İZMİR KÂTİP ÇELEBİ UNIVERSITY

2010

Deep Learning Project Report

Fashion Classification with Deep Networks

Instructor
Asst. Prof. Volkan Kılıç

Ali Egemen Taşören
160403034

Abstract

In this project, densenet, mobilenet, nasnet-mobile, nasnet, resnet18, resnet50, resnet101, shufflenet, yolov3 networks are trained using a customized version of deepfashion v1 dataset. With the customization process, backgrounds of the clothes and pictures with multiple items of clothing eliminated from the deepfashion v1 dataset. This customization significantly improved the training performance of the mentioned datasets. Most validation accuracy of 83.01% is obtained using the nasnet-mobile and shufflenet deep network. Least validation accuracy of 61.1% is obtained using a resnet101 deep network. Later a GUI has developed for fashion classification.

Introduction

Ongoing years, with the quick development of online trade and style-related applications, fashion picture investigation and comprehension have pulled in an increasing amount of consideration in the community. Broad investigations have been conducted in this field, for example, class arrangement, style or trait forecast, fashion landmark detection, and fashion picture synthesis. In this paper, it’s concentrated on fashion piece classification. Past works based on deep learning have demonstrated a lot of accomplishment in these fields. However, most of the fashion classification is done by the fashion-mnsit dataset which proves bad results thus not have any useful meaning in real life.
In this research, a bigger dataset which is deepfashion v1 is curated using computer programs and researchers eyes. This dataset proves better results for real-life situations.

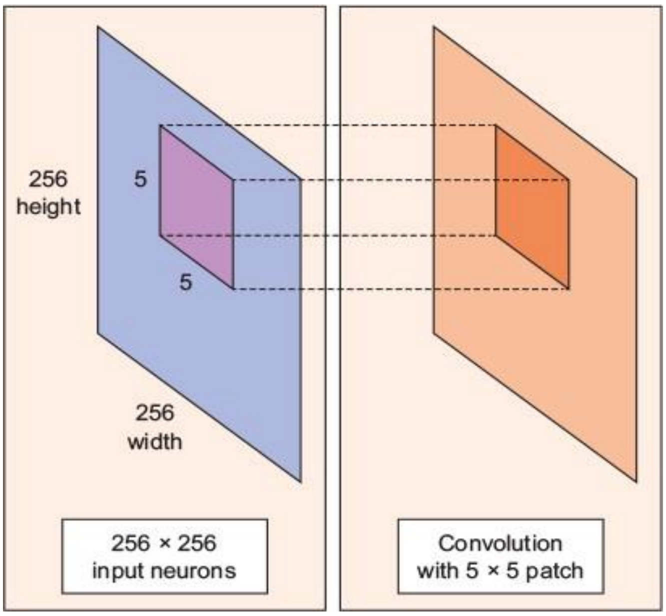
Image Classification

The issue of Image Classification goes this way: Given a lot of pictures that are marked with a solitary classification, we are approached to foresee these classifications for a novel arrangement of test pictures and measure the exactness of the expectations. There is an assortment of difficulties related to this assignment, including perspective variety, scale variety, intra-class variety, picture twisting, picture impediment, brightening conditions, foundation mess, and so on.

By what means may we approach composing a calculation that can order pictures into unmistakable classes? PC Vision analysts have thought of an information-driven way to deal with comprehend this. Rather than attempting to determine what all of the picture classifications of premium look like legitimately in code, they give the PC numerous instances of each picture class and afterward create learning calculations that take a gander at these models and find out about the visual appearance of each class. As it were, they initially gather a preparation dataset of marked pictures, at that point feed it to the PC with the end goal for it to get acquainted with the information.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is the most popular neural network model being used for image classification problem. The big idea behind CNNs is that a local understanding of an image is good enough. The practical benefit is that having fewer parameters greatly improves the time it takes to learn as well as reduces the amount of data required to train the model. Instead of a fully connected network of weights from each pixel, a CNN has just enough weights to look at a small patch of the image. It’s like reading a book by using a magnifying glass; eventually, you read the whole page, but you look at only a small patch of the page at any given time.



Consider a 256 x 256 image. CNN can efficiently scan it chunk by chunk — say, a 5 × 5 window. The 5 × 5 window slides along the image (usually

left to right, and top to bottom), as shown below. How "quickly" it slides is called its **stride length**. For example, a stride length of 2 means the 5 × 5 sliding window moves by 2 pixels at a time until it spans the entire image.
A **convolution** is a weighted sum of the pixel values of the image, as the window slides across the whole image. Turns out, this convolution process throughout an image with a weight matrix produces another image (of the same size, depending on the convention). Convolving is the process of applying a convolution.
The sliding-window shenanigans happen in the **convolution layer** of the neural network. A typical CNN has multiple convolution layers. Each convolutional layer typically generates many alternate convolutions, so the weight matrix is a tensor of 5 × 5 × n, where n is the number of convolutions.
As an example, let's say an image goes through a convolution layer on a weight matrix of 5 × 5 × 64. It generates 64 convolutions by sliding a 5 × 5 window. Therefore, this model has 5 × 5 × 64 (= 1,600) parameters, which are remarkably fewer parameters than a fully connected network, 256 × 256 (= 65,536).
The beauty of CNN is that the number of parameters is independent of the size of the original image. You can run the same CNN on a 300 × 300 image, and the number of parameters won't change in the convolution layer.

ShuffleNet

Megvii Inc (a.k.a Face++) introduced ShuffleNet, which they claim to be an extremely computation efficient CNN architecture, designed for mobile devices with computing power of 10–150 MFLOPs. The ShuffleNet utilizes pointwise group convolution and channel shuffle to reduce computation cost while maintaining accuracy. It manages to obtain lower top-1 error than the MobileNet system on ImageNet classification, and achieves ~13x actual speedup over AlexNet while maintaining comparable accuracy.

ResNet

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. An additional weight matrix may be used to learn the skip weights; these models are known as HighwayNets. In the context of residual neural networks, a non-residual network may be described as a plain network.

DenseNet

In Standard ConvNet, input image goes through multiple convolution and obtain high-level features. In ResNet, identity mapping is proposed to promote the gradient propagation. Element-wise addition is used. It can be viewed as algorithms with a state passed from one ResNet module to another one. In DenseNet, each layer obtains additional inputs from all preceding layers and passes on its feature-maps to all subsequent layers. Concatenation is used. Each layer is receiving a “collective knowledge” from all preceding layers.

NasNet

Neural architecture search (NAS) is a technique for automating the design of artificial neural networks (ANN), a widely used model in the field of machine learning. NAS has been used to design networks that are on par or outperform hand-designed architectures. Methods for NAS can be categorized according to the search space, search strategy, and performance estimation strategy used:

1. The search space defines the type(s) of ANN that can be designed and optimized.
2. The search strategy defines the approach used to explore the search space.
3. The performance estimation strategy evaluates the performance of a possible ANN from its design (without constructing and training it).

RL-based NAS requires thousands of GPU-days of searching/training to achieve state-of-the-art computer vision results as described in the NASNet, mNASNet, and MobileNetV3 papers.

MobileNet

MobileNet is an architecture that is more suitable for mobile and embedded based vision applications where there is a lack of computing power. This architecture was proposed by Google.

YOLO

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X, it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev.

Information on Dataset

In the training process, deepfashion v1 with customization is used. This customization includes:

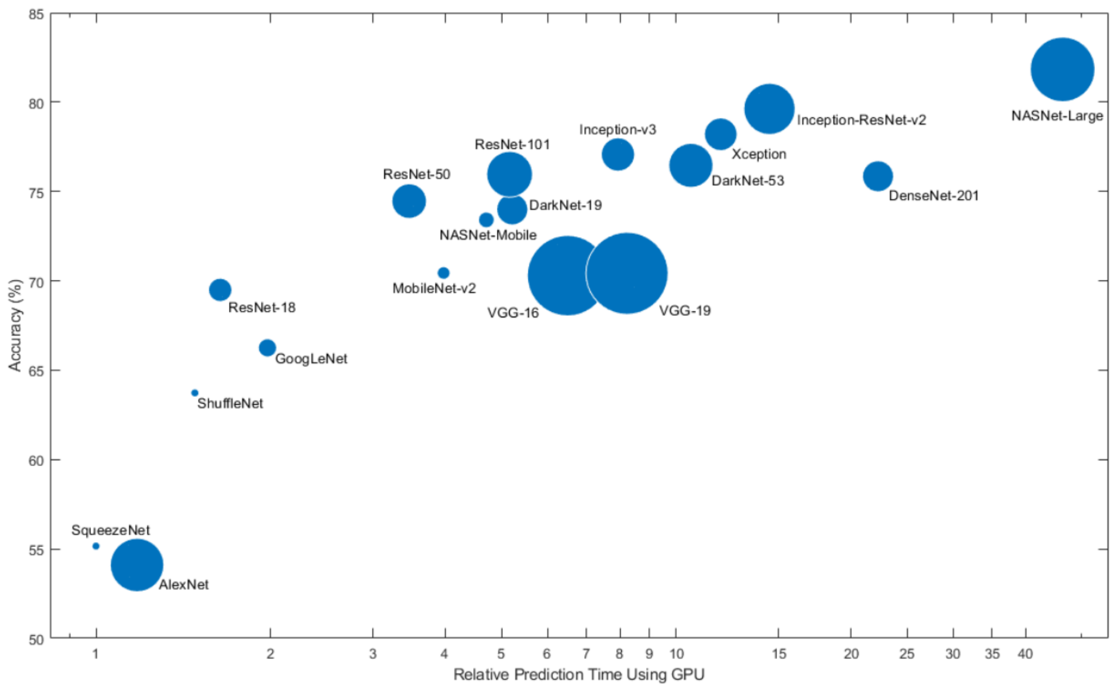
1. Removing unnecessary pictures from the dataset (such as miss classed pictures or pictures with extremely bad quality).
2. Removing backgrounds from some pictures that can harden the training process (such as pictures taken directly in the street).
3. Removing pictures include multiple pieces of clothing since a person can view tens of clothing pieces on him, the system will try to detect them as single clothing if those clothes exist.
4. Images augmented in x and y axes.

Dataset contains 868 images in total with classes as follows:

1. Blouse: 66 images.
2. Dress: 76 images.
3. Jacket: 43 images.
4. Pants: 161 images.
5. Shorts: 78 images.
6. Skirt: 94 images.
7. Sweater: 71 images.
8. Tank: 118 images.
9. T-Shirt: 161 images.

Comparison On ImageNet Dataset

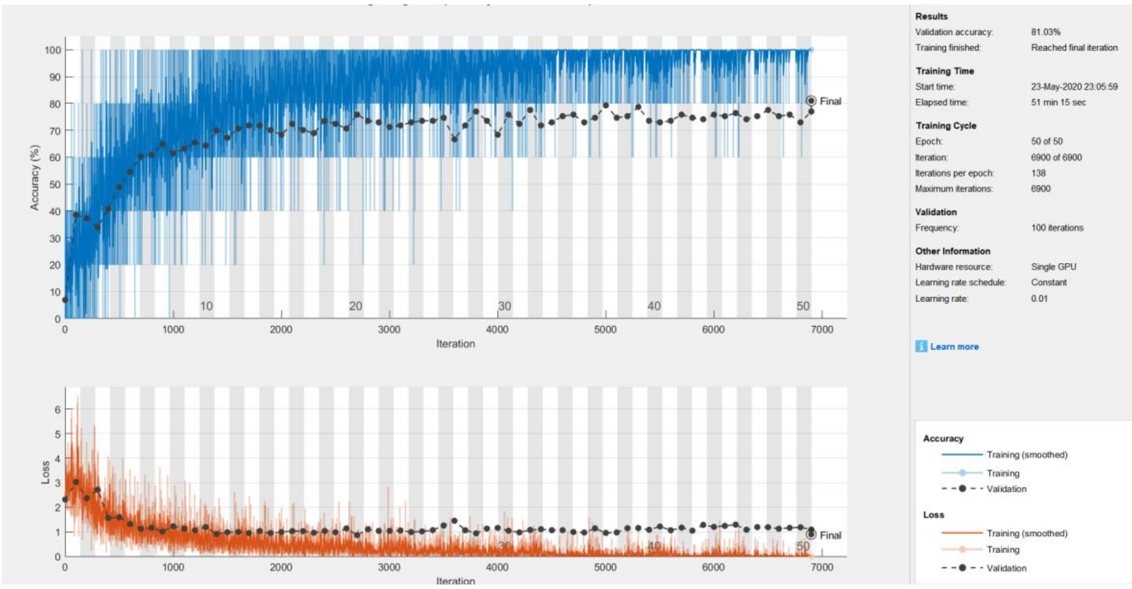
MATLAB researched the imagenet dataset and they obtained an accuracy vs GPU time chart which can be seen below.



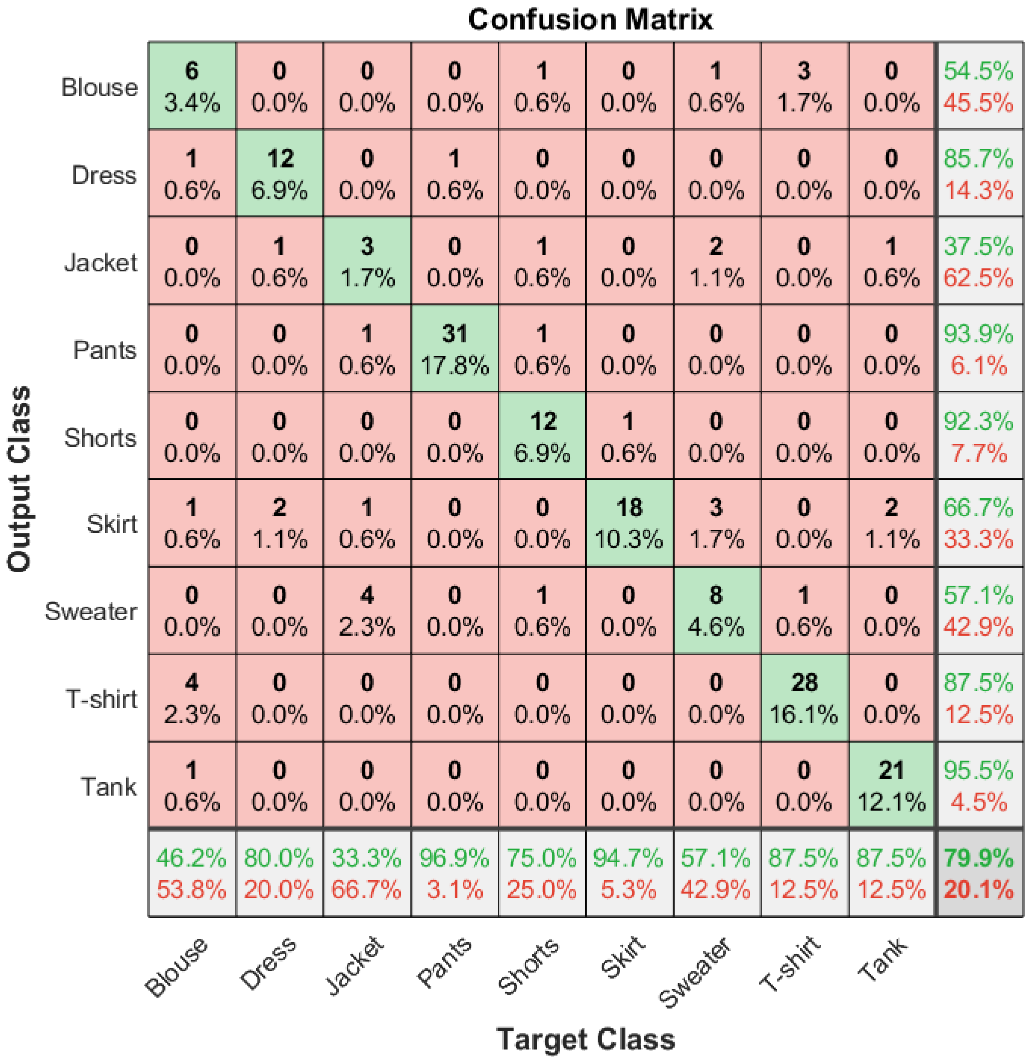
Customized Dataset Training Results

In this section, the training process, confusion matrixes, and random image detection chart is given for all the networks that are used.

MobileNet:

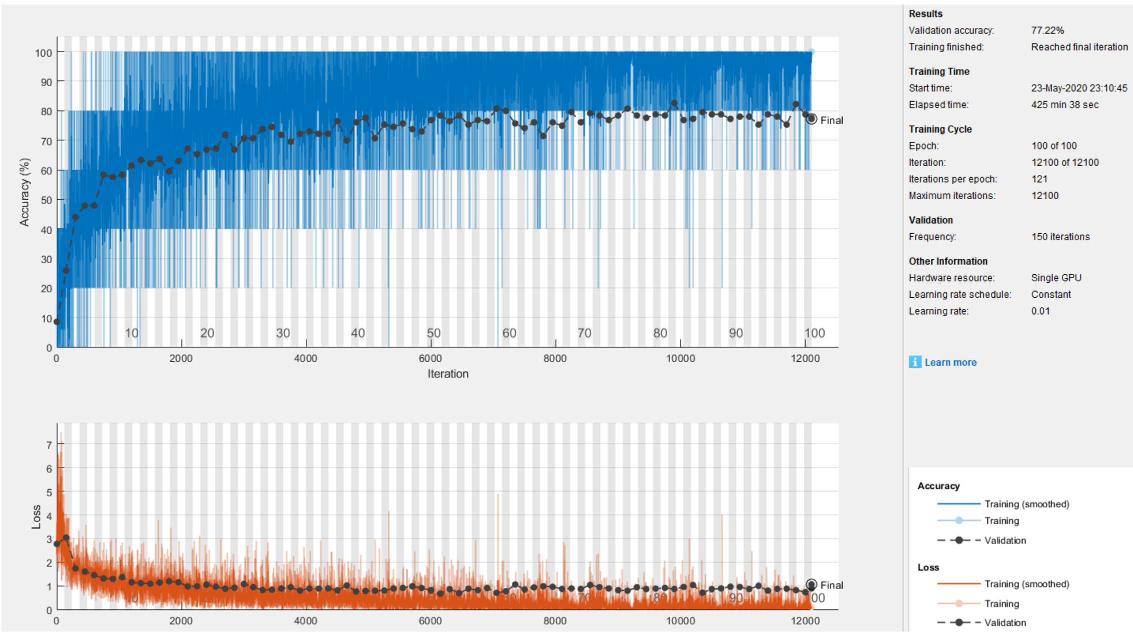


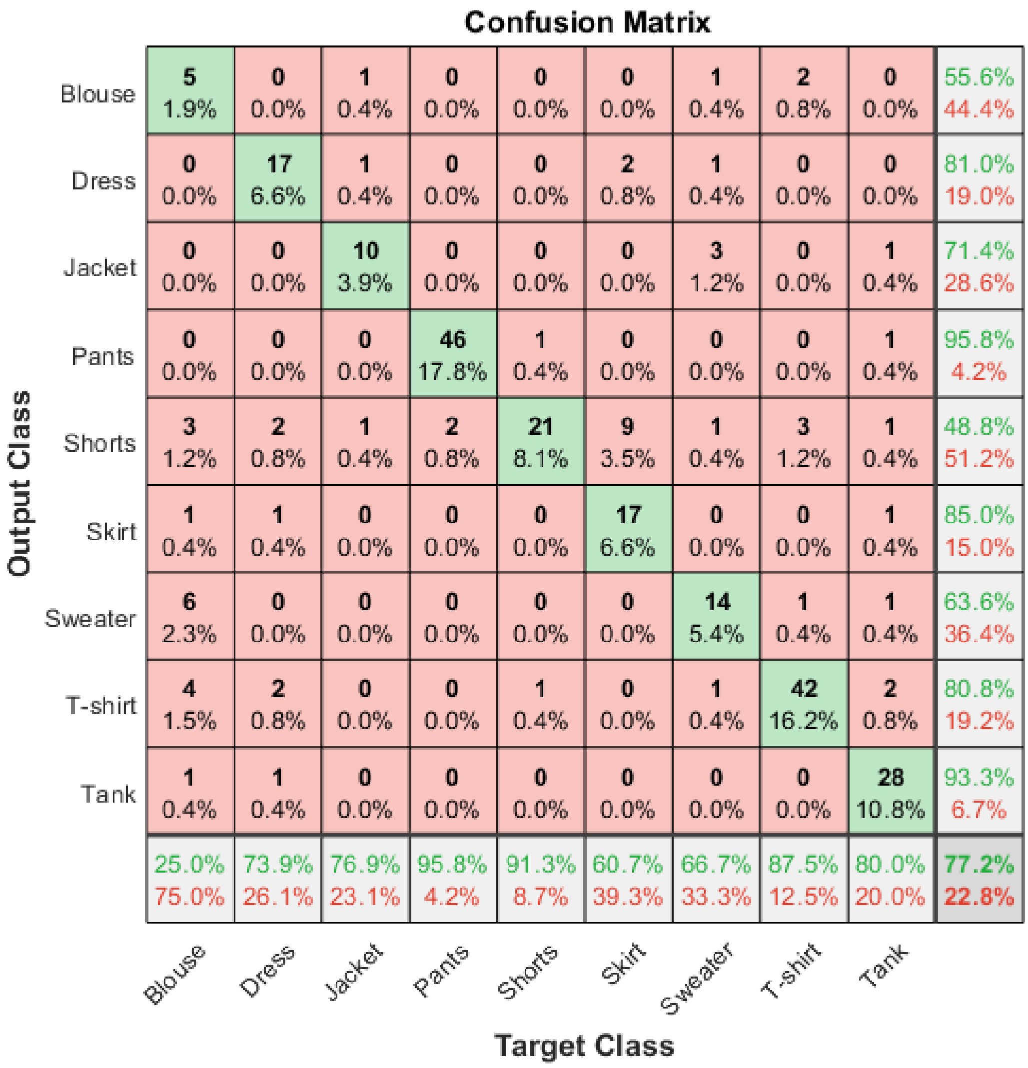
Accuracy (%)
Iteration
Final
Loss
Iteration
Final
Results
Validation accuracy: 81.03%
Training finished: Reached final iteration
Training Time
Start time: 23-May-2020 23:05:59
Elapsed time: 51 min 15 sec
Training Cycle
Epoch: 50 of 50
Iteration: 6900 of 6900
Iterations per epoch: 138
Maximum iterations: 6900
Validation
Frequency: 100 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation


Confusion Matrix
Output Class
Target Class
Blouse
Dress
Jacket
Pants
Shorts
Skirt
Sweater
T-shirt
Tank
6 3.4% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.6% | 0 0.0% | 1 0.6% | 3 1.7% | 0 0.0% | 54.5% 45.5%
1 0.6% | 12 6.9% | 0 0.0% | 1 0.6% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 85.7% 14.3%
0 0.0% | 1 0.6% | 3 1.7% | 0 0.0% | 1 0.6% | 0 0.0% | 2 1.1% | 0 0.0% | 1 0.6% | 37.5% 62.5%
0 0.0% | 0 0.0% | 1 0.6% | 31 17.8% | 1 0.6% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 93.9% 6.1%
0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 12 6.9% | 1 0.6% | 0 0.0% | 0 0.0% | 0 0.0% | 92.3% 7.7%
1 0.6% | 2 1.1% | 1 0.6% | 0 0.0% | 0 0.0% | 18 10.3% | 3 1.7% | 0 0.0% | 2 1.1% | 66.7% 33.3%
0 0.0% | 0 0.0% | 4 2.3% | 0 0.0% | 1 0.6% | 0 0.0% | 8 4.6% | 1 0.6% | 0 0.0% | 57.1% 42.9%
4 2.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 28 16.1% | 0 0.0% | 87.5% 12.5%
1 0.6% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 21 12.1% | 95.5% 4.5%
46.2% 53.8% | 80.0% 20.0% | 33.3% 66.7% | 96.9% 3.1% | 75.0% 25.0% | 94.7% 5.3% | 57.1% 42.9% | 87.5% 12.5% | 87.5% 12.5% | 79.9% 20.1%

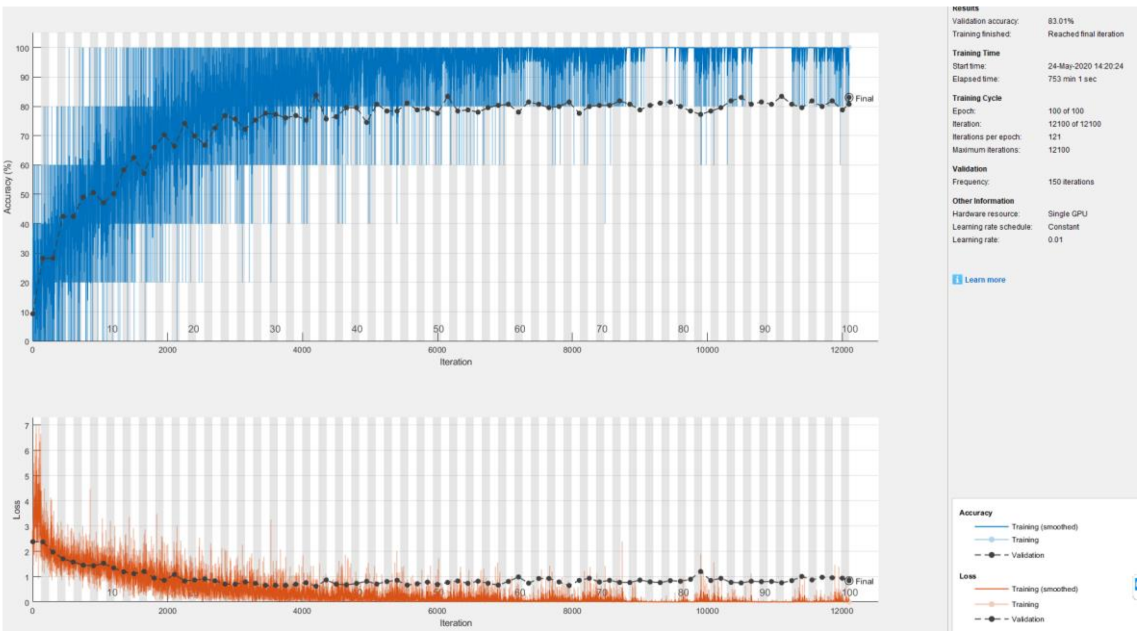

Skirt, 83.6%
Blouse, 90.8%
Skirt, 100%
T-shirt, 100%
Dress, 100%
Sweater, 66.6%
Skirt, 99.9%
Pants, 100%
Pants, 100%
T-shirt, 100%
T-shirt, 94.7%
T-shirt, 99.8%
Dress, 100%
Skirt, 97.4%
Tank, 100%
Sweater, 95.3%

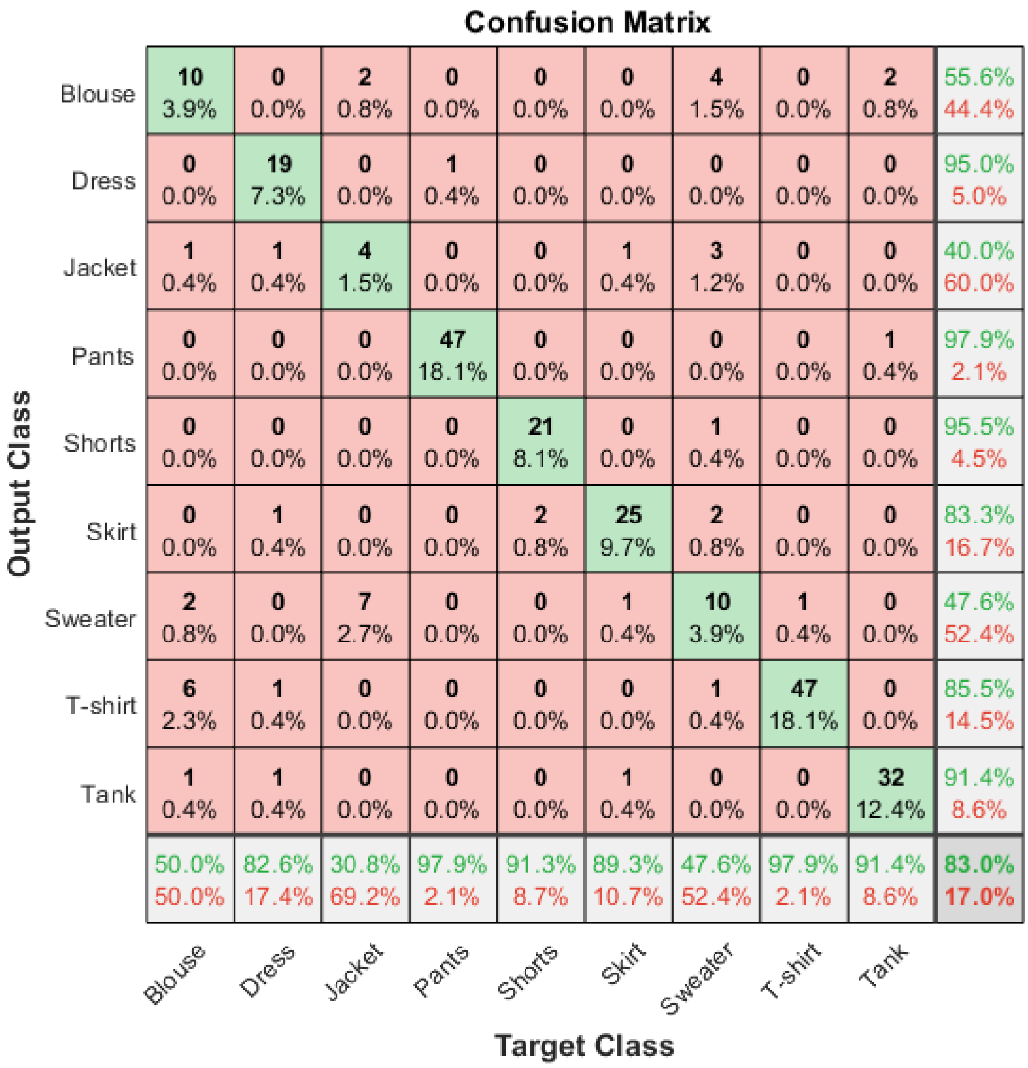
DenseNet:


Accuracy (%)
Iteration
Final
Loss
Results
Validation accuracy: 77.22%
Training finished: Reached final iteration
Training Time
Start time: 23-May-2020 23:10:45
Elapsed time: 425 min 38 sec
Training Cycle
Epoch: 100 of 100
Iteration: 12100 of 12100
Iterations per epoch: 121
Maximum iterations: 12100
Validation
Frequency: 150 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation


Confusion Matrix
Output Class
Target Class
Blouse
Dress
Jacket
Pants
Shorts
Skirt
Sweater
T-shirt
Tank
5 1.9% | 0 0.0% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.4% | 2 0.8% | 0 0.0% | 55.6% 44.4%
0 0.0% | 17 6.6% | 1 0.4% | 0 0.0% | 0 0.0% | 2 0.8% | 1 0.4% | 0 0.0% | 0 0.0% | 81.0% 19.0%
0 0.0% | 0 0.0% | 10 3.9% | 0 0.0% | 0 0.0% | 0 0.0% | 3 1.2% | 0 0.0% | 1 0.4% | 71.4% 28.6%
0 0.0% | 0 0.0% | 0 0.0% | 46 17.8% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.4% | 95.8% 4.2%
3 1.2% | 2 0.8% | 1 0.4% | 2 0.8% | 21 8.1% | 9 3.5% | 1 0.4% | 3 1.2% | 1 0.4% | 48.8% 51.2%
1 0.4% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 17 6.6% | 0 0.0% | 0 0.0% | 1 0.4% | 85.0% 15.0%
6 2.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 14 5.4% | 1 0.4% | 1 0.4% | 63.6% 36.4%
4 1.5% | 2 0.8% | 0 0.0% | 0 0.0% | 1 0.4% | 0 0.0% | 1 0.4% | 42 16.2% | 2 0.8% | 80.8% 19.2%
1 0.4% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 28 10.8% | 93.3% 6.7%
25.0% 75.0% | 73.9% 26.1% | 76.9% 23.1% | 95.8% 4.2% | 91.3% 8.7% | 60.7% 39.3% | 66.7% 33.3% | 87.5% 12.5% | 80.0% 20.0% | 77.2% 22.8%

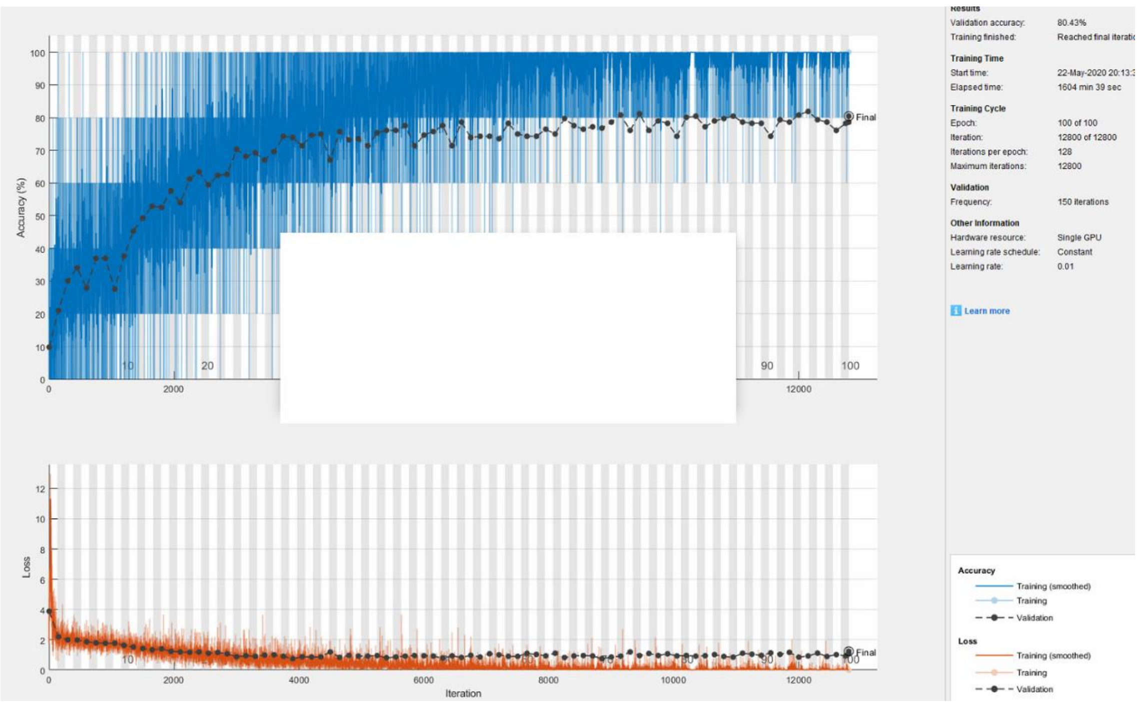
Nasnet-Mobile:

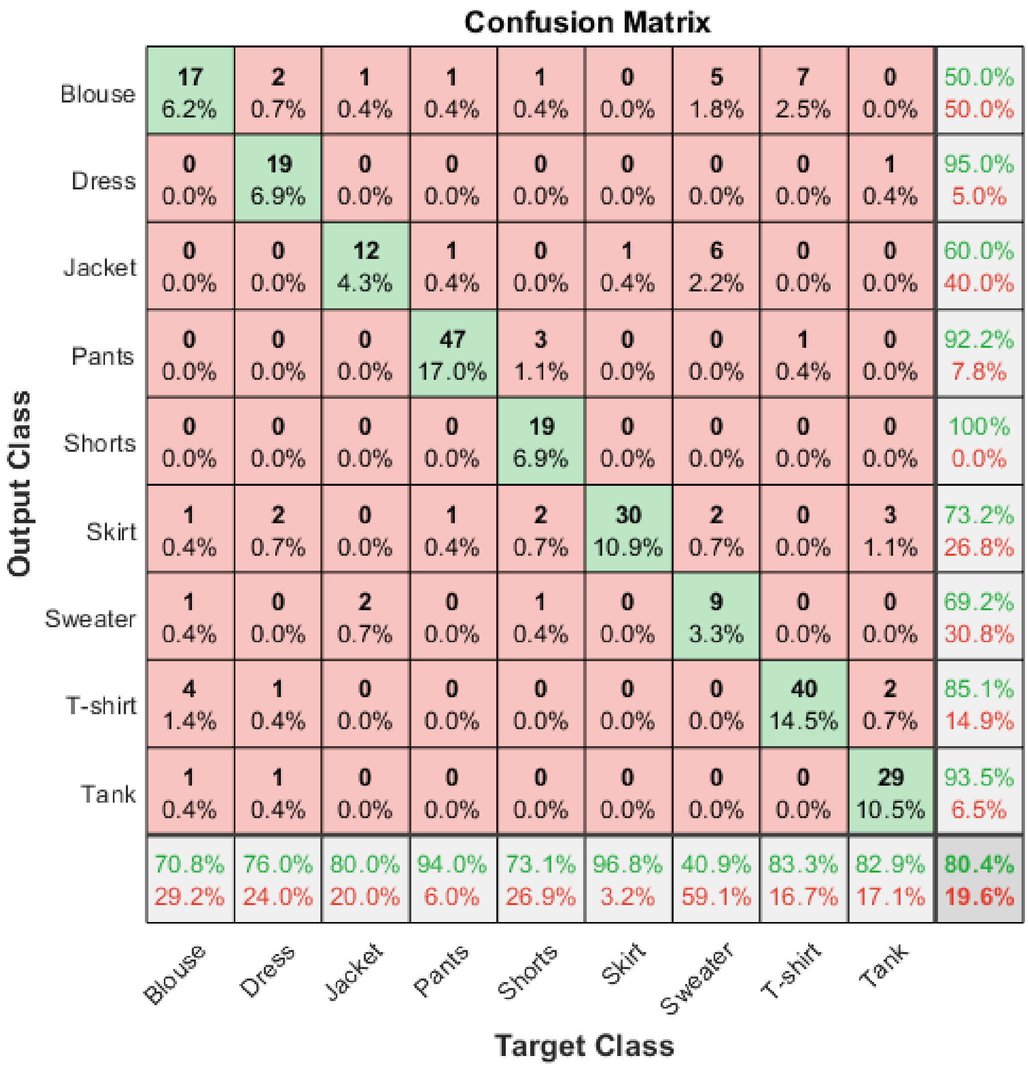



Confusion Matrix
Output Class
Target Class
Blouse
Dress
Jacket
Pants
Shorts
Skirt
Sweater
T-shirt
Tank
10 3.9% | 0 0.0% | 2 0.8% | 0 0.0% | 0 0.0% | 0 0.0% | 4 1.5% | 0 0.0% | 2 0.8% | 55.6% 44.4%
0 0.0% | 19 7.3% | 0 0.0% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 95.0% 5.0%
1 0.4% | 1 0.4% | 4 1.5% | 0 0.0% | 0 0.0% | 1 0.4% | 3 1.2% | 0 0.0% | 0 0.0% | 40.0% 60.0%
0 0.0% | 0 0.0% | 0 0.0% | 47 18.1% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.4% | 97.9% 2.1%
0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 21 8.1% | 0 0.0% | 1 0.4% | 0 0.0% | 0 0.0% | 95.5% 4.5%
0 0.0% | 1 0.4% | 0 0.0% | 0 0.0% | 2 0.8% | 25 9.7% | 2 0.8% | 0 0.0% | 0 0.0% | 83.3% 16.7%
2 0.8% | 0 0.0% | 7 2.7% | 0 0.0% | 0 0.0% | 1 0.4% | 10 3.9% | 1 0.4% | 0 0.0% | 47.6% 52.4%
6 2.3% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.4% | 47 18.1% | 0 0.0% | 85.5% 14.5%
1 0.4% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.4% | 0 0.0% | 0 0.0% | 32 12.4% | 91.4% 8.6%
50.0% 50.0% | 82.6% 17.4% | 30.8% 69.2% | 97.9% 2.1% | 91.3% 8.7% | 89.3% 10.7% | 47.6% 52.4% | 97.9% 2.1% | 91.4% 8.6% | 83.0% 17.0%


Blouse, 65.9%
T-shirt, 98.2%
Sweater, 99.8%
Pants, 100%
Sweater, 100%
Tank, 95.9%
Pants, 100%
T-shirt, 100%
Sweater, 95.4%
Skirt, 99.6%
Pants, 100%
Tank, 99.4%
Sweater, 49.8%
Tank, 100%
Pants, 100%
Blouse, 81.3%

NasNet:

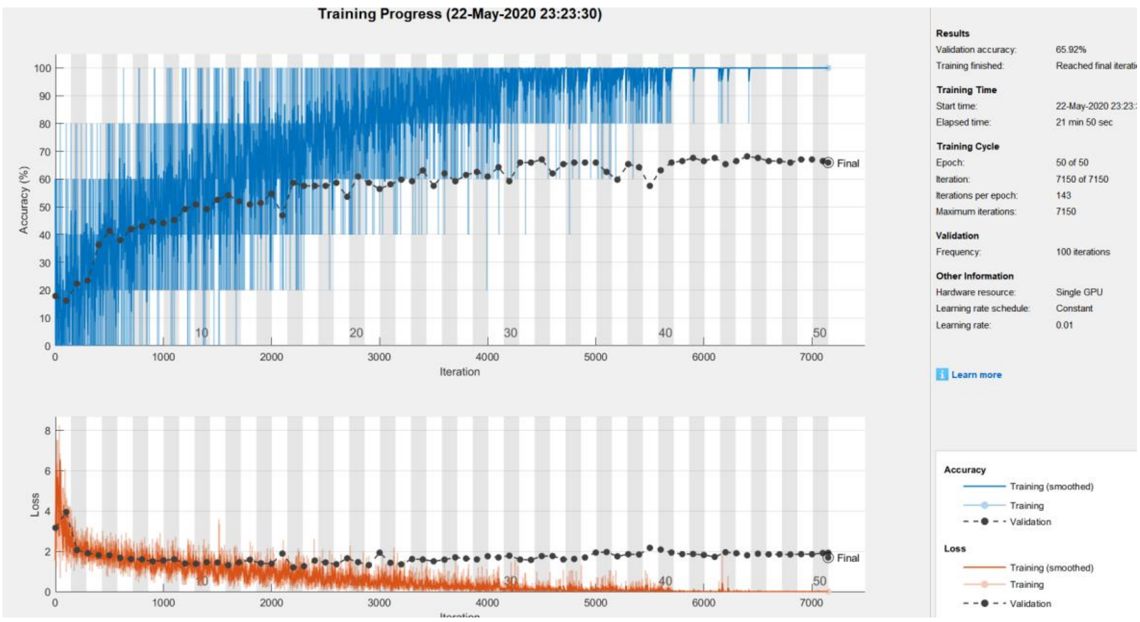

Accuracy (%)
Loss
Iteration
Final
Validation accuracy: 80.43%
Training Time
Start time:
Elapsed time: 1604 min 39 sec
Training Cycle
Epoch: 100 of 100
Iteration: 12800 of 12800
Iterations per epoch: 128
Maximum iterations: 12800
Validation
Frequency: 150 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation

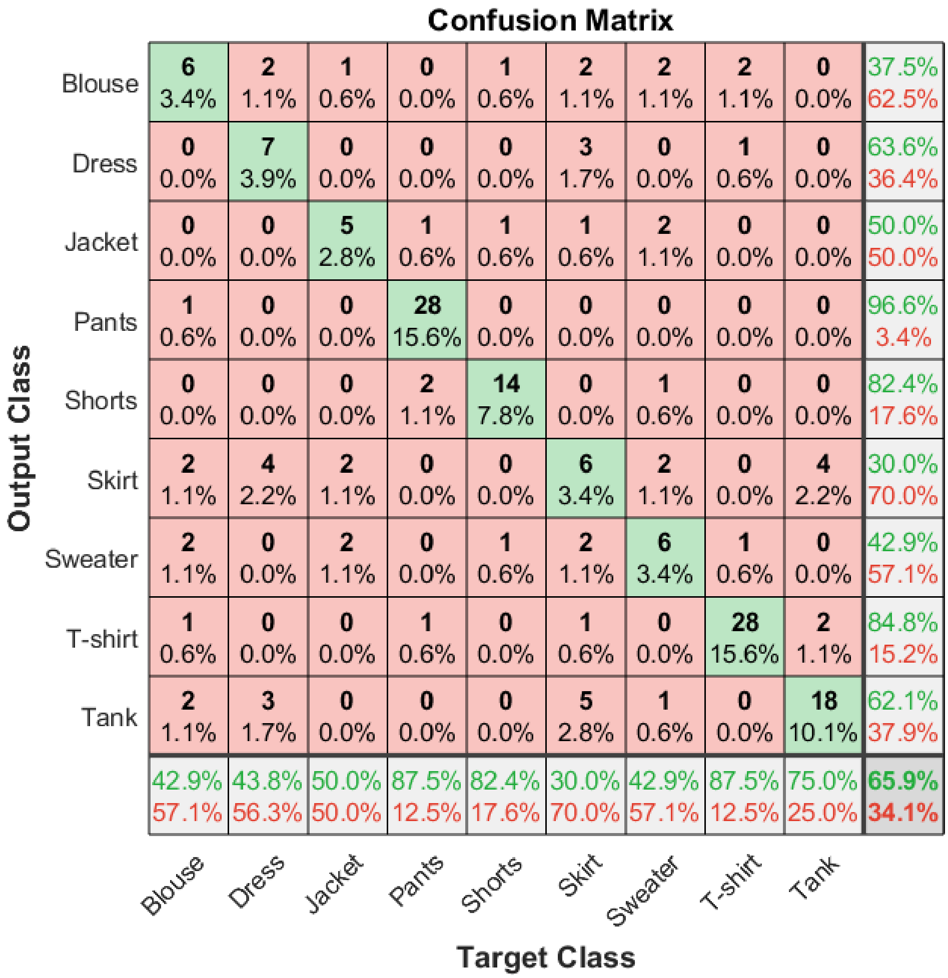


Confusion Matrix
Output Class
Target Class
Blouse
Dress
Jacket
Pants
Shorts
Skirt
Sweater
T-shirt
Tank
17 6.2% | 2 0.7% | 1 0.4% | 1 0.4% | 1 0.4% | 0 0.0% | 5 1.8% | 7 2.5% | 0 0.0% | 50.0% 50.0%
0 0.0% | 19 6.9% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.4% | 95.0% 5.0%
0 0.0% | 0 0.0% | 12 4.3% | 1 0.4% | 0 0.0% | 1 0.4% | 6 2.2% | 0 0.0% | 0 0.0% | 60.0% 40.0%
0 0.0% | 0 0.0% | 0 0.0% | 47 17.0% | 3 1.1% | 0 0.0% | 0 0.0% | 1 0.4% | 0 0.0% | 92.2% 7.8%
0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 19 6.9% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 100% 0.0%
1 0.4% | 2 0.7% | 0 0.0% | 1 0.4% | 2 0.7% | 30 10.9% | 2 0.7% | 0 0.0% | 3 1.1% | 73.2% 26.8%
1 0.4% | 0 0.0% | 2 0.7% | 0 0.0% | 1 0.4% | 0 0.0% | 9 3.3% | 0 0.0% | 0 0.0% | 69.2% 30.8%
4 1.4% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 40 14.5% | 2 0.7% | 85.1% 14.9%
1 0.4% | 1 0.4% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 29 10.5% | 93.5% 6.5%
70.8% 29.2% | 76.0% 24.0% | 80.0% 20.0% | 94.0% 6.0% | 73.1% 26.9% | 96.8% 3.2% | 40.9% 59.1% | 83.3% 16.7% | 82.9% 17.1% | 80.4% 19.6%


Shorts, 68%
Jacket, 100%
Blouse, 96.8%
Sweater, 86.1%
Blouse, 57.3%
T-shirt, 100%
Dress, 100%
Blouse, 77.9%
Pants, 100%
Skirt, 100%
Skirt, 52.6%
T-shirt, 100%
T-shirt, 99.3%
Pants, 100%
Skirt, 100%
Skirt, 100%

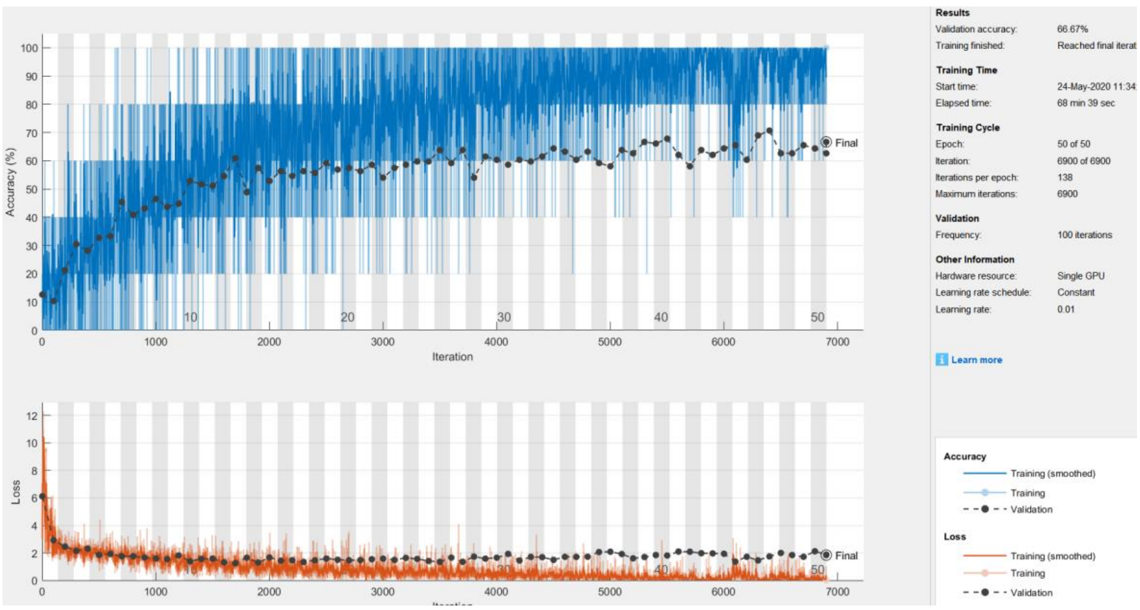
Resnet-18:

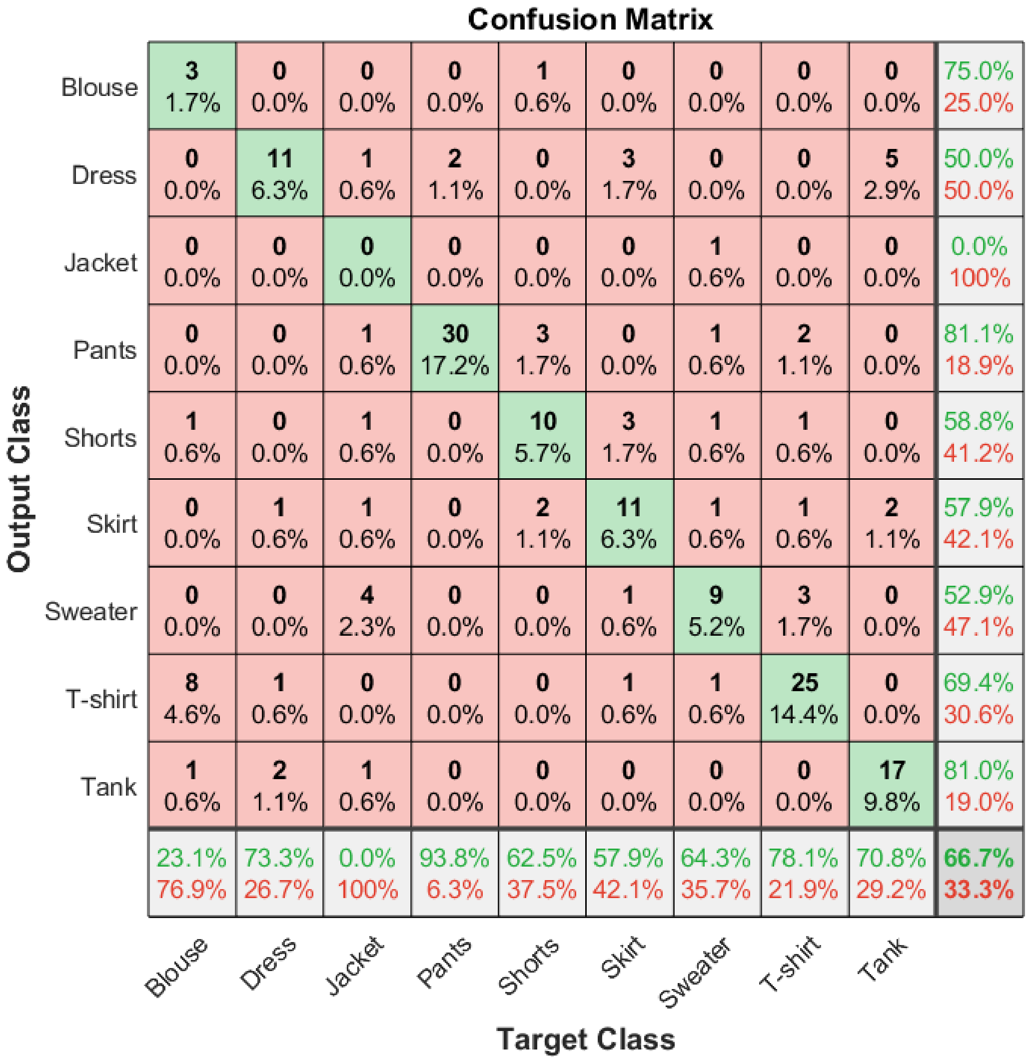

Training Progress (22-May-2020 23:23:30)
Accuracy (%)
Iteration
Final
Loss
Final
Results
Validation accuracy: 65.92%
Training finished: Reached final iterati
Training Time
Start time: 22-May-2020 23:23:
Elapsed time: 21 min 50 sec
Training Cycle
Epoch: 50 of 50
Iteration: 7150 of 7150
Iterations per epoch: 143
Maximum iterations: 7150
Validation
Frequency: 100 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation


Confusion Matrix
Output Class
Target Class
Blouse
Dress
Jacket
Pants
Shorts
Skirt
Sweater
T-shirt
Tank
6 3.4% | 2 1.1% | 1 0.6% | 0 0.0% | 1 0.6% | 2 1.1% | 2 1.1% | 2 1.1% | 0 0.0% | 37.5% 62.5%
0 0.0% | 7 3.9% | 0 0.0% | 0 0.0% | 0 0.0% | 3 1.7% | 0 0.0% | 1 0.6% | 0 0.0% | 63.6% 36.4%
0 0.0% | 0 0.0% | 5 2.8% | 1 0.6% | 1 0.6% | 1 0.6% | 2 1.1% | 0 0.0% | 0 0.0% | 50.0% 50.0%
1 0.6% | 0 0.0% | 0 0.0% | 28 15.6% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 96.6% 3.4%
0 0.0% | 0 0.0% | 0 0.0% | 2 1.1% | 14 7.8% | 0 0.0% | 1 0.6% | 0 0.0% | 0 0.0% | 82.4% 17.6%
2 1.1% | 4 2.2% | 2 1.1% | 0 0.0% | 0 0.0% | 6 3.4% | 2 1.1% | 0 0.0% | 4 2.2% | 30.0% 70.0%
2 1.1% | 0 0.0% | 2 1.1% | 0 0.0% | 1 0.6% | 2 1.1% | 6 3.4% | 1 0.6% | 0 0.0% | 42.9% 57.1%
1 0.6% | 0 0.0% | 0 0.0% | 1 0.6% | 0 0.0% | 1 0.6% | 0 0.0% | 28 15.6% | 2 1.1% | 84.8% 15.2%
2 1.1% | 3 1.7% | 0 0.0% | 0 0.0% | 0 0.0% | 5 2.8% | 1 0.6% | 0 0.0% | 18 10.1% | 62.1% 37.9%
42.9% 57.1% | 43.8% 56.3% | 50.0% 50.0% | 87.5% 12.5% | 82.4% 17.6% | 30.0% 70.0% | 42.9% 57.1% | 87.5% 12.5% | 75.0% 25.0% | 65.9% 34.1%


Sweater, 100%
Pants, 99.8%
Dress, 100%
Skirt, 95.9%
T-shirt, 100%
T-shirt, 100%
Tank, 100%
Jacket, 88.9%
Skirt, 100%
T-shirt, 100%
Dress, 100%
T-shirt, 100%
T-shirt, 100%
Jacket, 97%
Blouse, 100%
Shorts, 100%

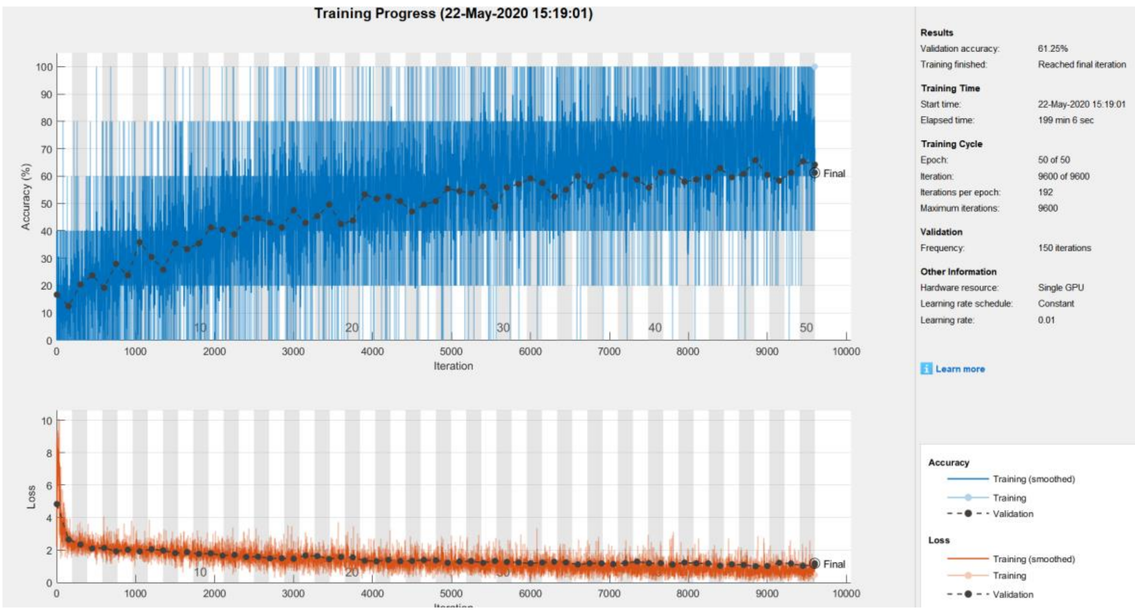
Resnet-50:

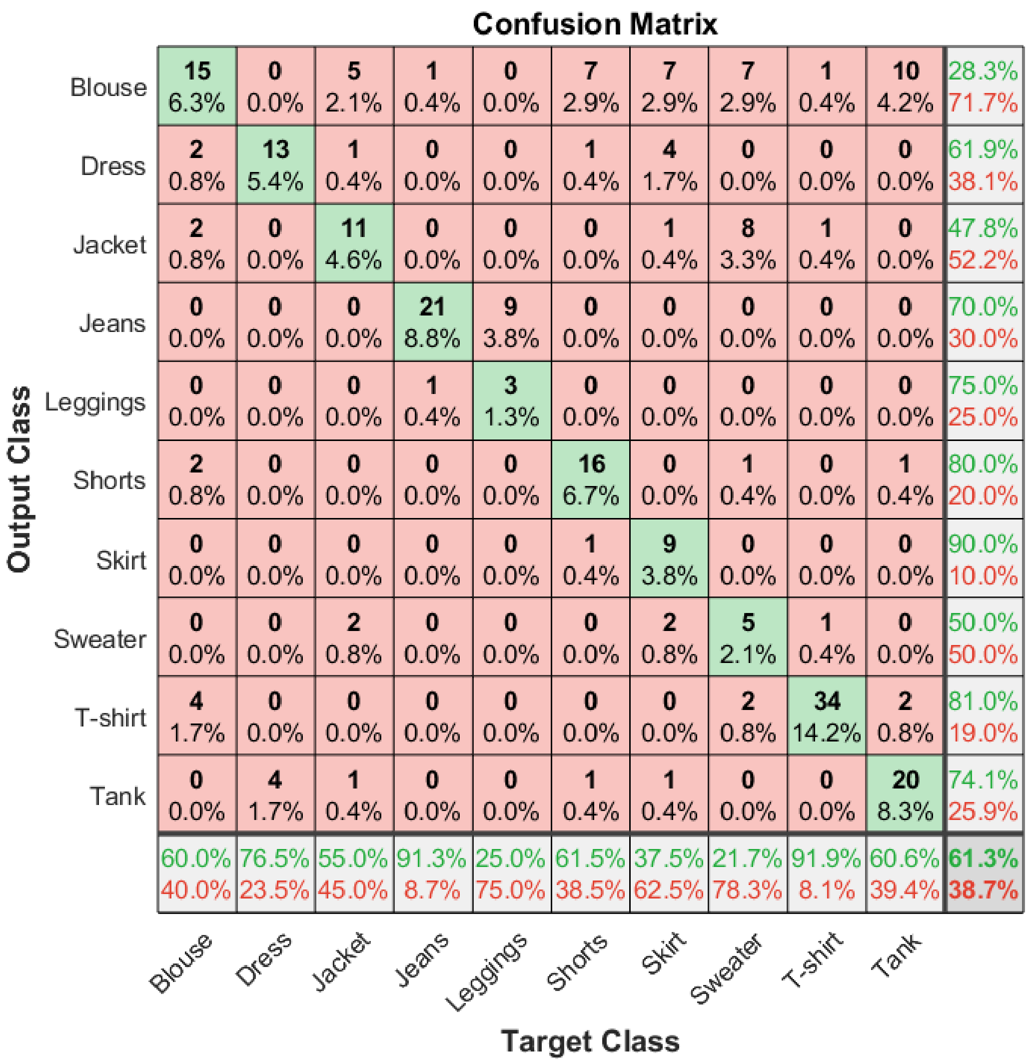

Accuracy (%)
Iteration
Loss
Final
Results
Validation accuracy: 66.67%
Training finished: Reached final iterat
Training Time
Start time: 24-May-2020 11:34
Elapsed time: 68 min 39 sec
Training Cycle
Epoch: 50 of 50
Iteration: 6900 of 6900
Iterations per epoch: 138
Maximum iterations: 6900
Validation
Frequency: 100 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation


Confusion Matrix
Output Class
Target Class
Blouse
Dress
Jacket
Pants
Shorts
Skirt
Sweater
T-shirt
Tank
3 1.7% 0 0.0% 0 0.0% 0 0.0% 1 0.6% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 75.0% 25.0%
0 0.0% 11 6.3% 1 0.6% 2 1.1% 0 0.0% 3 1.7% 0 0.0% 0 0.0% 5 2.9% 50.0% 50.0%
0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 1 0.6% 0 0.0% 0 0.0% 0.0% 100%
0 0.0% 0 0.0% 1 0.6% 30 17.2% 3 1.7% 0 0.0% 1 0.6% 2 1.1% 0 0.0% 81.1% 18.9%
1 0.6% 0 0.0% 1 0.6% 0 0.0% 10 5.7% 3 1.7% 1 0.6% 1 0.6% 0 0.0% 58.8% 41.2%
0 0.0% 1 0.6% 1 0.6% 0 0.0% 2 1.1% 11 6.3% 1 0.6% 1 0.6% 2 1.1% 57.9% 42.1%
0 0.0% 0 0.0% 4 2.3% 0 0.0% 0 0.0% 1 0.6% 9 5.2% 3 1.7% 0 0.0% 52.9% 47.1%
8 4.6% 1 0.6% 0 0.0% 0 0.0% 0 0.0% 1 0.6% 1 0.6% 25 14.4% 0 0.0% 69.4% 30.6%
1 0.6% 2 1.1% 1 0.6% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 17 9.8% 81.0% 19.0%
23.1% 76.9% 73.3% 26.7% 0.0% 100% 93.8% 6.3% 62.5% 37.5% 57.9% 42.1% 64.3% 35.7% 78.1% 21.9% 70.8% 29.2% 66.7% 33.3%

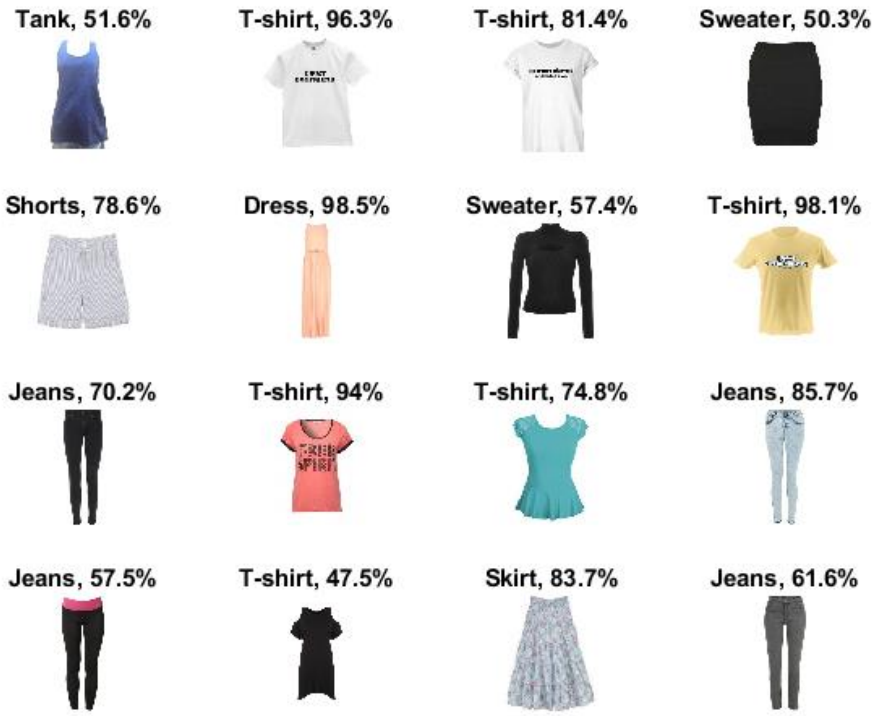

T-shirt, 64%
T-shirt, 100%
Pants, 100%
Sweater, 100%
Pants, 99.8%
Sweater, 84%
Shorts, 58.2%
Blouse, 89.3%
Skirt, 87%
Sweater, 67.9%
Tank, 83.3%
Shorts, 98.9%
Skirt, 98%
Pants, 100%
Pants, 99.9%
T-shirt, 100%

Resnet-101:

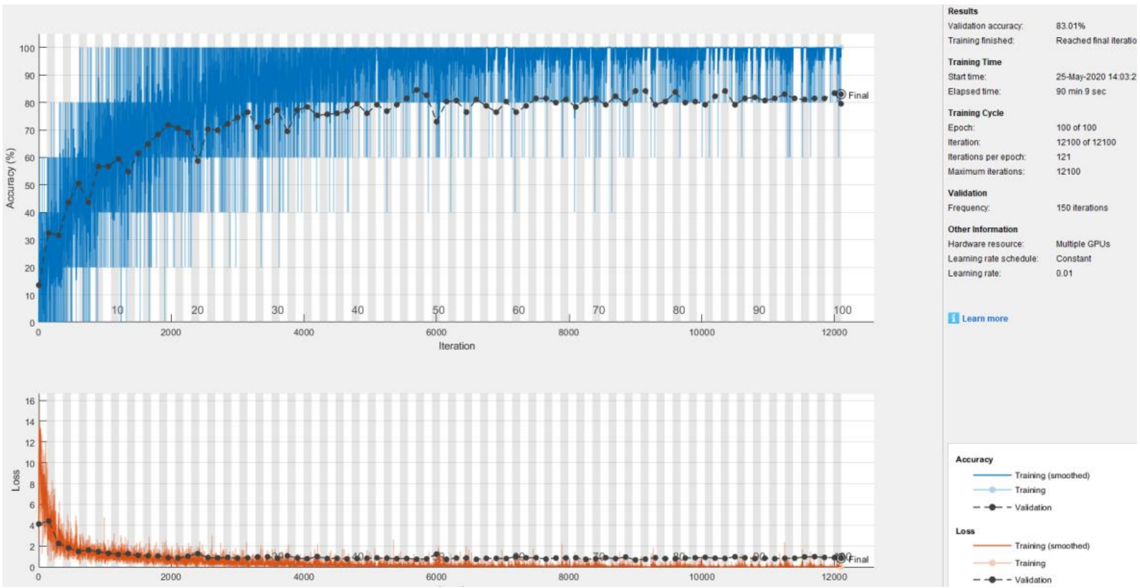

Training Progress (22-May-2020 15:19:01)
Accuracy (%)
Iteration
Loss
Final
Results
Validation accuracy: 61.25%
Training finished: Reached final iteration
Training Time
Start time: 22-May-2020 15:19:01
Elapsed time: 199 min 6 sec
Training Cycle
Epoch: 50 of 50
Iteration: 9600 of 9600
Iterations per epoch: 192
Maximum iterations: 9600
Validation
Frequency: 150 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation

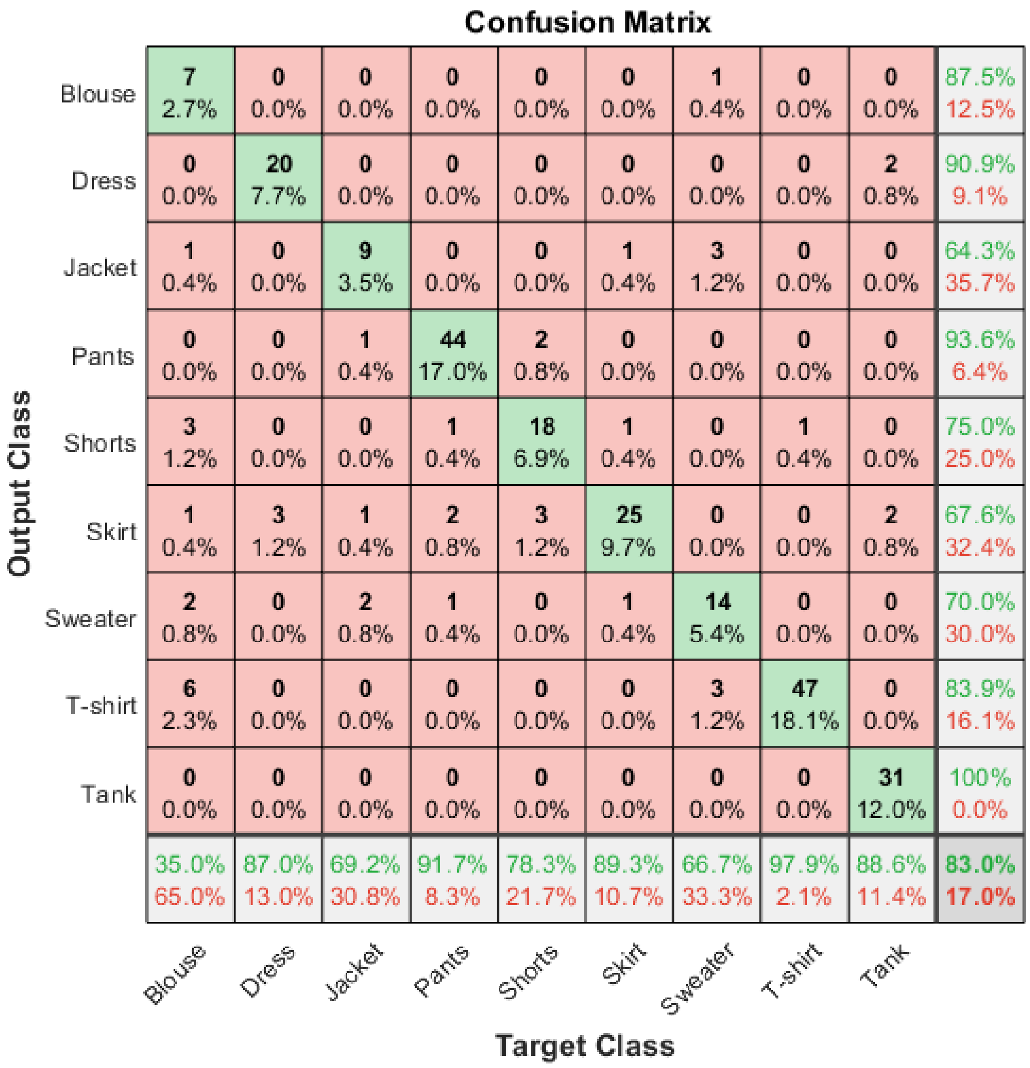


Confusion Matrix
Output Class
Target Class
Blouse 15 6.3% 0 0.0% 5 2.1% 1 0.4% 0 0.0% 7 2.9% 7 2.9% 7 2.9% 1 0.4% 10 4.2% 28.3% 71.7%
Dress 2 0.8% 13 5.4% 1 0.4% 0 0.0% 0 0.0% 1 0.4% 4 1.7% 0 0.0% 0 0.0% 0 0.0% 61.9% 38.1%
Jacket 2 0.8% 0 0.0% 11 4.6% 0 0.0% 0 0.0% 0 0.0% 1 0.4% 8 3.3% 1 0.4% 0 0.0% 47.8% 52.2%
Jeans 0 0.0% 0 0.0% 0 0.0% 21 8.8% 9 3.8% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 70.0% 30.0%
Leggings 0 0.0% 0 0.0% 0 0.0% 1 0.4% 3 1.3% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 75.0% 25.0%
Shorts 2 0.8% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 16 6.7% 0 0.0% 1 0.4% 0 0.0% 1 0.4% 80.0% 20.0%
Skirt 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 1 0.4% 9 3.8% 0 0.0% 0 0.0% 0 0.0% 90.0% 10.0%
Sweater 0 0.0% 0 0.0% 2 0.8% 0 0.0% 0 0.0% 0 0.0% 2 0.8% 5 2.1% 1 0.4% 0 0.0% 50.0% 50.0%
T-shirt 4 1.7% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 2 0.8% 34 14.2% 2 0.8% 81.0% 19.0%
Tank 0 0.0% 4 1.7% 1 0.4% 0 0.0% 0 0.0% 1 0.4% 1 0.4% 0 0.0% 0 0.0% 20 8.3% 74.1% 25.9%
60.0% 40.0% 76.5% 23.5% 55.0% 45.0% 91.3% 8.7% 25.0% 75.0% 61.5% 38.5% 37.5% 62.5% 21.7% 78.3% 91.9% 8.1% 60.6% 39.4% 61.3% 38.7%
Blouse Dress Jacket Jeans Leggings Shorts Skirt Sweater T-shirt Tank


Tank, 51.6%
T-shirt, 96.3%
T-shirt, 81.4%
Sweater, 50.3%
Shorts, 78.6%
Dress, 98.5%
Sweater, 57.4%
T-shirt, 98.1%
Jeans, 70.2%
T-shirt, 94%
T-shirt, 74.8%
Jeans, 85.7%
Jeans, 57.5%
T-shirt, 47.5%
Skirt, 83.7%
Jeans, 61.6%

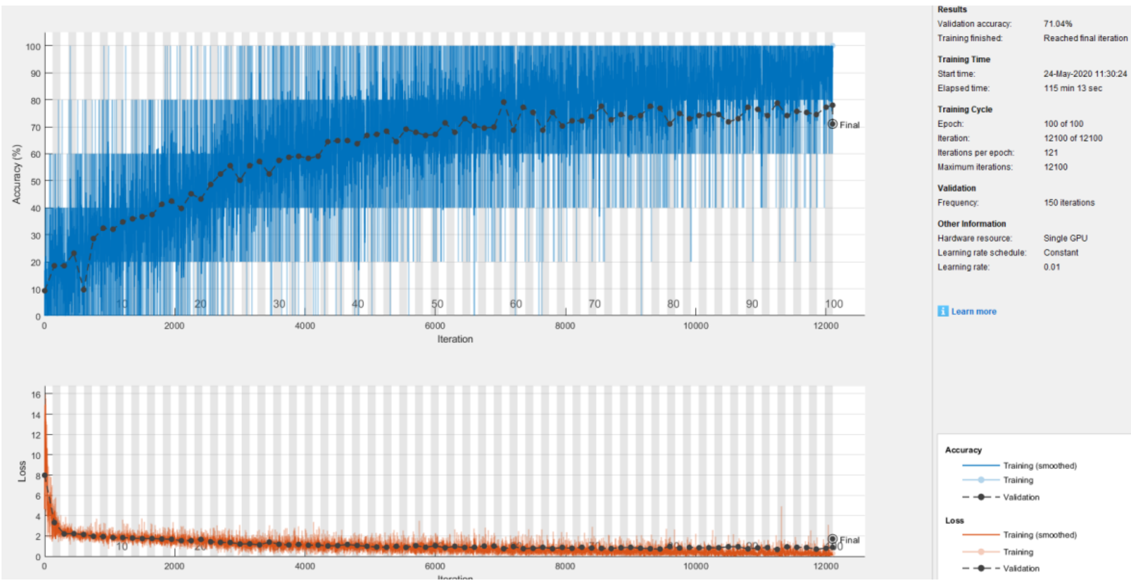
ShuffleNet:

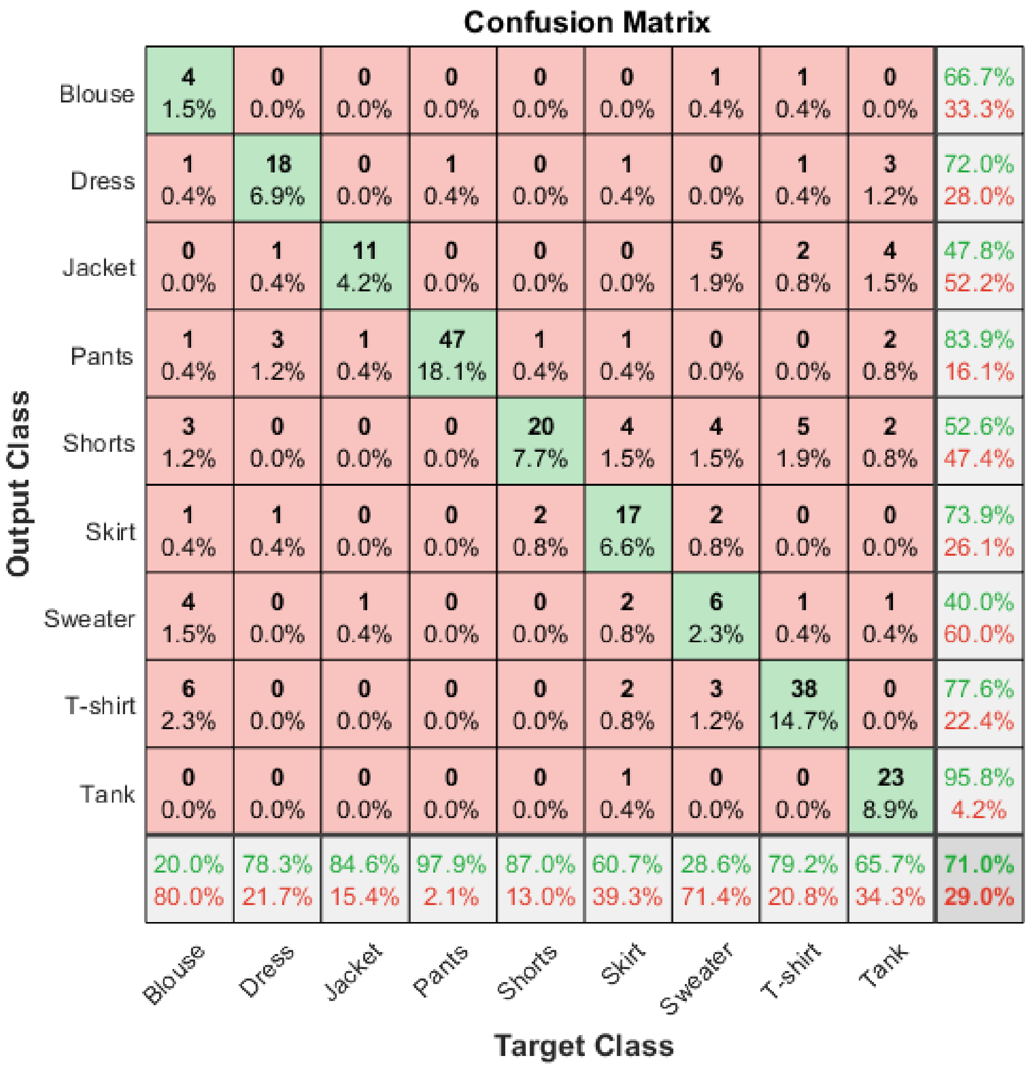

Accuracy (%)
Iteration
Loss
Final
Results
Validation accuracy: 83.01%
Training finished: Reached final iteratio
Training Time
Start time: 25-May-2020 14:03:2
Elapsed time: 90 min 9 sec
Training Cycle
Epoch: 100 of 100
Iteration: 12100 of 12100
Iterations per epoch: 121
Maximum iterations: 12100
Validation
Frequency: 150 iterations
Other Information
Hardware resource: Multiple GPUs
Learning rate schedule: Constant
Learning rate: 0.01
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation


Confusion Matrix
Output Class
Blouse 7 2.7% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 1 0.4% 0 0.0% 0 0.0% 87.5% 12.5%
Dress 0 0.0% 20 7.7% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 2 0.8% 90.9% 9.1%
Jacket 1 0.4% 0 0.0% 9 3.5% 0 0.0% 0 0.0% 1 0.4% 3 1.2% 0 0.0% 0 0.0% 64.3% 35.7%
Pants 0 0.0% 0 0.0% 1 0.4% 44 17.0% 2 0.8% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 93.6% 6.4%
Shorts 3 1.2% 0 0.0% 0 0.0% 1 0.4% 18 6.9% 1 0.4% 0 0.0% 1 0.4% 0 0.0% 75.0% 25.0%
Skirt 1 0.4% 3 1.2% 1 0.4% 2 0.8% 3 1.2% 25 9.7% 0 0.0% 0 0.0% 2 0.8% 67.6% 32.4%
Sweater 2 0.8% 0 0.0% 2 0.8% 1 0.4% 0 0.0% 1 0.4% 14 5.4% 0 0.0% 0 0.0% 70.0% 30.0%
T-shirt 6 2.3% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 3 1.2% 47 18.1% 0 0.0% 83.9% 16.1%
Tank 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 31 12.0% 100% 0.0%
35.0% 65.0% 87.0% 13.0% 69.2% 30.8% 91.7% 8.3% 78.3% 21.7% 89.3% 10.7% 66.7% 33.3% 97.9% 2.1% 88.6% 11.4% 83.0% 17.0%
Blouse Dress Jacket Pants Shorts Skirt Sweater T-shirt Tank
Target Class


Skirt, 100%
Pants, 100%
Skirt, 82.4%
Sweater, 91.8%
Shorts, 38%
Pants, 100%
Dress, 99.5%
Skirt, 84.1%
Pants, 100%
Dress, 91.7%
Sweater, 100%
Jacket, 74%
Dress, 99.9%
Skirt, 39.2%
T-shirt, 100%
Sweater, 100%

YoloV3:

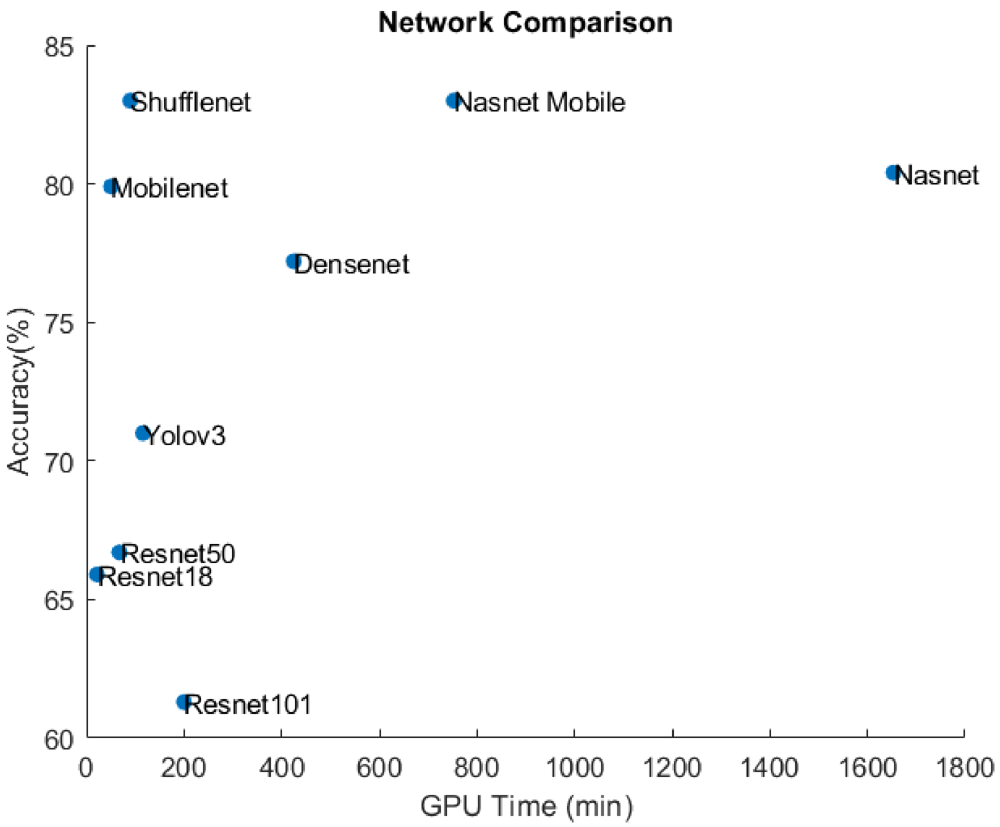

Accuracy (%)
Iteration
Loss
Final
Results
Validation accuracy: 71.04%
Training finished: Reached final iteration
Training Time
Start time: 24-May-2020 11:30:24
Elapsed time: 115 min 13 sec
Training Cycle
Epoch: 100 of 100
Iteration: 12100 of 12100
Iterations per epoch: 121
Maximum iterations: 12100
Validation
Frequency: 150 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation


Confusion Matrix
Blouse
Dress
Jacket
Pants
Shorts
Skirt
Sweater
T-shirt
Tank
4 1.5% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.4% | 1 0.4% | 0 0.0% | 66.7% 33.3%
1 0.4% | 18 6.9% | 0 0.0% | 1 0.4% | 0 0.0% | 1 0.4% | 0 0.0% | 1 0.4% | 3 1.2% | 72.0% 28.0%
0 0.0% | 1 0.4% | 11 4.2% | 0 0.0% | 0 0.0% | 0 0.0% | 5 1.9% | 2 0.8% | 4 1.5% | 47.8% 52.2%
1 0.4% | 3 1.2% | 1 0.4% | 47 18.1% | 1 0.4% | 1 0.4% | 0 0.0% | 0 0.0% | 2 0.8% | 83.9% 16.1%
3 1.2% | 0 0.0% | 0 0.0% | 0 0.0% | 20 7.7% | 4 1.5% | 4 1.5% | 5 1.9% | 2 0.8% | 52.6% 47.4%
1 0.4% | 1 0.4% | 0 0.0% | 0 0.0% | 2 0.8% | 17 6.6% | 2 0.8% | 0 0.0% | 0 0.0% | 73.9% 26.1%
4 1.5% | 0 0.0% | 1 0.4% | 0 0.0% | 0 0.0% | 2 0.8% | 6 2.3% | 1 0.4% | 1 0.4% | 40.0% 60.0%
6 2.3% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 2 0.8% | 3 1.2% | 38 14.7% | 0 0.0% | 77.6% 22.4%
0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 0 0.0% | 1 0.4% | 0 0.0% | 0 0.0% | 23 8.9% | 95.8% 4.2%
20.0% 80.0% | 78.3% 21.7% | 84.6% 15.4% | 97.9% 2.1% | 87.0% 13.0% | 60.7% 39.3% | 28.6% 71.4% | 79.2% 20.8% | 65.7% 34.3% | 71.0% 29.0%
Output Class
Target Class

Comparison Table and Graph:

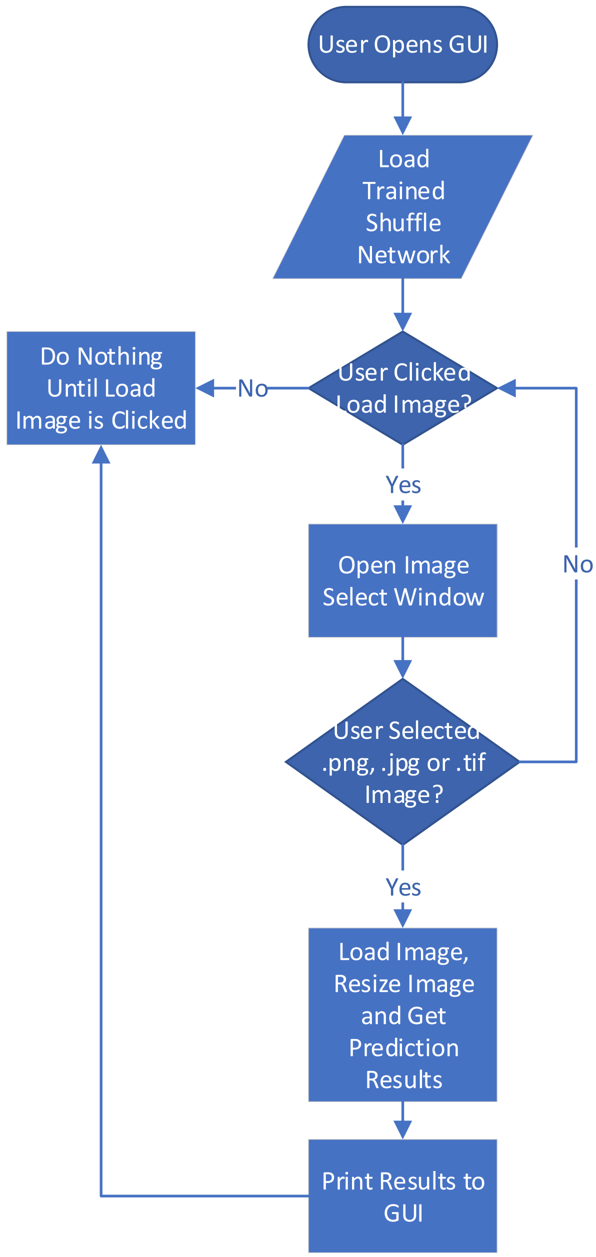
Method	Nasnet	Shufflenet	Mobilenet	Nasnet Mobile	Resnet18
Validation Accuracy (%)	80.4	83.0	79.9	83.0	65.9
GPU Time (min)	1653	90	51	753	22
Method	**Resnet50**	**Resnet101**	**Densenet**	**Yolov3**	
Validation Accuracy (%)	66.7	61.3	77.2	71.0	
GPU Time (min)	68	200	425	116	

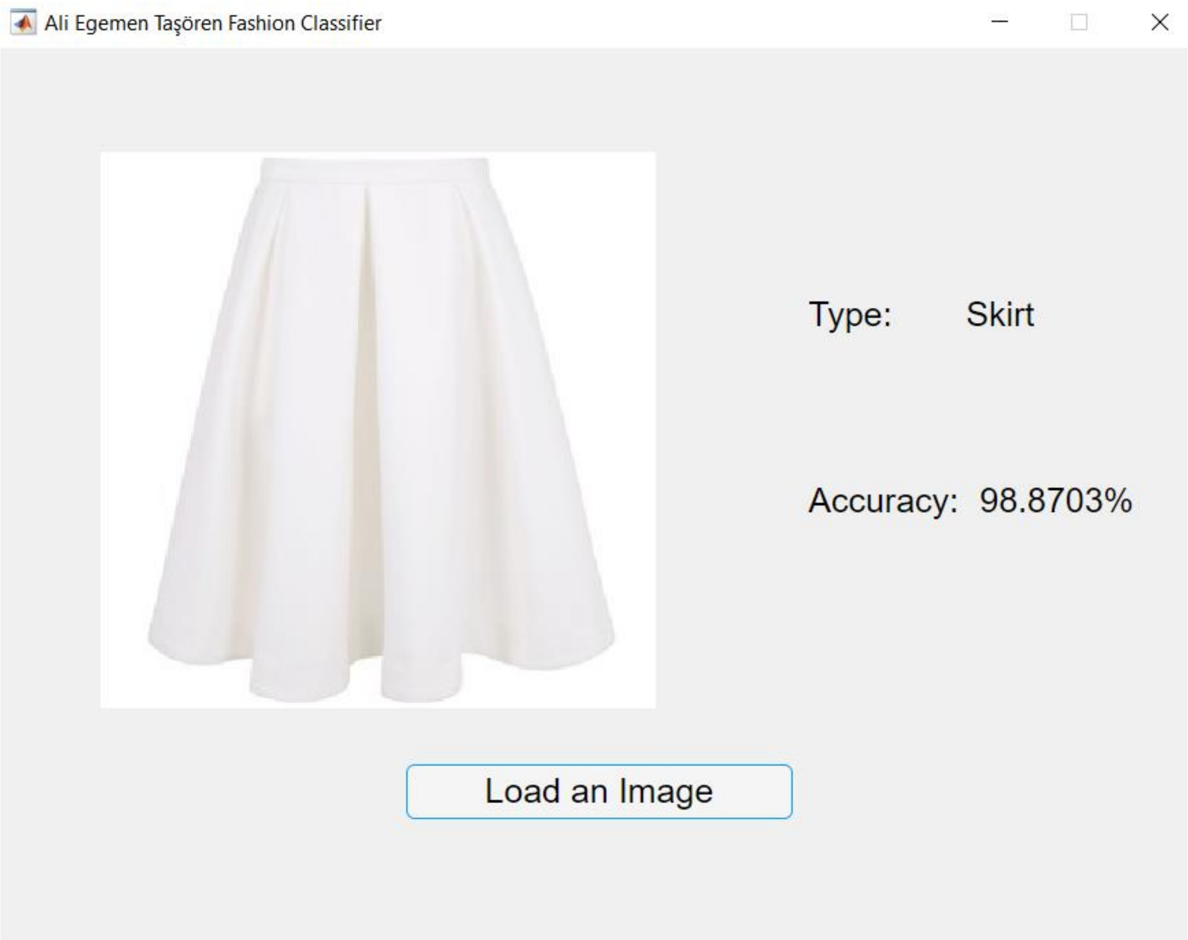
Here GPU Time is the time it takes to train a network with Nvidia 2080Ti 11 GB GPU.



Designed GUI:

A GUI is designed for the project. With this GUI user can select any .jpg, .tif, and .png file in their computer and check it's class and prediction accuracy using shufflenet. The selection of the shufflenet depends on its highest accuracy and low GPU memory needs compared to other models. Below the flowchart and a picture can be seen.


User Opens GUI
Load Trained Shuffle Network
User Clicked Load Image?
No
Do Nothing Until Load Image is Clicked
Yes
Open Image Select Window
No
User Selected .png, .jpg or .tif Image?
Yes
Load Image, Resize Image and Get Prediction Results
Print Results to GUI



Conclusion:

Deepfashion v1 dataset is customized using the customization methods which are told in the dataset section, and a new dataset is curated with these customizations. This dataset later used in the training process of 9 different networks which are densenet, mobilenet, nasnet-mobile, nasnet, resnet18, resnet50, resnet101, shufflenet, yolov3.
From the results, Shufflenet proved its promise to work efficiently with extremely small datasets. Shufflenet shares the most accurate network position with Nasnet-Mobile. Even if the mobilenet proves low accuracy in MATLAB's imagenet test, with the custom dataset it provided near 80% accuracy. Nas-net still proved good, even if the custom dataset is very small compared to imagenet with 80% accuracy. Nasnet and mobile network is being followed by Densenet with 77% accuracy. In this test, however, yolov3 and resnet methods did not provide accurate findings compared to other methods. Yolov3 managed to pass 70% with a score of

71% meanwhile Resnet18 and resnet50 gave around 66% accuracy. The lowest accuracy achieved by the resnet101 method with 61.3%.
Foundings suggests with different networks, accuracy results might be different. Also, a bigger network (like resnet101) can perform worse than a smaller one (resnet18 or resnet50) even if it is not expected.
This can be caused by overfitting which came from the depth of the network.
Also smaller networks provided better accuracy compared to other models, this probably caused by pictures of the dataset being rather small since they don't have any pixels beside pictures pixels.
Foundings suggest that small datasets prove better results with shufflenet or mobilenet. Also, these networks use less GPU time compared to most of the methods they performed better from. From GPU times it can be said both of the Nasnet variants use way too much GPU power and let alone the small networks, it will be extremely long to train big networks with it. Slowest networks were Nasnet and Nasnet Mobile with 1653 and 753 minutes respectively. The fastest networks was Mobilenet and Resnet18 with 51 and 68 minutes for customized dataset. Even if the accuracy of the Mobilenet was one of the highest, Resnet18 failed to achieve high enough accuracy compared to the other methods. For this dataset, most suited networks are Mobilenet for fastening the GPU time, and Shufflenet for maximum accuracy without any compromises from GPU time.
Using Shufflenet, a GUI is built to classify fashion images. With this GUI any .jpg, .png, or .tif image can be classified for its clothing type.