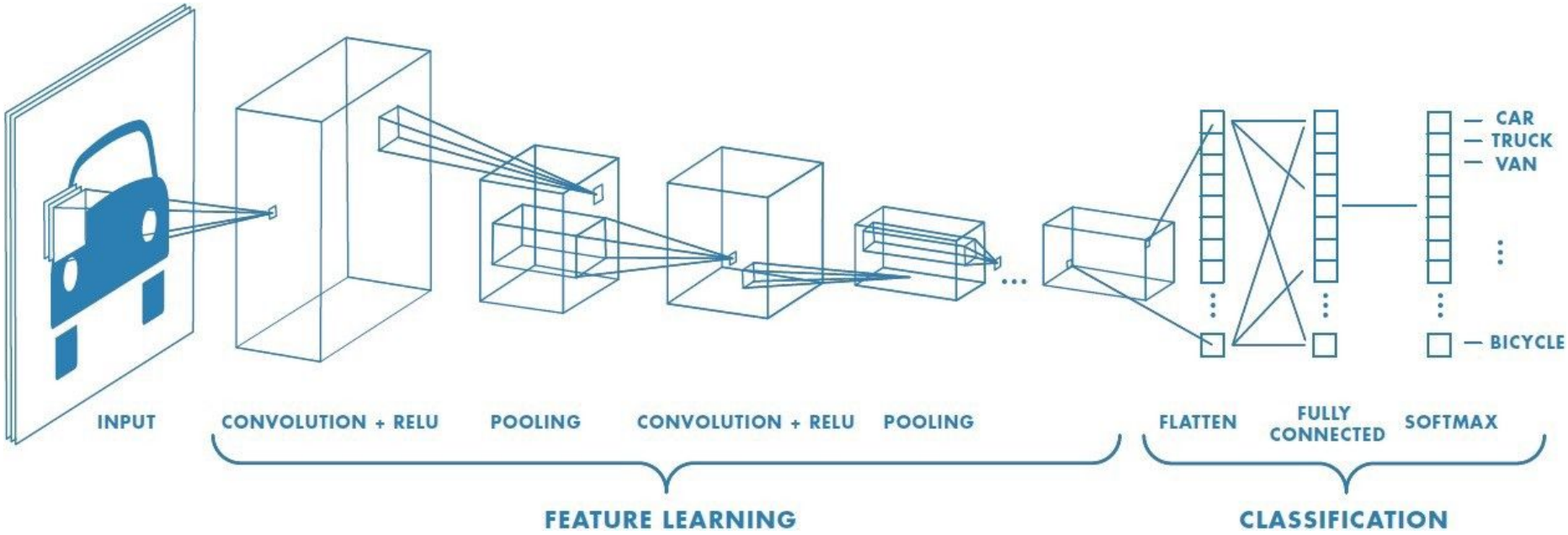
MNIST Fashion classification using a CNN

Dino Anastasopoulos: 1900661
Philani Mpofu: 1848751
Chloë Smith: 1877342

What is a CNN?


CAR
TRUCK
VAN
BICYCLE
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
FEATURE LEARNING
CLASSIFICATION

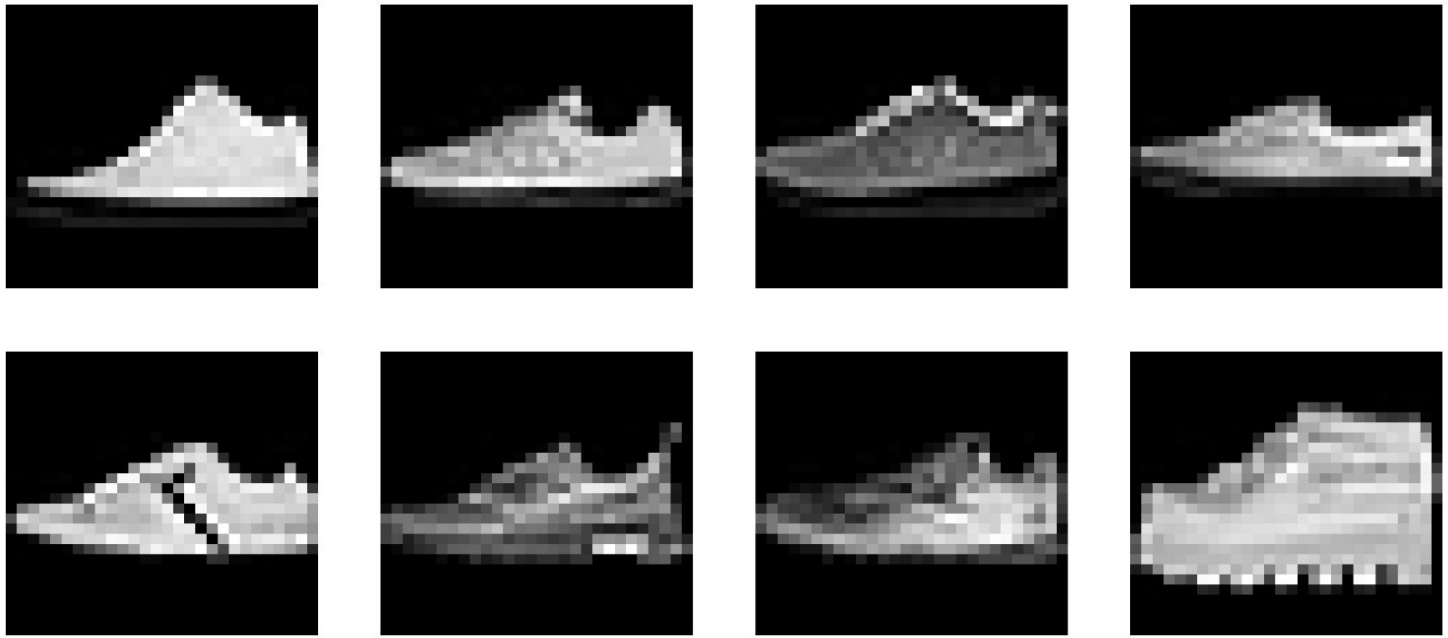
Dataset Description

- 70 000 images of clothing articles
- 28x28 gray-scale images
- Images have 10 different labels

0. T-Shirt
1. Pair of Trousers
2. Pullover
3. Dress
4. Coat
5. Pair of Sandals
6. Shirt or Top
7. Pair of Sneakers
8. Bag
9. Pair of Ankle Boots

Examples of data

Examples of sneaker class



Train, Validation, Test Data Split

Training: 68%

Feature Matrix:
(48000, 28, 28)

Target Vector:

(48000,)

Validation: 17%

Feature Matrix:
(12000, 28, 28)

Target Vector:
(12000,)

Testing: 15%

Feature Matrix:
(10000, 28, 28)

Target Vector:
(10000,)

Preprocessing

Input Feature Matrix:

Keras requires a 3 dimensional vector including the number of colour channels.

- Training Data Feature Matrix: (48000, 28, 28, 1)
- Validation Data Feature Matrix: (12000, 28, 28, 1)
- Testing Data Feature Matrix: (10000, 28, 28, 1)

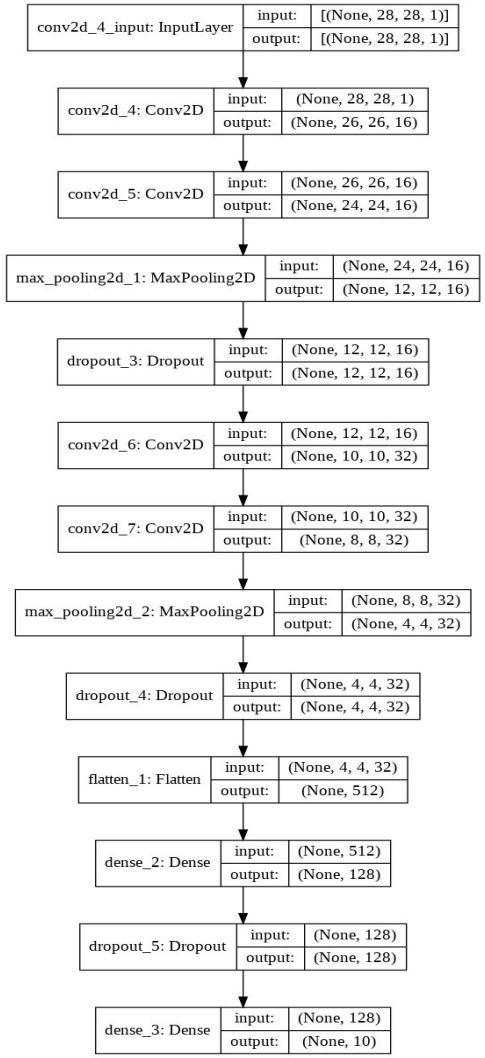
Classes:

One hot encoding

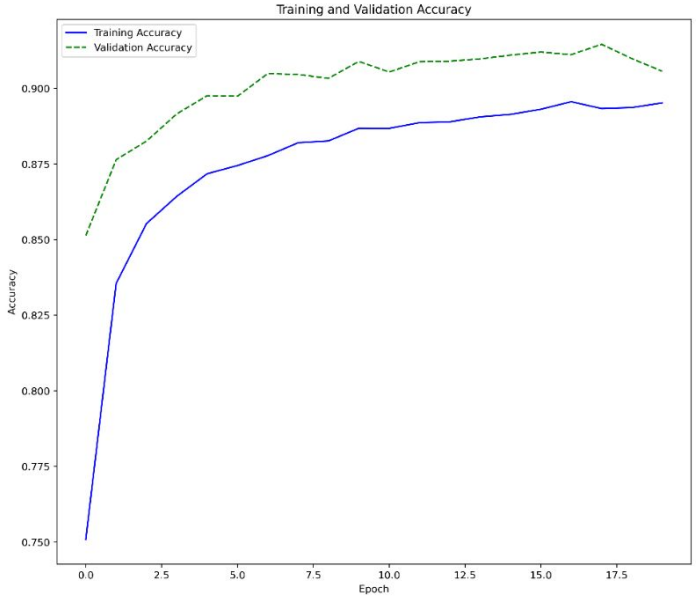
Example: Class 8 (Bag) ➜ (0,0,0,0,0,0,0,0,1,0)

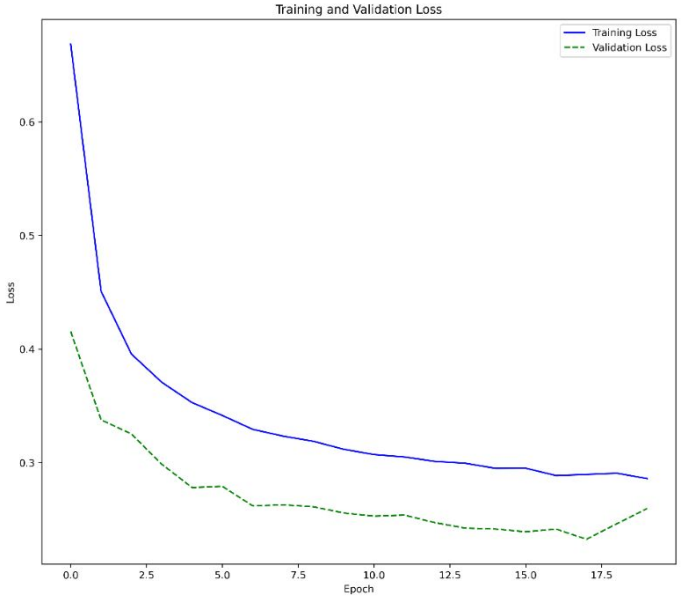
Baseline Implementation

1. 3 Convolutional Layers
2. 1 Max Pooling Layer
3. 1 Dropout
4. 2 Convolutional Layers
5. 1 Max Pooling Layer
6. 1 Dropout
7. 1 Flatten
8. 1 Dense Layer
9. 1 Dropout
10. Dense Layer



Baseline Model Results





Hyperparameter Tuning

Number of trials: 20

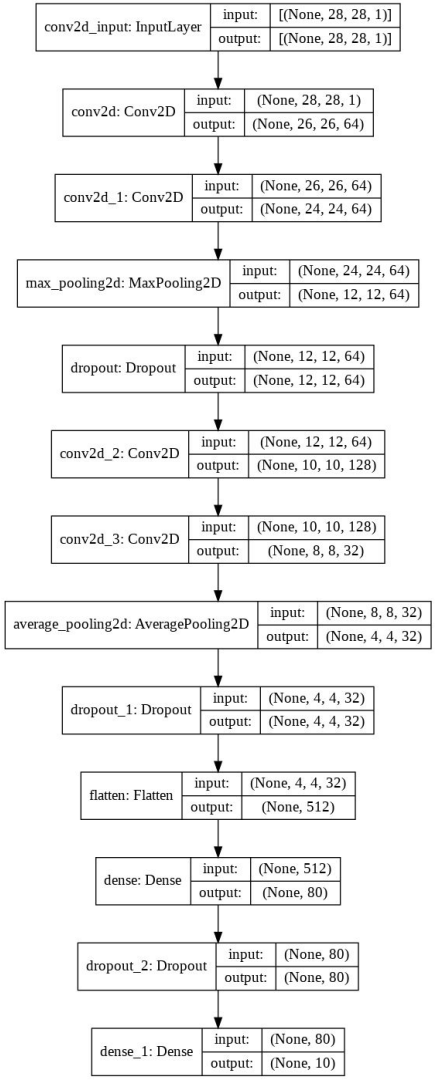
Number of Epochs Per Search: 10

Executions Per Trial: 2

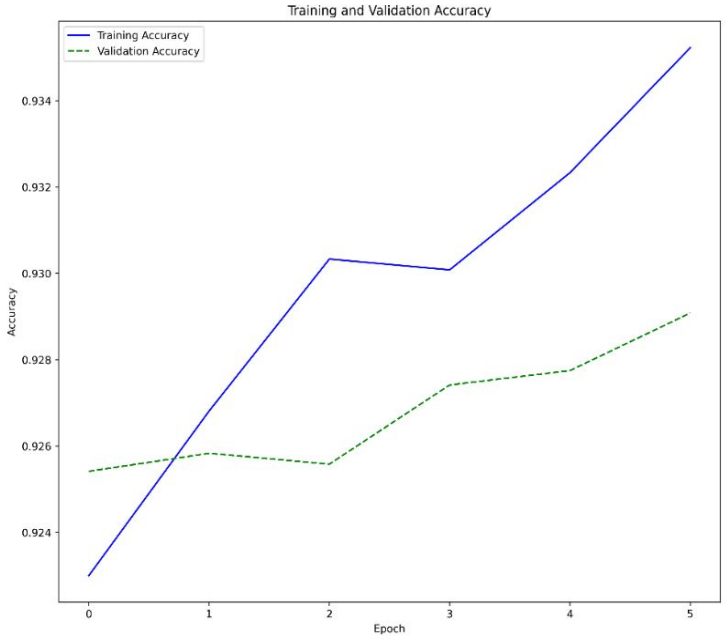
```
Search space summary
Default search space size: 12
Conv_Filters_1 (Choice)
{'default': 16, 'conditions': [], 'values': [16, 32, 64], 'ordered': True}
Conv_Filters_2 (Choice)
{'default': 16, 'conditions': [], 'values': [16, 32, 64], 'ordered': True}
Conv_Activation_2 (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'sigmoid'], 'ordered': False}
Dropout_Rate_1 (Float)
{'default': 0.25, 'conditions': [], 'min_value': 0.0, 'max_value': 0.5, 'step': 0.05, 'sampling': None}
Conv_Filters_3 (Choice)
{'default': 32, 'conditions': [], 'values': [32, 64, 128], 'ordered': True}
Conv_Activation_3 (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'sigmoid'], 'ordered': False}
Conv_Filters_4 (Choice)
{'default': 64, 'conditions': [], 'values': [32, 64, 128], 'ordered': True}
Conv_Activation_4 (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'sigmoid'], 'ordered': False}
Dropout_Rate_2 (Float)
{'default': 0.25, 'conditions': [], 'min_value': 0.0, 'max_value': 0.5, 'step': 0.05, 'sampling': None}
Last_Dense_NumUnits (Int)
{'default': 64, 'conditions': [], 'min_value': 16, 'max_value': 128, 'step': 16, 'sampling': None}
Dropout_Rate_3 (Float)
{'default': 0.25, 'conditions': [], 'min_value': 0.0, 'max_value': 0.5, 'step': 0.05, 'sampling': None}
Adam_Optim_LR (Float)
{'default': 0.001, 'conditions': [], 'min_value': 0.0001, 'max_value': 0.01, 'step': None, 'sampling': 'log'}
```

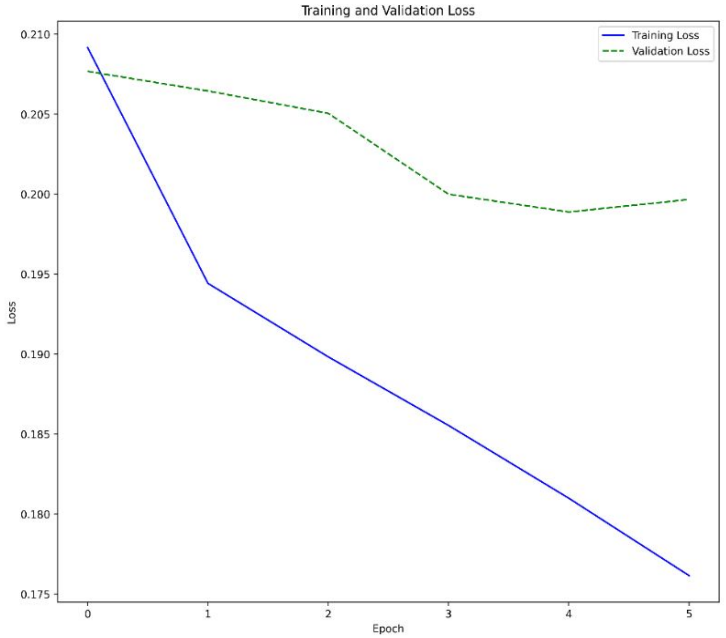
Hyperparameter Process

Layer	Hyperparameter	Before Tuning	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
conv2d	Number of filters	16	64	64	32	64	64
conv2d_1	Number of filters	16	64	64	16	32	16
	Activation function	ReLU	ReLU	ReLU	sigmoid	sigmoid	sigmoid
dropout	Dropout Rate	0.5	0.25	0.25	0.30	0.2	0.45
conv2d_2	Number of filters	32	128	32	32	32	128
	Activation Function	ReLU	ReLU	sigmoid	ReLU	ReLU	ReLU
conv2d_3	Number of filters	32	32	32	128	128	32
	Activation Function	ReLU	ReLU	ReLU	ReLU	ReLU	sigmoid
dropout_1	Dropout rate	0.3	0.25	0.4	0.30	0.45	0.25
dense	Number of nodes	128	80	32	48	48	48
dropout_2	Dropout rate	0.2	0.15	0.2	0.15	0.05	0.2

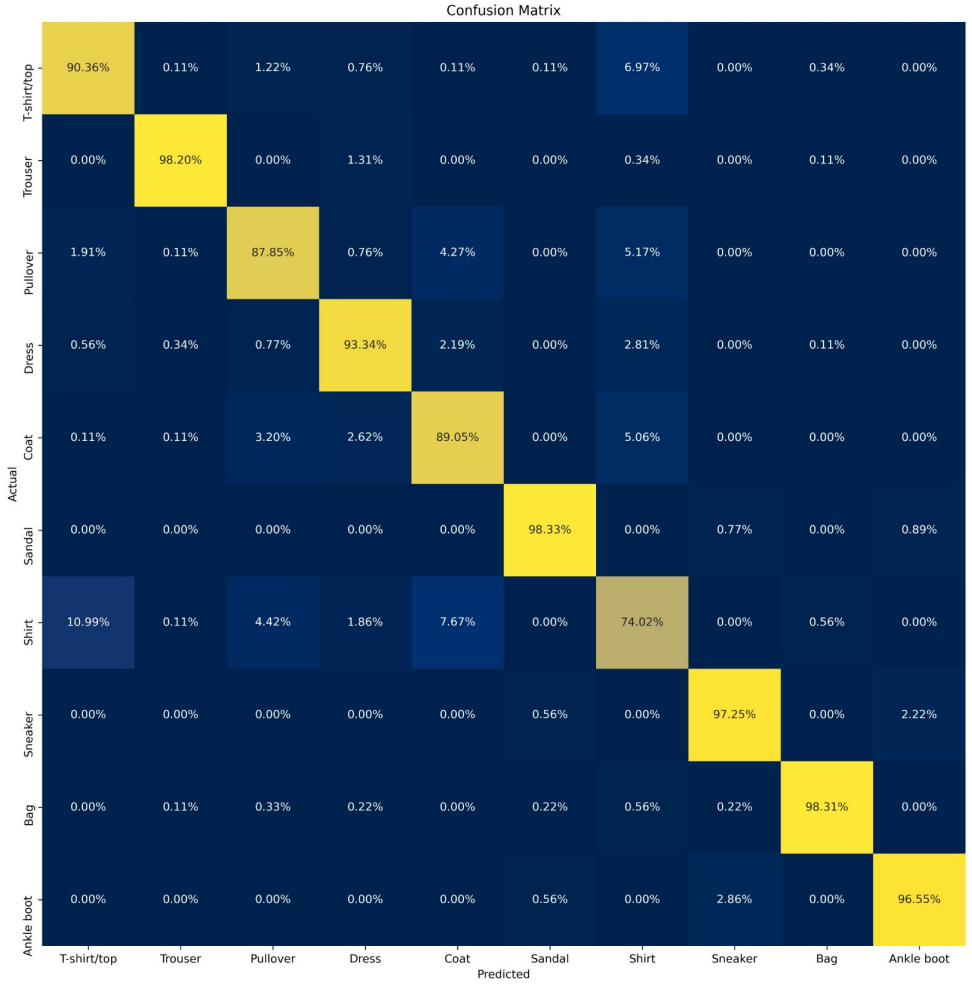


Tuned Model Results

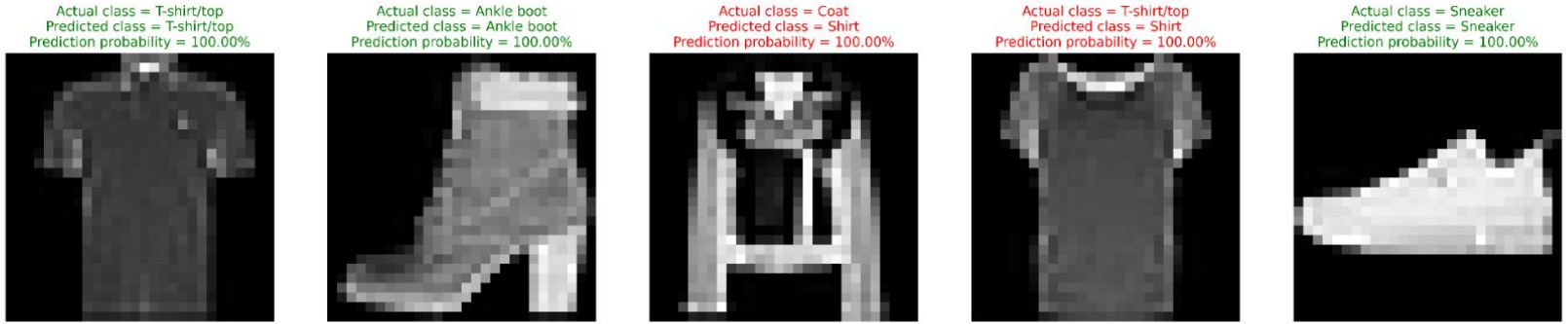




Confusion Matrix



Prediction Visualizations



Applications

- Automatic labelling of clothing items for e-commerce retail stores.
- Automatic price determination for retail stores.

THE END