****

**Ahsanullah University of Science and Technology**

**Department of Computer Science and Engineering**

**Course No. :** CSE 4238

**Course Name :** Soft Computing Lab

**Assignment No. :** 02

**Submitted By:**

Name : Emdadul Haque

ID No. : 17 01 04 028

Session : Fall - 2020

Section    :       A (A2)

Table of Contents

[1. Dataset Creation 4](#_Toc81514004)

[1.1. Data Downloading: 5](#_Toc81514005)

[1.2. Creating Dataset Proper Format 6](#_Toc81514006)

[2. Training Process 7](#_Toc81514007)

[2.1. Experiment 1 7](#_Toc81514008)

[2.2. Experiment 2 9](#_Toc81514009)

[2.2.1. Model-1 V2 9](#_Toc81514010)

[2.2.1. Model-1 V3 11](#_Toc81514011)

[2.2.2. Model-2-CNN 13](#_Toc81514012)

[2.2.3. Model-3-RSNET152 15](#_Toc81514013)

[2.3. Training Flow 16](#_Toc81514014)

[2.3.1. Hyper parameter 16](#_Toc81514015)

[2.3.2. Load Dataset 17](#_Toc81514016)

[2.3.3. Model Creation 17](#_Toc81514017)

[2.3.4. Model train and save the model 18](#_Toc81514018)

[3. Training Results 19](#_Toc81514019)

[3.1. Experiment 1 19](#_Toc81514020)

[3.1.1. Own(training-a) 19](#_Toc81514021)

[3.1.2. Fashion-MNIST 20](#_Toc81514022)

[3.2. Experiment 2 – Model 1-V2 21](#_Toc81514023)

[3.2.1. Own(training-a) 21](#_Toc81514024)

[3.2.2. Fashion-MNIST 22](#_Toc81514025)

[3.3. Experiment 2 – Model 1-V3 23](#_Toc81514026)

[3.3.1. Own(training-a) 23](#_Toc81514027)

[3.3.2. Fashion-MNIST 24](#_Toc81514028)

[3.4. Experiment 2 – Model 2 25](#_Toc81514029)

[3.4.1. Own(training-a) 25](#_Toc81514030)

[3.4.1.1. Input Channel 3(RGB-Image) 25](#_Toc81514031)

[3.4.1.2. Input Channel 1(GRAY-Image) 26](#_Toc81514032)

[3.4.2. Fashion-MNIST 27](#_Toc81514033)

[3.5. Experiment 2 – Model 3 28](#_Toc81514034)

[3.5.1. Own(training-a) 28](#_Toc81514035)

[3.5.1.1. Input Channel 3(RGB-Image) 28](#_Toc81514036)

[3.5.1.2. Input Channel 1(GRAY-Image) 29](#_Toc81514037)

[3.5.2. Fashion-MNIST 30](#_Toc81514038)

[4. Reference 31](#_Toc81514039)

Figure of Content

[Figure 1: Raw Dataset 5](#_Toc81514497)

[Figure 2: Data Download Code 5](#_Toc81514498)

[Figure 3: After format data 6](#_Toc81514499)

[Figure 4: Data format code 6](#_Toc81514500)

[Figure 5: Experiment 1, Model 1 7](#_Toc81514501)

[Figure 6: Model 1 summary 8](#_Toc81514502)

[Figure 7: Experiment 2, Model 1 V2, This is based on linear layer 9](#_Toc81514503)

[Figure 8: Model V2 flow 10](#_Toc81514504)

[Figure 9:Model 1 V3. 11](#_Toc81514505)

[Figure 10: Model 1 V3 Flow summary 12](#_Toc81514506)

[Figure 11: Model 3, Based on the CNN 13](#_Toc81514507)

[Figure 12: Model 2 flow summary 14](#_Toc81514508)

[Figure 13:Model 3, Based on the pretrained and flow of summary 15](#_Toc81514509)

[Figure 14: Training Parameter’s 16](#_Toc81514510)

[Figure 15: Dataset Transform 17](#_Toc81514511)

[Figure 16: Basic Visualization 17](#_Toc81514512)

[Figure 17: Model saved code 18](#_Toc81514513)

[Figure 18: Model Train Function 18](#_Toc81514514)

[Figure 19: Accuracy Graph Exp-1 20](#_Toc81514515)

[Figure 20: Loss Graph Exp-1 20](#_Toc81514516)

[Figure 21: Performance Matrice of exp-1 20](#_Toc81514517)

[Figure 22:Loss graph exp-2 21](#_Toc81514518)

[Figure 23: Accuracy graph exp-2 21](#_Toc81514519)

[Figure 24: Performance matrices of exp-2 model v2 21](#_Toc81514520)

[Figure 25: F-MINIST Exp-2 Loss 22](#_Toc81514521)

[Figure 26:F-MINIST Exp-2 Accuracy 22](#_Toc81514522)

[Figure 27: Performance matrices F- MINIST 22](#_Toc81514523)

[Figure 28: Loss graph exp-2 23](#_Toc81514524)

[Figure 29:Accuracy graph exp-2 23](#_Toc81514525)

[Figure 30: Performance Matrices 23](#_Toc81514526)

[Figure 31: Loss Graph of Model-3 28](#_Toc81514527)

[Figure 32:Accuracy Graph of Model 3 28](#_Toc81514528)

[Figure 33: Performance Matrices 28](#_Toc81514529)

[Figure 34:FMINIST loss in mode3 30](#_Toc81514530)

[Figure 35:Accuracy Model3 30](#_Toc81514531)

[Figure 36: Performance Matrices 30](#_Toc81514532)

# 1. Dataset Creation

In this section, we know about the creation process of the dataset.

For the data creation process, we divide it into two different sub-processes.

1. Data Download
2. Creating Dataset Proper Format

## Data Downloading:

In the data downloading process, first, we need to download the files based on the ID. My is 170104028. So, it is even ID. That’s why I need to download the [Dataset A](https://drive.google.com/file/d/1txyKhs1Zt5AKswGGK9VI_jE0JNHuQT85/view). In dataset we saw that this is Bangla Hand Written digit. After extracting the dataset, we can saw that there are two files. One is train training-a folder training-a.csv. In folder we see all the image. **training-a.csv** file have the information of the image. The two files content is-

|  |  |
| --- | --- |
|  |  |

Figure 1: Raw Dataset

Form this, dataset we saw that this is not properly formatted. That’s why we need to format the dataset.

For download the dataset I used gdown package. Because this is google drive link. For downloading I used given code-

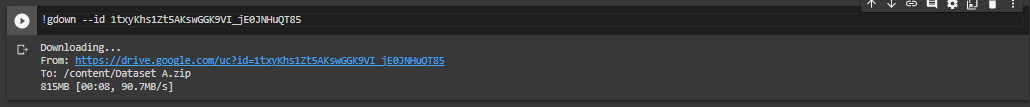


Figure 2: Data Download Code

## Creating Dataset Proper Format

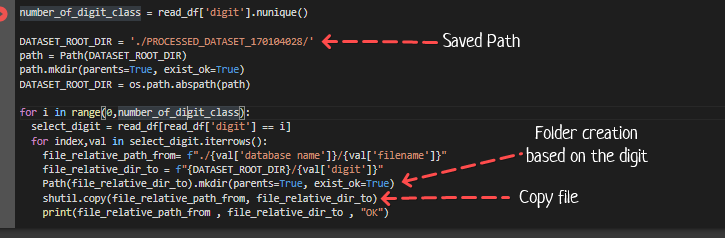
After Collecting the raw data. We need to generate our dataset. In to a good format. Because, next time when we train, we don’t need to change anything our dataset. Also, future, we can train our dataset easily. After formatted dataset the dataset looks like-

|  |  |
| --- | --- |
|  |  |

Figure 3: After format data

To format the dataset, we used the given code -





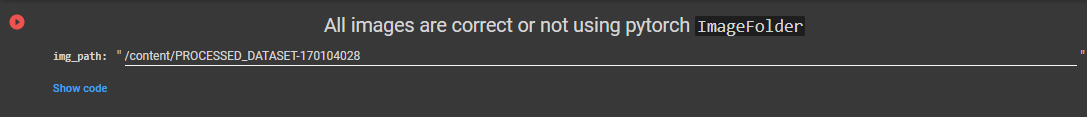




Figure 4: Data format code

# Training Process

For training process, we need to follow two parts. First one is Experiment 1 and second one is Experiment 2.

## Experiment 1

For experiment 1, must follow some rules. The rules are, we must use 6 hidden layer, 200 nodes, 0.01 learning rate etc. For model creation I use this code -

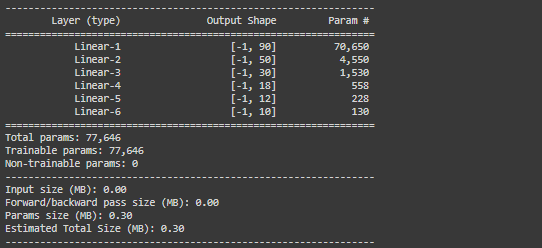
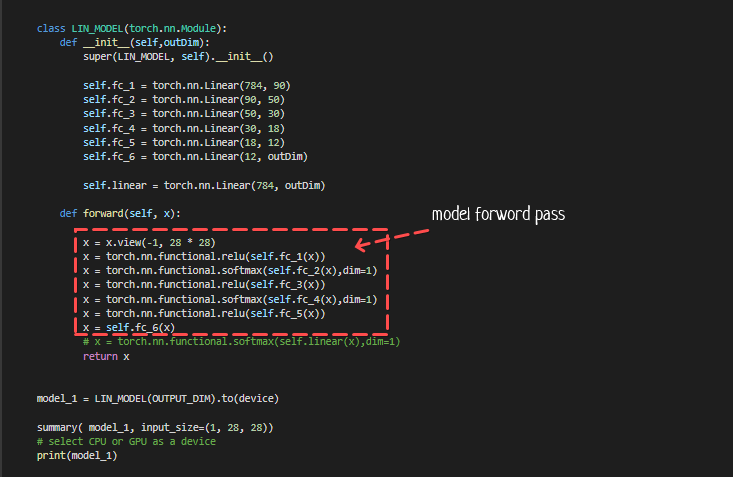


Figure 5: Experiment 1, Model 1

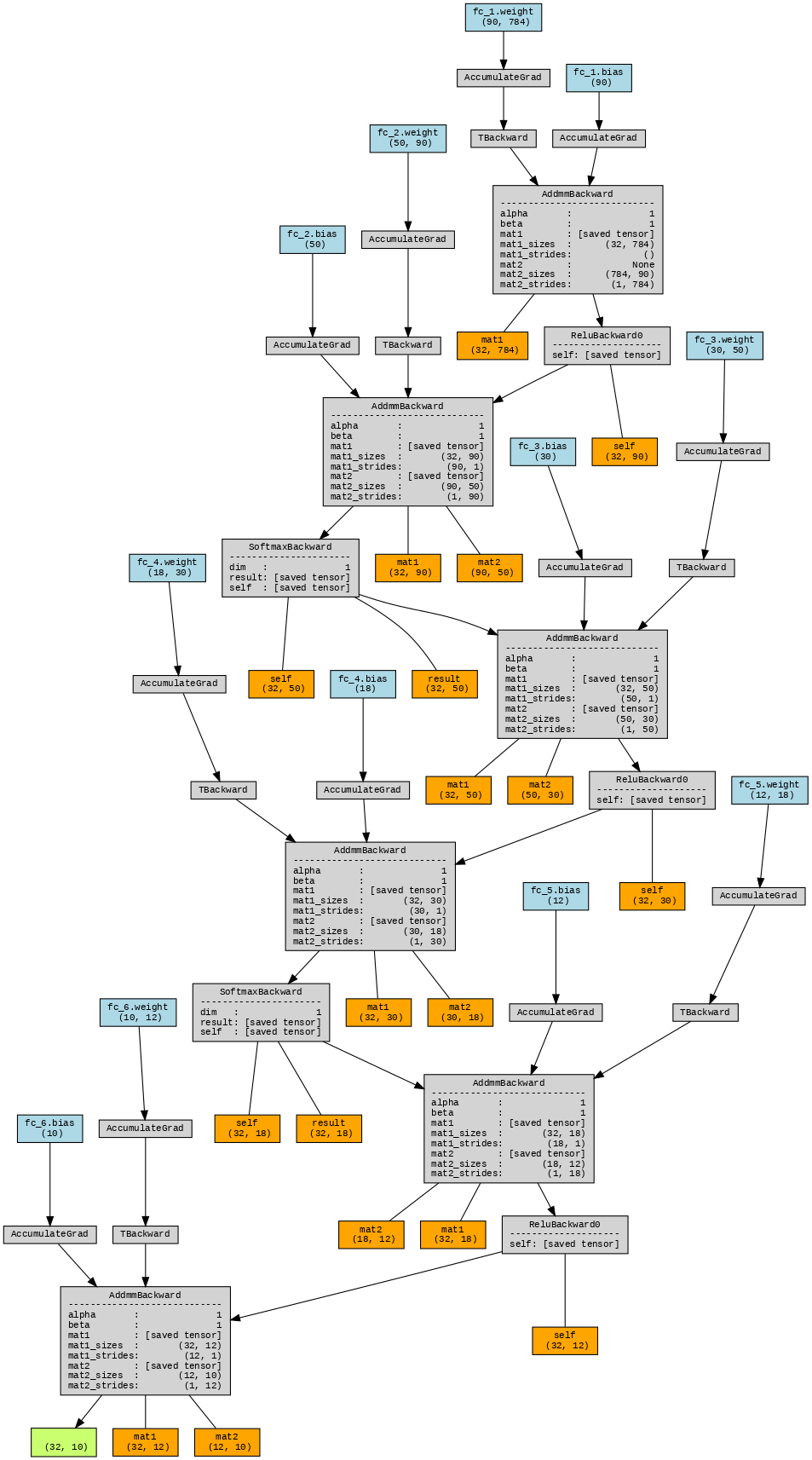
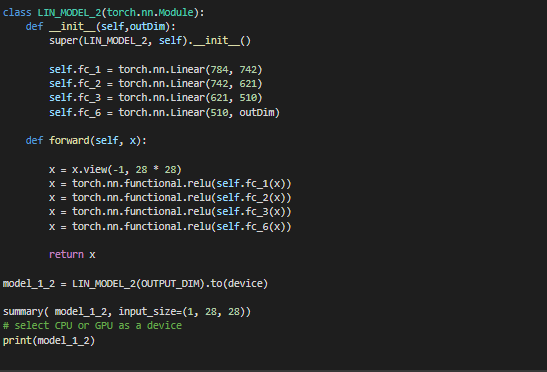


Figure 6: Model 1 summary

## Experiment 2

For experiment 2, must follow the 85% accuracy of dataset.

### 2.2.1. Model-1 V2



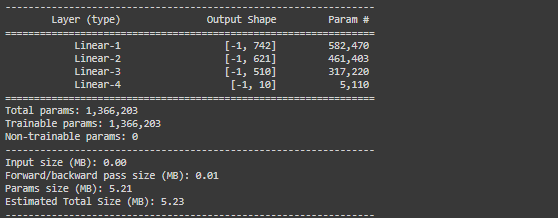


Figure 7: Experiment 2, Model 1 V2, This is based on linear layer

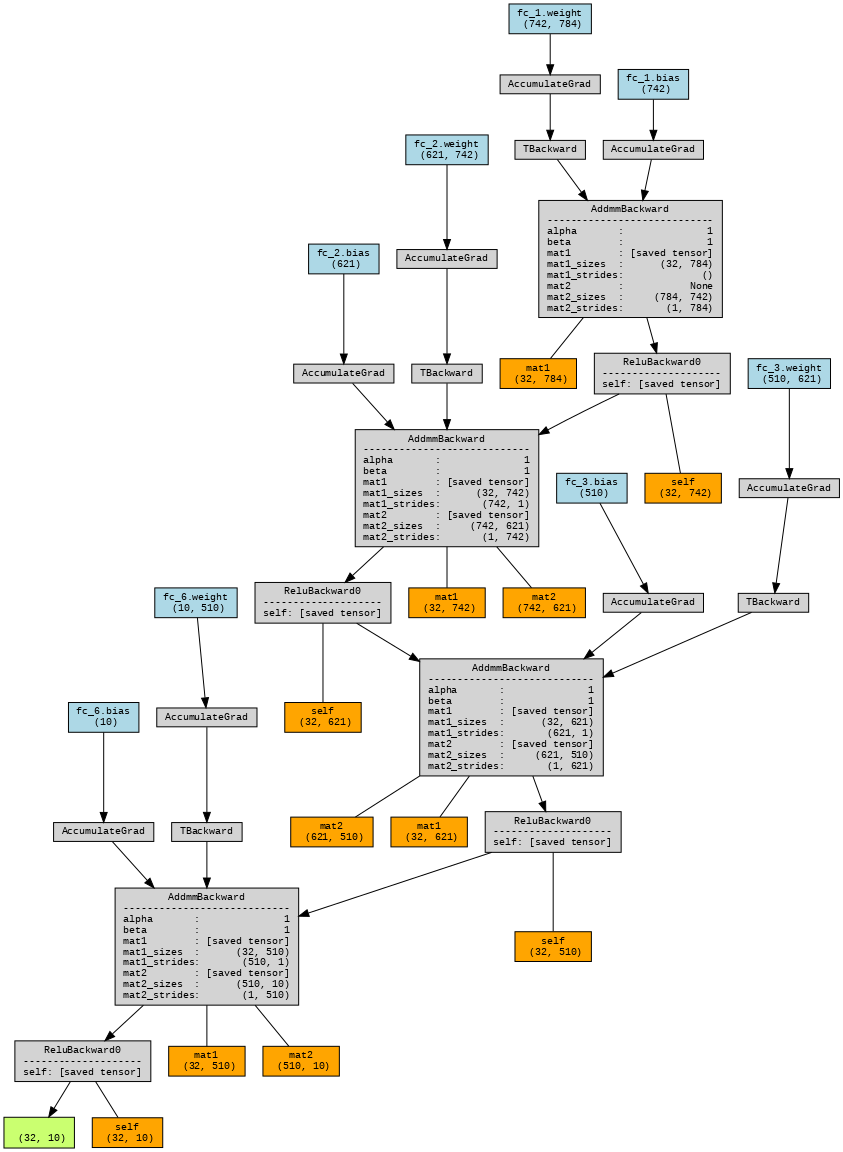


Figure 8: Model V2 flow

### Model-1 V3



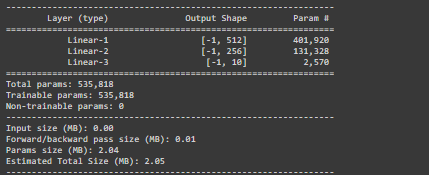


Figure 9:Model 1 V3.

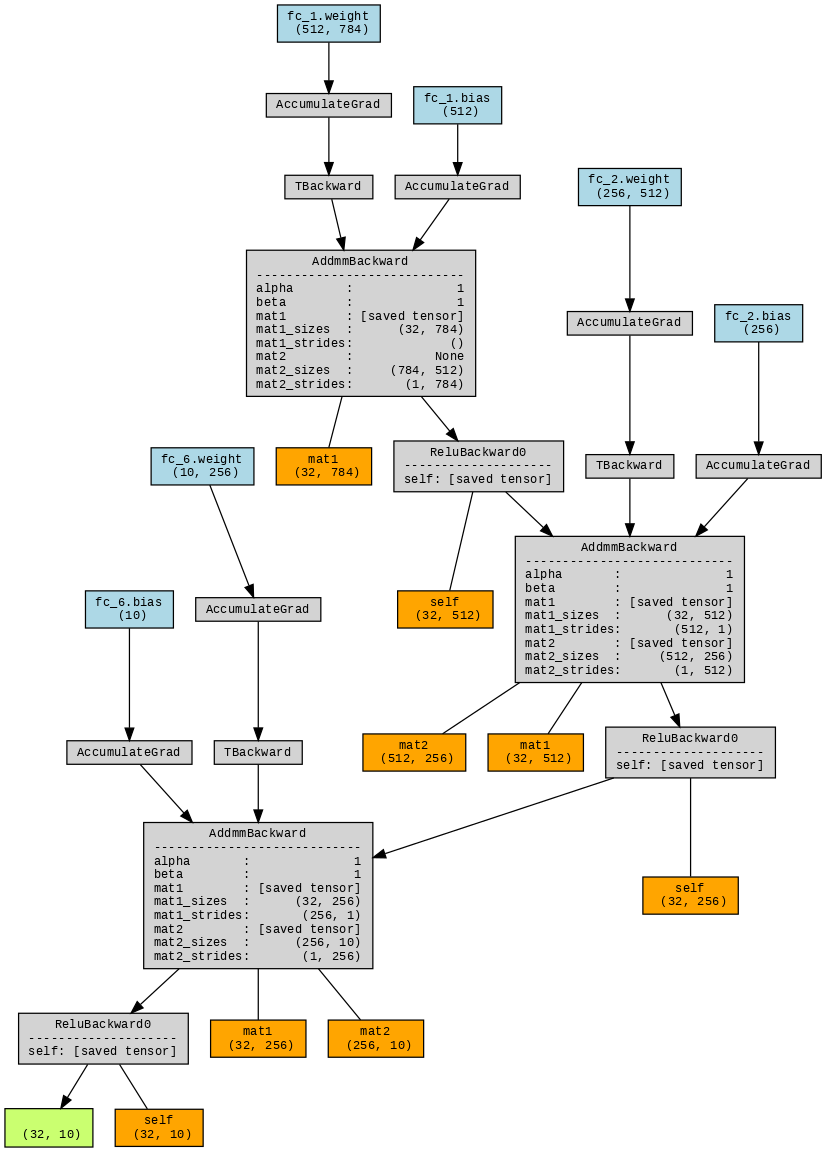
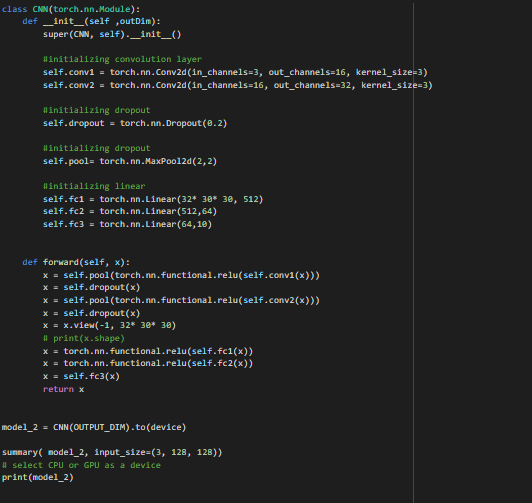


Figure 10: Model 1 V3 Flow summary

### 2.2.2. Model-2-CNN



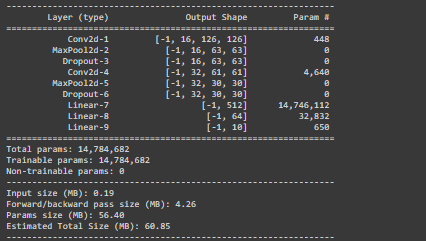
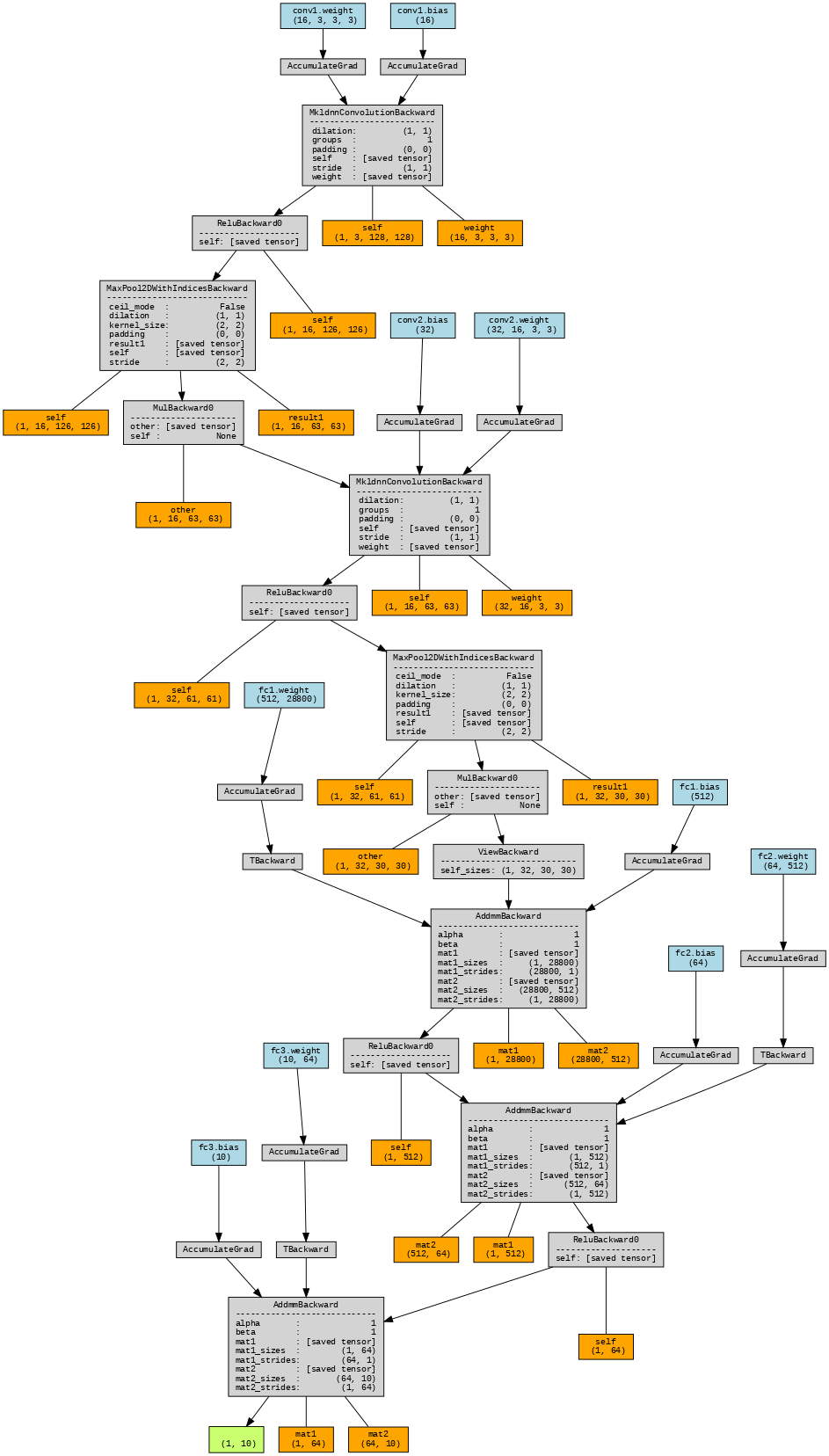
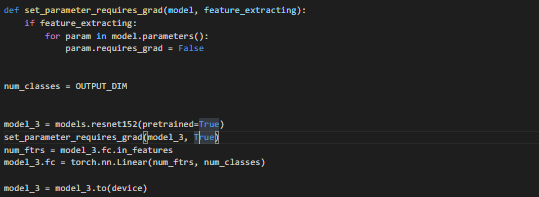


Figure 11: Model 3, Based on the CNN

Figure 12: Model 2 flow summary

### 2.2.3. Model-3-RSNET152



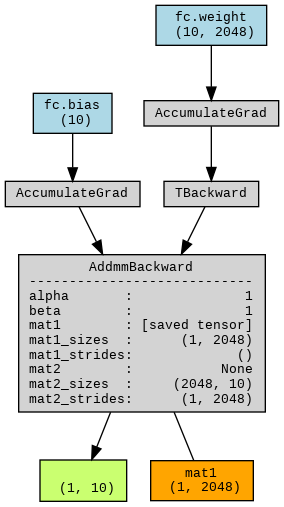


Figure 13:Model 3, Based on the pretrained and flow of summary

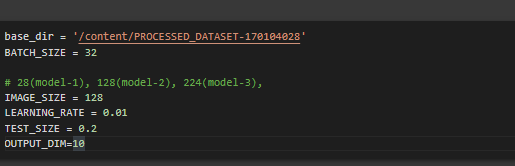
## Training Flow

For training the mode. I use 4 basic flows.

* 1. Hyper parameter and variables
  2. Load dataset
  3. Model Creation
  4. Model train and save the model

### Hyper parameter

For model hyper parameter and variables. I used this code.



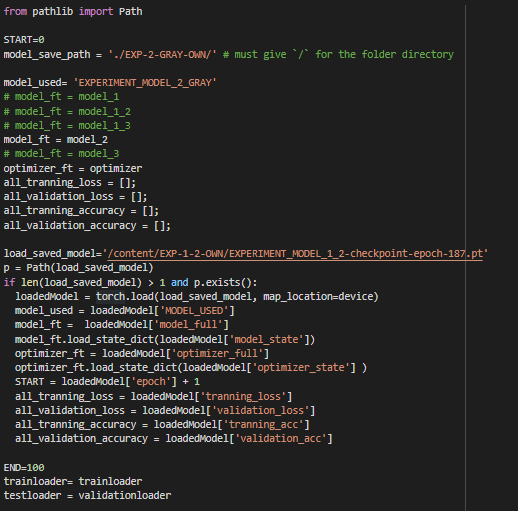


Figure 14: Training Parameter’s

### Load Dataset

For loading the dataset. I use ImageFolder pytorch function. And I use transforms of the dataset based on the model parameters. Mainly I used three shape of image there are 1\*28\*28, 1\*128\*128, 3\*128\*128, 3\*224\*224, 1\*224\*224. The main code is-

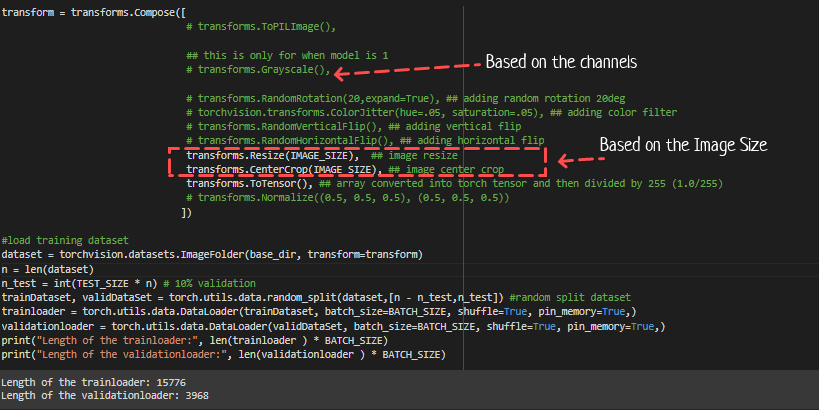


Figure 15: Dataset Transform

The basic visualization of dataset-



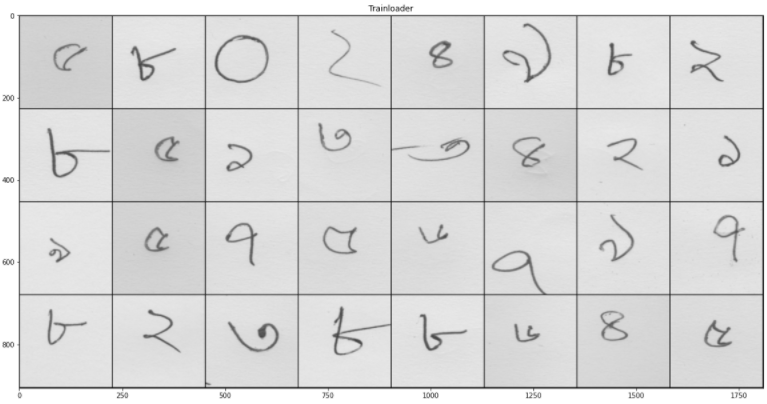
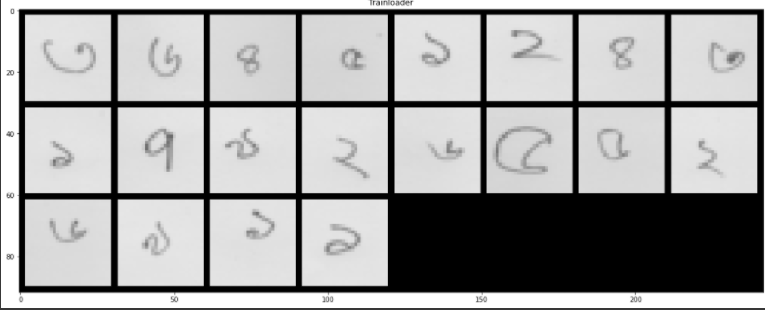


Figure 16: Basic Visualization

### Model Creation

In model creation I divide into 2 parts first one is the model class creation and other one optimizer & criterion. In Section2.1 and Section2.2 we saw all the model information. And the optimizer & criterion is given bellow-



### Model train and save the model

For training the model I use two functions first one is Model save and second one is model training function. For model saving the I use this function-

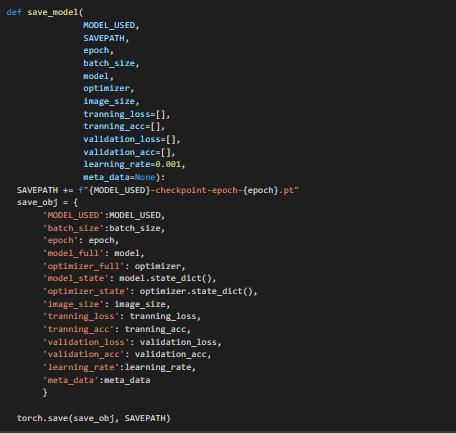
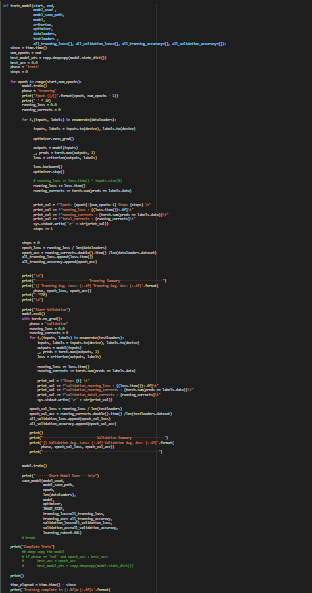
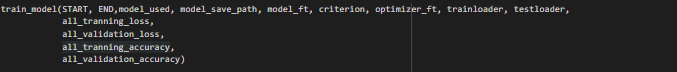


Figure 17: Model saved code

And the model train function is –



Figure 18: Model Train Function

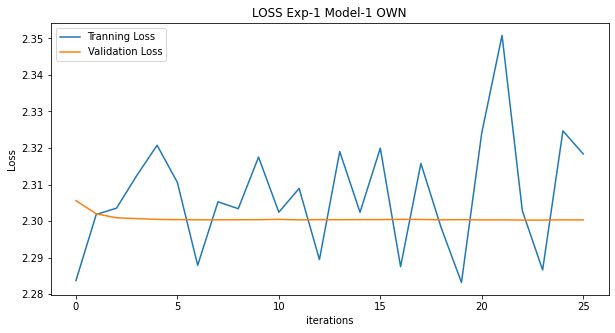
# Training Results

Here we can saw all the training result of model. Also you can see he parameters of the model training process.

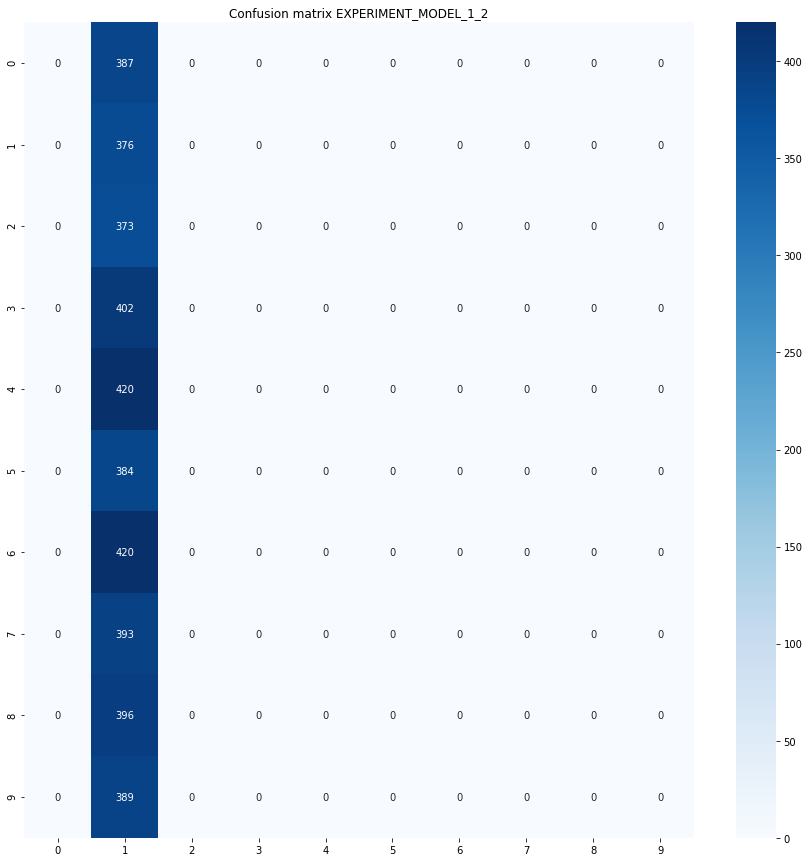
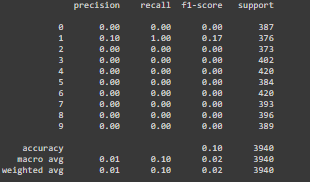
## Experiment 1

### Own(training-a)

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 19702 |
| Learning rate | 0.01 |
| Batch Size | 20 |
| Iteration | 20490 |
| Training and Testing Split | 80% and 20% |
| Image Shape | 1 \* 28 \* 28 |
| Optimizer | SGD |





### Fashion-MNIST

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 60000 |
| Learning rate | 0.01 |
| Batch Size | 20 |
| Iteration | 65000 |
| Image Shape | 1 \* 28 \* 28 |
| Optimizer | SGD |

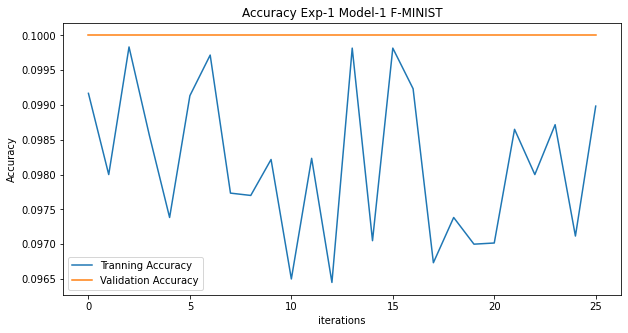
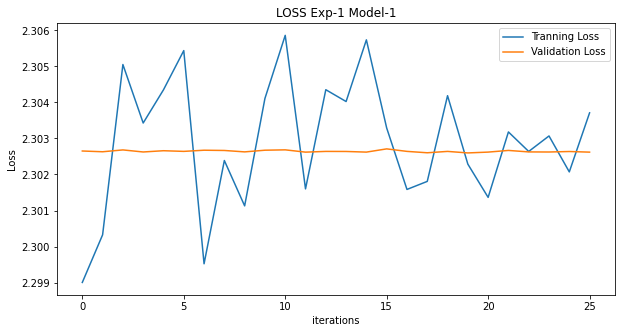


Figure 19: Accuracy Graph Exp-1

Figure 20: Loss Graph Exp-1

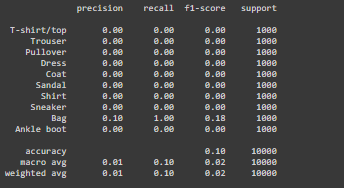


Figure 21: Performance Matrice of exp-1

## Experiment 2 – Model 1-V2

### Own(training-a)

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 19702 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 125715 |
| Training and Testing Split | 80% and 20% |
| Image Shape | 1 \* 28 \* 28 |
| Epochs | 255 |
| Optimizer | SGD |

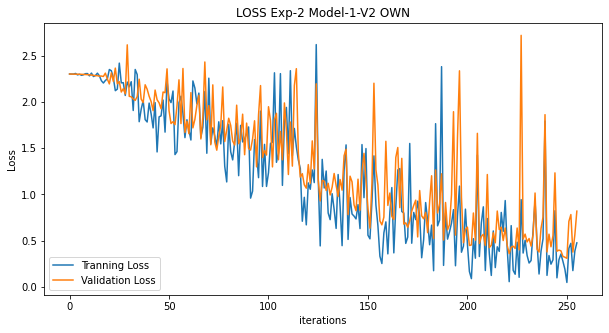
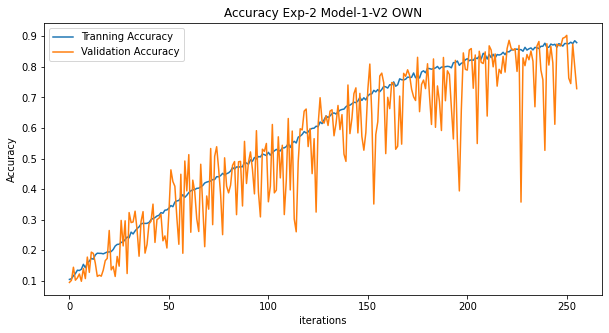
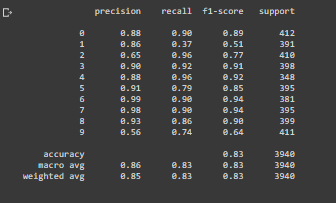
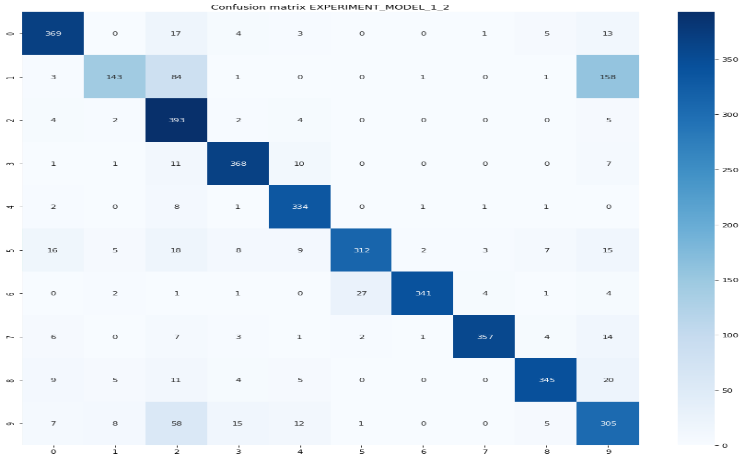
Figure 22:Loss graph exp-2

Figure 23: Accuracy graph exp-2

Figure 24: Performance matrices of exp-2 model v2

### Fashion-MNIST

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 60000 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 478125 |
| Training and Testing Split | 80% and 20% |
| Image Shape | 1 \* 28 \* 28 |
| Epochs | 255 |
| Optimizer | SGD |

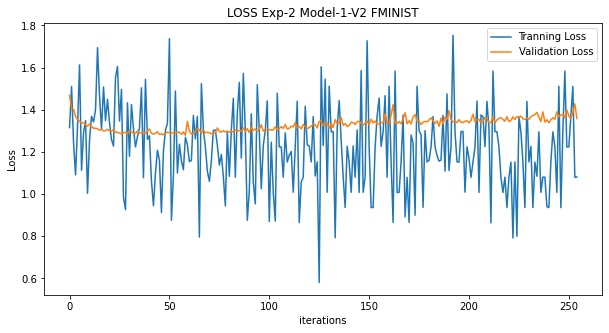


Figure 25: F-MINIST Exp-2 Loss

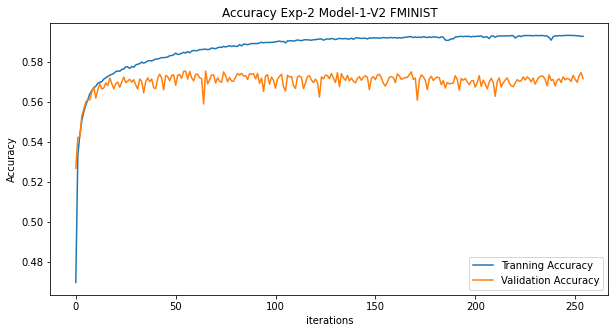
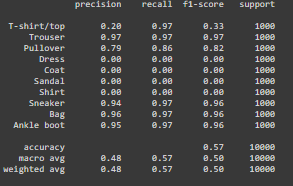
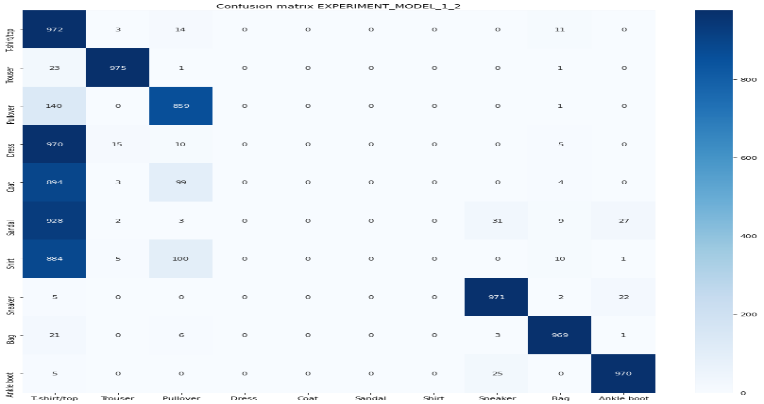


Figure 26:F-MINIST Exp-2 Accuracy

Figure 27: Performance matrices F- MINIST

## Experiment 2 – Model 1-V3

### Own(training-a)

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 19702 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 125715 |
| Training and Testing Split | 80% and 20% |
| Epochs | 299 |
| Image Shape | 1 \* 28 \* 28 |
| Optimizer | SGD |

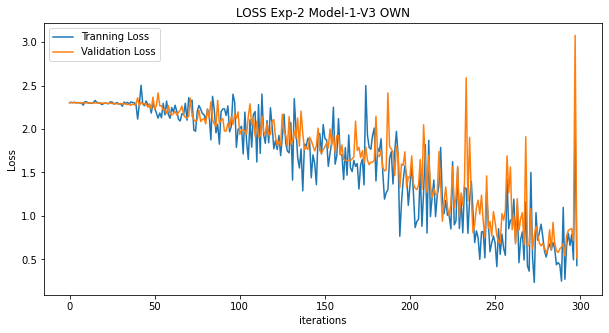


Figure 28: Loss graph exp-2

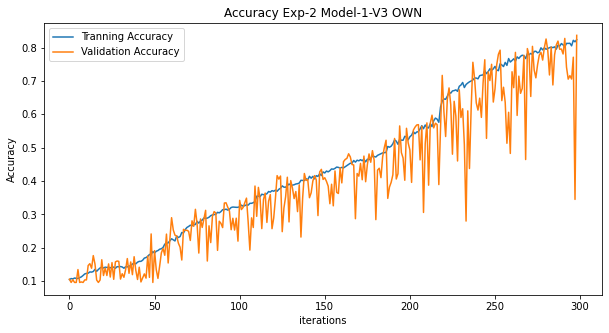
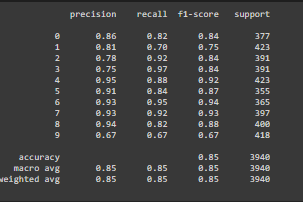
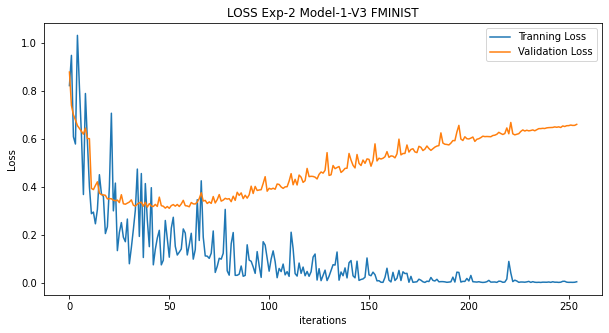


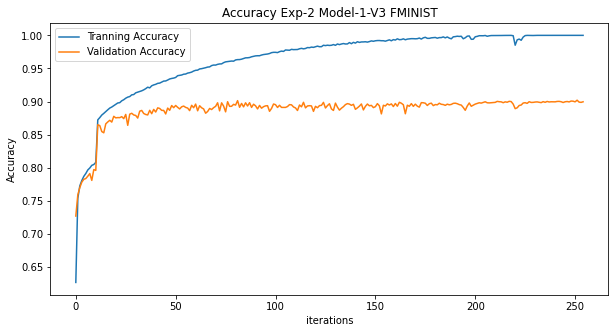
Figure 29:Accuracy graph exp-2

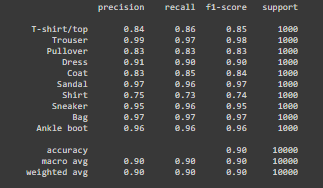
Figure 30: Performance Matrices

### Fashion-MNIST

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 60000 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 478125 |
| Training and Testing Split | 80% and 20% |
| Image Shape | 1 \* 28 \* 28 |
| Epochs | 255 |
| Optimizer | SGD |





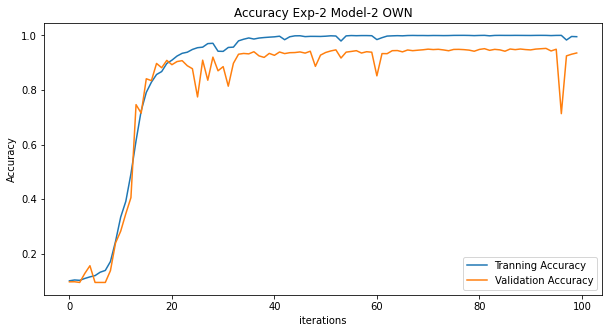
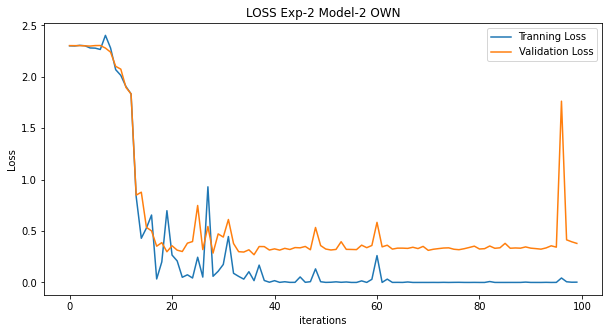


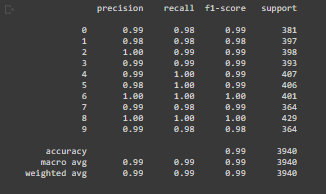
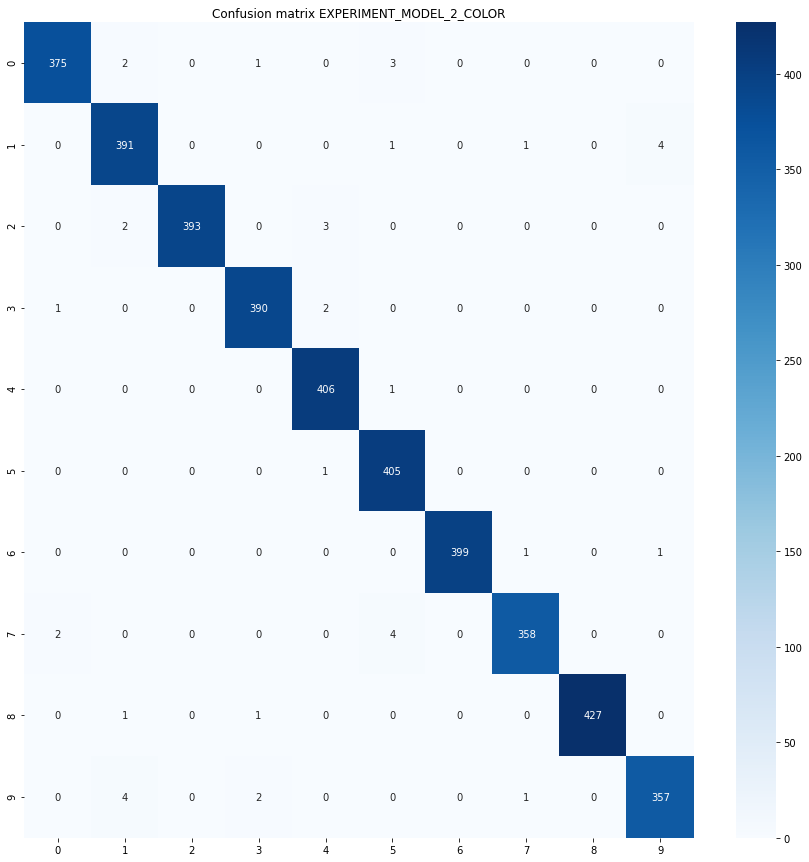
## Experiment 2 – Model 2

### Own(training-a)

#### Input Channel 3(RGB-Image)

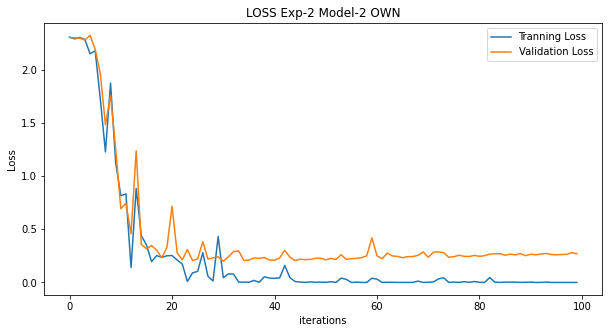
|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 19702 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 49300 |
| Training and Testing Split | 80% and 20% |
| Epochs | 100 |
| Image Shape | 3 \* 128 \* 128 |
| Optimizer | SGD |

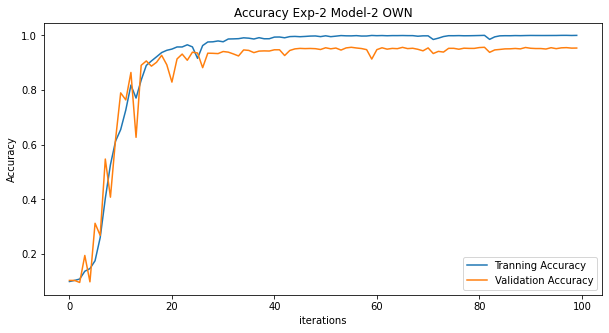


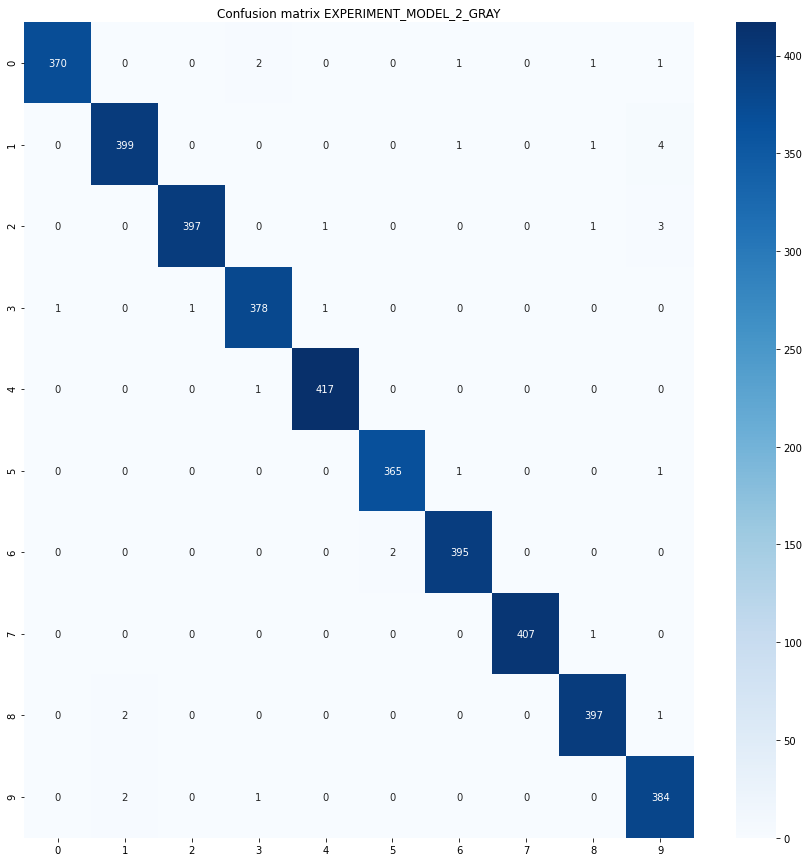
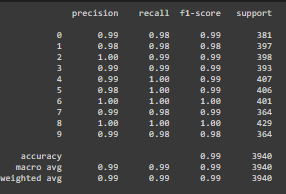


#### Input Channel 1(GRAY-Image)

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 19702 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 49300 |
| Training and Testing Split | 80% and 20% |
| Epochs | 100 |
| Image Shape | 1 \* 128 \* 128 |
| Optimizer | SGD |

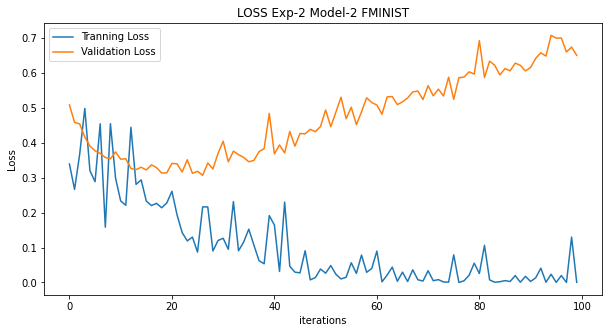


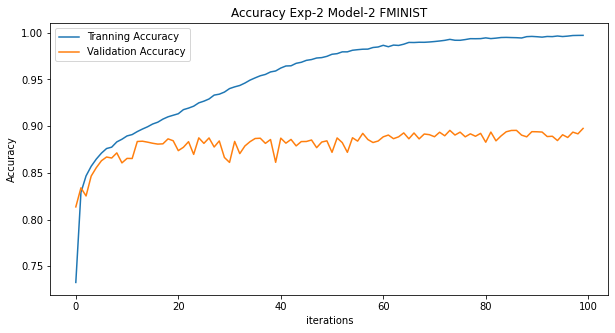


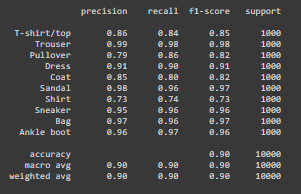
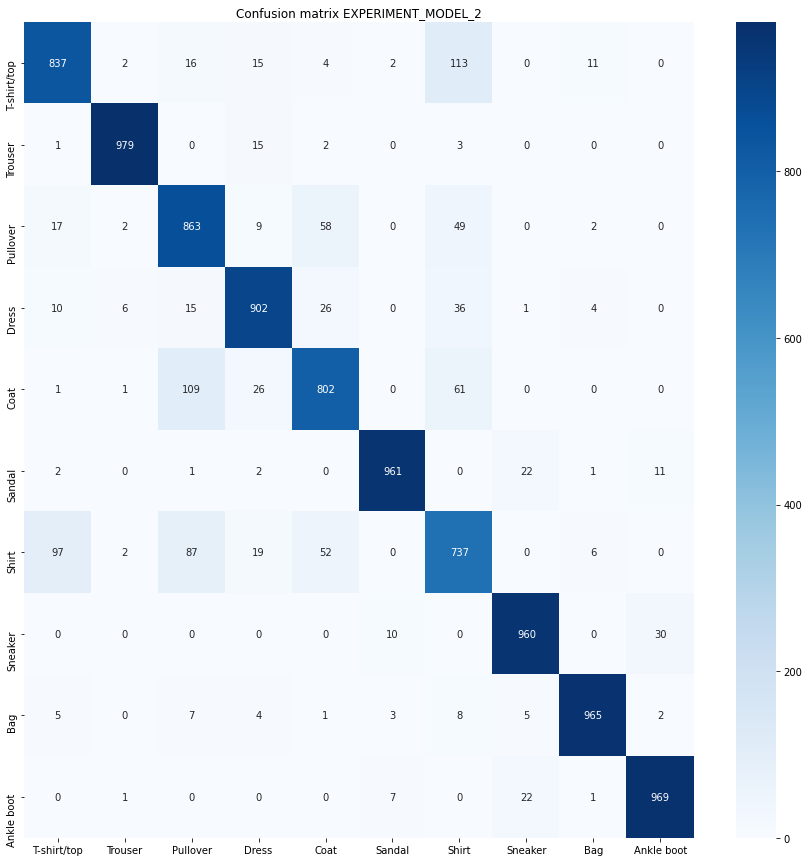
 

### Fashion-MNIST

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 60000 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 187500 |
| Training and Testing Split | 80% and 20% |
| Image Shape | 1 \* 128 \* 128 |
| Epochs | 100 |
| Optimizer | SGD |







## Experiment 2 – Model 3

### Own(training-a)

#### Input Channel 3(RGB-Image)

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 19702 |
| Learning rate | 0.001 |
| Batch Size | 32 |
| Iteration | 12325 |
| Training and Testing Split | 80% and 20% |
| Epochs | 25 |
| Image Shape | 3 \* 224 \* 224 |
| Optimizer | SGD |

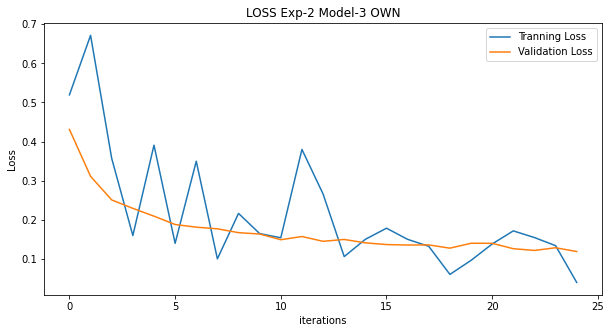


Figure 31: Loss Graph of Model-3

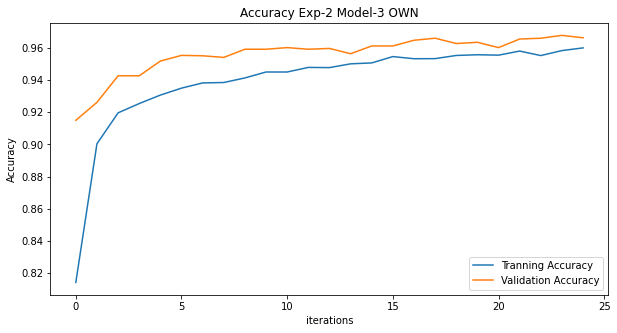
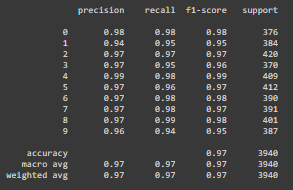
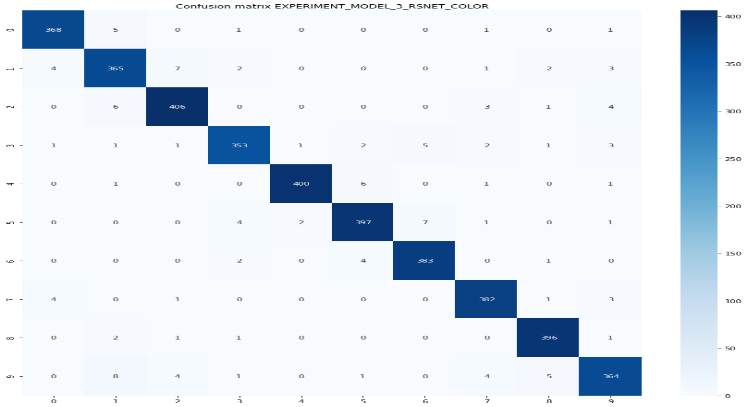
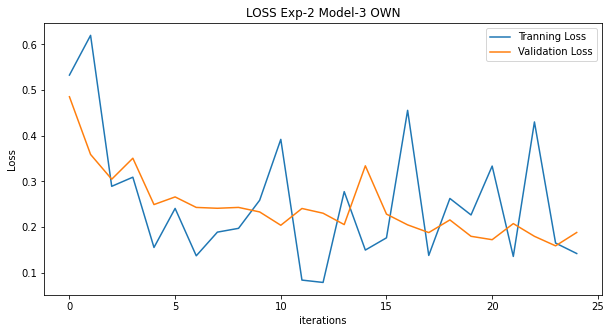


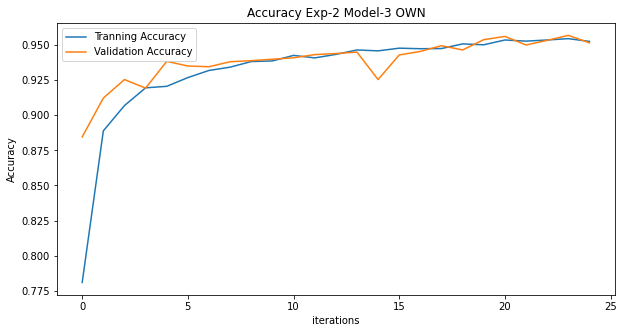
Figure 32:Accuracy Graph of Model 3

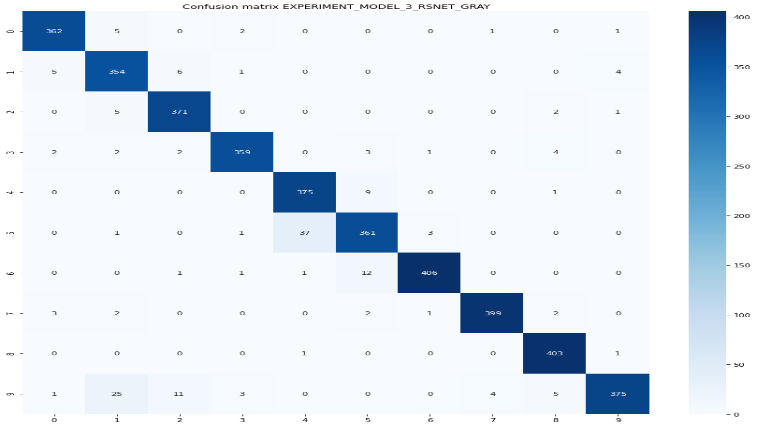
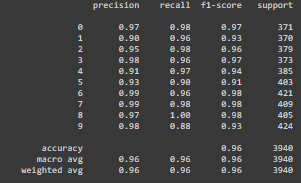
Figure 33: Performance Matrices

#### Input Channel 1(GRAY-Image)

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 19702 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 49300 |
| Training and Testing Split | 80% and 20% |
| Epochs | 100 |
| Image Shape | 1 \* 244 \* 244 |
| Optimizer | SGD |





### Fashion-MNIST

|  |  |
| --- | --- |
| **Name** | **Values** |
| Total dataset | 60000 |
| Learning rate | 0.01 |
| Batch Size | 32 |
| Iteration | 45000 |
| Training and Testing Split | 80% and 20% |
| Image Shape | 1 \* 224 \* 224 |
| Epochs | 24 |
| Optimizer | SGD |

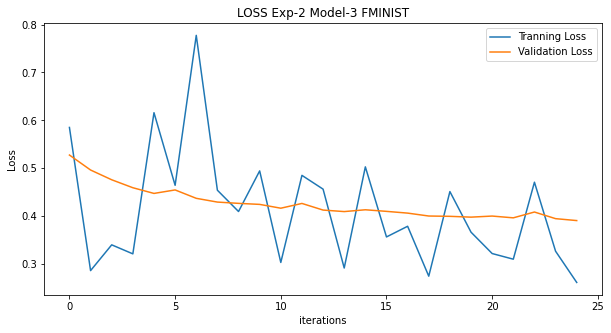


Figure 34:FMINIST loss in mode3

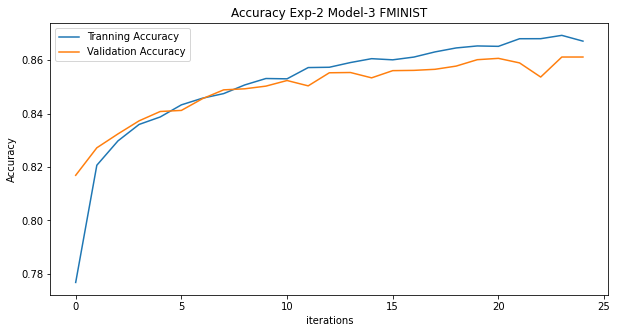


Figure 35:Accuracy Model3

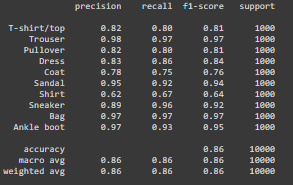
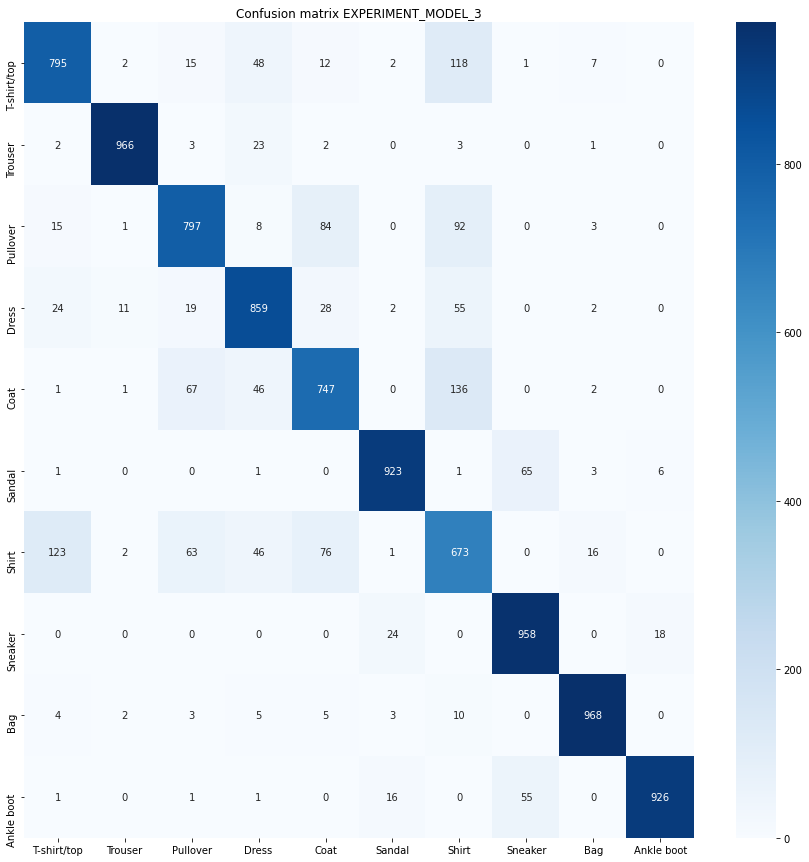


Figure 36: Performance Matrices

# Reference

Source code

<https://colab.research.google.com/drive/1mAg2xBR7uTMOr0k6Qjcl9jwMDJJWoyOx?usp=sharing>

<https://colab.research.google.com/drive/1NI0o0kO7aOmjh4enWf-JDu6XxzZnwdYo?usp=sharing>

<https://colab.research.google.com/drive/1FFhvZVv8sMc4jYovDvXp7GbN9pLwuJ4j?usp=sharing>

<https://colab.research.google.com/drive/1pSmlRV21bNGXygJSJgsO4uP7qfid87H5?usp=sharing>

Given Dataset Trained Model

<https://drive.google.com/file/d/1-7a44W76z-joINkLC_0ffcdBkPkRY_y8/view?usp=sharing>

<https://drive.google.com/file/d/1-8AKgpmaqLDT5K6CGmfZ-SrgC5vtxqYG/view?usp=sharing>

<https://drive.google.com/file/d/1-8x2xQQ0NOxpkF1MMe0k2Sz-PjnAS8Lc/view?usp=sharing>

<https://drive.google.com/file/d/1IxM0bieCLAaWWe_r67sd_Jz-oS8qXVfN/view?usp=sharing>

<https://drive.google.com/file/d/1LwOS_LP56Pm1xcbjXRzf_EYClpz167PK/view?usp=sharing>

<https://drive.google.com/file/d/1VLPrGSLzQBbL_G1rTufcs7EmMZ_YJGH-/view?usp=sharing>

<https://drive.google.com/file/d/1qpsII4q2Xk1mdSrS3KW4K8Pe7ugY8rs3/view?usp=sharing>

F-MNIST Trained Model

<https://drive.google.com/file/d/1-92tqY9DwsAHgUikt5B_9dMPc7Oq1sWH/view?usp=sharing>

<https://drive.google.com/file/d/1GQpI5HSHF_cg8mvBsBxc89ohSa5fDqEv/view?usp=sharing>

<https://drive.google.com/file/d/1K73sdUVBVx29VX8cBABH0AInev920sT0/view?usp=sharing>

<https://drive.google.com/file/d/1PBTq5xRgadmgc_MocZIHGhCFY80QBAg2/view?usp=sharing>

<https://drive.google.com/file/d/1VzWoWbjjMeptGSLgqqBxSdwiQ28CFmp9/view?usp=sharing>