2 Convolutional Layer
- CNN with 3 Convolutional Layer

Convolutional Neural Networks (CNN)

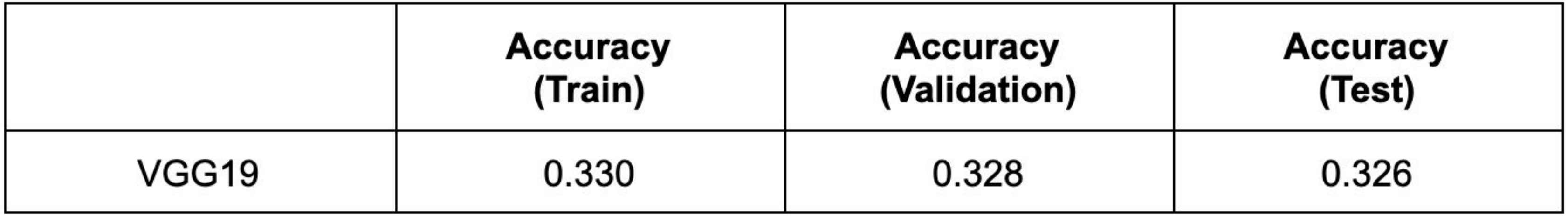


CNN Model	Accuracy (Train)	Accuracy (Validation)	Accuracy (Test)
1 convolutional layer, 1 max pooling layer	0.948	0.922	0.914
2 convolutional layers, 2 max pooling layers	0.906	0.911	0.907
3 convolutional layers, 2 max pooling layers	0.949	0.931	0.928

Convolutional Neural Networks (CNN)

- The CNN model with 3 convolutional layer has the highest accuracy
- Test set: 92.8%
- Much higher than that of Random Forest Classifier
- Still a little bit overfitting, but acceptable

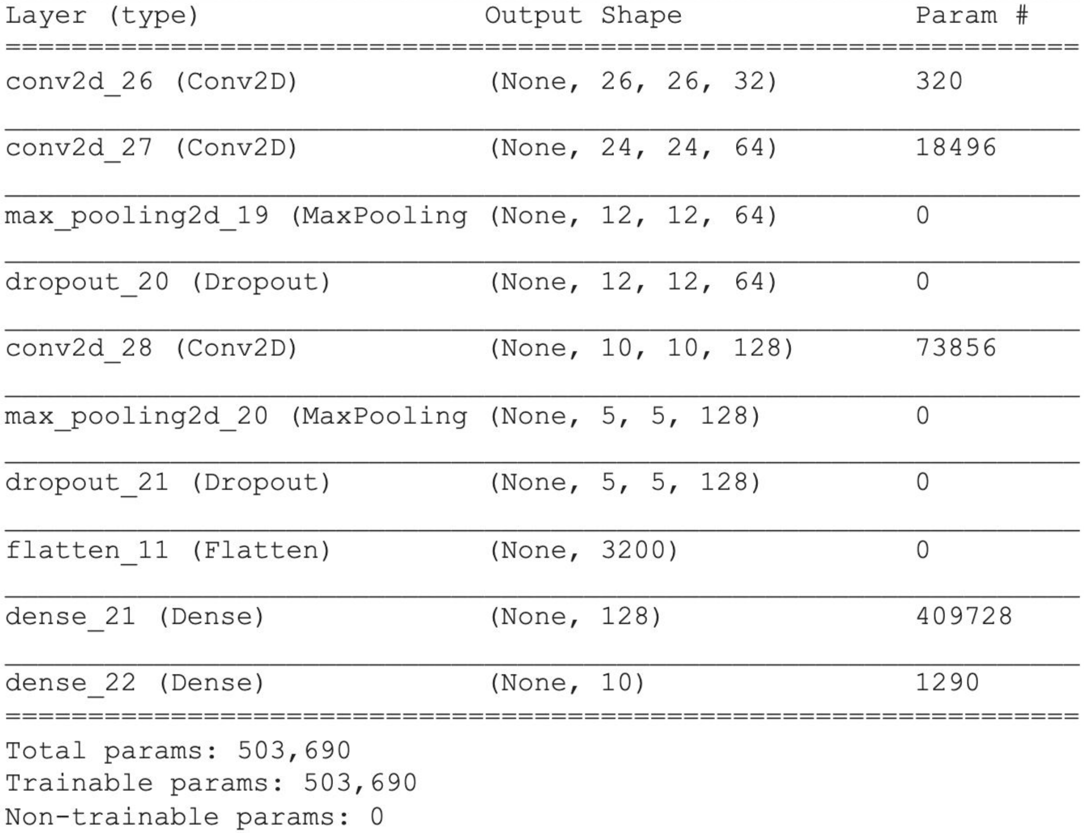
Transfer Learning



	Accuracy (Train)	Accuracy (Validation)	Accuracy (Test)
VGG19	0.330	0.328	0.326

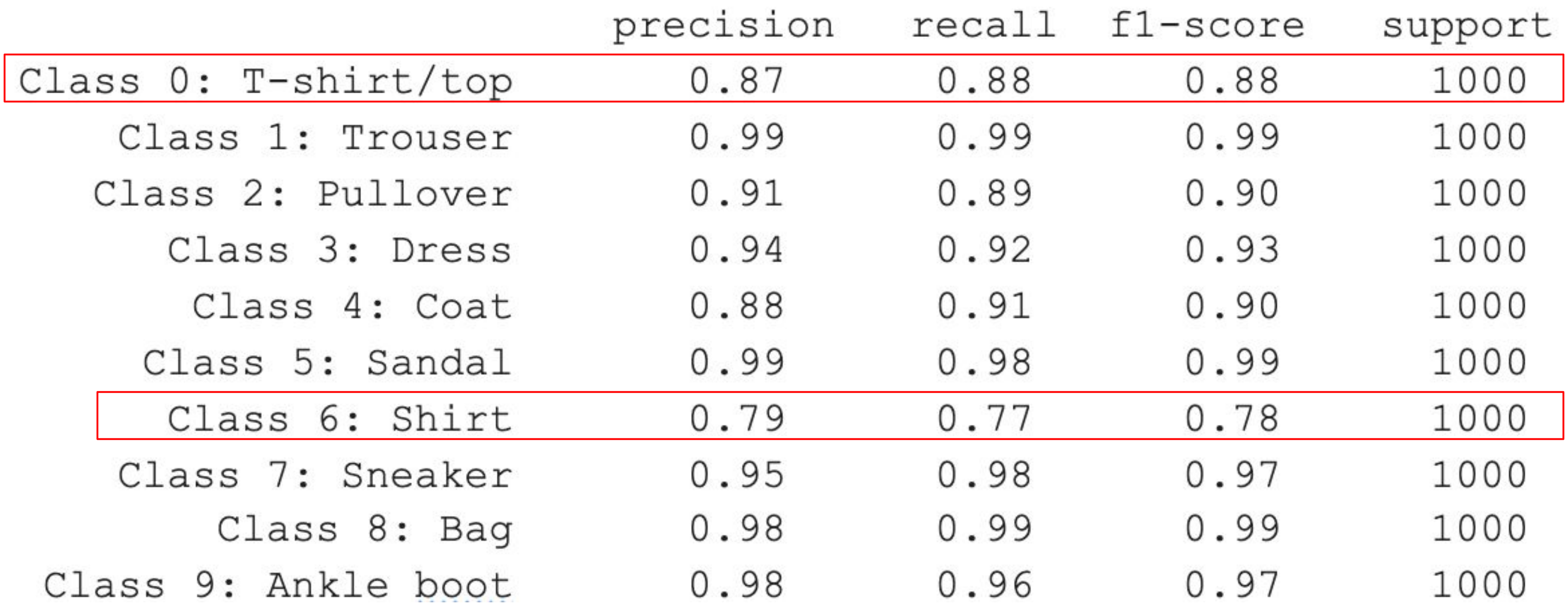
Final Model

CNN with 3 Convolutional Layers



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_26 (Conv2D)           (None, 26, 26, 32)        320
_________________________________________________________________
conv2d_27 (Conv2D)           (None, 24, 24, 64)        18496
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 12, 12, 64)        0
_________________________________________________________________
dropout_20 (Dropout)         (None, 12, 12, 64)        0
_________________________________________________________________
conv2d_28 (Conv2D)           (None, 10, 10, 128)       73856
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 5, 5, 128)         0
_________________________________________________________________
dropout_21 (Dropout)         (None, 5, 5, 128)         0
_________________________________________________________________
flatten_11 (Flatten)         (None, 3200)              0
_________________________________________________________________
dense_21 (Dense)             (None, 128)               409728
_________________________________________________________________
dense_22 (Dense)             (None, 10)                1290
=================================================================
Total params: 503,690
Trainable params: 503,690
Non-trainable params: 0
```

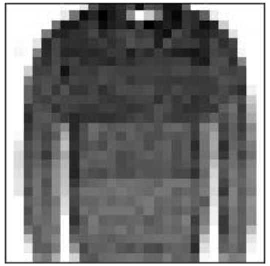
Classification Report



```
                        precision    recall  f1-score   support
Class 0: T-shirt/top         0.87      0.88      0.88      1000
     Class 1: Trouser        0.99      0.99      0.99      1000
    Class 2: Pullover        0.91      0.89      0.90      1000
       Class 3: Dress        0.94      0.92      0.93      1000
        Class 4: Coat        0.88      0.91      0.90      1000
      Class 5: Sandal        0.99      0.98      0.99      1000
       Class 6: Shirt        0.79      0.77      0.78      1000
     Class 7: Sneaker        0.95      0.98      0.97      1000
         Class 8: Bag        0.98      0.99      0.99      1000
  Class 9: Ankle boot        0.98      0.96      0.97      1000
```

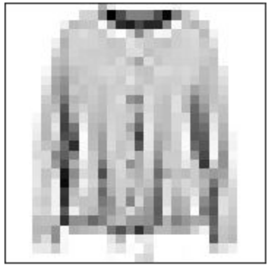
The model is not good at classify Shirt and T-shirt/top

Example of Mistakes

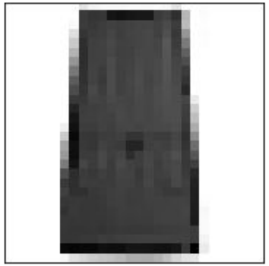


True: Pullover
Predicted: Shirt

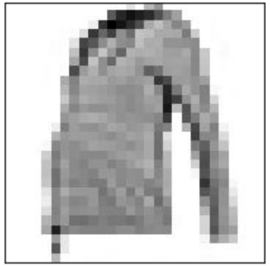
True: Ankle boot
Predicted: Sandal



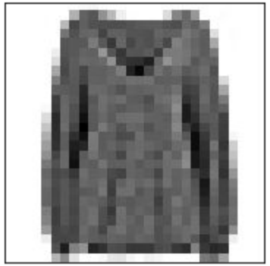
True: Coat
Predicted: Shirt



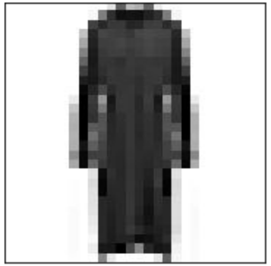
True: Dress
Predicted: T-shirt/top



True: Pullover
Predicted: Shirt



True: Pullover
Predicted: Shirt



True: Dress
Predicted: Coat

True: Ankle boot
Predicted: Sneaker

Some are difficult to classify even for human beings!

Conclusion

Conclusion

- Fashion MNIST is a good starting point for Machine Learning and Deep Learning beginners.
- By Machine Learning models, I got a classification accuracy of 85% on the test data using Random Forest Classifier to predict.
- By Deep Learning methods, the best model CNN with 3 convolutional layers reaches an accuracy of **93%** on the test data!

Limitations and Next Steps

- The CNN model with 3 convolutional layers is underperforming for class 0 T-shirt/Top and class 6 Shirt. Next step can be a model that can learn more from mistakes, therefore improve the accuracy on the two classes mentioned above.

- Due to the capacity of my device, I was only able to process data in a small batch size, increasing the batch size may improve the model trained, but will definitely be more time and resource consuming. Possible solutions can be trying out more pre-trained models and running model training on GPUs.