Reed Ballesteros
MSDS 458-DL: Artificial Intelligence & Deep Learning
Spring 2023
Prof. Syamala Srinivasan, Ph.D.
June 4, 2023

A.4: Fourth Research/Programming Assignment: Convolutional Neural Networks (CNNs) and the Fashion-MNIST Dataset

Abstract

This study details the development of several convolutional neural networks (CNNs) trained and tested for their performance accuracy using the Fashion-MNIST image dataset. The most optimal model would be integrated as part of a fashion retailer's recommendation system to offer other products to customers browsing their website. While we find a combination of regularization techniques applied on a CNN can yield significant gains in accuracy, we also compare them against the performance of various pre-trained models in which we find do not classify the dataset as well as our own models. Our most optimal CNN containing 9 total layers in 3 clustered groups was able to achieve a 93.7% accuracy against the Fashion-MNIST test dataset.

Introduction

This report documents the development of a convolutional neural network (CNN) with the goal of classifying clothing articles. The resulting model would be integrated into a fashion retailer's website in which it would scan and classify images that are displayed while a customer is browsing products on the website and its recommendation system would refer other items for them as well. Image classification accuracy for the system is important so the website can properly recommend to a visitor similar or related products that are popular, in-stock/available, are part of a current marketing campaign or promotion, or upsell to higher-end products. Displaying unrelated products could be jarring to the user experience and could possibly discourage them from continuing shopping on the website and thus a missed opportunity on a completed online sales transaction and collecting user information. We will choose the model configuration that yields the highest accuracy in image classification using the Fashion-MNIST dataset for model training, validation, and testing. We will also use this opportunity to build and compare several pre-trained CNNs on the same dataset.

Literature Review

There have been several studies on convolutional neural network (CNN) development using the Fashion-MNIST dataset. One study compared several CNNs trained with the dataset to create a model which

could be integrated into service robot technologies that could assist the elderly by identifying objects in their home such as clothing (Nocentini et al. 2022). The research led to the creation of the Multiple Convolutional Network (MCNN15) containing 15 CNNs achieving a test accuracy of 94% on the Fashion-MNIST dataset, surpassing other previous pre-trained models such as AlexNet and ResNet. Inspired by MCNN15, we created CNN models using sets of grouped convolutional layers and are able to achieve a 93.7% accuracy with the Fashion-MNIST test dataset.

Another study compared several pre-trained models using the Fashion-MNIST dataset in which the authors were able to yield a 96.2% test accuracy using the PyramidNet pre-trained model (Tang et al. 2020). The study also used a version of the ResNet pre-trained model where its optimal model was able to yield 94.2% accuracy. While our study includes the use of pre-trained models, including a version of ResNet, we were not able to achieve similar results with our ResNet pre-trained model.

Methods

Research Design

We will conduct our research by first downloading the Fashion-MNIST dataset available in the Keras framework and perform exploratory data analysis (EDA) on it to understand the data and its corresponding labels we will be using to build, train, and test our models. Each model is represented as an experiment. Our first section of experiments will involve building CNN models using 3 isolated convolution layers followed by 2 fully connected layers and a final softmax classification layer. Model variations will be created by adding regularization components such as batch normalization, dropout, and ridge regression (also called 'L2 Regularization') along with hyperparameter tuning of these components. We will also create a 4-layer CNN model and compare its performance to the 3-layer CNN models.

The second section of models contains a more complex architecture utilizing 3 groups of 3 convolutional layers in which each group is separated by a max pooling layer. Model variations in this section will be based on tuning regularization hyperparameters. The third section of models will utilize pre-trained models such as VGG16, VGG19, Inception V3 and ResNet152 V3, each of which will use a transformed Fashion-MNIST dataset to meet their respective image format requirements. The models in both of these sections will also be followed by 2 fully connected layers and a final softmax classification layer.

For each model we will record their accuracy and loss against the test subset, as well as precision, recall, F1 score, and RMSE. We will also plot each model's training and validation loss and accuracy over their respective number of epochs of training. Training for each model will automatically stop after validation accuracy does not improve after a given number of epochs, in which the best training epoch will be saved before the model overfits to the training data. Confusion matrices for each model will also be charted to find how well they perform for each image classification.

Implementation

Our models will be created using the Keras and TensorFlow frameworks available in Python. These are the most common tools used in developing such models based on their ease of use, customization options available, and their scalability for each convolutional or dense layer. Each convolutional and dense hidden layer will use the ReLU activation function as its simplicity in calculation can make the feedforward and backpropagation process fast, which will be helpful when we train deep models that contain a large number of parameters. The final output 10-way classification layer will use a softmax activation function.

The Keras model compiler will be configured to use the Adam optimizer, calculate cross entropy loss using the SparseCategorialCrossentropy class, and model performance will be based on accuracy metrics. For reproducibility and stabilize the random nature of the model training process, we will set a system seed of 43.

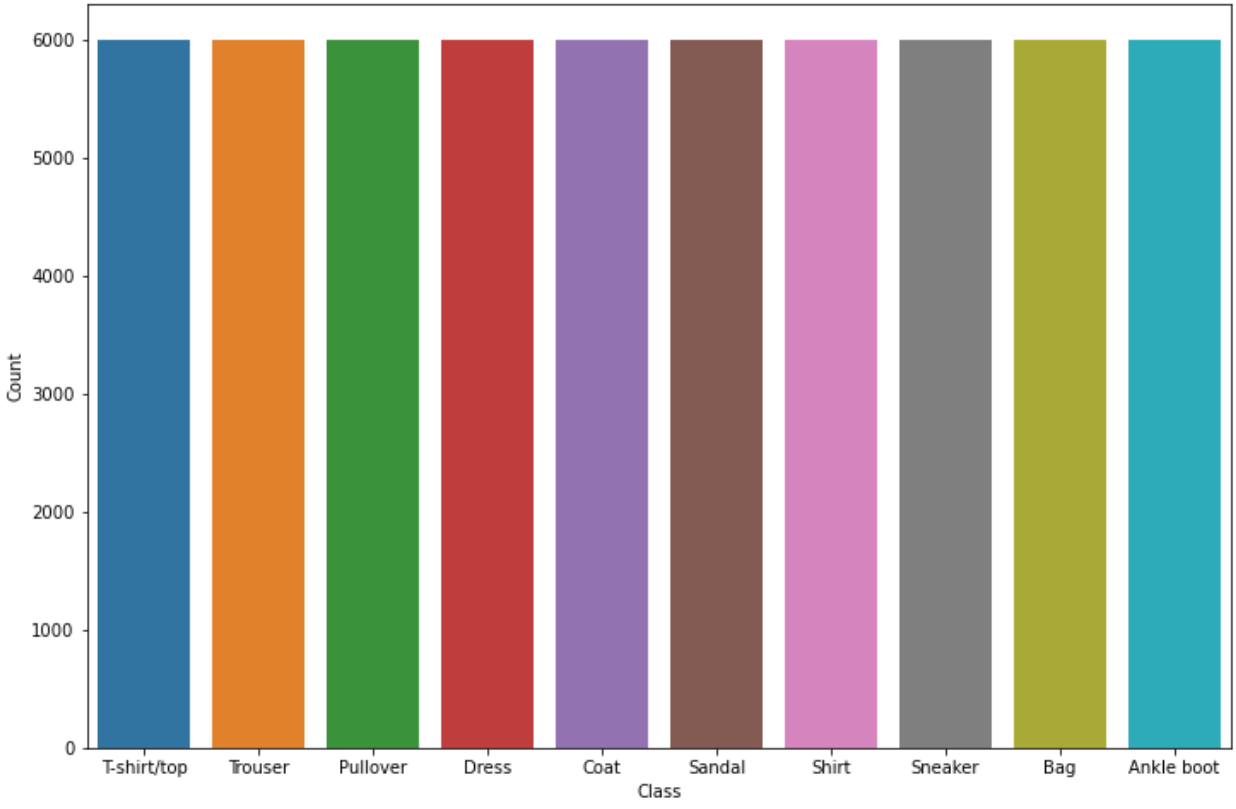
Dataset: Fashion-MNIST

The Fashion-MNIST dataset provided by German online retailer Zalando is a collection of images (60000 for training, 10000 for testing), each categorized as 1 of 10 classes (t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot). The dataset is commonly used for benchmarking machine learning models. Each image is represented as a 28x28 matrix over a single greyscale channel, where each value in the matrix represents a pixel integer value between 0 to 255. We will transform the integer pixel data to a standardized value between 0 and 1 before feeding the image into the models. We will also transform the dataset into three-channel color images and resize them to a larger scale to meet the minimum image size requirements for each respective pre-trained model.

Exploratory Data Analysis (EDA)

As described above, the Fashion-MNIST dataset is a collection of 70000 28x28 color images, each classified as one of 10 categories: t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot. Below is a sample of images from the dataset:

The training dataset contains 6000 samples of each classification.

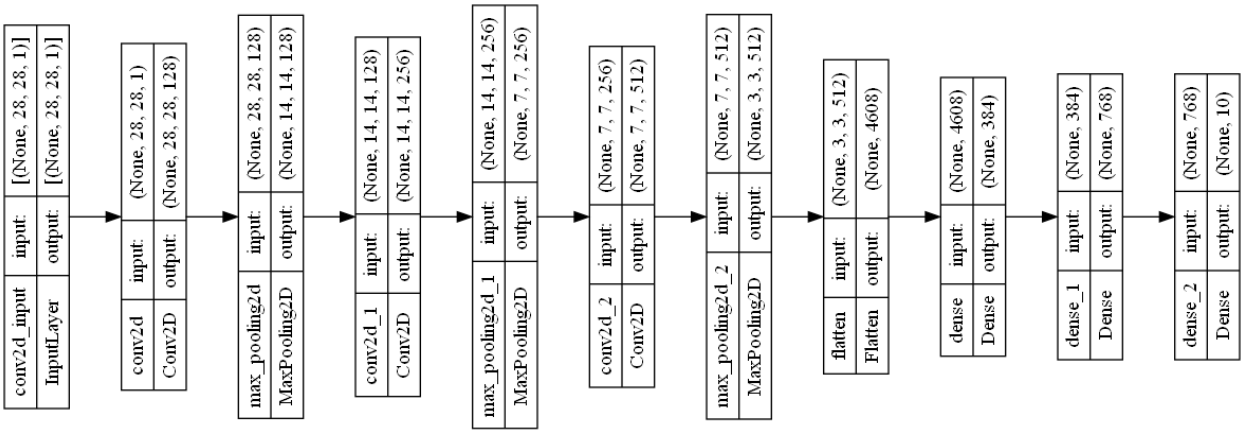


Results

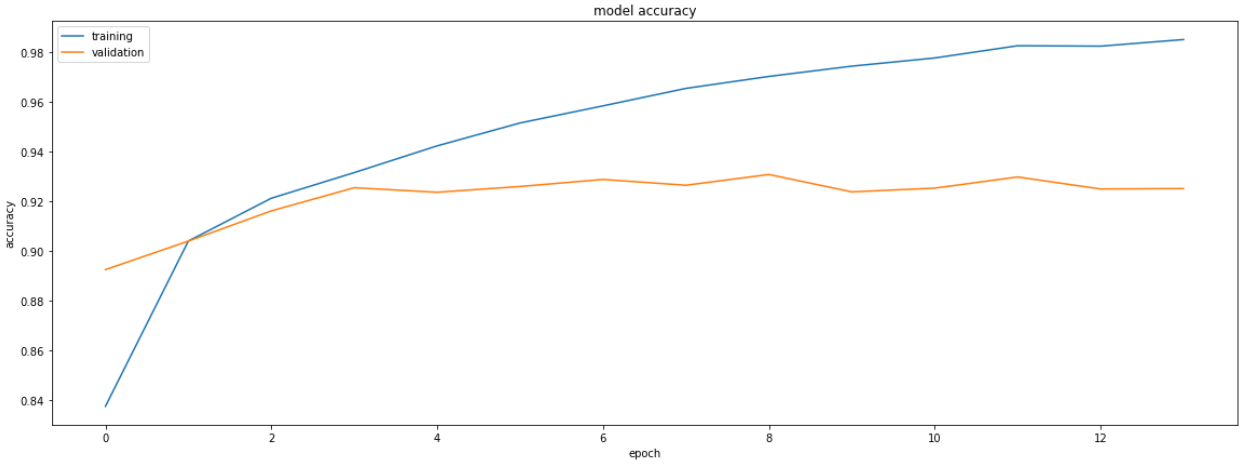
Section One: Convolutional Models

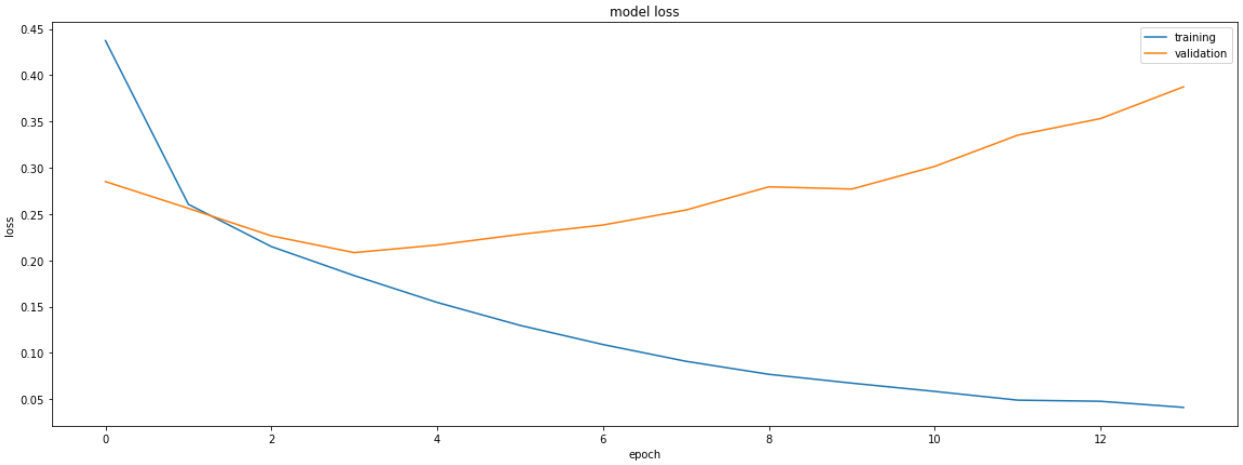
Experiment 1: CNN with 3 convolution/max pooling layers & 2 fully connected layers – no regularization

In this experiment we use a CNN consisting of three convolutional layers using a kernel size of 3x3 with the first layer containing 128 filters, the second layer containing 256 filters, and the third layer containing 512 filters, each separated by a 2x2 max pooling layer. The convolutional base then feeds into a DNN containing 2 fully connected layers, with the first layer containing 384 units and the second layer containing 768 units, followed by the 10-way softmax classification output layer.



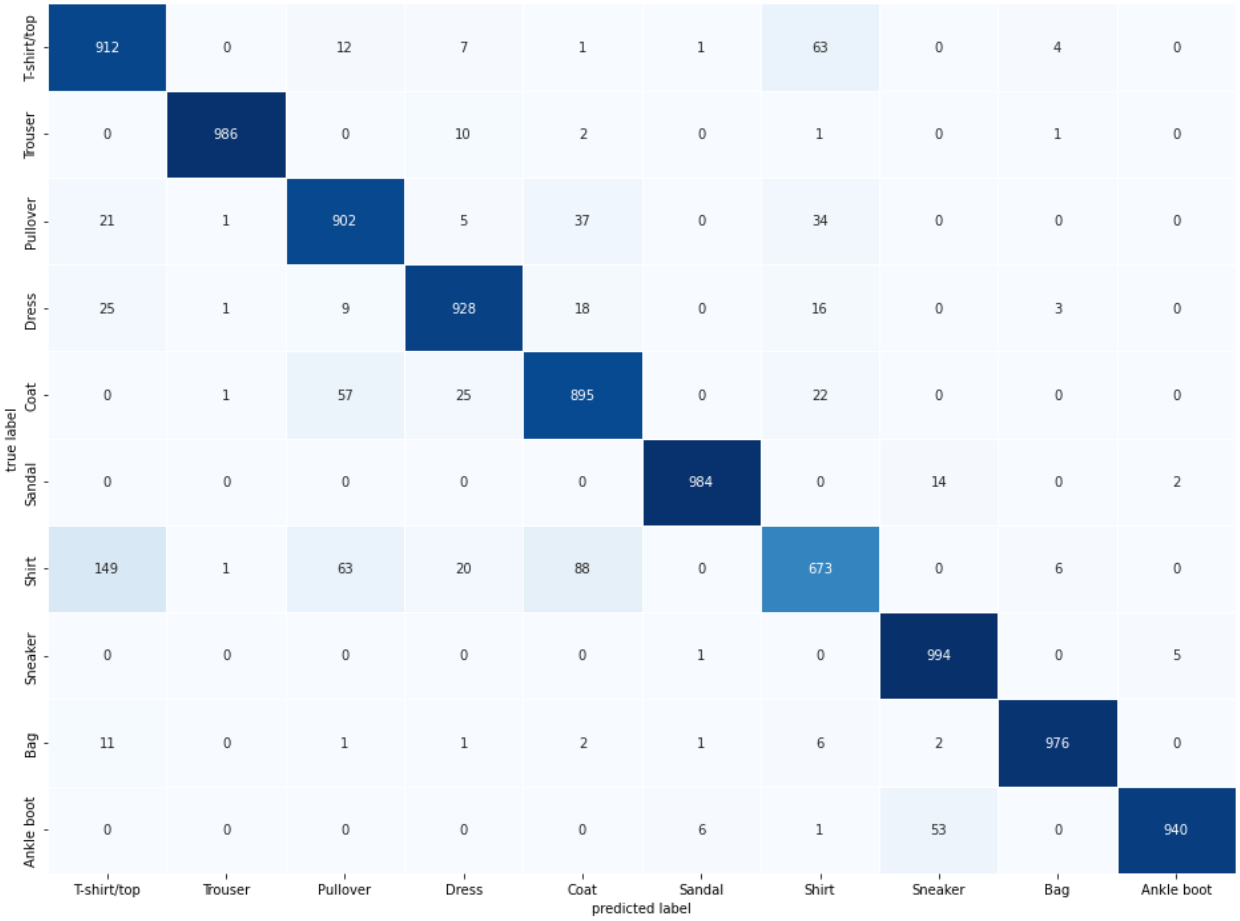
Training and validation accuracy and cross entropy loss charts are below.





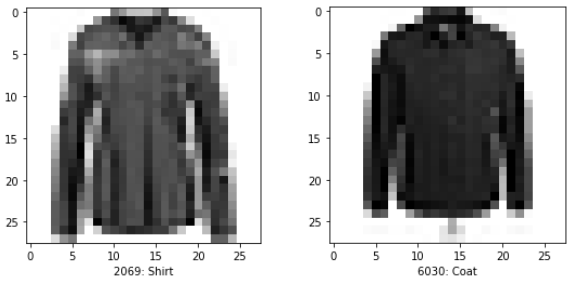
We see validation results start to diverge from the training results within 5 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 91.9%, which is a high start in what we would expect our other models to be in within a similar range.

The experiment yields the following confusion matrix:

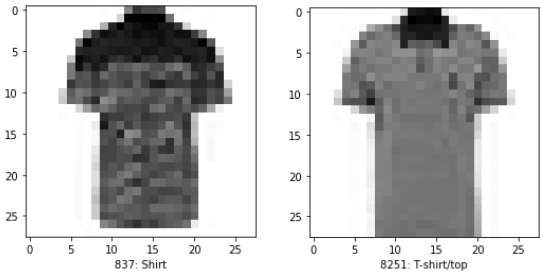


From the matrix above we can see how well the model can correctly classify images from the Fashion-MNIST dataset, with the exception of shirts. Shirt misclassification would be a common theme shown in all confusion matrices for each model in this study, as images in the dataset would understandably look very similar to other clothing categories like tops, pullovers, and coats.

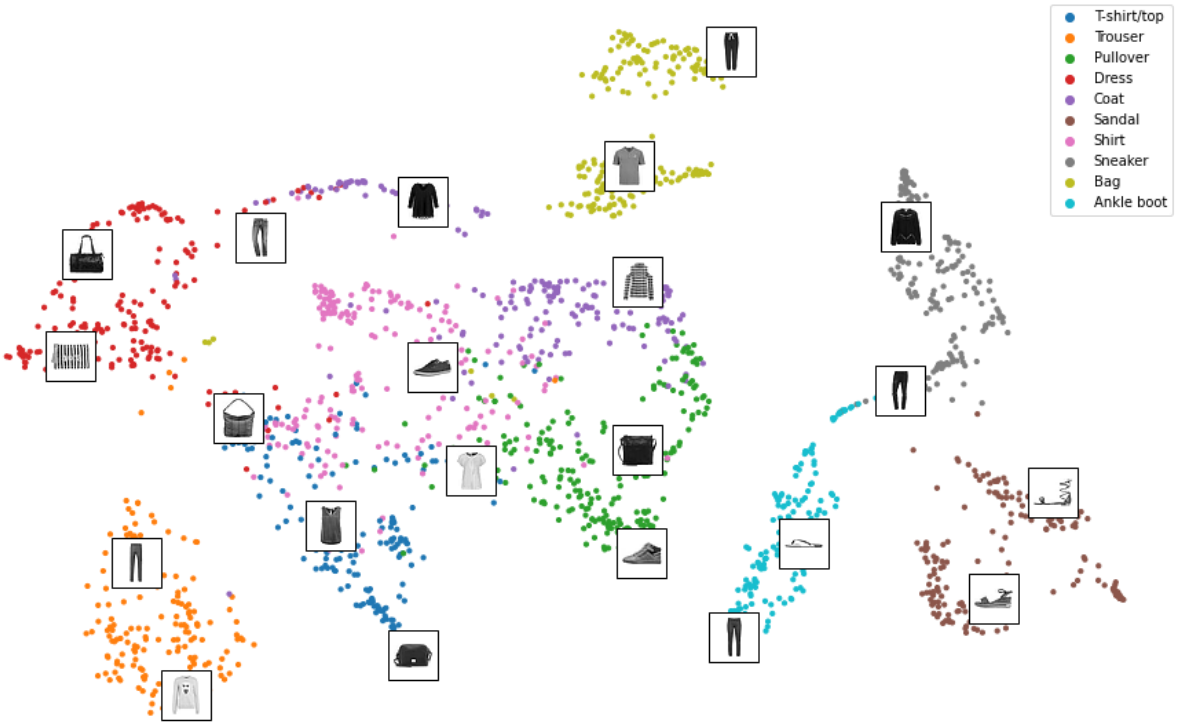
Images labeled ‘shirt’ and ‘coat’ in the dataset can visibly look similar such as these examples:



As well as between images labeled ‘shirt’ and ‘t-shirt/top’:



We can reduce the model to 2 features and create a scatterplot to observe the distribution of a sample of model classifications.



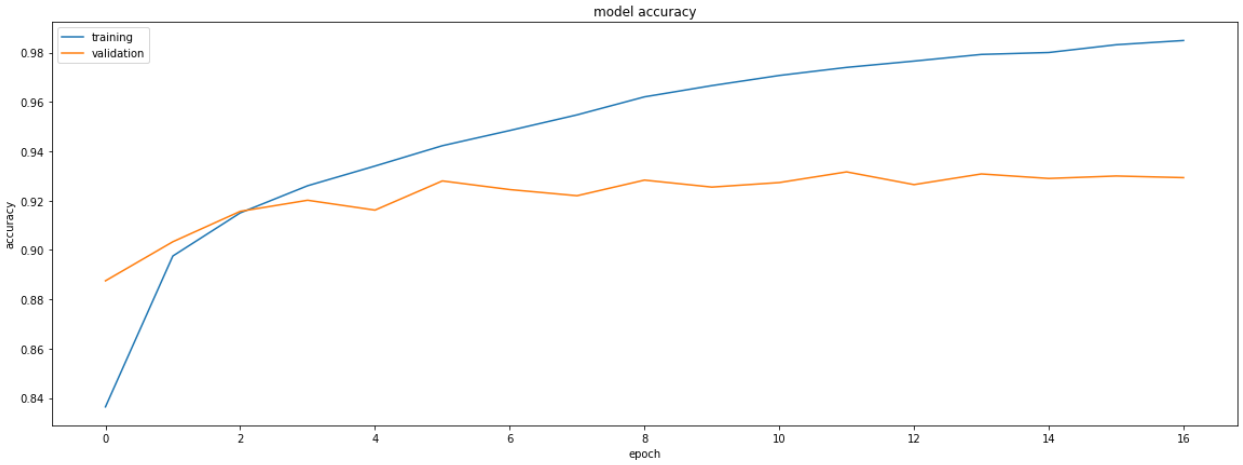
We can see distinct groups form in this scatterplot sample due to the model's high test accuracy. We've observed similar clustering in our other models as well as they're also able to achieve high test accuracy of about 90%.

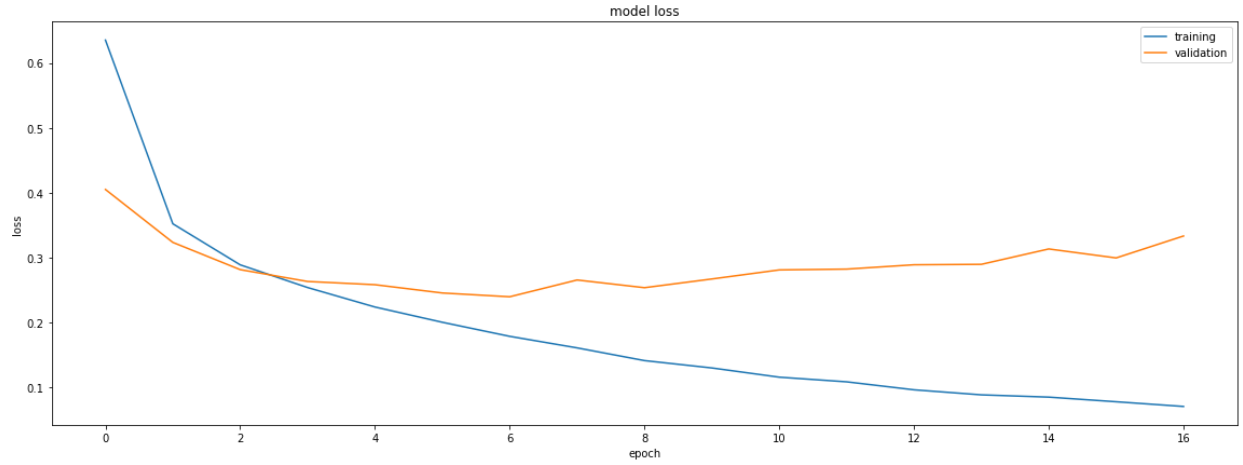
Experiment 2: CNN with 3 convolution/max pooling layers & 2 fully connected layers – L2 Regularization (0.001)

Experiment 2 is similar to Experiment 1, containing a CNN consisting of three 3x3 convolutional layers with the first layer containing 128 filters, the second layer containing 256 filters, and the third layer containing 512 filters. The convolutional base then feeds into a DNN containing 2 fully connected layers,

with the first layer containing 384 units and the second layer containing 768 units, followed by the 10-way softmax classification output layer. All fully connected layers utilize L2 regularization (0.001).

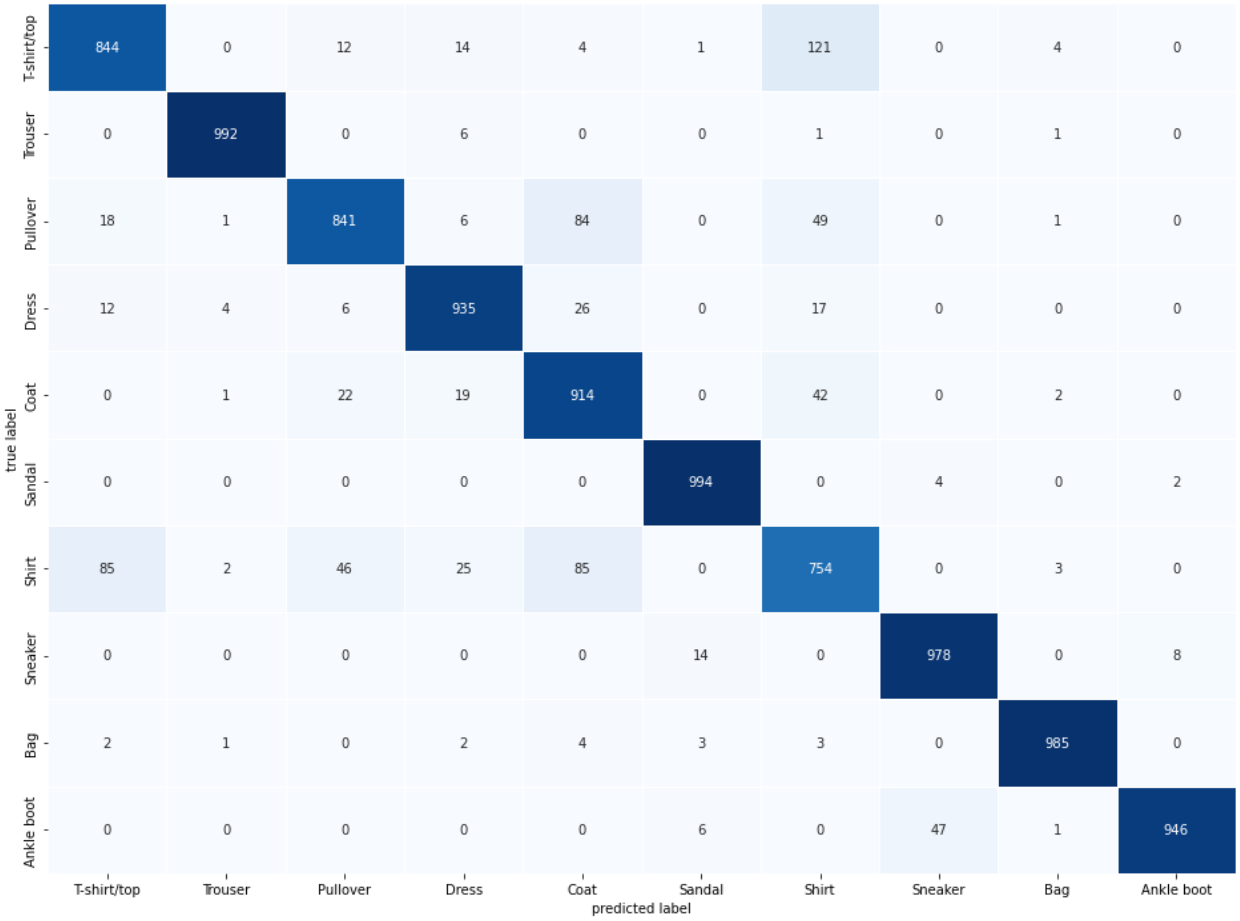
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results within 4 epochs. The resulting test data accuracy is 91.83%, just 0.07% less than Experiment 1's accuracy.

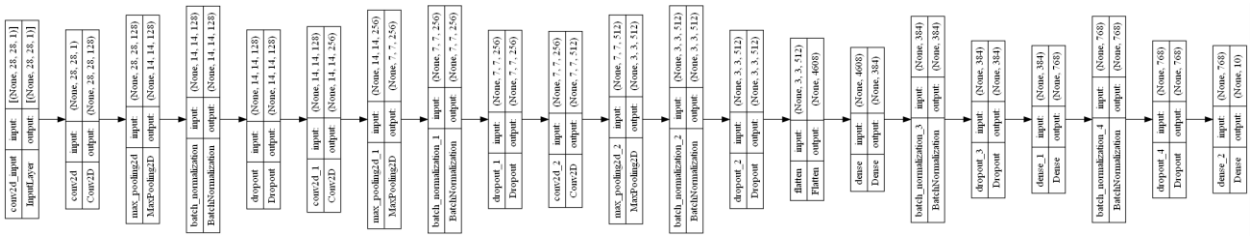
The experiment yields the following confusion matrix:



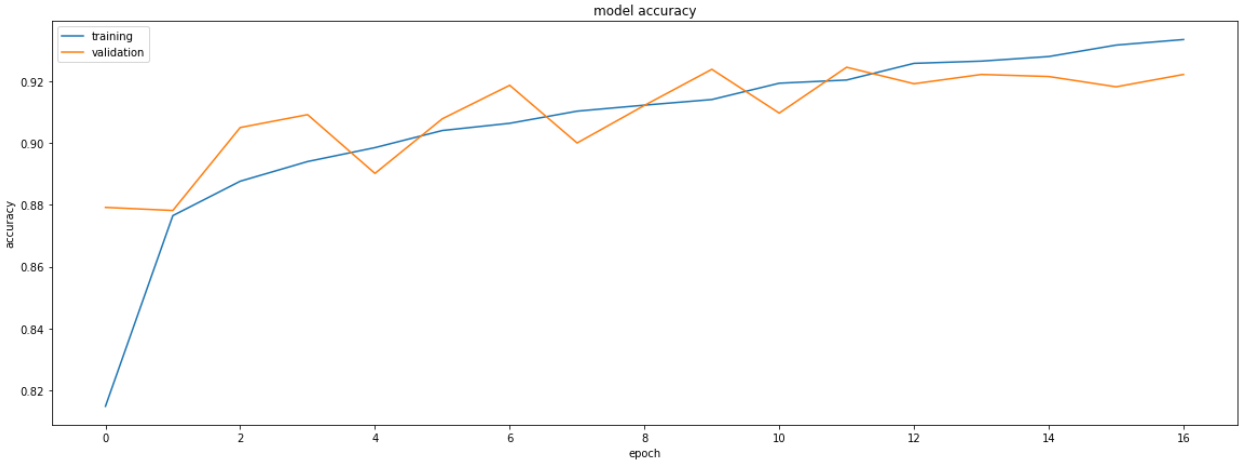
As expected, we see that this experiment also has issues classifying shirts as well.

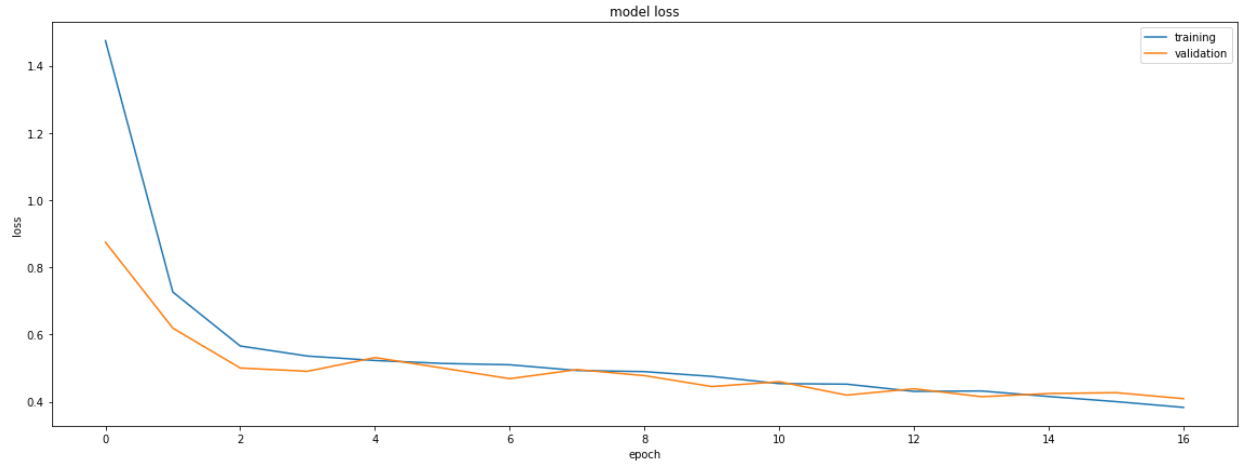
Experiment 3: CNN with 3 convolution/max pooling layers & 2 fully connected layers – Add Batch Normalization and Dropout (0.3)

Experiment 3 adds batch normalization and dropout (0.3) regularization processes to all convolution and fully connected layers. A diagram of the full architecture is below.



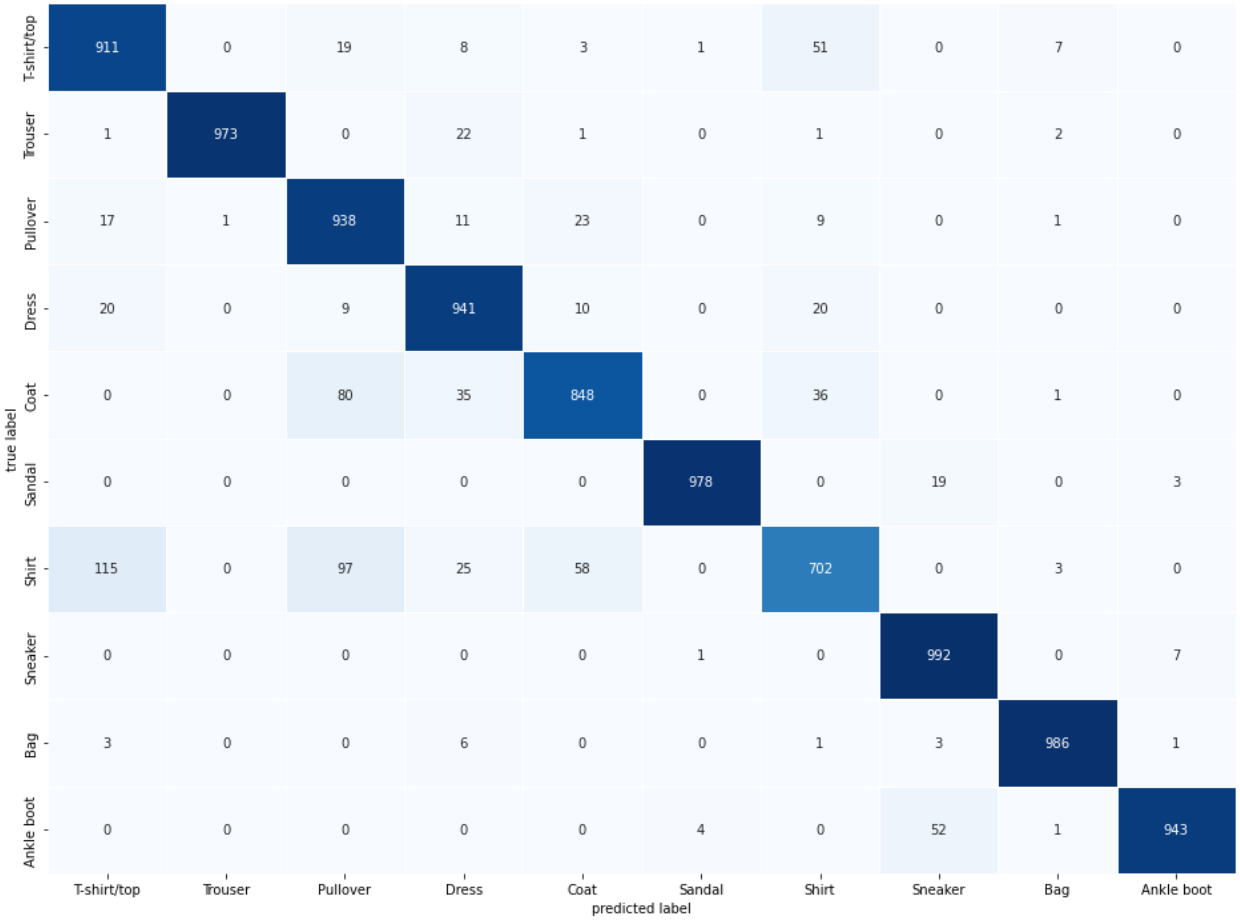
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results within 12 epochs. The resulting test data accuracy is 92.12%, exceeding Experiment 1's test accuracy. We see that adding regularization processes can improve model accuracy.

The experiment yields the following confusion matrix:

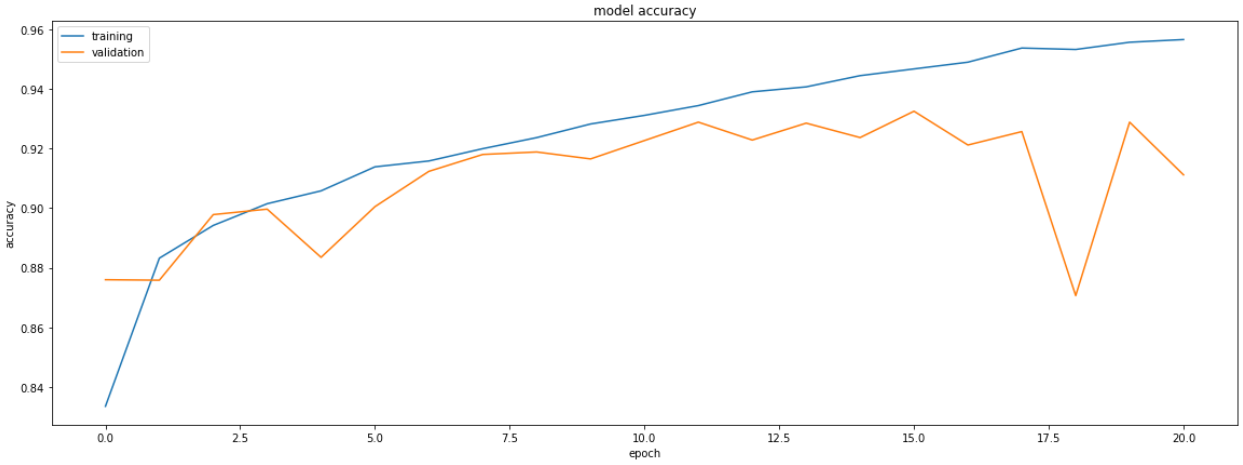


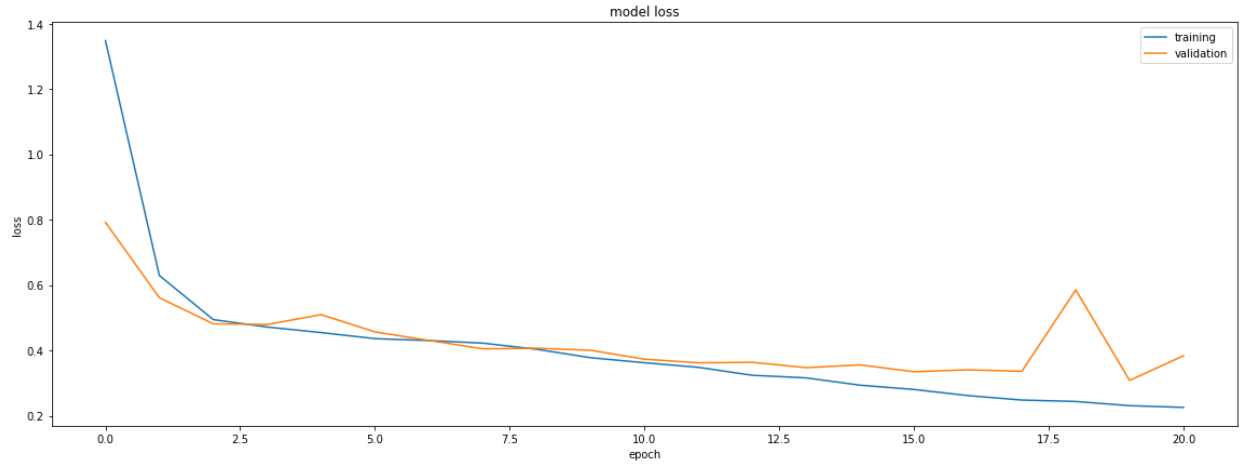
While the model can classify most images fairly well, it too has issues in classifying shirts.

Experiment 4: CNN with 3 convolution/max pooling layers & 2 fully connected layers – Change Dropout (0.2)

Experiment 4 is similar to Experiment 3 but changes the dropout rate to 0.2.

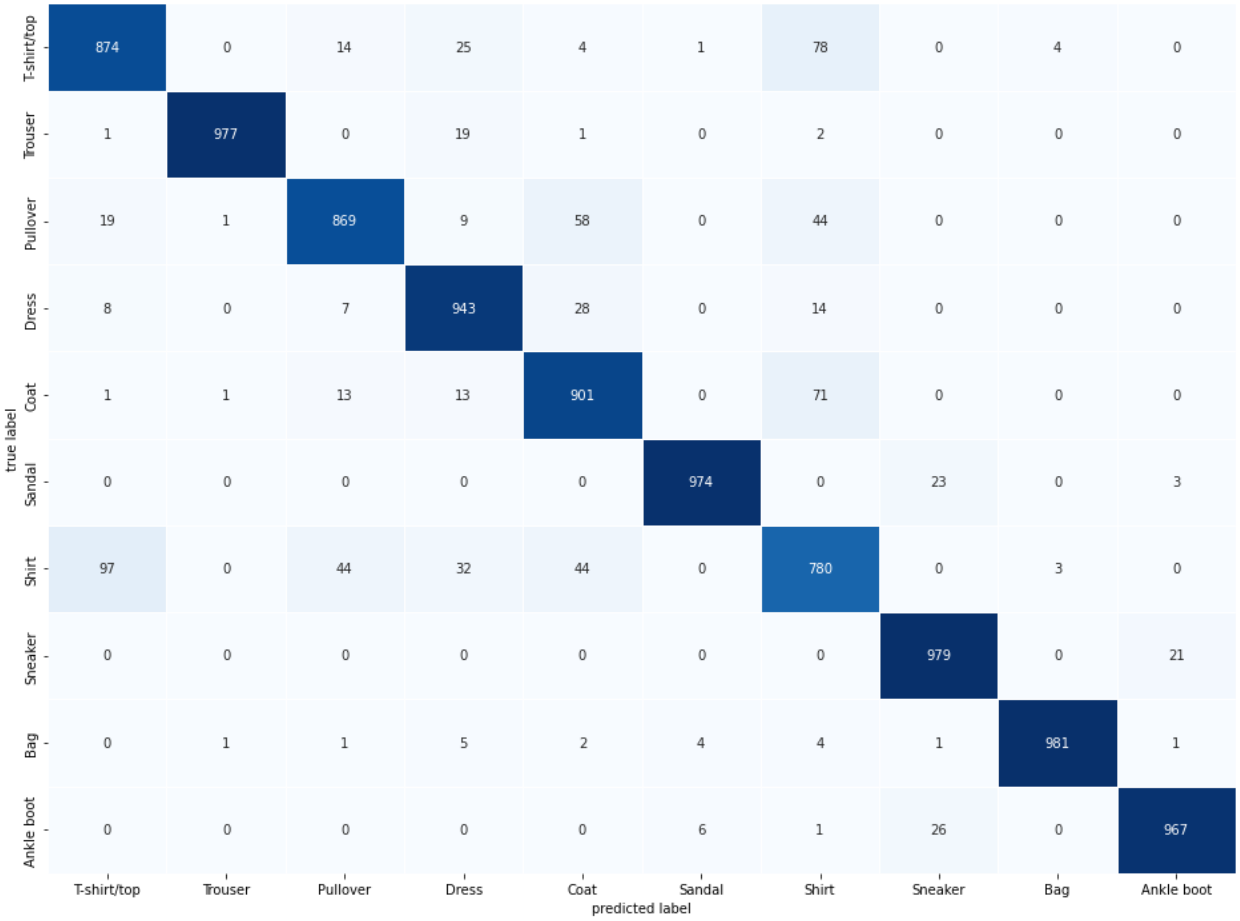
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results within 7 epochs. The resulting test data accuracy is 92.45%, exceeding Experiment 3's test accuracy.

The experiment yields the following confusion matrix:

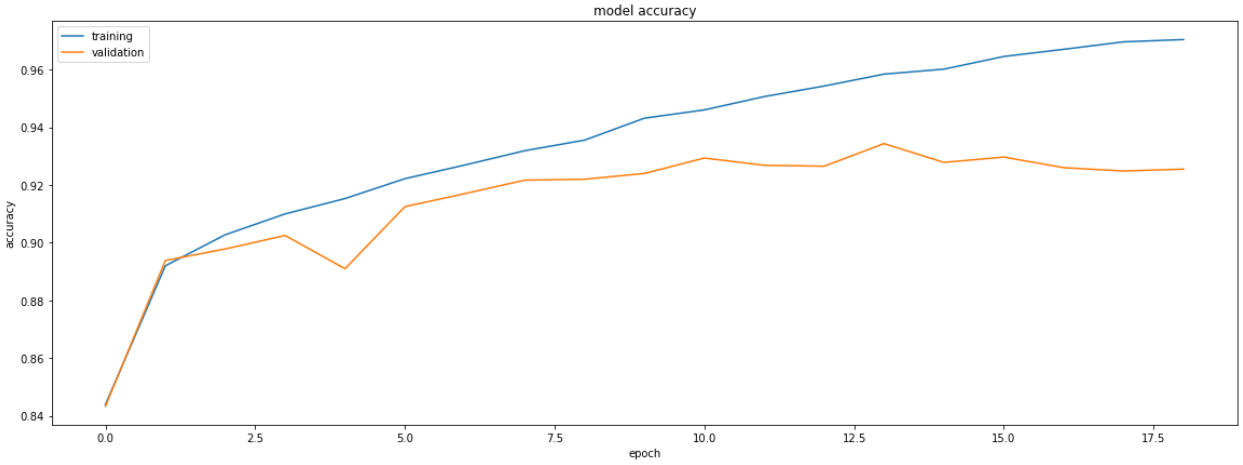


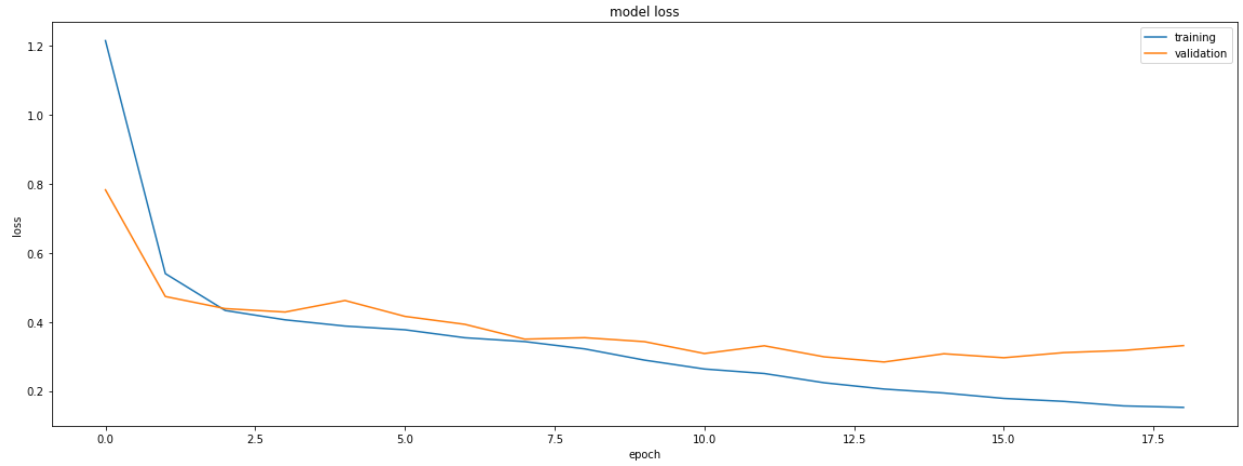
Shirt misclassification is still present compared to other categories.

Experiment 5: CNN with 3 convolution/max pooling layers & 2 fully connected layers – Change Dropout (0.1)

Experiment 5 is similar to Experiment 3 but changes the dropout rate to 0.1.

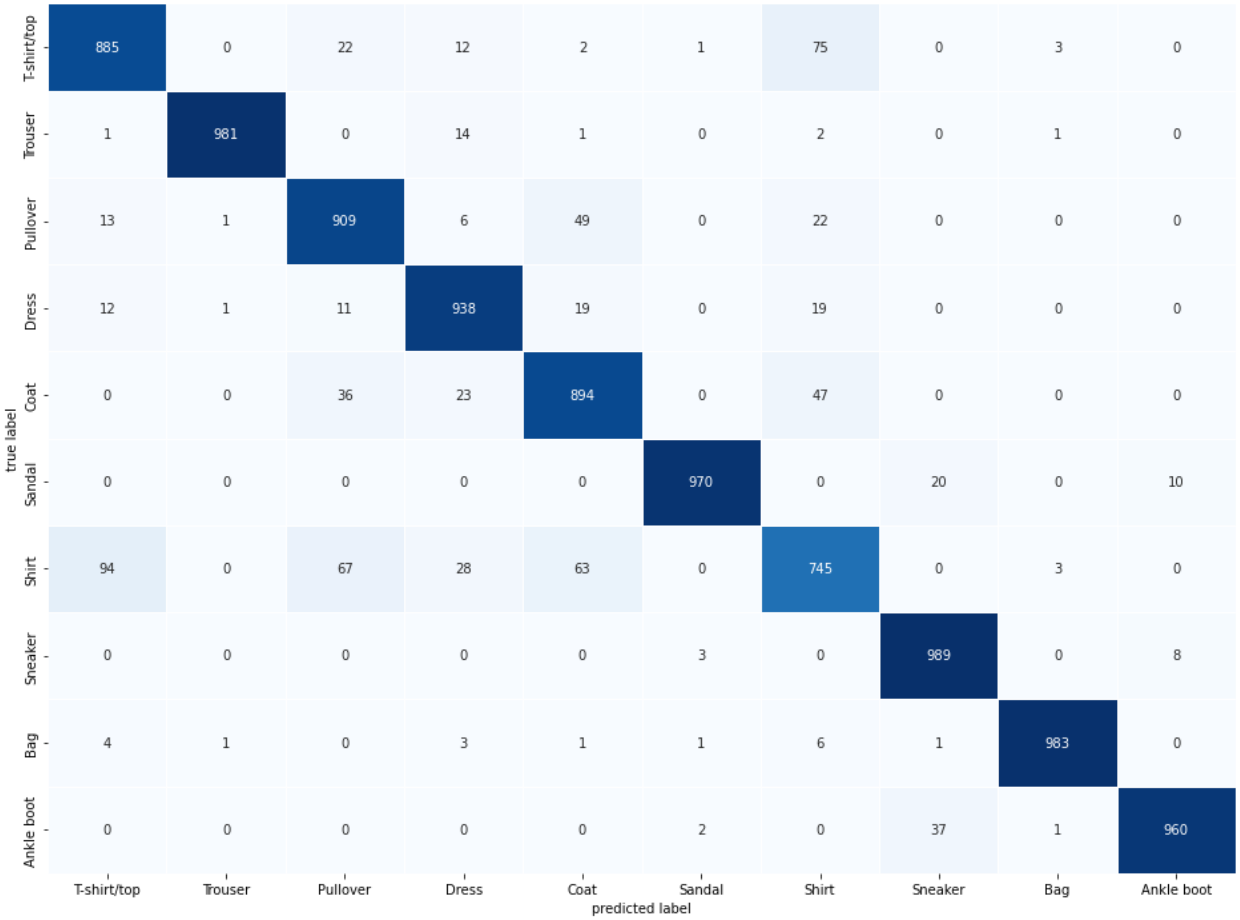
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results by 10 epochs. The resulting test data accuracy is 92.54%, exceeding Experiment 4's test accuracy.

The experiment yields the following confusion matrix:

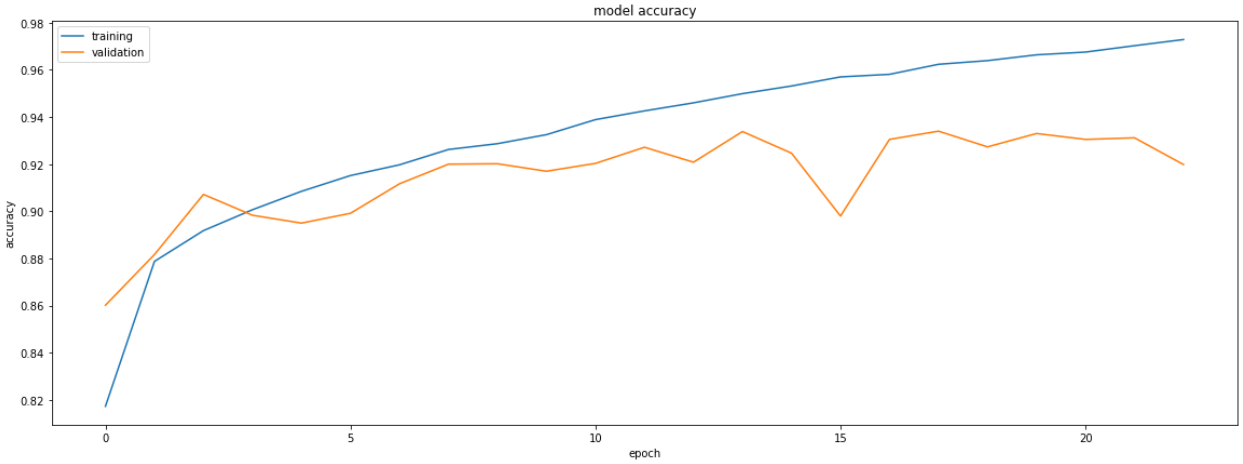


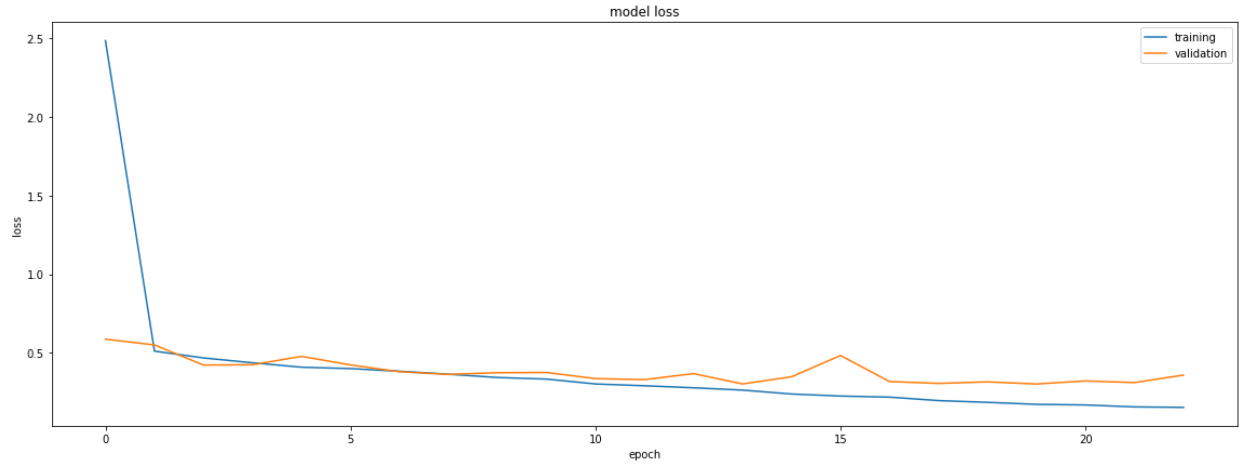
As usual, the model shows signs of shirt misclassification.

Experiment 6: CNN with 3 convolution/max pooling layers & 2 fully connected layers – Change Dropout (0.2), L2 Regularization (0.01)

Experiment 6 is similar to Experiment 4 but changes L2 regularization learning rate to 0.01.

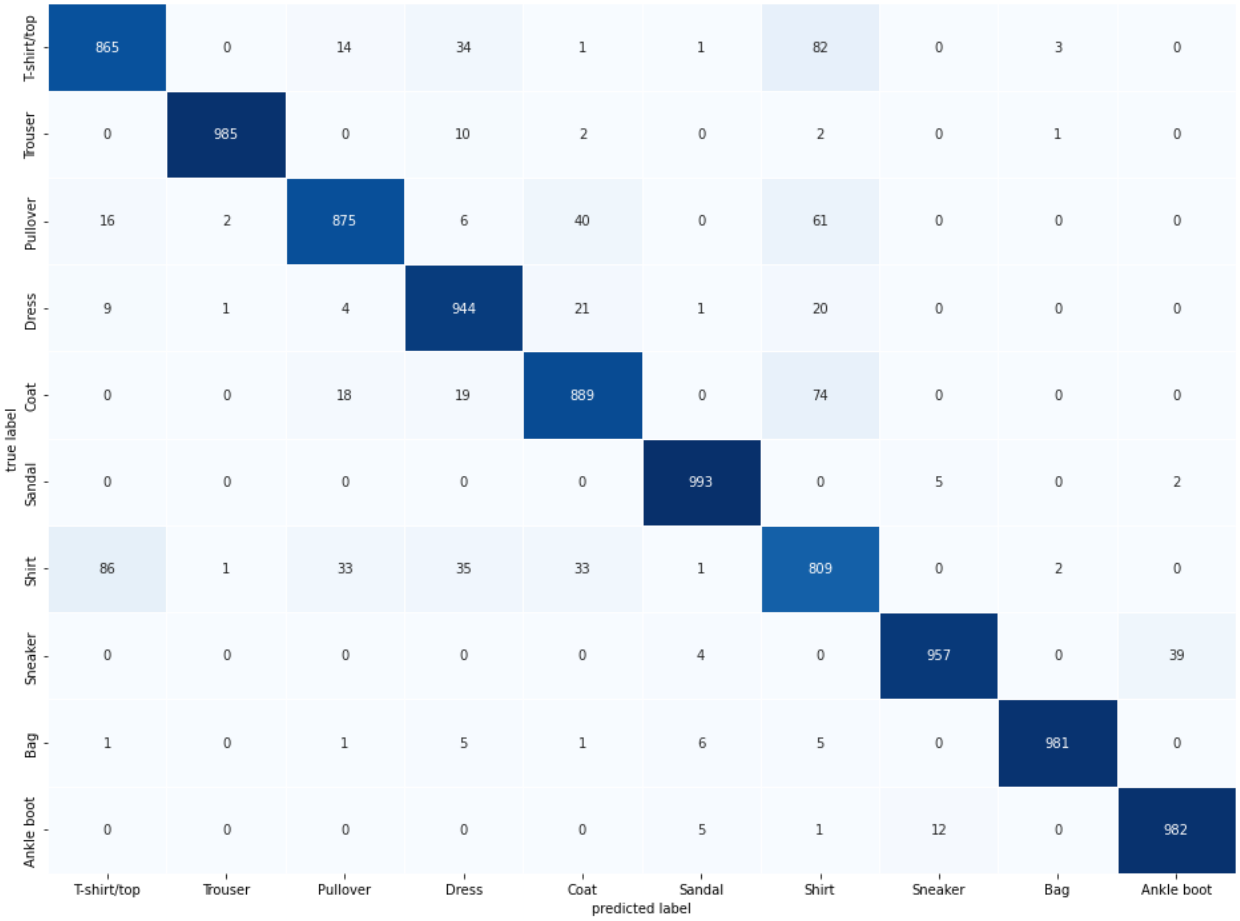
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after 14 epochs. The resulting test data accuracy is 92.8%, our best-performing model so far.

The experiment yields the following confusion matrix:

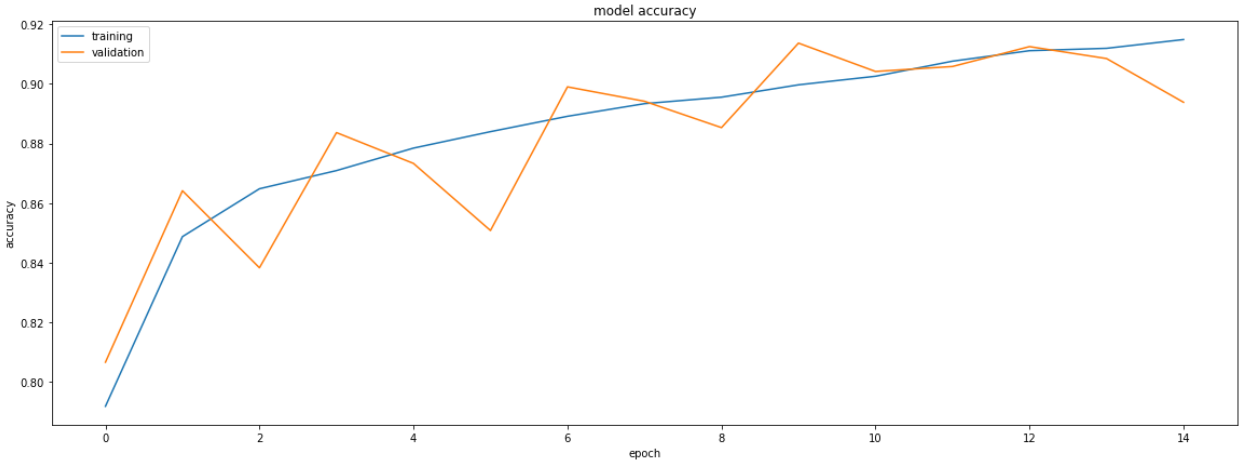


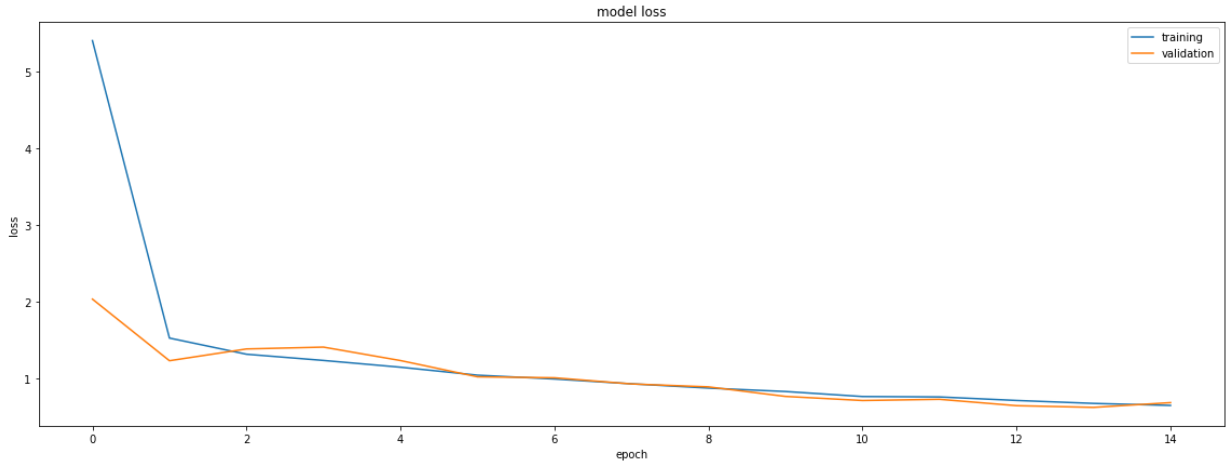
Shirt classification is somewhat improved compared to previous models, but still underperforms compared to other categories.

Experiment 7: CNN with 3 convolution/max pooling layers & 2 fully connected layers – Change L2 Regularization (0.1)

Experiment 7 is similar to Experiment 6 but changes the L2 regularization learning rate to 0.1.

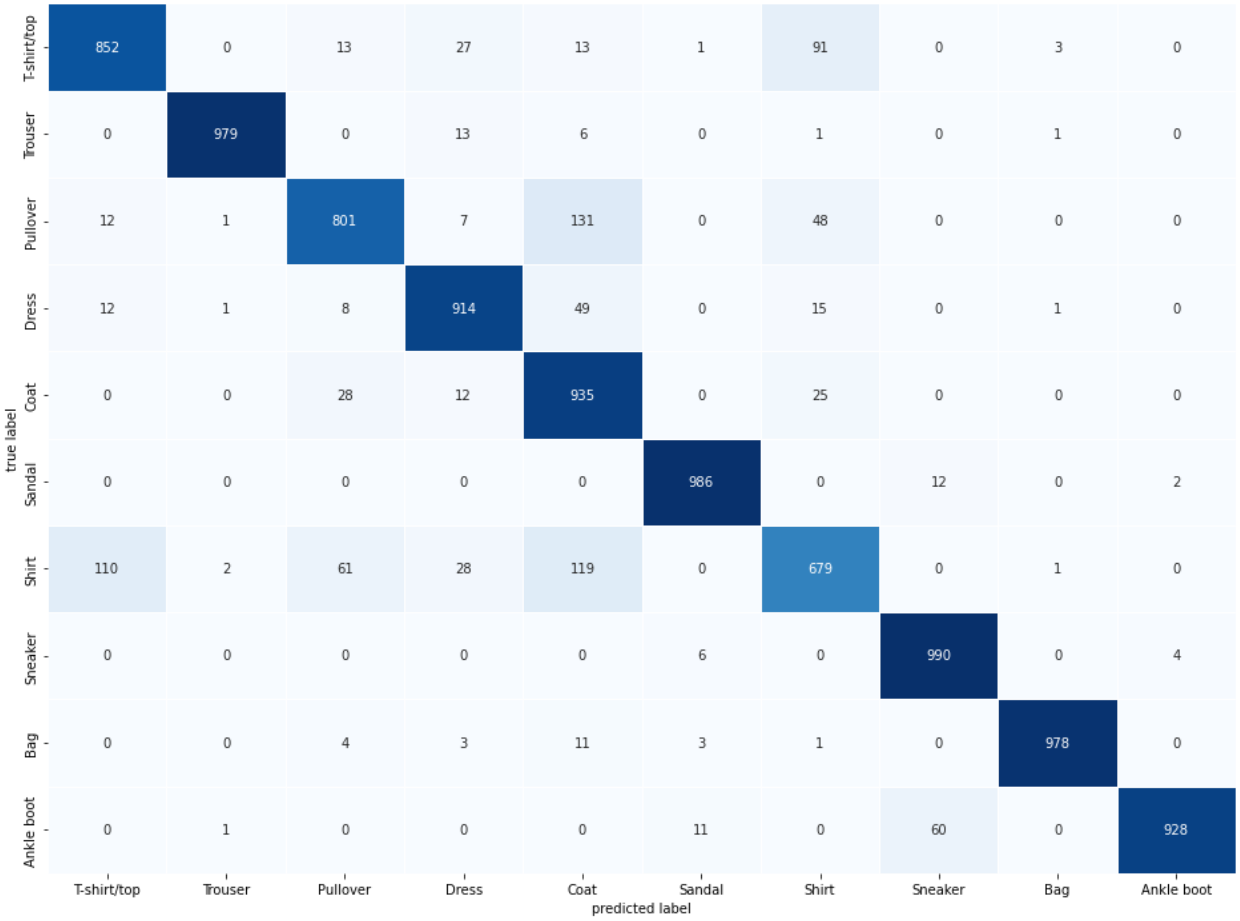
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after 12 epochs. The resulting test data accuracy is 91.9%. Compared to Experiment 6, our best-performing model so far, Experiment 7's accuracy is off by about 1%.

The experiment yields the following confusion matrix:

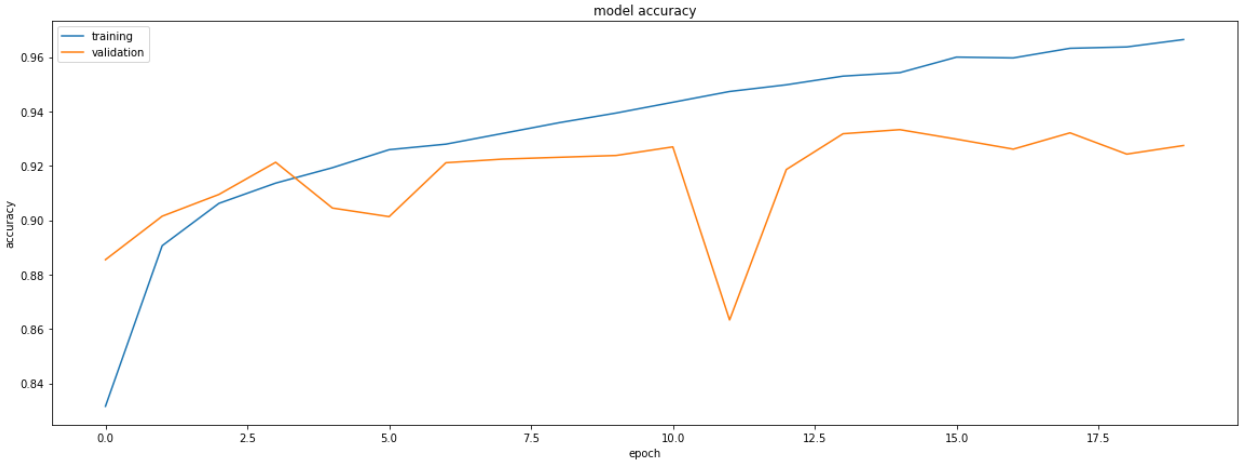


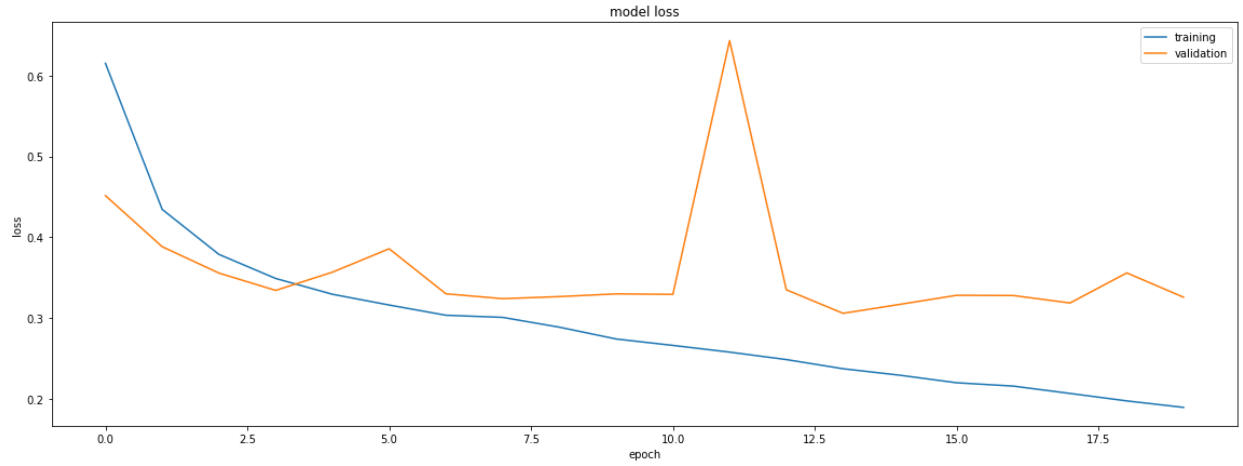
Experiment 7 seems to have more difficulty classifying shirts compared to previous models.

Experiment 8: CNN with 3 convolution/max pooling layers & 2 fully connected layers – Change L2 Regularization (0.0001)

Experiment 8 is similar to Experiment 6 but changes the L2 regularization learning rate to 0.0001.

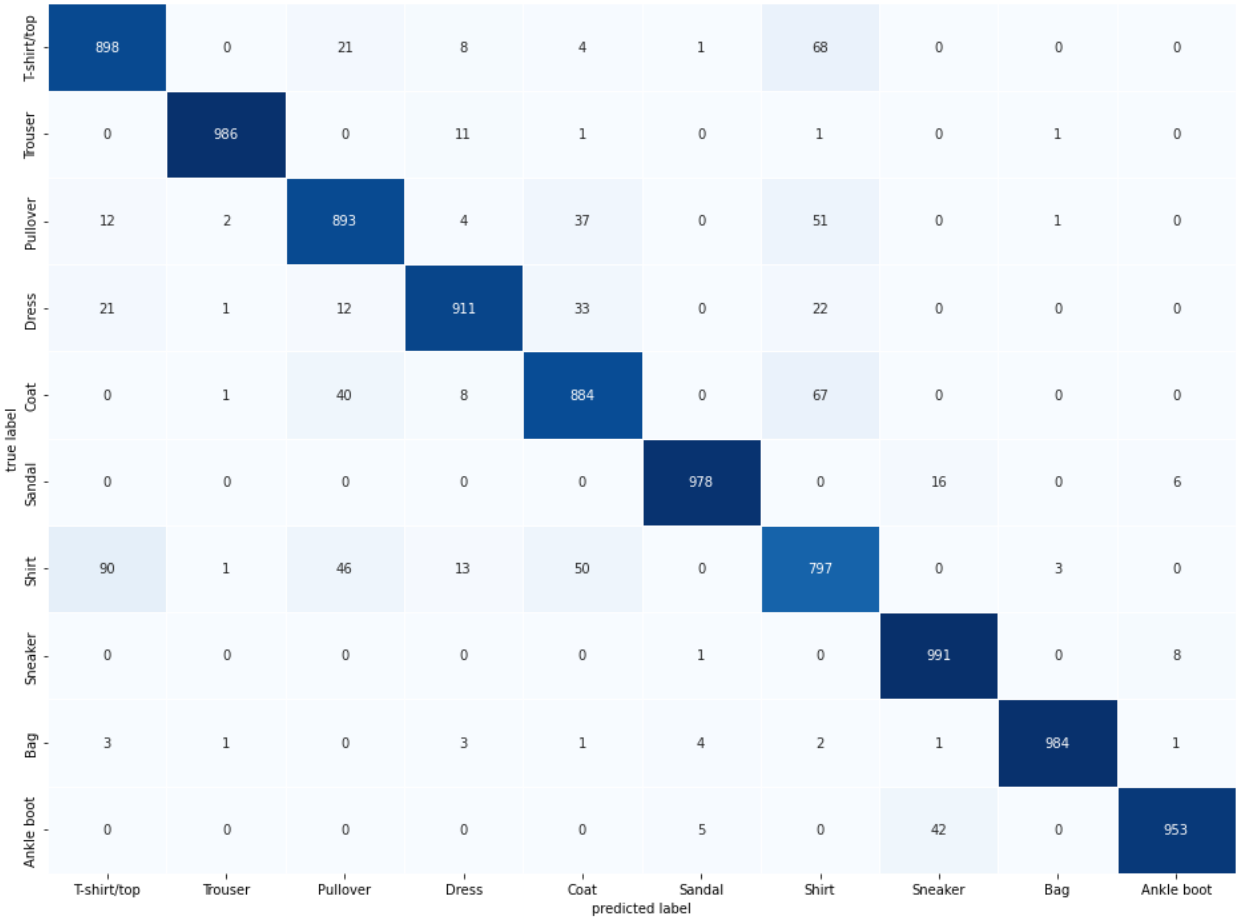
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after about 9 epochs. The resulting test data accuracy is 92.75%, which misses out Experiment 6's accuracy by just 0.05%.

The experiment yields the following confusion matrix:

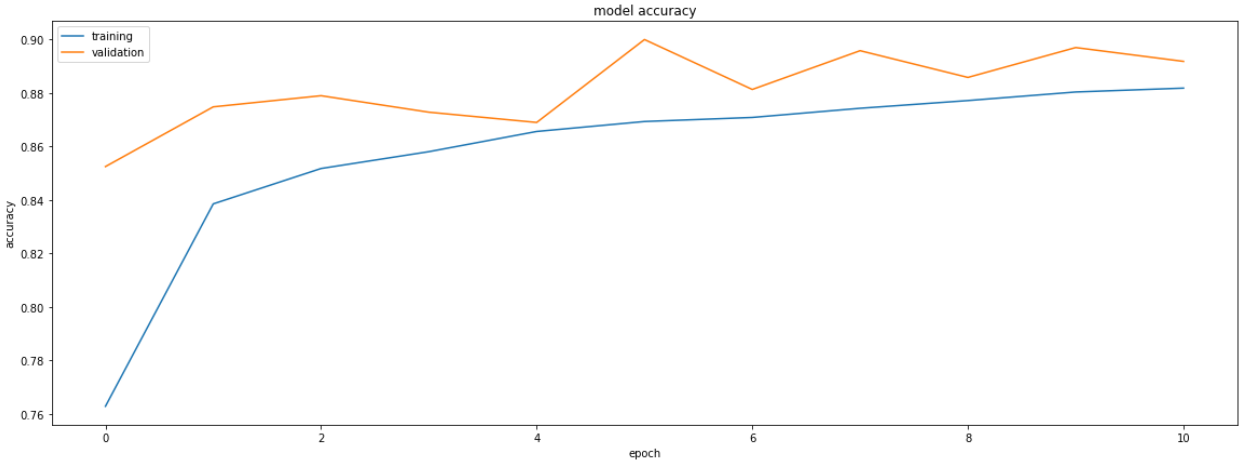


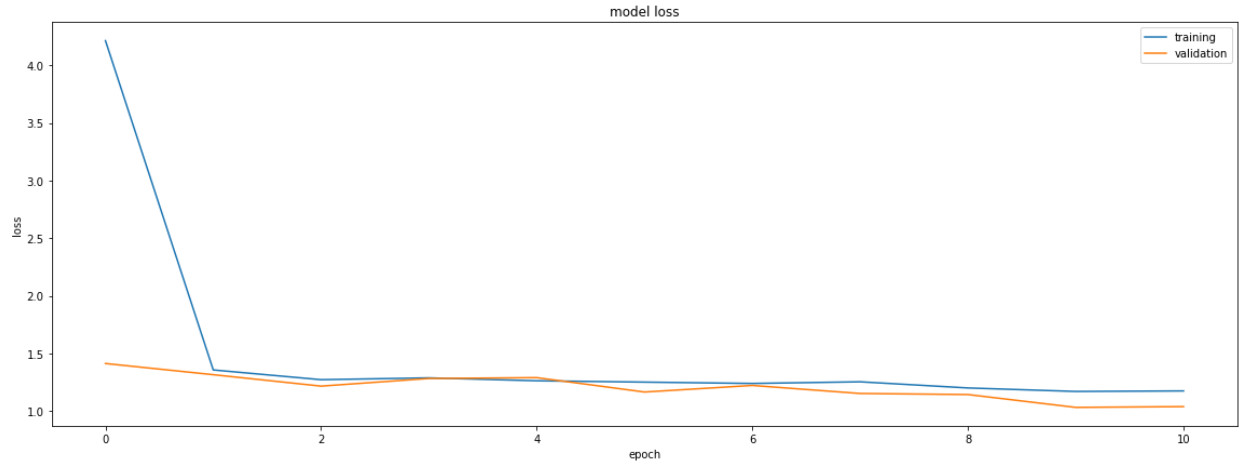
Shirt misclassification is still somewhat present in this model, but not as bad compared to most of the previous models.

Experiment 9: CNN with 3 convolution/max pooling layers & 2 fully connected layers – Change L2 Regularization (0.01), Dropout (0.5)

Experiment 9 is similar to Experiment 6, but changes the dropout rate to 0.5.

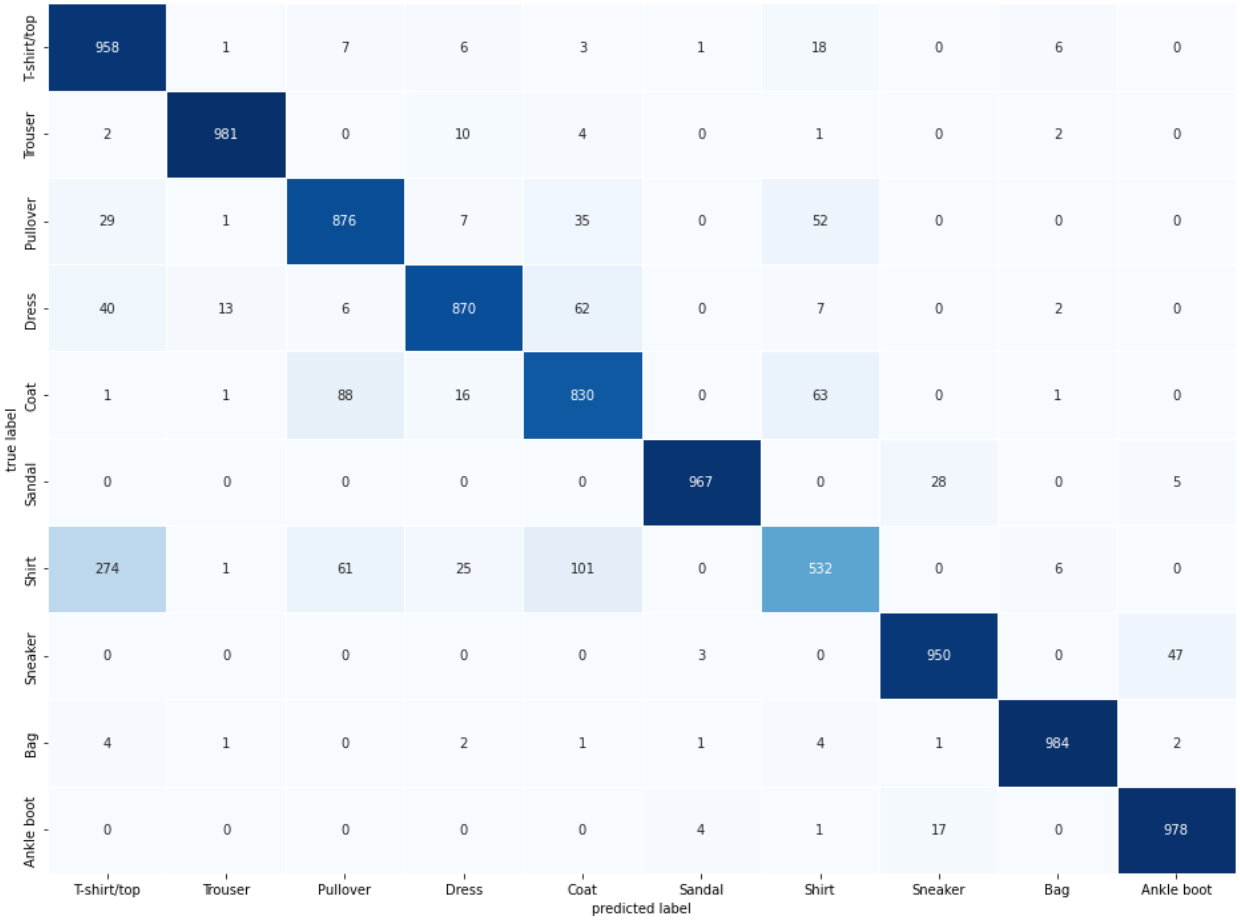
Training and validation accuracy and cross entropy loss charts are below.





While we see validation results keep up fairly well with training results, the whole training process only lasts for 10 epochs. The resulting test accuracy is 89.3%, well off our best test accuracy of 92.8% by 2.5%. We find that using larger dropout rates in this CNN model architecture does not result in better test accuracy.

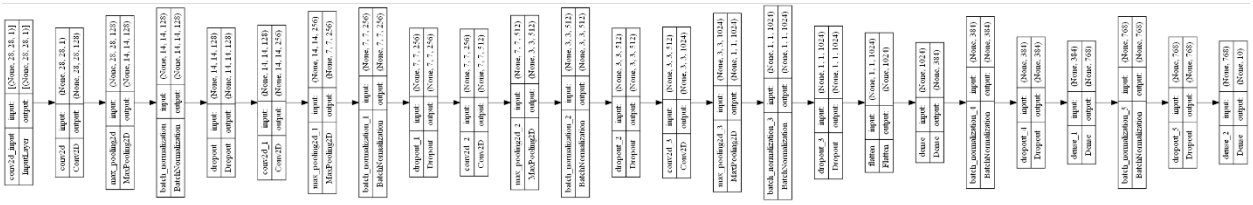
The experiment yields the following confusion matrix:



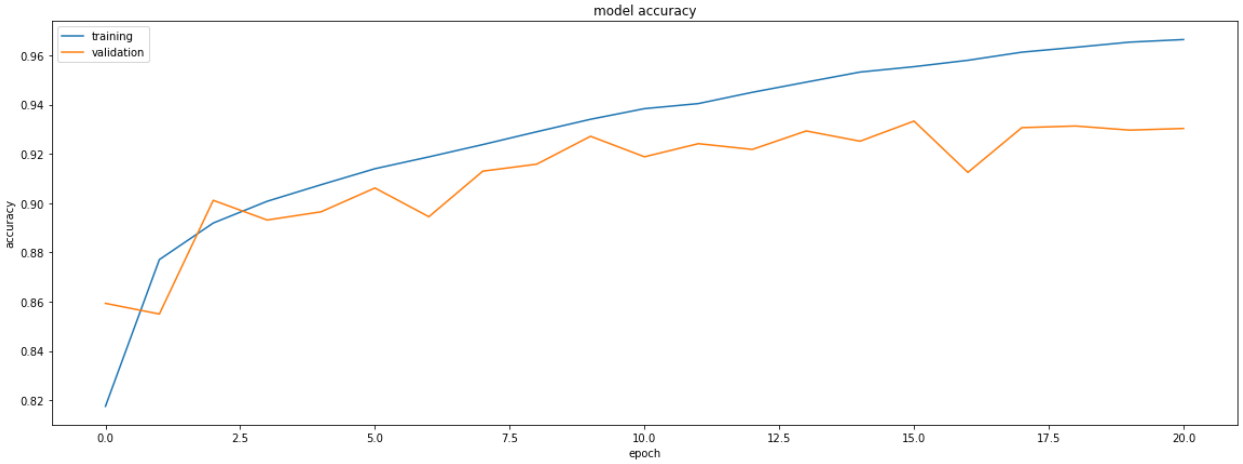
We see that using a larger dropout rate results in a worse shirt categorization process.

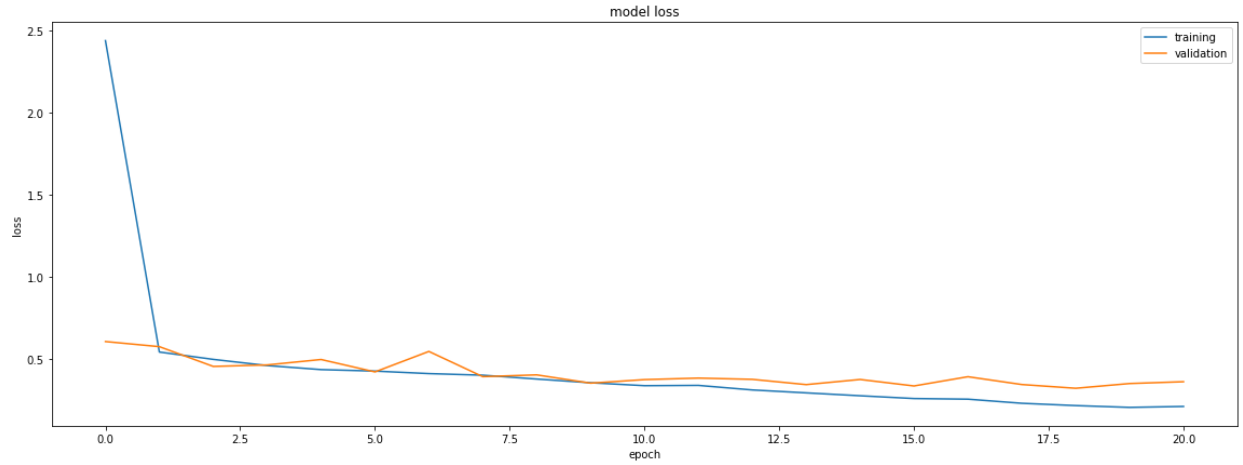
Experiment 10: CNN with 4 convolution/max pooling layers & 2 fully connected layers

Experiment 10 is similar to Experiment 6, our best-performing model so far, but adds another 3x3 convolutional layer which uses 1024 filters. A diagram of the full architecture is below.



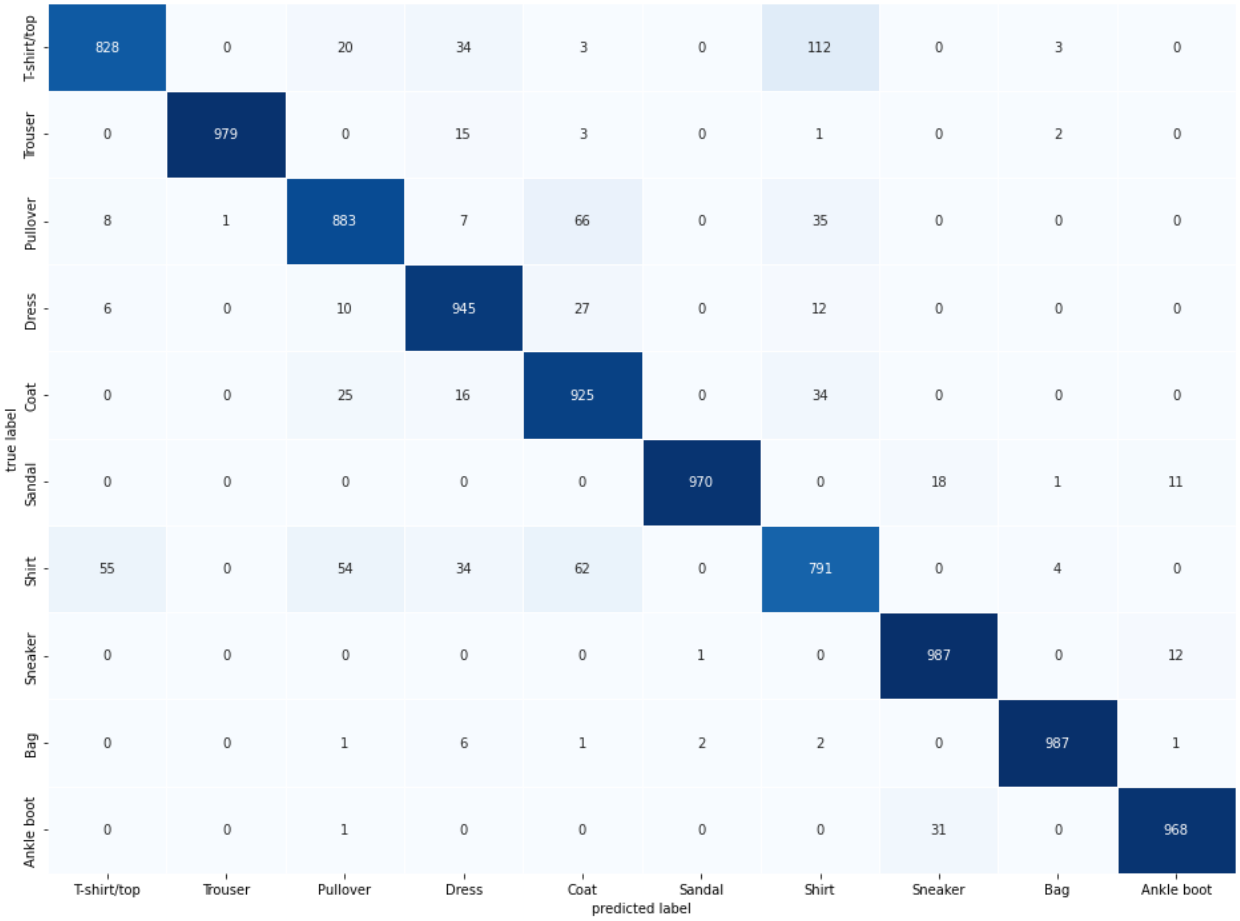
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after 5 epochs. The resulting test data accuracy is 92.6%, which doesn't exceed the accuracy performance of Experiment 6, despite the added convolutional layer. Since Fashion-MNIST is a dataset which contains small images, there could be a ceiling of convolutional or max pooling layers, and if exceeded, could potentially lose important feature information (Nocentini et al. 2022). Adding an additional convolutional and max pooling layer in this case shows diminishing returns despite the added model complexity.

The experiment yields the following confusion matrix:



Like in previous experiments, this model shows some signs of shirt misclassification.

The results of our convolutional model experiments in this section are summarized in the table below, with the best one highlighted (Experiment 6).

Experiment	Conv2d Layers	DNNs	L2 Reg	Batch Norm	Dropout	Test Accuracy	Test Loss	Precision	Recall	F1-Score	RMSE
1	3	2	None	None	None	0.919	0.2288	0.92	0.92	0.92	1.124
2	3	2	**0.001**	None	None	0.9183	0.2666	0.92	0.92	0.92	1.096
3	3	2	0.001	**Yes**	**0.3**	0.9212	0.4229	0.92	0.92	0.92	1.045
4	3	2	0.001	Yes	**0.2**	0.9245	0.331	0.92	0.92	0.92	1.03
5	3	2	0.001	Yes	**0.1**	0.9254	0.3061	0.93	0.93	0.93	1.022
6	**3**	**2**	**0.01**	**Yes**	**0.2**	**0.928**	**0.3182**	**0.93**	**0.93**	**0.93**	**1.019**
7	3	2	**0.1**	Yes	0.2	0.9188	0.4982	0.91	0.9	0.9	1.132
8	3	2	**0.0001**	Yes	0.2	0.9275	0.332	0.93	0.93	0.93	0.995
9	3	2	0.01	Yes	**0.5**	0.8926	1.0441	0.89	0.89	0.89	1.272
10	**4**	2	0.01	Yes	0.2	0.9263	0.3276	0.93	0.93	0.93	0.983

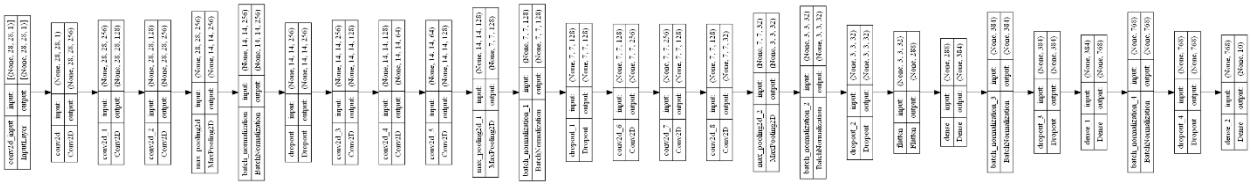
We were able to achieve a test accuracy of 92.8% in our group of convolutional models by tuning various hyperparameters, in which Experiment 6 was our most optimal model. While the last experiment in this section added another convolution layer we started to see diminishing returns as its test accuracy did not exceed Experiment 6.

Section Two: Grouped Convolutional Models (3x3 Stacks)

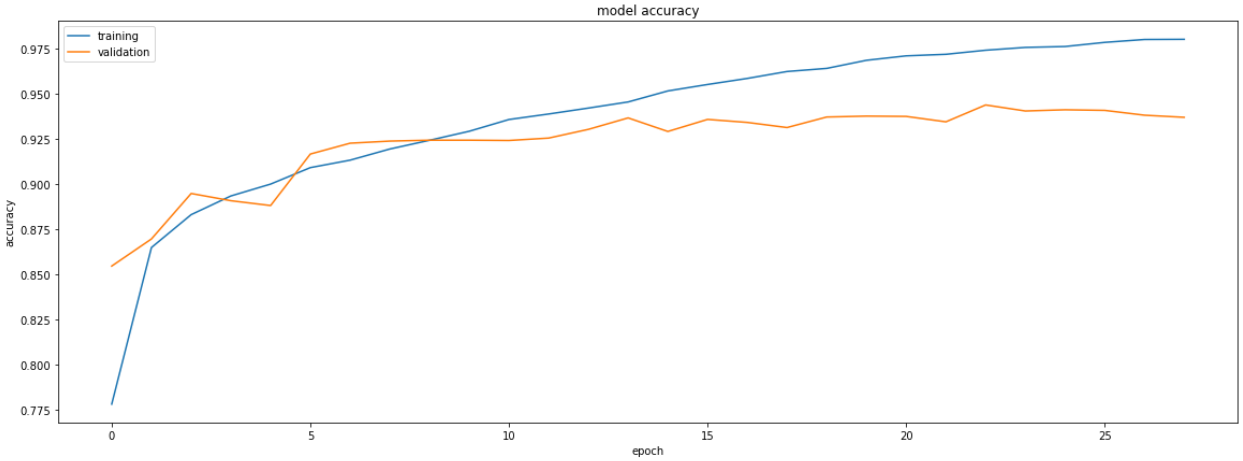
Experiment 1: CNN with 3 groups of 3 convolution layers & 2 fully connected layers – Add Batch Normalization, Dropout (0.2), L2 Regularization (0.01)

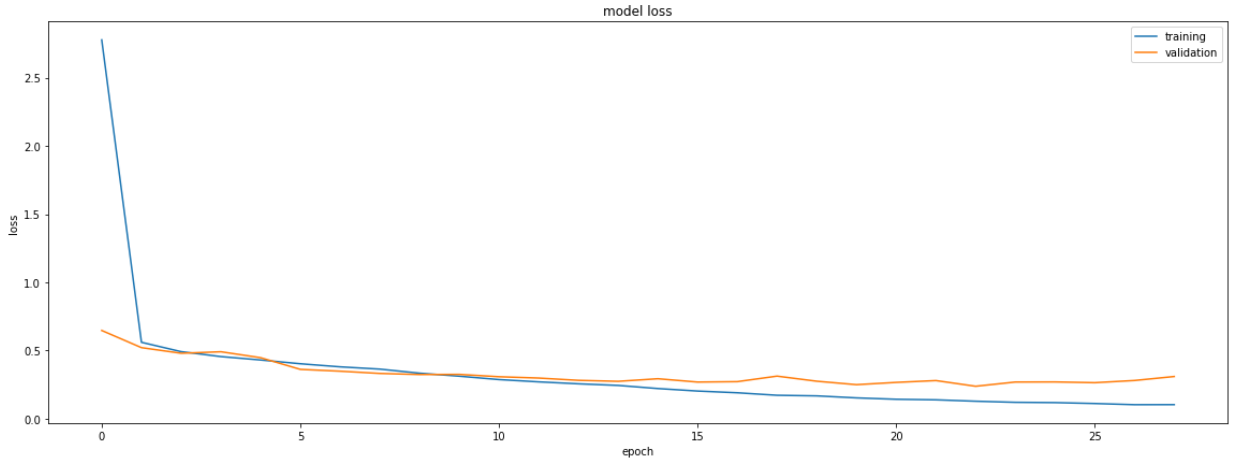
We model a CNN consisting of 3 groups of 3x3 convolutional layers, with each group separated by a 2x2 max pooling layer. The first group contains layers with 256, 128, and 256 filters respectively, the second group contains layers with 128, 64, and 128 filters respectively, and the third group contains layers with 256, 128, and 32 filters respectively. The convolutional base then feeds into a DNN containing 2 fully connected layers, with the first layer containing 384 units and the second layer containing 768 units, followed by the 10-way softmax classification output layer. Batch normalization and dropout (0.2) processes follow all max pooling layers and fully connected layers. All fully connected layers utilize L2 regularization configured with a learning rate of 0.01.

The model diagram below shows the full architecture used by all models in this section.



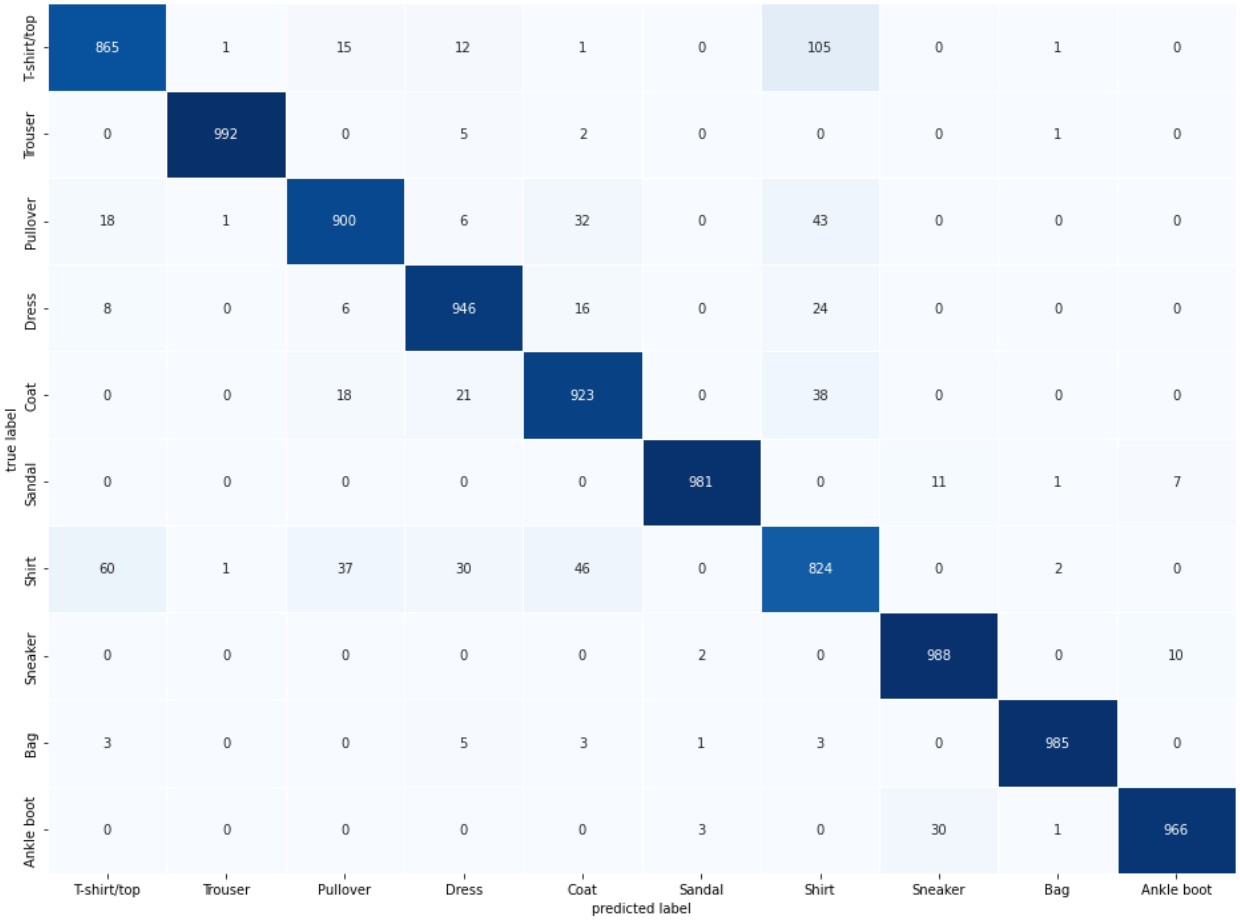
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after 7 epochs. The resulting test data accuracy is 93.7%, our best-performing model so far, outperforming all models in the first section.

The experiment yields the following confusion matrix:

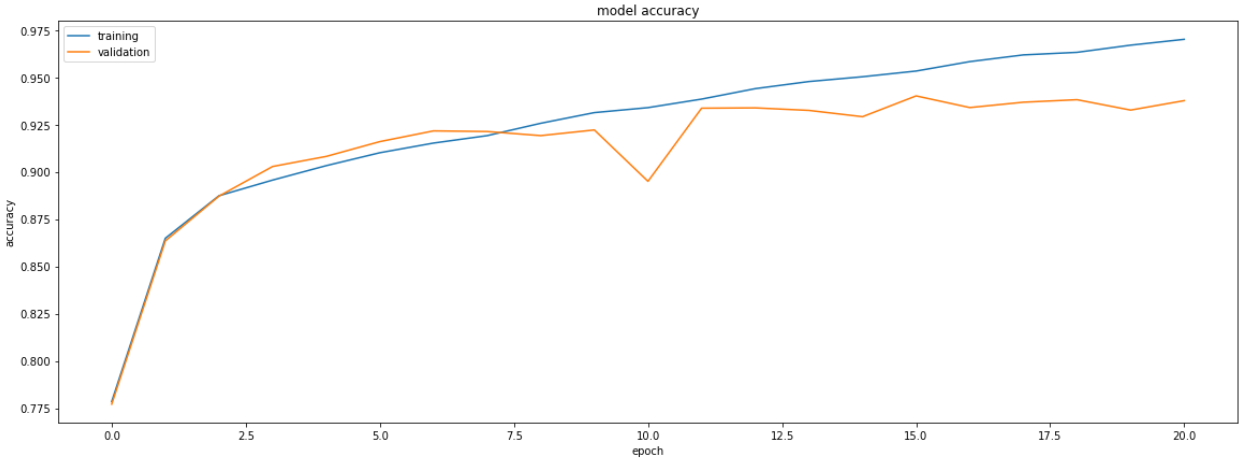


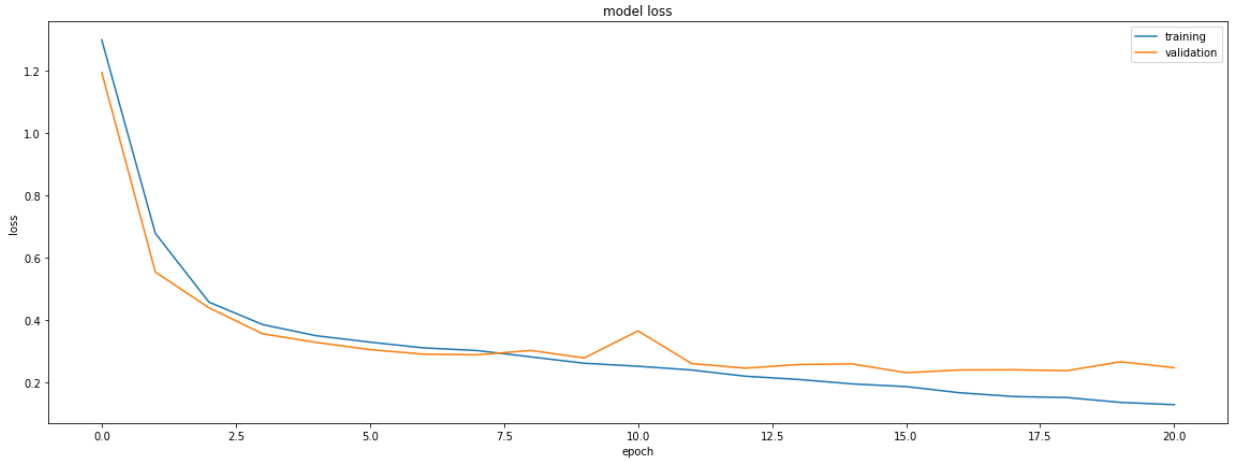
While the model is able to classify most garments in the test dataset with well over 90% precision, it still has issues with classifying shirts with 79% precision.

Experiment 2: CNN with 3 groups of 3 convolution layers & 2 fully connected layers – Change L2 Regularization (0.001)

Experiment 2 follows the same architecture as Experiment 1 but changes L2 Regularization to a learning rate of 0.001.

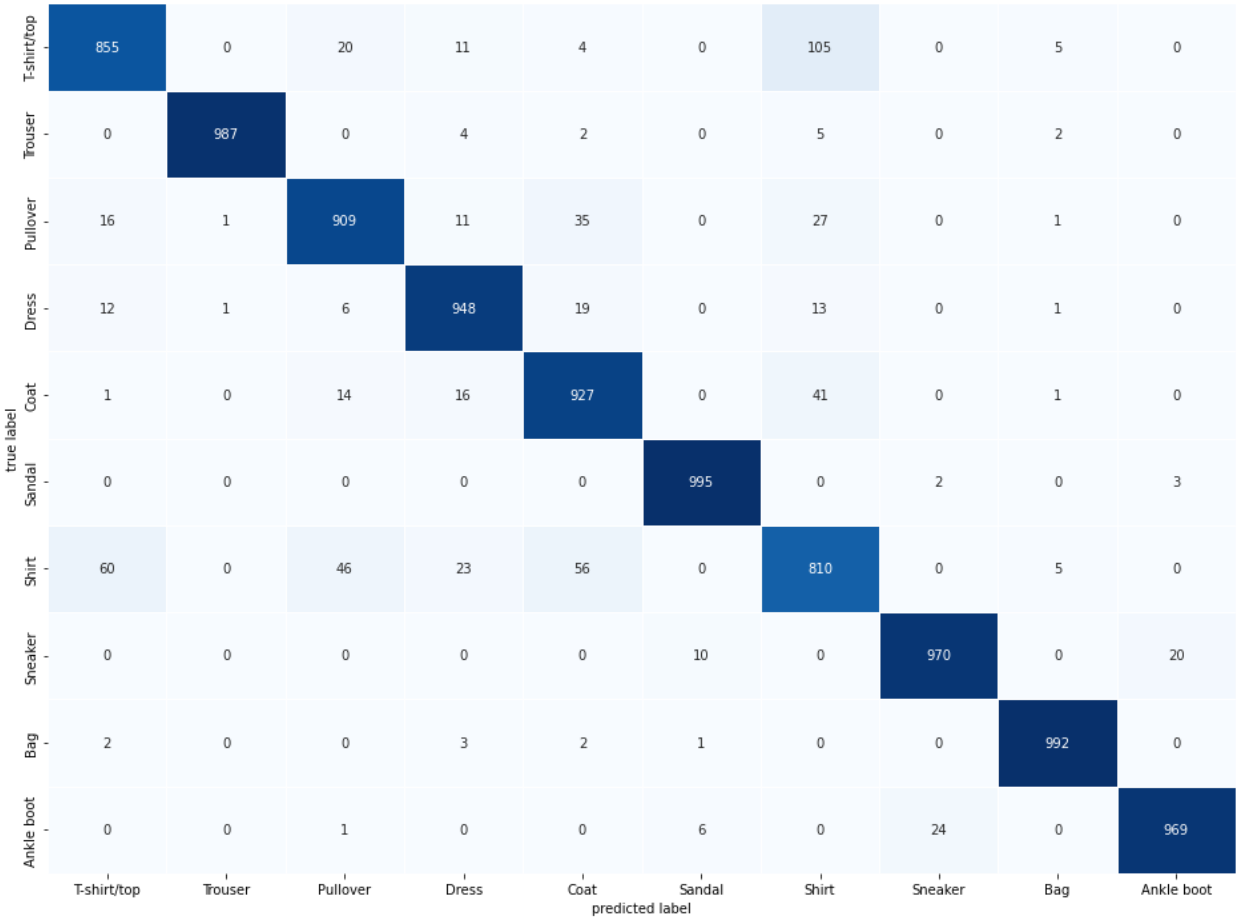
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after about 8 epochs. The resulting test data accuracy is 93.6%, which is just lower than Experiment 1's test accuracy by 0.1%.

The experiment yields the following confusion matrix:

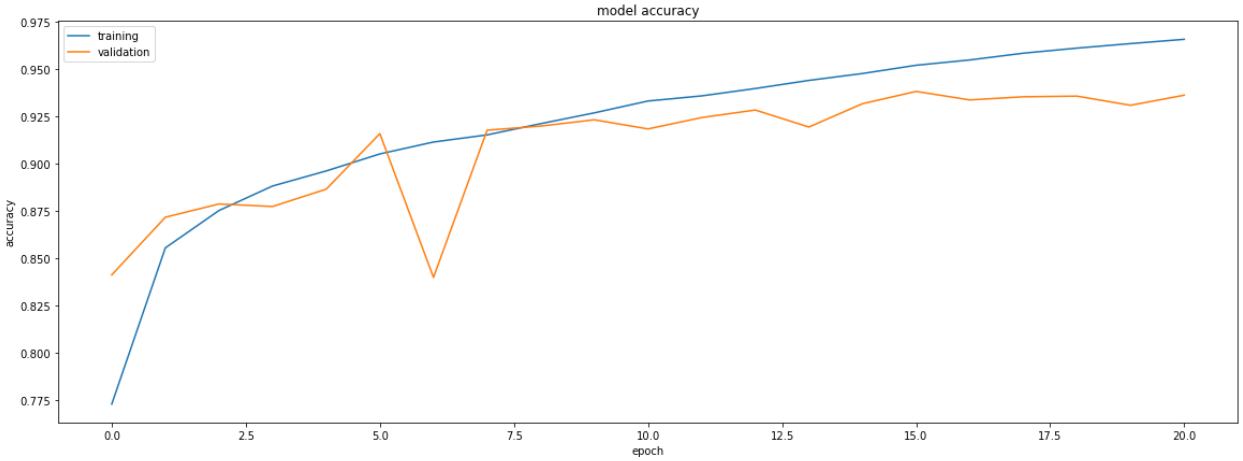


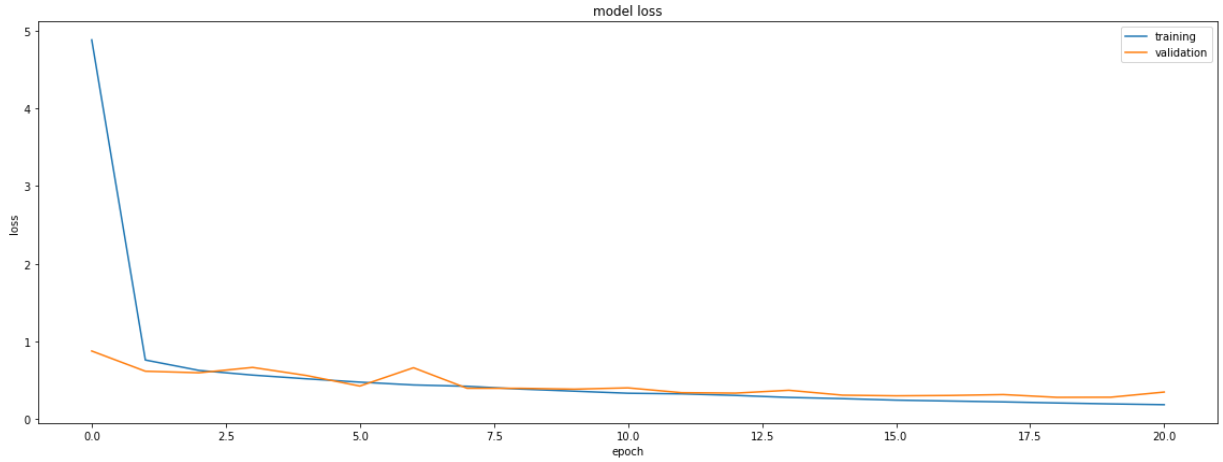
While the model can correctly classify most images well, like Experiment 1 it too doesn't classify tops and shirt categories compared to other categories.

Experiment 3: CNN with 3 groups of 3 convolution layers & 2 fully connected layers – Change L2 Regularization (0.1)

Experiment 3 follows the same architecture as Experiment 1 but changes L2 Regularization to a learning rate of 0.1.

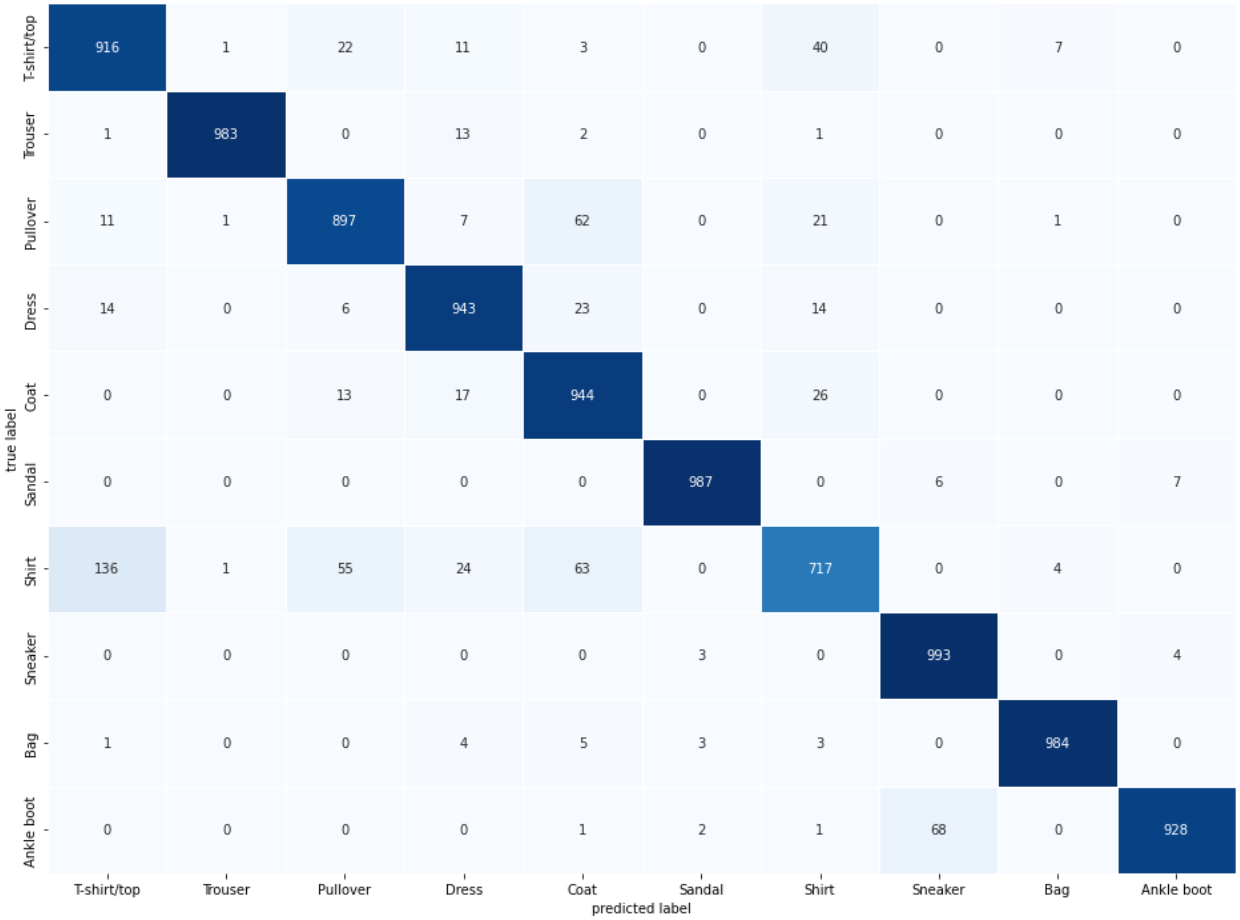
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after about 9 epochs. The resulting test data accuracy is 92.9%, which is just lower than Experiment 2's test accuracy by about 0.7%.

The experiment yields the following confusion matrix:

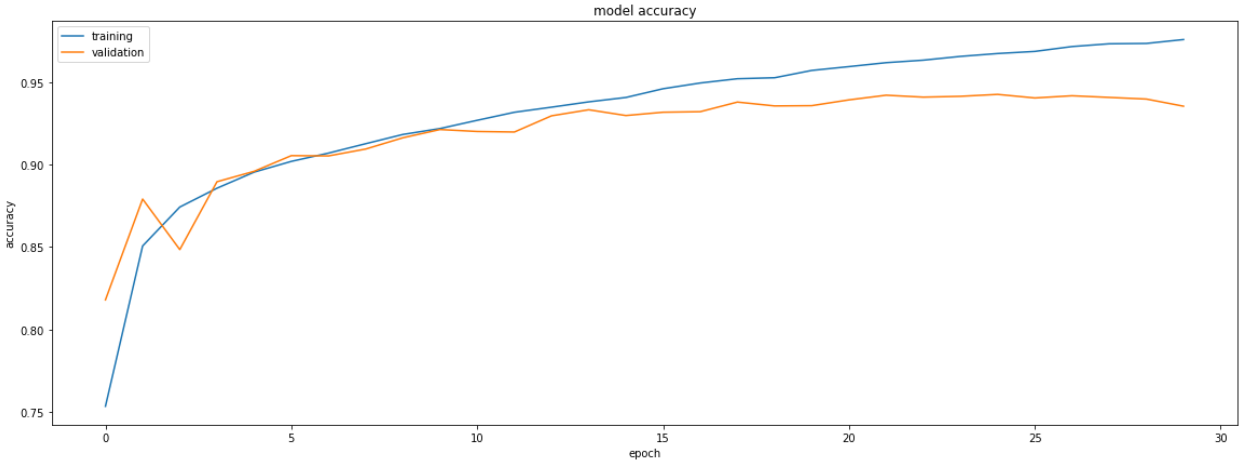


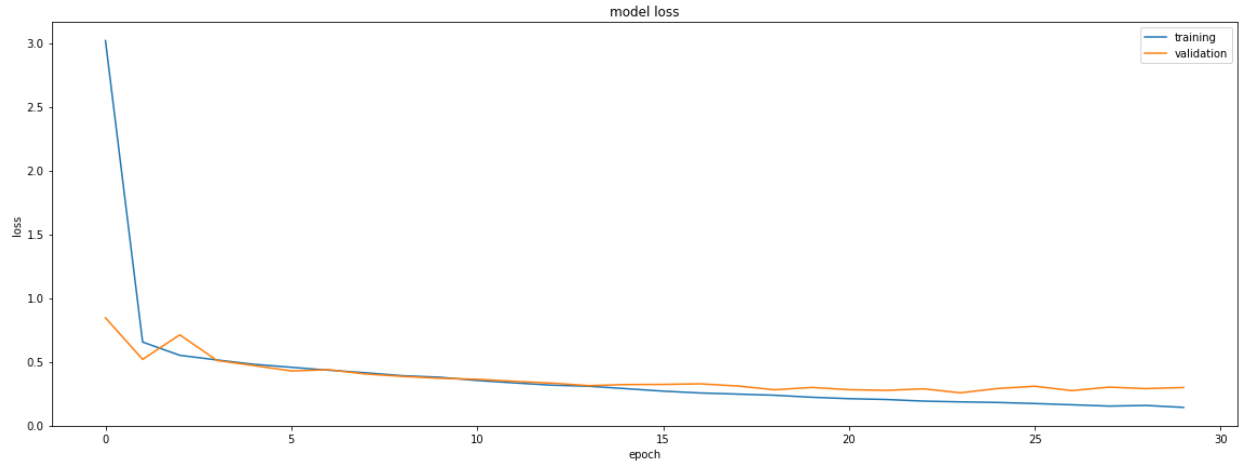
The theme of shirt misclassification continues in this experiment despite its high overall test accuracy.

Experiment 4: CNN with 3 groups of 3 convolution layers & 2 fully connected layers – Change L2 Regularization (0.2), Dropout (0.3)

Experiment 4 follows the same architecture as Experiment 1 but changes dropout rate to 0.3.

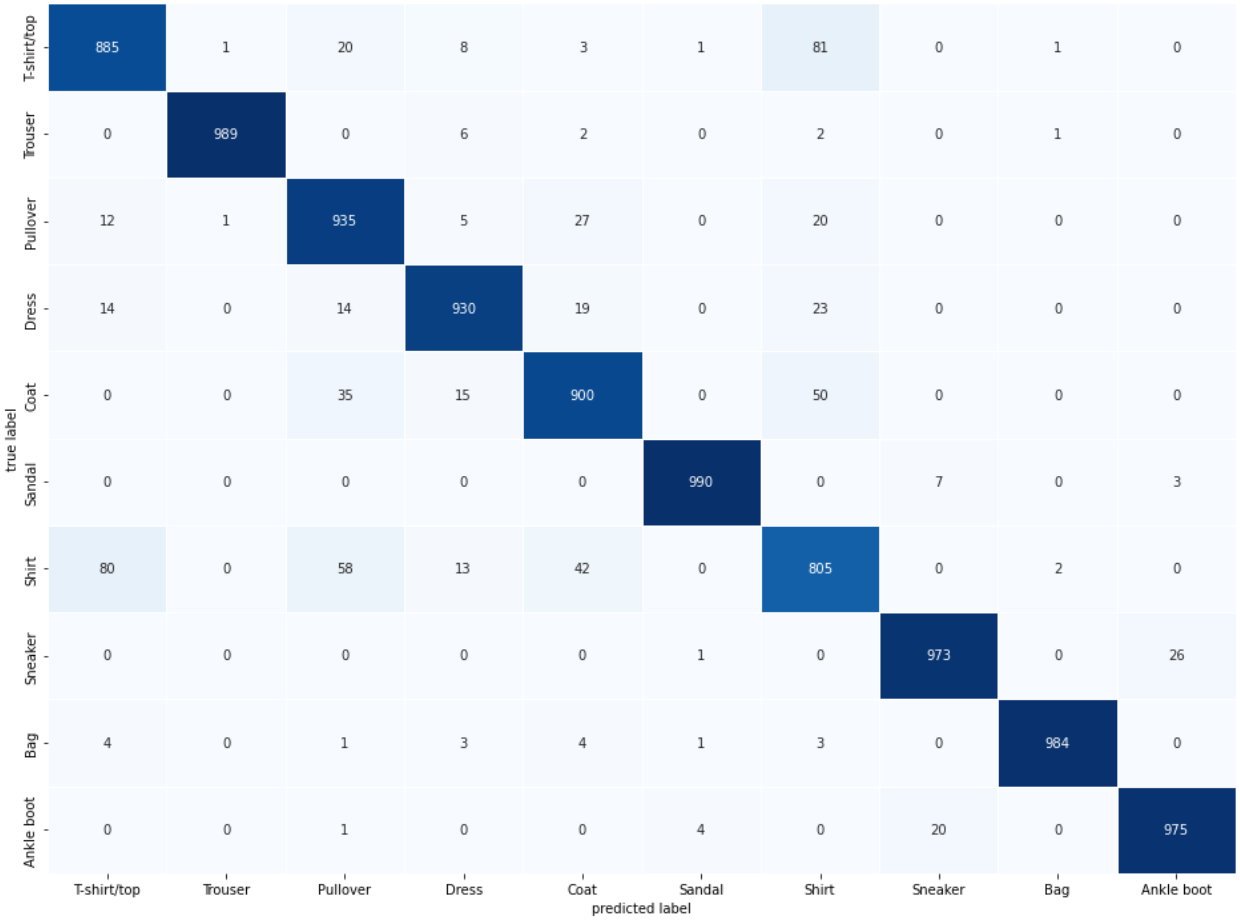
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after about 12 or 13 epochs. The resulting test data accuracy is 93.66%, which is slightly higher than Experiment 2's test accuracy by about 0.04% but still lower than Experiment 1 by 0.04%.

The experiment yields the following confusion matrix:

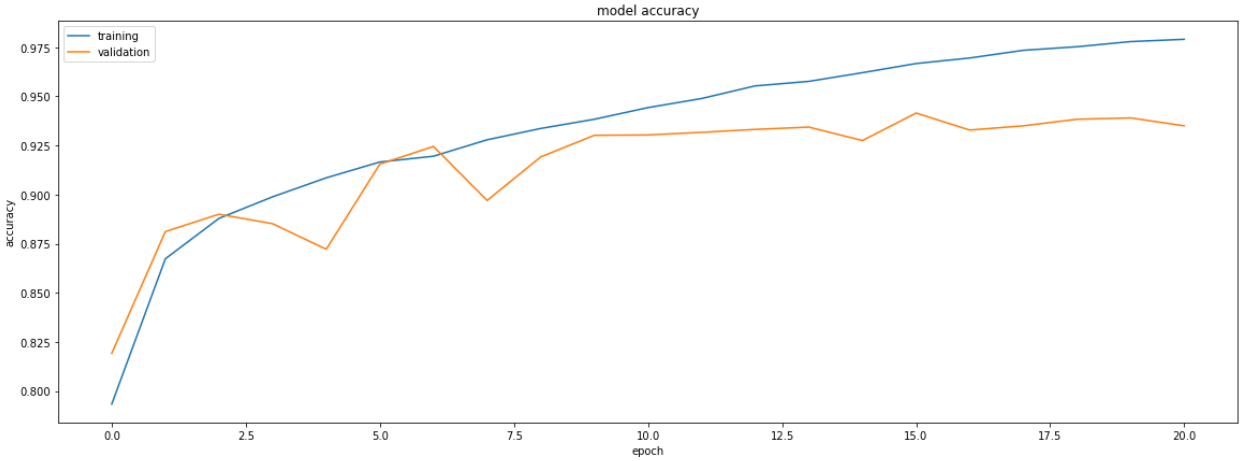


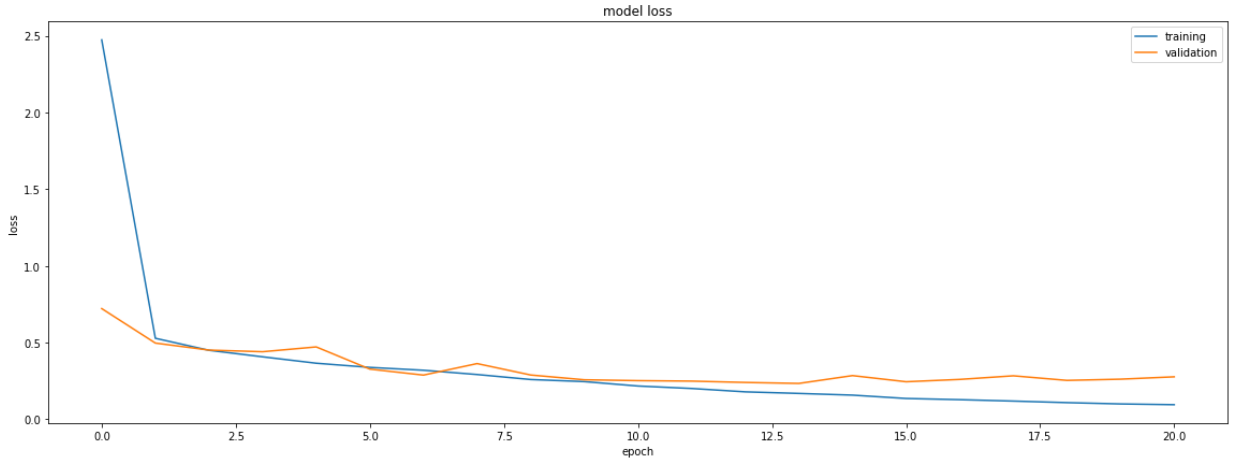
The model can classify most categories well but still has some trouble categorizing shirts.

Experiment 5: CNN with 3 groups of 3 convolution layers & 2 fully connected layers – Change Dropout (0.1)

Experiment 5 follows the same architecture as Experiment 1 but changes dropout rate to 0.1.

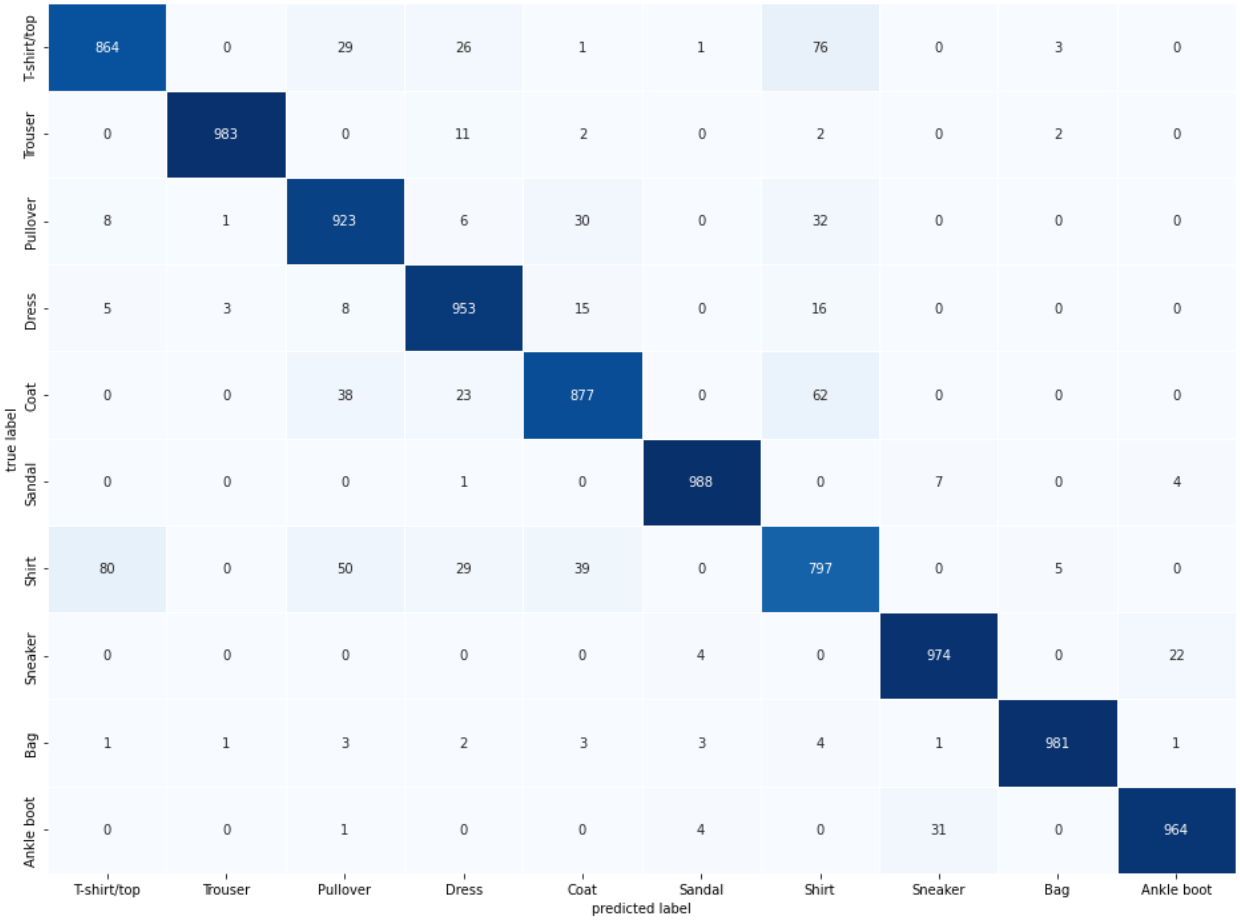
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after 7 epochs. The resulting test data accuracy is 93%, which is the worst-performing model in this section but still out-performs all models in Section One.

The experiment yields the following confusion matrix:



Like the previous models, it too has some trouble classifying shirt images.

The results of our grouped convolutional model experiments in this section are summarized in the table below, with the best one highlighted (Experiment 1).

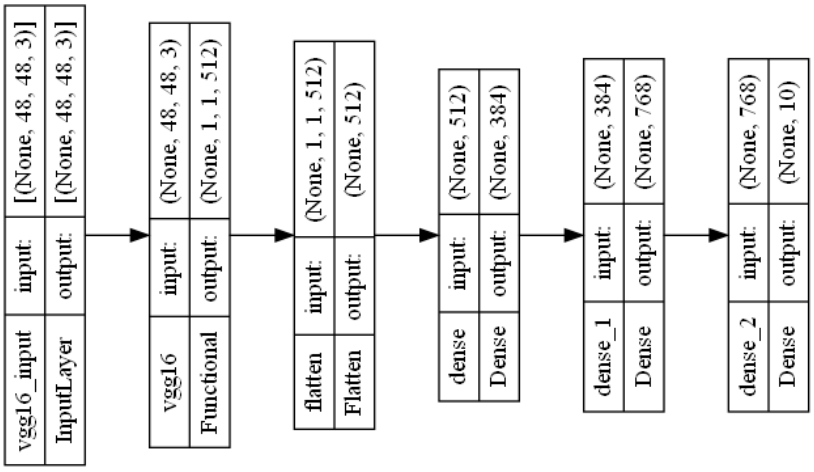
Experiment	Conv2d Layers	DNNs	L2 Reg	Batch Norm	Dropout	Test Accuracy	Test Loss	Precision	Recall	F1-Score	RMSE
1	**9 (3x3 Groups)**	**2**	**0.01**	**Yes**	**0.2**	**0.937**	**0.2698**	**0.94**	**0.94**	**0.94**	**0.978**
2	9 (3x3 Groups)	2	**0.001**	Yes	0.2	0.9362	0.2472	0.94	0.94	0.94	0.992
3	9 (3x3 Groups)	2	**0.1**	Yes	0.2	0.9292	0.3083	0.93	0.93	0.93	1.021
4	9 (3x3 Groups)	2	0.01	Yes	**0.3**	0.9366	0.2799	0.94	0.94	0.94	0.974
5	9 (3x3 Groups)	2	0.01	Yes	**0.1**	0.9304	0.2545	0.93	0.93	0.93	0.988

While we tuned dropout and L2 regularization hyperparameters above and below the values set in Experiment 1, none of the models were able to exceed its test accuracy performance. We found Experiment 1 to have the most optimal hyperparameter tuning settings and was able to achieve the best-performing model so far. All models in this section outperformed all models in the first section.

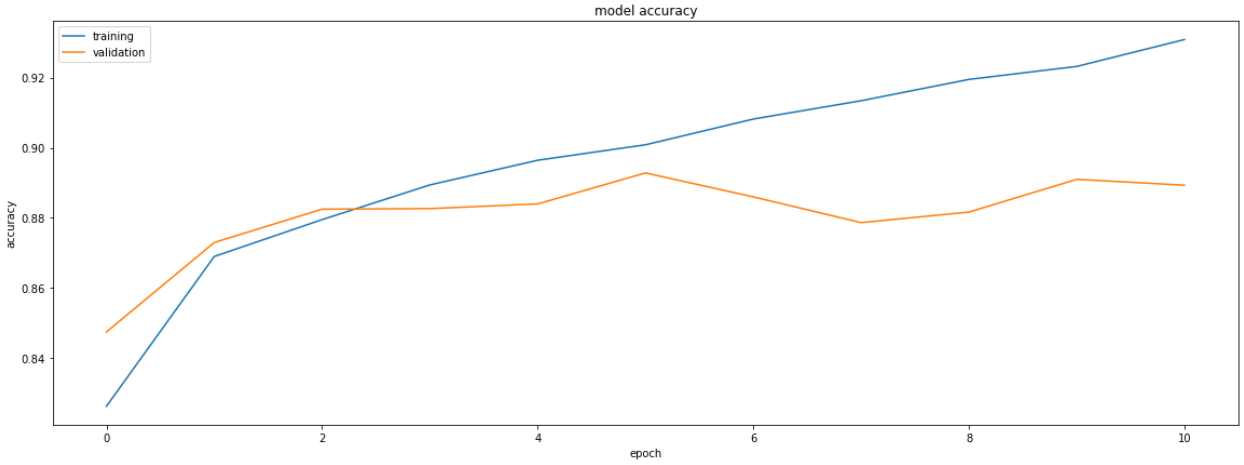
Section 3: Pre-Trained Models

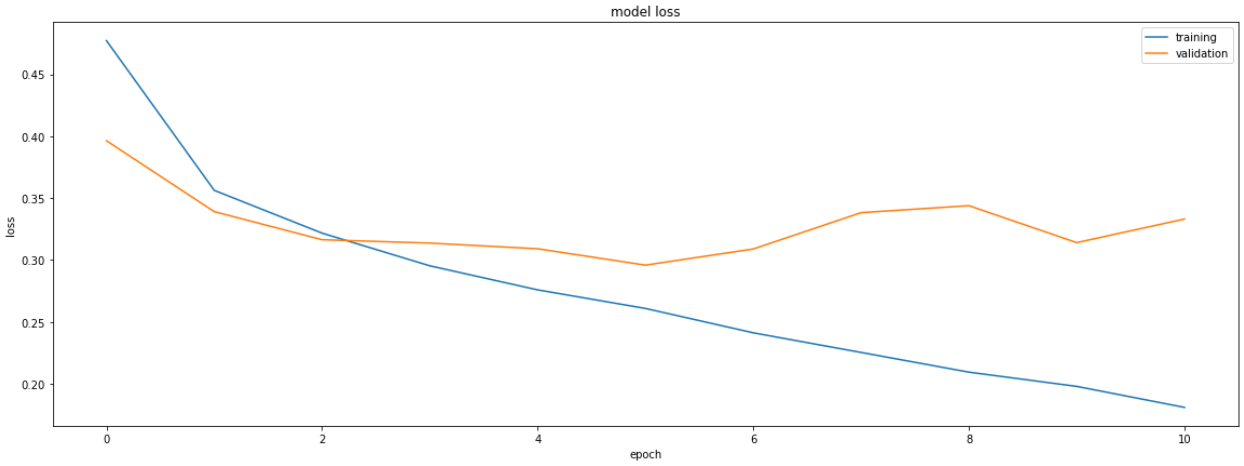
Experiment 1: VGG16 with 2 fully connected layers

The VGG16 pre-trained model base feeds into a DNN containing 2 fully connected layers, with the first layer containing 384 units and the second layer containing 768 units, followed by the 10-way softmax classification output layer.



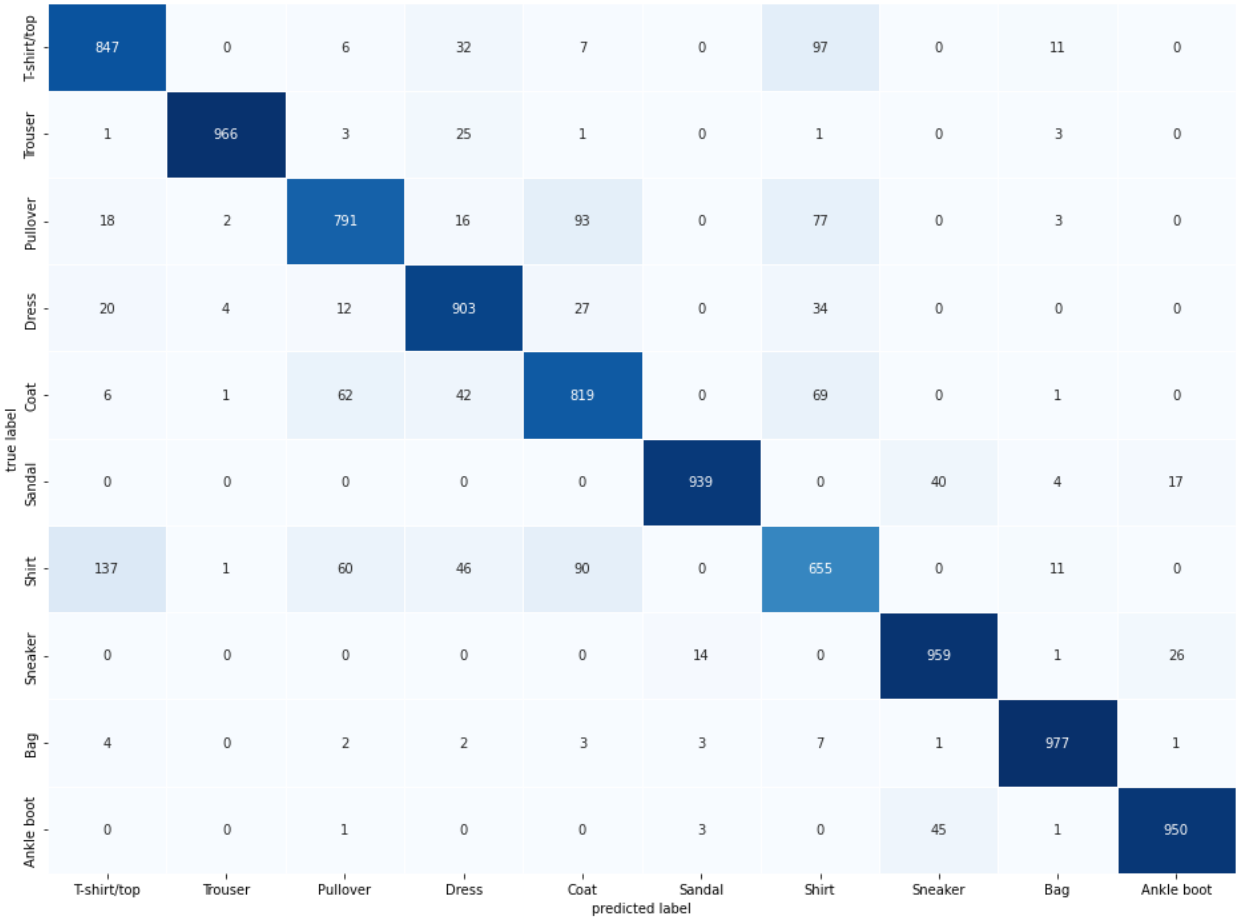
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after 5 epochs. The resulting test data accuracy is 88.1%, over 5% worse than our best-performing model that does not use a pre-trained model base.

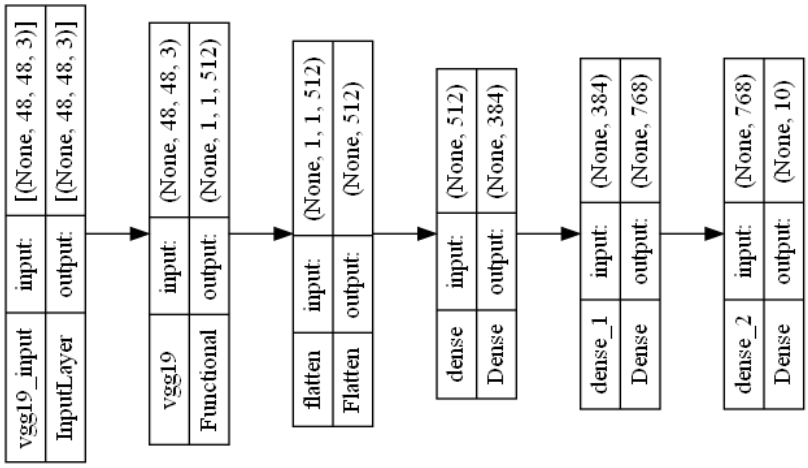
The experiment yields the following confusion matrix:



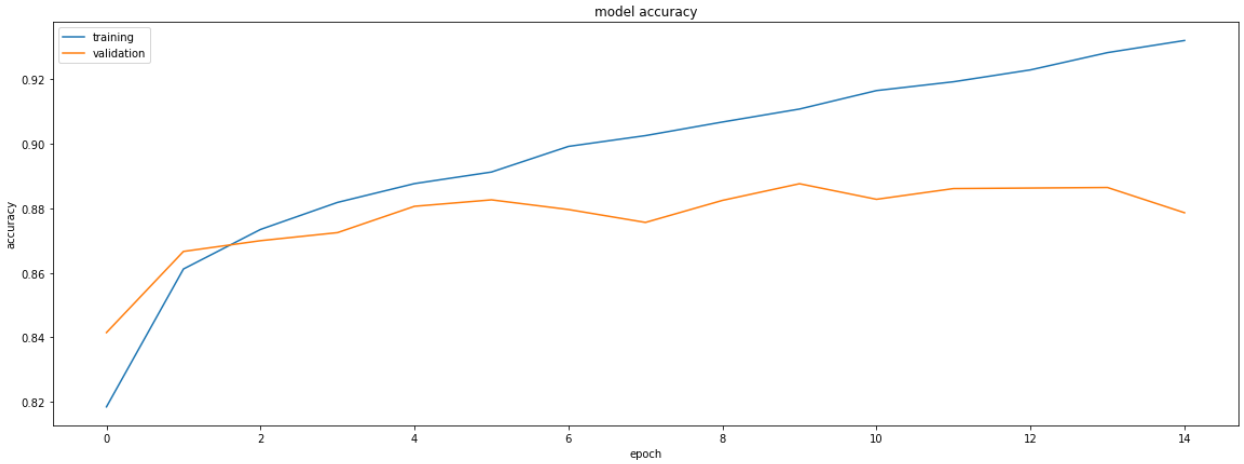
While the model exhibits the common theme of shirt misclassification, we see the model also has difficulties classifying between coats and pullovers.

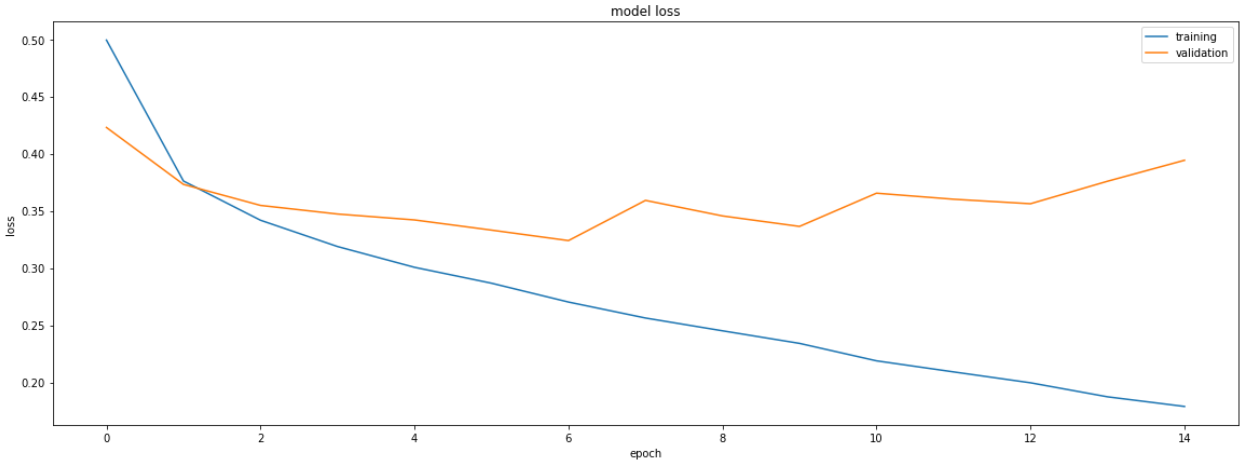
Experiment 2: VGG19 with 2 fully connected layers

Like the VGG16 model above, the VGG19 pre-trained base feeds into a DNN containing 2 fully connected layers, with the first layer containing 384 units and the second layer containing 768 units, followed by the 10-way softmax classification output layer.



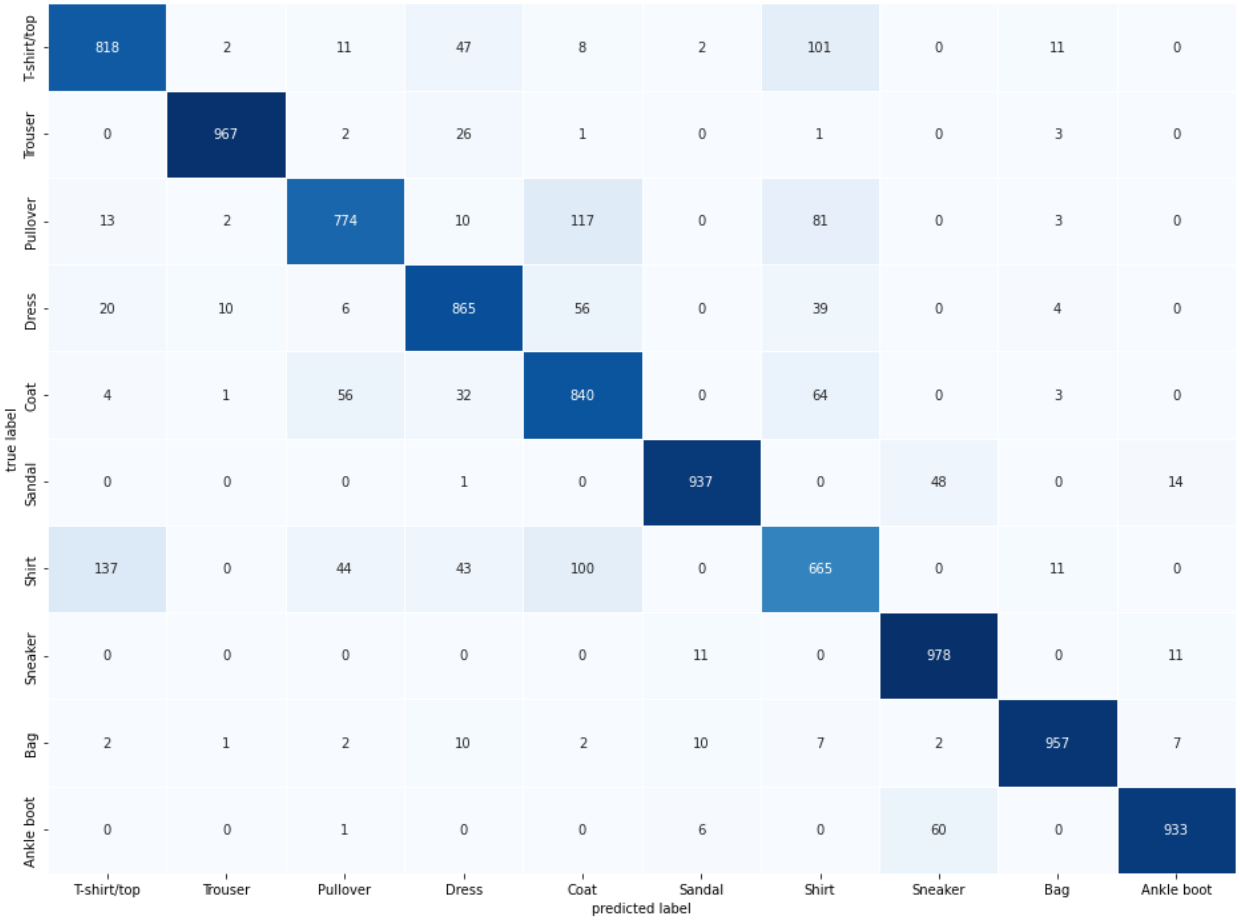
Training and validation accuracy and cross entropy loss charts are below.





We see validation results start to diverge from the training results after 5 epochs. The resulting test data accuracy is 87.3%, 0.7% worse than the less complex VGG16-based model.

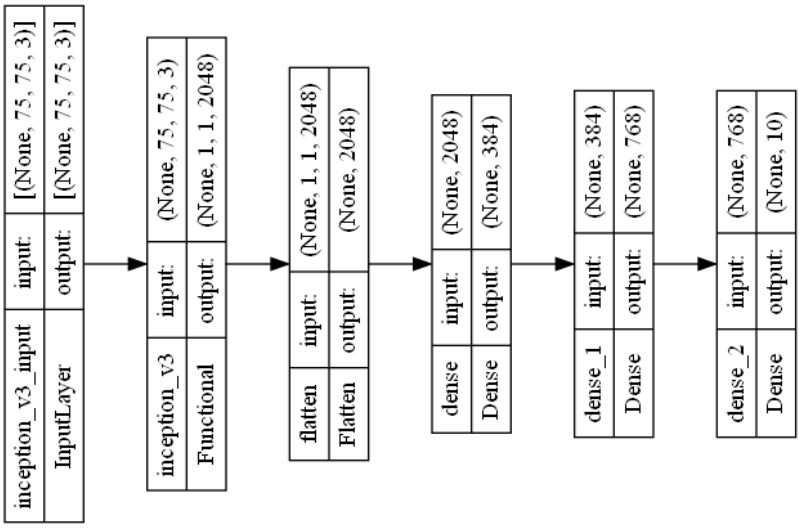
The experiment yields the following confusion matrix:



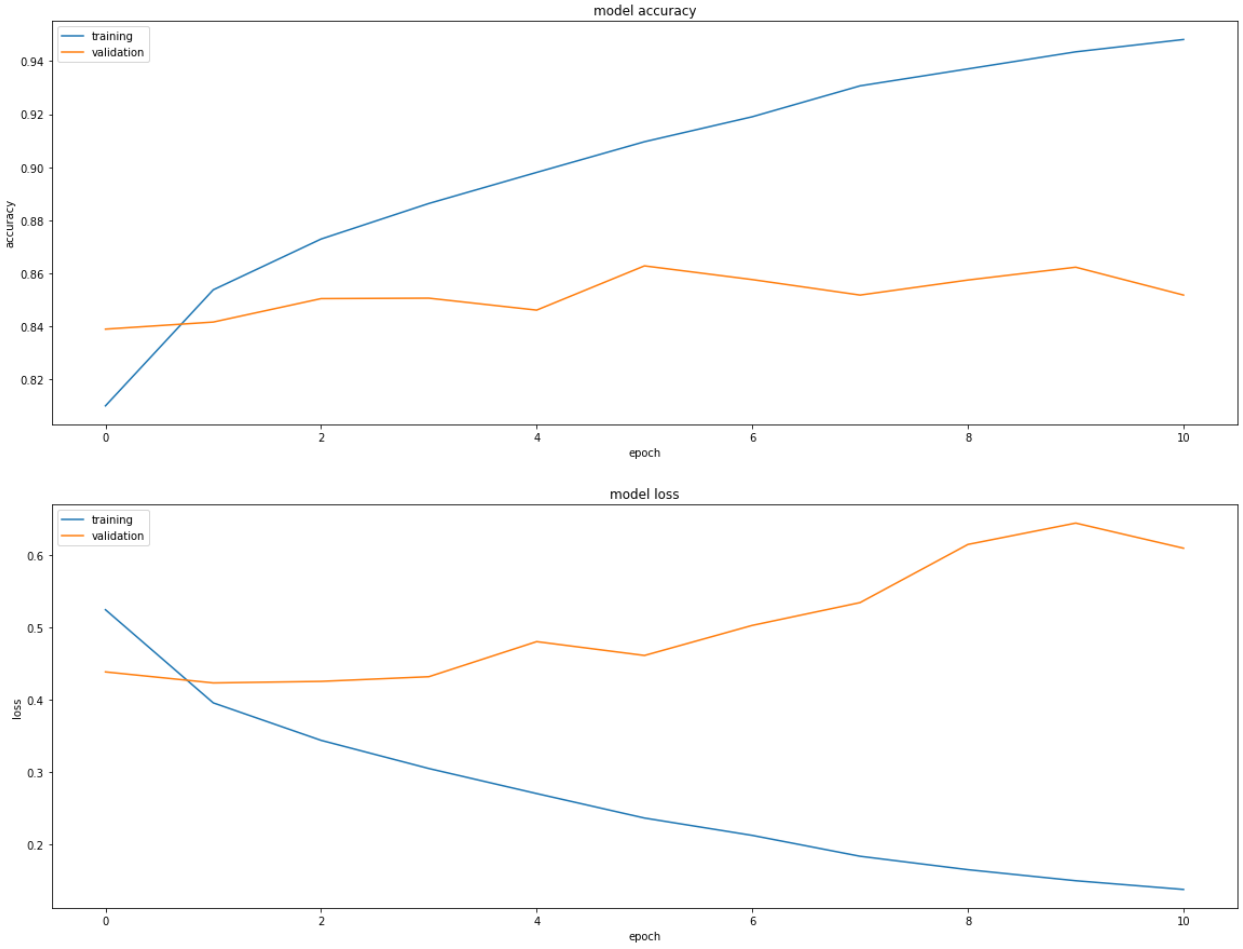
Along with shirt misclassification, VGG19 also has difficulties classifying between coats, dresses, and pullovers, having more trouble than its smaller VGG16 counterpart.

Experiment 3: Inception V3 with 2 fully connected layers

The Inception V3 pre-trained base feeds into a DNN containing 2 fully connected layers, with the first layer containing 384 units and the second layer containing 768 units, followed by the 10-way softmax classification output layer.

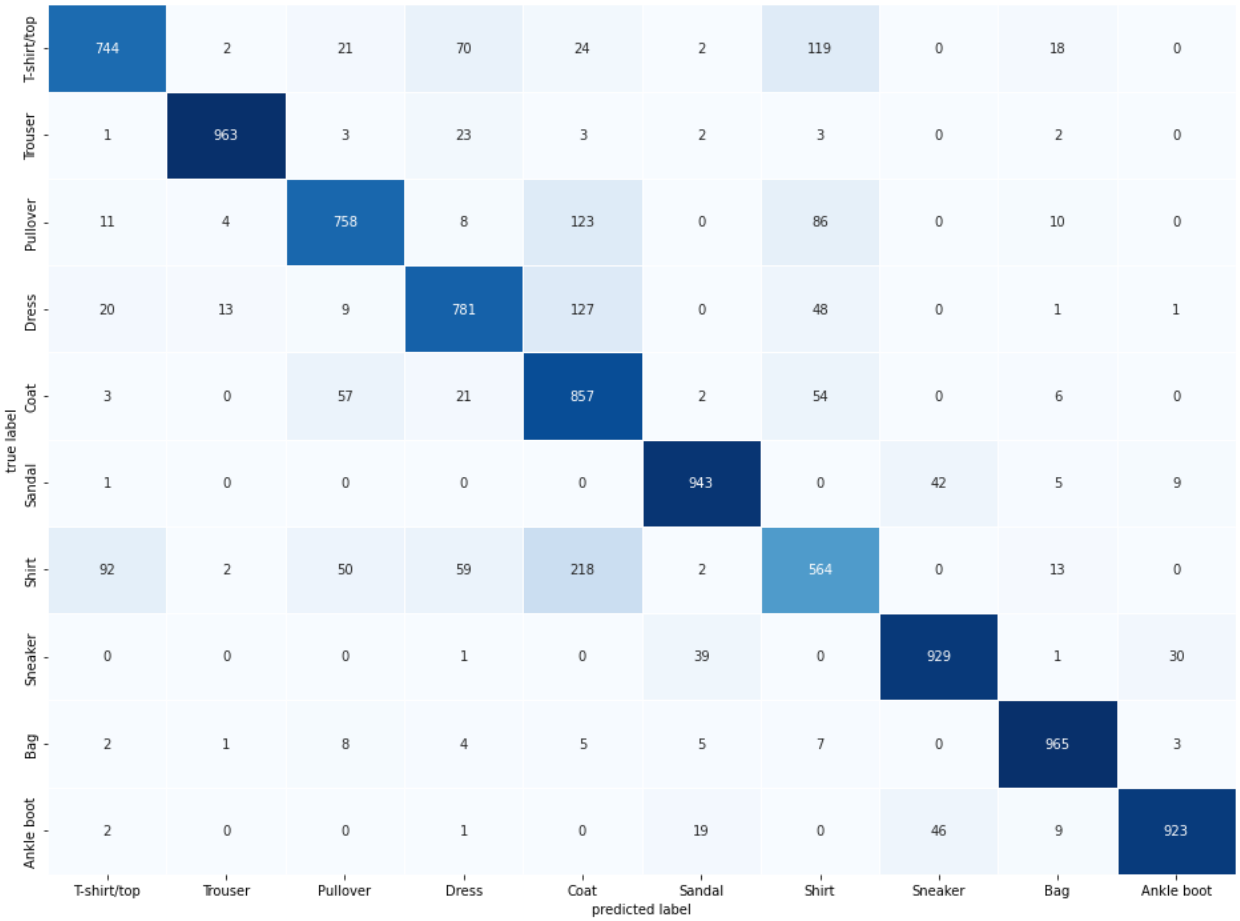


Training and validation accuracy and cross entropy loss charts are below.



We see validation results already start to diverge from the training results just after 2 or 3 epochs. The resulting test data accuracy is 84.3%, 3% worse than the VGG19 model.

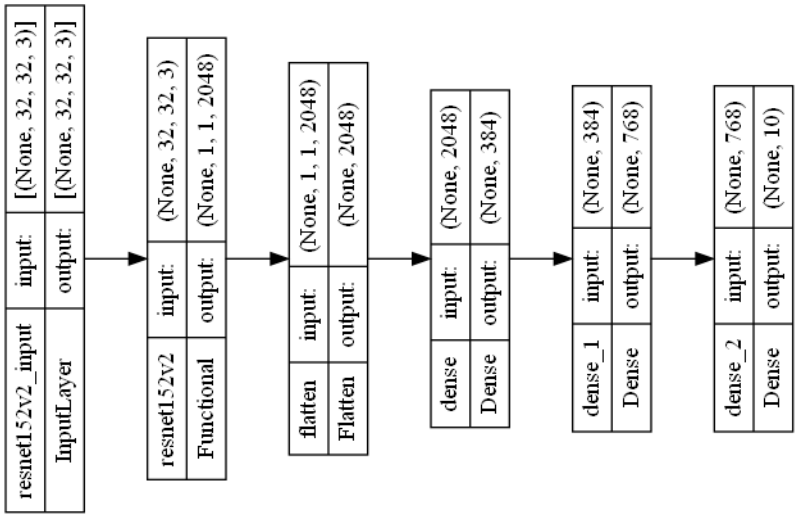
The experiment yields the following confusion matrix:



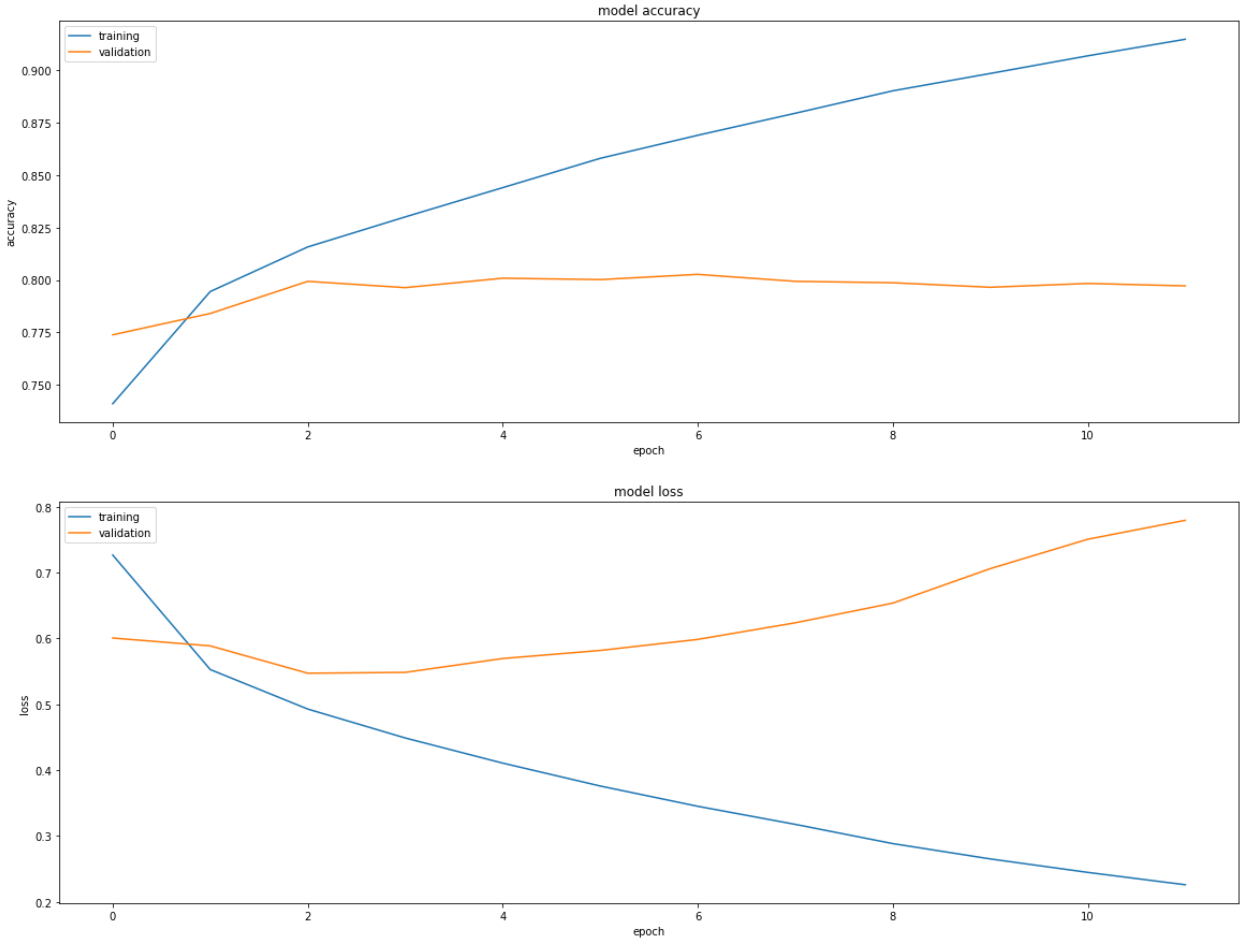
The Inception-based model has some trouble classifying any article of clothing worn on the upper body, performing worse than the VGG-based models.

Experiment 4: ResNet152 V3 with 2 fully connected layers

The ResNet152 V3 pre-trained base feeds into a DNN containing 2 fully connected layers, with the first layer containing 384 units and the second layer containing 768 units, followed by the 10-way softmax classification output layer.

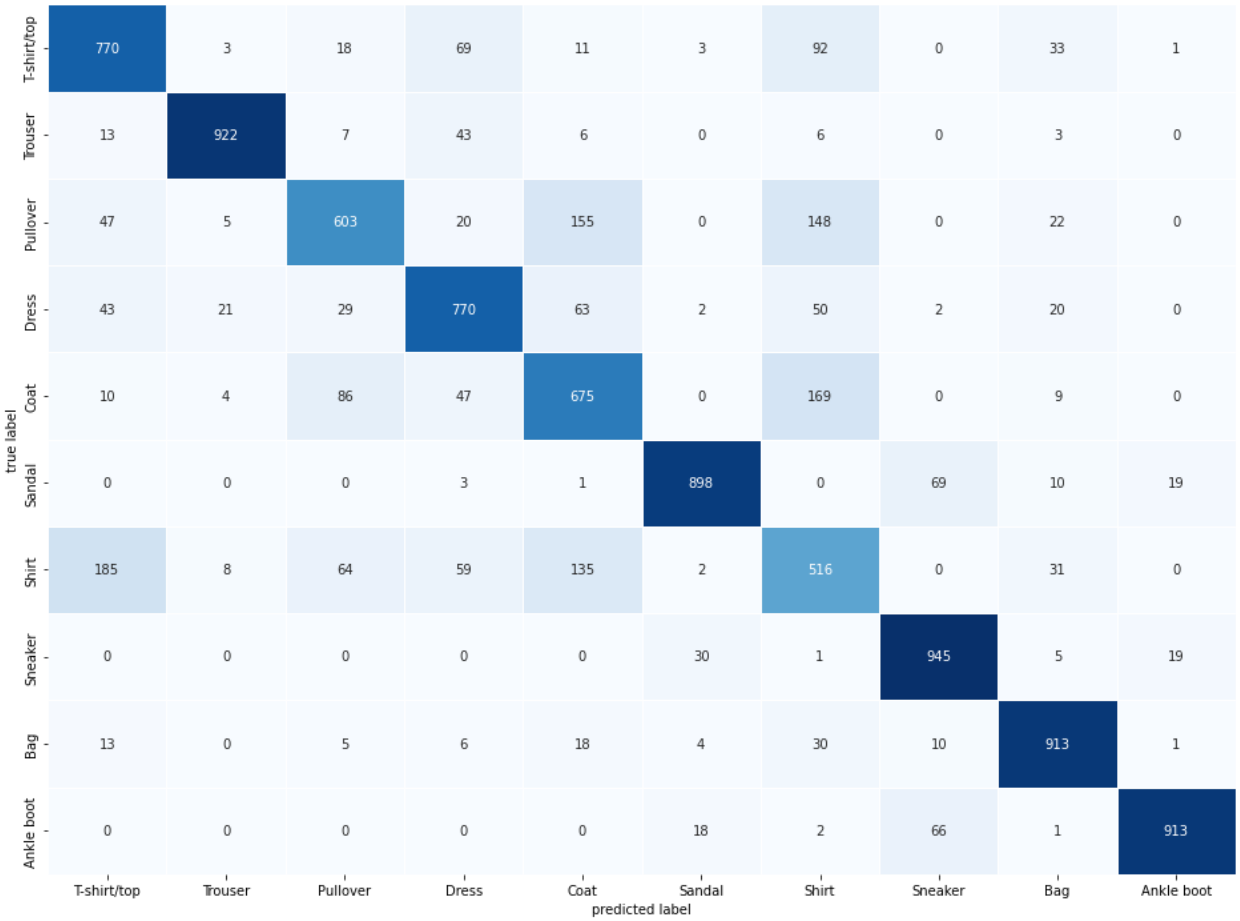


Training and validation accuracy and cross entropy loss charts are below.



We see validation results already start to diverge from the training results just after 2 or 3 epochs. The resulting test data accuracy is 79.3%, our worst overall model.

The experiment yields the following confusion matrix:



The ResNet-based pre-trained model correctly classifies shirts just over half of the time, which is not great. It performs worst overall out of all the models we've tested, as it has trouble categorizing upper-body garments in general.

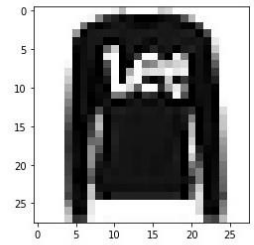
The results of our pre-trained model experiments are summarized in the table below, with the best one highlighted (Experiment 1, using VGG16).

Experiment	Model	DNNs	L2 Reg	Batch Norm	Dropout	Test Accuracy	Test Loss	Precision	Recall	F1-Score	RMSE
1	**VGG16**	**2**	**None**	**None**	**None**	**0.8806**	**0.3338**	**0.88**	**0.88**	**0.88**	**1.267**
2	**VGG19**	2	None	None	None	0.8734	0.3533	0.87	0.87	0.87	1.288
3	**Inception V3**	2	None	None	None	0.8427	0.4356	0.85	0.84	0.84	1.356
4	**ResNet152V3**	2	None	None	None	0.7925	0.564	0.79	0.79	0.79	1.616

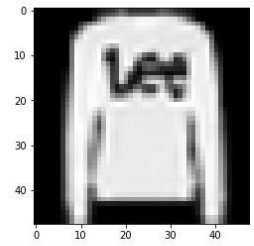
The reason why these pre-trained models did not perform as well as our own CNNs in this study could be due to the image transformation performed on the Fashion-MNIST dataset, as they require images to be in a 3-channel color format and need to be larger than their 28x28 resolution. While we are able to perform such transformations, the overall image quality might have been compromised in the process.

The figure below shows a comparison of an original Fashion-MNIST image with its transformed counterparts used for training and testing the pre-trained models. While the image colors are inversed, we see a blurrier image as its resolution is increased, thus a degradation in image quality.

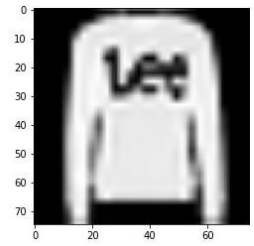
```
# original B/W image
plt.imshow(x_test[1], cmap=plt.cm.binary)
plt.show()
```



```
# transformed image: 3-channel RGB, inversed, 48x48 resolution
plt.imshow(x_test_tf[1], cmap=plt.cm.binary)
plt.show()
```



```
# transformed image: 3-channel RGB, inversed, 75x75 resolution
plt.imshow(x_test_tf_inception[1], cmap=plt.cm.binary)
plt.show()
```



With ResNet requiring a 75x75 resolution image and our transformation process results in a blurrier image, it is understandable why the model would perform worse than other models as it is more difficult for it to distinguish features, despite its complex architecture.

Overall, we found the best performing model to be Experiment 1 in the Grouped Convolutional Models section, yielding a 93.7% accuracy on the Fashion-MNIST test dataset.

Conclusion

We were able to build a convolutional neural network model that can classify images from the Fashion-MNIST test dataset with 93.7% accuracy. Incorporating regularization techniques to our models such as batch normalization, dropout, and ridge regression along with hyperparameter tuning can improve accuracy by almost 1%. We also saw diminishing returns when we created a CNN using 4 isolated convolutional layers did not improve test accuracy. Utilization of pre-trained models such as ResNet did

not achieve similar results compared to other studies that used them on the Fashion-MNIST dataset, but it could be due to the image quality of the transformed dataset. Our most optimal CNN-based model uses a complex architecture using multiple groups of convolutional layers. Given the 93.7% accuracy, we feel this model is a good first step towards integrating a classifier into the retailer's recommendation system. While we feel the model's accuracy is enough to be deployed for A/B testing on a beta version of the retail website, there is still more room for improvement. We can expect to optimize model accuracy over time, either with more nuanced tuning of our own developed models, creating a more complex architecture of grouped convolution layers, deep-diving into the details behind pre-trained models, or improving our image transformation processes.

Resources

Nocentini, O., Kim, J., Bashir, M. Z., & Cavallo, F. (2022). Image Classification Using Multiple Convolutional Neural Networks on the Fashion-MNIST Dataset. *Sensors*, *22*(23), 9544. https://doi.org/10.3390/s22239544

Tang, Y., Hanguo ., & Shuyong L. (2020). Optimal Design of Deep Residual Network Based on Image Classification of Fashion-MNIST Dataset. *Journal of Physics: Conference Series*, 1624.5 (2020), 52011. http://doi.org/10.1088/1742-6596/1624/5/052011