Report Project 3

Name: Alhiet Orbegoso

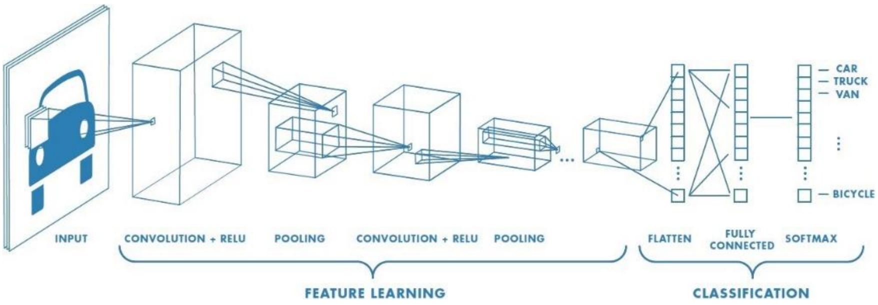
Unity ID: 200322491

Class: ECE 763 Computer Vision

A. Introduction

The project objective is to classify an image as face or non-face. This task is executed using the convolutional neural networks.

The image will pass though convolutional layers, then flatten to a single vector and finally dense layers will collapse it to single outputs. Each output represents the probability to belong to a specific class.



Multiclass convolutional layer

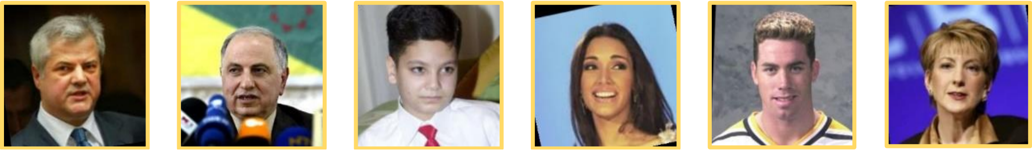
The project performs experiments with many configurations in neural network by including or not: data normalization, batch normalization, linear learning rate decay, Xavier parameters initialization and additional convolutional layers. Moreover, it has been tested if image size has an impact in accuracy and overall performance.

The report describes data preprocessing step, experimental setup, results and comments.

B. Dataset

The dataset is ensembled from 2 sources, dataset **Labeled Faces in the Wild** (http://vis-www.cs.umass.edu/lfw/) from University of Massachusetts Amherst and dataset **Caltech 101** (http://www.vision.caltech.edu/Image_Datasets/Caltech101) from California Institute of Technology.

- **Labeled Faces in the Wild:** This dataset provides aligned face images from uncontrolled environment, challenging the classification model. From this dataset has been extracted 7198 face images (1).



Face samples from dataset

- **Caltech 101:** This dataset provides the non-face images: random objects photos and drawings. From this dataset has been extracted 7125 non-face images.

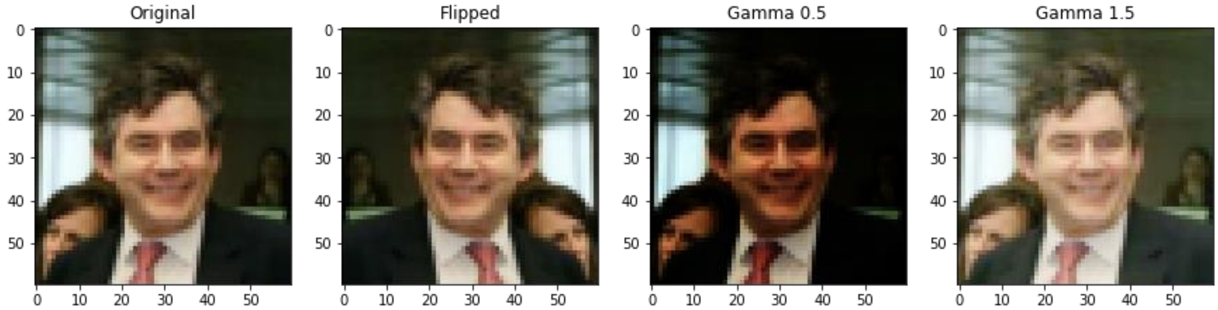
Non-faces samples from dataset

(1) Original dataset size is 11997, but to make both classes more equiprobable, the number of faces images was reduced to 7198.

C. Experimental setup

All images have a preprocessing step. First the size is reduced to 60x60, then all values are converted in float to a range from 0 to 1. In addition, the samples are randomly reordered so in every experiment the dataset is not identical.

In order to improve the model learning the data augmentation scheme has been adopted. For each image it has been generated 3 more images with the following modifications, flip image, gamma modification to 0.5 and 1.5.



Data augmentation samples

For this experiment, dataset is divided in train, validation and test samples. The train/validation data set represents the 70% of the total of samples and test is 30%. Internally train/validation has been again split in train samples 80% and validation samples 20%. The data augmentation configuration has been only applied to train and validation samples. Following chart resumes the number of images per type:

Dataset	Images	Data Augmentation
Train	32709	YES
Validation	8021	YES
Test	4298	NO

Number of images

Neural network architecture consists of 2 convolutional layers and 3 dense layers. The last layer has 2 outputs, each one provides the probability to belong to face class or to non-face class:

	Type	Inputs	Outputs	Kernel	Comments
1	Convolutional	3	16	5x5	Stride=1, No padding, Act=ReLU
2	Maxpool	-	-	2x2	Stride=2
4	Convolutional	16	32	13x13	Stride=1, No padding, Act=ReLU
5	Maxpool	-	-	2x2	Stride=2
7	Dense	2048	256	-	Act=ReLU
9	Dense	256	32	-	Act=ReLU
10	Dense	32	2	-	

Neural network layers

The loss function is cross entropy and the optimizer is stochastic gradient descent. The implementation of Pytorch loss function has already SoftMax layer so it was not added that layer. It was found that 3 convolutional layers does not improve the accuracy and model started to overfit (validation accuracy is reduced).

The code has been developed in Python 3.7.4; libraries installed are listed below:

- NumPy 1.17.1
- Matplotlib 3.1.2
- Pytorch 1.4.0
- Scikit-learn 0.22.1
- Seaborn 0.9.0
- OpenCV 4.0.1.25

If available, a GPU will be selected to speed up the training process.

D. Configurations Evaluated

The following configurations were tested for the best selection:

1. Experiment was executed with and without data normalization. There was not a relevant difference in accuracy and loss among both configurations.
2. Experiment was executed with and without batch normalization, with data normalization convergence was too fast, and test accuracy is reduced.
3. Experiment adopts the linear learning rate decay, it showed more stable loss function and accuracy during the training.
4. Experiment uses Xavier initialization for convolutional and dense layers, it showed more stable learning process and faster convergence. Accuracy is the same as original experiment.
5. It was not possible to test with bigger images due to memory restrictions. By the other hand, it was tested with 30x30, this size implicated to change kernel size in convolutional layers to 3x3 both. It shows similar results to original experiment.
6. Adding a new convolutional layer does not improve the accuracy. In addition, it was observed an increase in loss for validation.

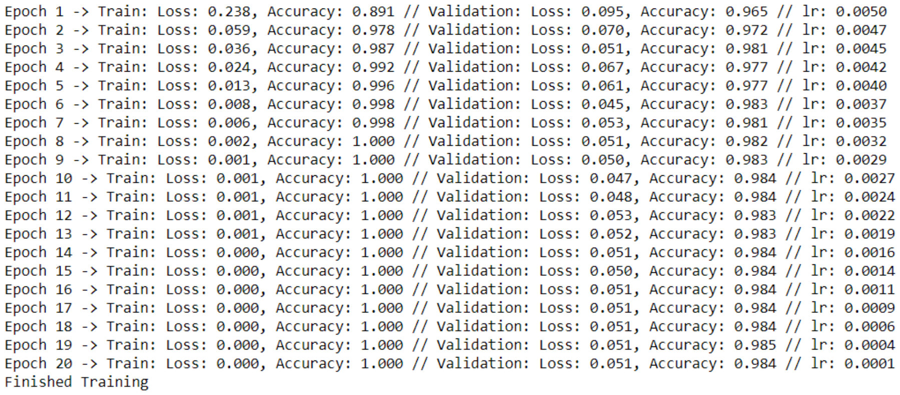
For all tests, it was verified that only 15 epochs are enough for convergence, nevertheless it was increased to 20 epochs to verify static loss and accuracy behavior of validation when train loss is almost 0. For training process in all configurations, the batch size is 100 elements.

E. Results

First is presented the result with the best configuration in terms of accuracy, loss, training stability and simplicity.

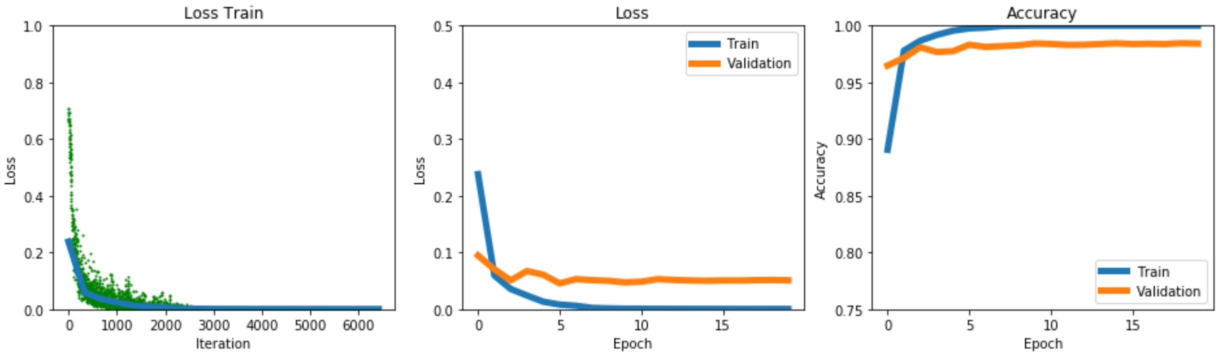
Configuration	Used	Configuration	Used	Configuration	Used
Data Normalization	NO	Linear Decay	YES	Image Size	60x60
Batch Norm	NO	Xavier Init	YES	Number CNN	2

Configurations used

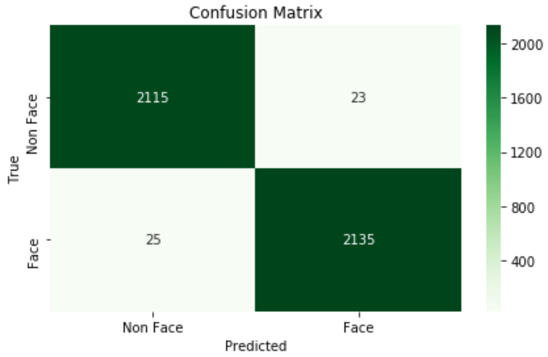


```
Epoch 1 -> Train: Loss: 0.238, Accuracy: 0.891 // Validation: Loss: 0.095, Accuracy: 0.965 // lr: 0.0050
Epoch 2 -> Train: Loss: 0.059, Accuracy: 0.978 // Validation: Loss: 0.070, Accuracy: 0.972 // lr: 0.0047
Epoch 3 -> Train: Loss: 0.036, Accuracy: 0.987 // Validation: Loss: 0.051, Accuracy: 0.981 // lr: 0.0045
Epoch 4 -> Train: Loss: 0.024, Accuracy: 0.992 // Validation: Loss: 0.067, Accuracy: 0.977 // lr: 0.0042
Epoch 5 -> Train: Loss: 0.013, Accuracy: 0.996 // Validation: Loss: 0.061, Accuracy: 0.977 // lr: 0.0040
Epoch 6 -> Train: Loss: 0.008, Accuracy: 0.998 // Validation: Loss: 0.045, Accuracy: 0.983 // lr: 0.0037
Epoch 7 -> Train: Loss: 0.006, Accuracy: 0.998 // Validation: Loss: 0.053, Accuracy: 0.981 // lr: 0.0035
Epoch 8 -> Train: Loss: 0.002, Accuracy: 1.000 // Validation: Loss: 0.051, Accuracy: 0.982 // lr: 0.0032
Epoch 9 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.050, Accuracy: 0.983 // lr: 0.0029
Epoch 10 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.047, Accuracy: 0.984 // lr: 0.0027
Epoch 11 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.048, Accuracy: 0.984 // lr: 0.0024
Epoch 12 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.053, Accuracy: 0.983 // lr: 0.0022
Epoch 13 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.052, Accuracy: 0.983 // lr: 0.0019
Epoch 14 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.051, Accuracy: 0.984 // lr: 0.0016
Epoch 15 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.050, Accuracy: 0.984 // lr: 0.0014
Epoch 16 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.051, Accuracy: 0.984 // lr: 0.0011
Epoch 17 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.051, Accuracy: 0.984 // lr: 0.0009
Epoch 18 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.051, Accuracy: 0.984 // lr: 0.0006
Epoch 19 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.051, Accuracy: 0.985 // lr: 0.0004
Epoch 20 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.051, Accuracy: 0.984 // lr: 0.0001
Finished Training
```

Training Log



Loss and Accuracy plots



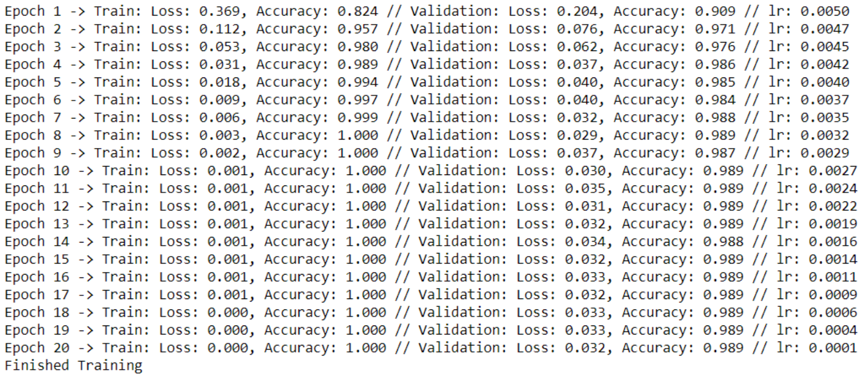
Confusion Matrix Test Samples

Here it is presented the results of experiments according the configurations described in section D. All experiments were tested 3 times and the best result is presented.

1. Data Normalization

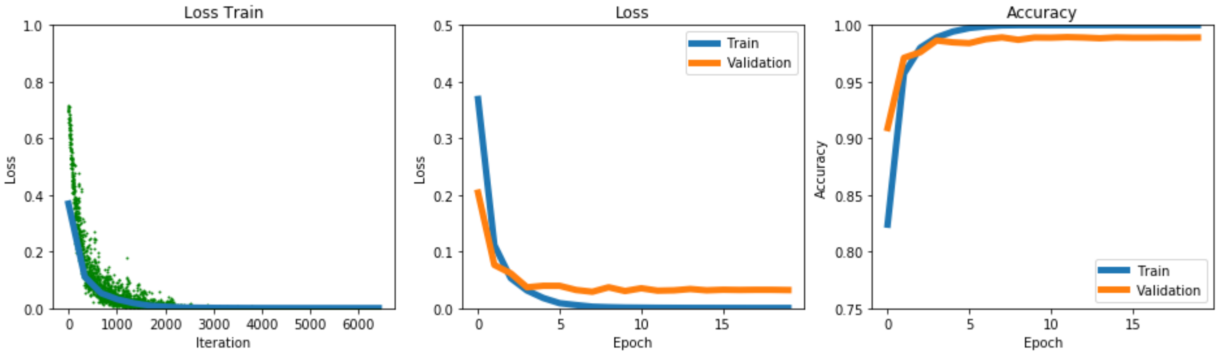
Configuration	Used	Configuration	Used	Configuration	Used
Data Normalization	YES	Linear Decay	YES	Image Size	60x60
Batch Norm	NO	Xavier Init	YES	Number CNN	2

Configurations used

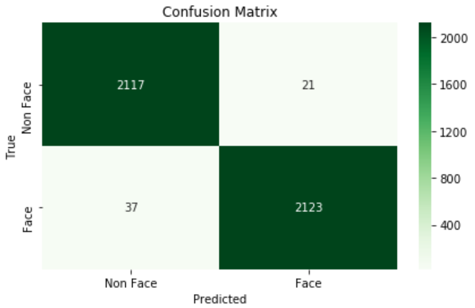


```
Epoch 1 -> Train: Loss: 0.369, Accuracy: 0.824 // Validation: Loss: 0.204, Accuracy: 0.909 // lr: 0.0050
Epoch 2 -> Train: Loss: 0.112, Accuracy: 0.957 // Validation: Loss: 0.076, Accuracy: 0.971 // lr: 0.0047
Epoch 3 -> Train: Loss: 0.053, Accuracy: 0.980 // Validation: Loss: 0.062, Accuracy: 0.976 // lr: 0.0045
Epoch 4 -> Train: Loss: 0.031, Accuracy: 0.989 // Validation: Loss: 0.037, Accuracy: 0.986 // lr: 0.0042
Epoch 5 -> Train: Loss: 0.018, Accuracy: 0.994 // Validation: Loss: 0.040, Accuracy: 0.985 // lr: 0.0040
Epoch 6 -> Train: Loss: 0.009, Accuracy: 0.997 // Validation: Loss: 0.040, Accuracy: 0.984 // lr: 0.0037
Epoch 7 -> Train: Loss: 0.006, Accuracy: 0.999 // Validation: Loss: 0.032, Accuracy: 0.988 // lr: 0.0035
Epoch 8 -> Train: Loss: 0.003, Accuracy: 1.000 // Validation: Loss: 0.029, Accuracy: 0.989 // lr: 0.0032
Epoch 9 -> Train: Loss: 0.002, Accuracy: 1.000 // Validation: Loss: 0.037, Accuracy: 0.987 // lr: 0.0029
Epoch 10 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.030, Accuracy: 0.989 // lr: 0.0027
Epoch 11 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.035, Accuracy: 0.989 // lr: 0.0024
Epoch 12 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.031, Accuracy: 0.989 // lr: 0.0022
Epoch 13 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.032, Accuracy: 0.989 // lr: 0.0019
Epoch 14 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.034, Accuracy: 0.988 // lr: 0.0016
Epoch 15 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.032, Accuracy: 0.989 // lr: 0.0014
Epoch 16 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.033, Accuracy: 0.989 // lr: 0.0011
Epoch 17 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.032, Accuracy: 0.989 // lr: 0.0009
Epoch 18 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.033, Accuracy: 0.989 // lr: 0.0006
Epoch 19 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.033, Accuracy: 0.989 // lr: 0.0004
Epoch 20 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.032, Accuracy: 0.989 // lr: 0.0001
Finished Training
```

Training Log



Loss and Accuracy plots

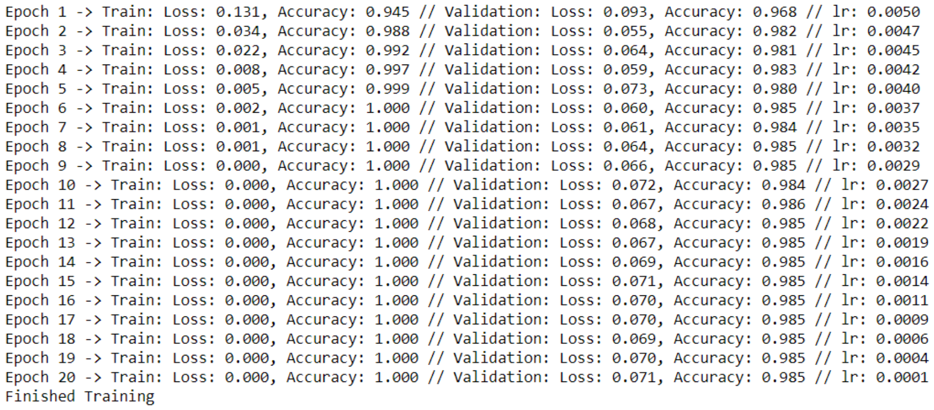


Confusion Matrix Test Samples

2. Batch Normalization

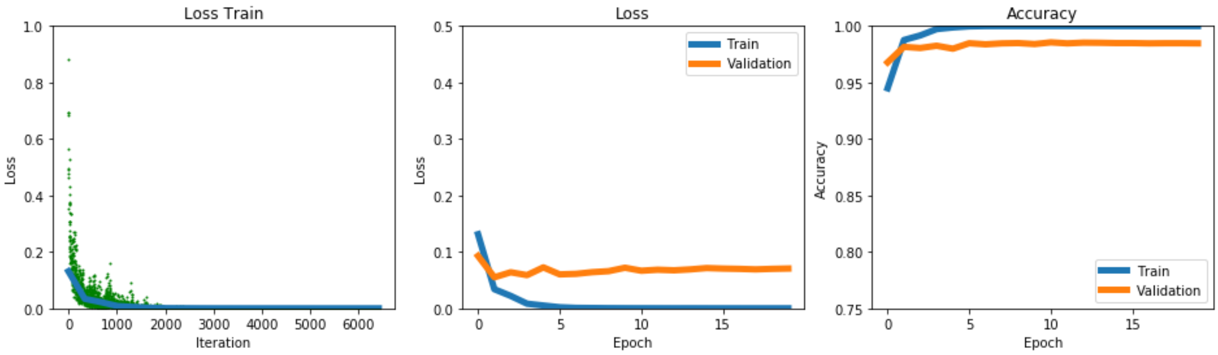
Configuration	Used	Configuration	Used	Configuration	Used
Data Normalization	YES	Linear Decay	YES	Image Size	60x60
Batch Norm	YES	Xavier Init	YES	Number CNN	2

Configurations used

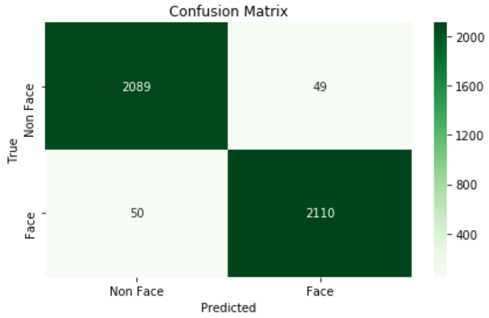


```
Epoch 1 -> Train: Loss: 0.131, Accuracy: 0.945 // Validation: Loss: 0.093, Accuracy: 0.968 // lr: 0.0050
Epoch 2 -> Train: Loss: 0.034, Accuracy: 0.988 // Validation: Loss: 0.055, Accuracy: 0.982 // lr: 0.0047
Epoch 3 -> Train: Loss: 0.022, Accuracy: 0.992 // Validation: Loss: 0.064, Accuracy: 0.981 // lr: 0.0045
Epoch 4 -> Train: Loss: 0.008, Accuracy: 0.997 // Validation: Loss: 0.059, Accuracy: 0.983 // lr: 0.0042
Epoch 5 -> Train: Loss: 0.005, Accuracy: 0.999 // Validation: Loss: 0.073, Accuracy: 0.980 // lr: 0.0040
Epoch 6 -> Train: Loss: 0.002, Accuracy: 1.000 // Validation: Loss: 0.060, Accuracy: 0.985 // lr: 0.0037
Epoch 7 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.061, Accuracy: 0.984 // lr: 0.0035
Epoch 8 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.064, Accuracy: 0.985 // lr: 0.0032
Epoch 9 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.066, Accuracy: 0.985 // lr: 0.0029
Epoch 10 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.072, Accuracy: 0.984 // lr: 0.0027
Epoch 11 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.067, Accuracy: 0.986 // lr: 0.0024
Epoch 12 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.068, Accuracy: 0.985 // lr: 0.0022
Epoch 13 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.067, Accuracy: 0.985 // lr: 0.0019
Epoch 14 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.069, Accuracy: 0.985 // lr: 0.0016
Epoch 15 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.071, Accuracy: 0.985 // lr: 0.0014
Epoch 16 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.070, Accuracy: 0.985 // lr: 0.0011
Epoch 17 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.070, Accuracy: 0.985 // lr: 0.0009
Epoch 18 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.069, Accuracy: 0.985 // lr: 0.0006
Epoch 19 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.070, Accuracy: 0.985 // lr: 0.0004
Epoch 20 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.071, Accuracy: 0.985 // lr: 0.0001
Finished Training
```

Training Log



Loss and Accuracy plots



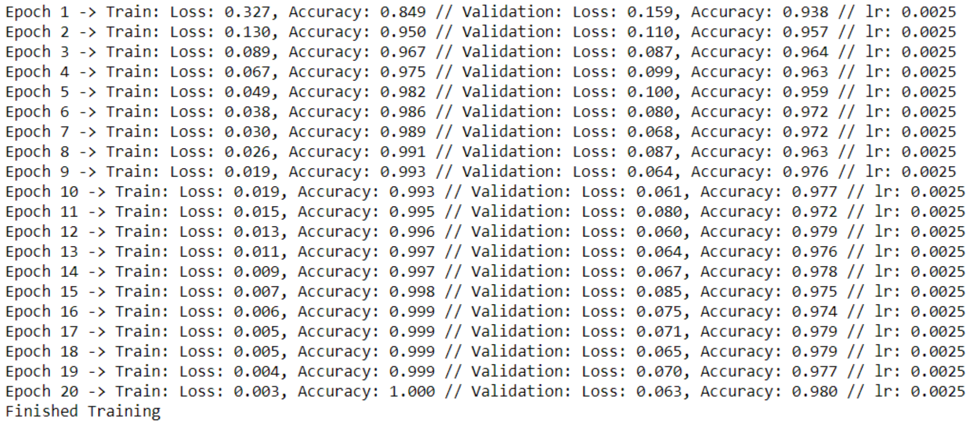
Confusion Matrix Test Samples

3. Without Linear Learning Rate Decay

It is used learning rate of 0.0025 fixed.

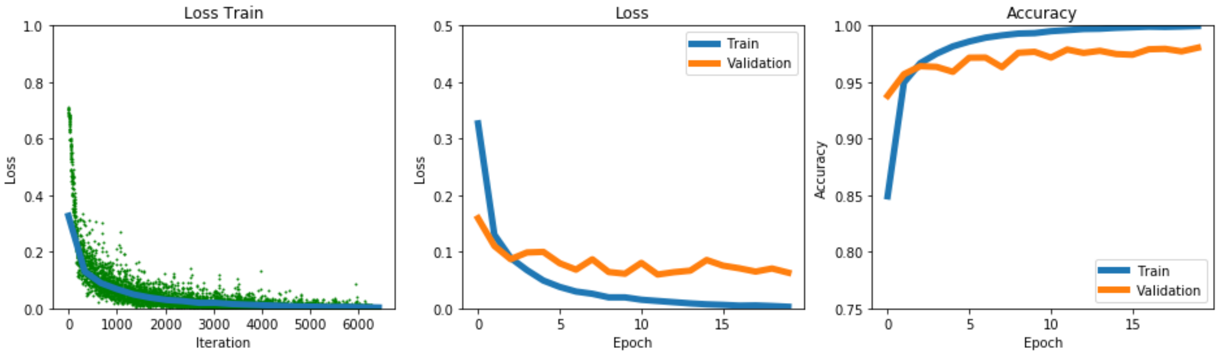
Configuration	Used	Configuration	Used	Configuration	Used
Data Normalization	NO	Linear Decay	NO	Image Size	60x60
Batch Norm	NO	Xavier Init	YES	Number CNN	2

Configurations used

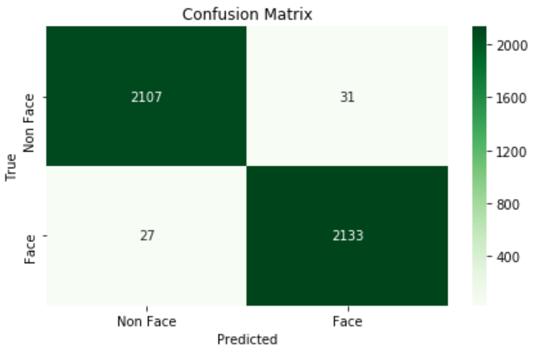


```
Epoch 1 -> Train: Loss: 0.327, Accuracy: 0.849 // Validation: Loss: 0.159, Accuracy: 0.938 // lr: 0.0025
Epoch 2 -> Train: Loss: 0.130, Accuracy: 0.950 // Validation: Loss: 0.110, Accuracy: 0.957 // lr: 0.0025
Epoch 3 -> Train: Loss: 0.089, Accuracy: 0.967 // Validation: Loss: 0.087, Accuracy: 0.964 // lr: 0.0025
Epoch 4 -> Train: Loss: 0.067, Accuracy: 0.975 // Validation: Loss: 0.099, Accuracy: 0.963 // lr: 0.0025
Epoch 5 -> Train: Loss: 0.049, Accuracy: 0.982 // Validation: Loss: 0.100, Accuracy: 0.959 // lr: 0.0025
Epoch 6 -> Train: Loss: 0.038, Accuracy: 0.986 // Validation: Loss: 0.080, Accuracy: 0.972 // lr: 0.0025
Epoch 7 -> Train: Loss: 0.030, Accuracy: 0.989 // Validation: Loss: 0.068, Accuracy: 0.972 // lr: 0.0025
Epoch 8 -> Train: Loss: 0.026, Accuracy: 0.991 // Validation: Loss: 0.087, Accuracy: 0.963 // lr: 0.0025
Epoch 9 -> Train: Loss: 0.019, Accuracy: 0.993 // Validation: Loss: 0.064, Accuracy: 0.976 // lr: 0.0025
Epoch 10 -> Train: Loss: 0.019, Accuracy: 0.993 // Validation: Loss: 0.061, Accuracy: 0.977 // lr: 0.0025
Epoch 11 -> Train: Loss: 0.015, Accuracy: 0.995 // Validation: Loss: 0.080, Accuracy: 0.972 // lr: 0.0025
Epoch 12 -> Train: Loss: 0.013, Accuracy: 0.996 // Validation: Loss: 0.060, Accuracy: 0.979 // lr: 0.0025
Epoch 13 -> Train: Loss: 0.011, Accuracy: 0.997 // Validation: Loss: 0.064, Accuracy: 0.976 // lr: 0.0025
Epoch 14 -> Train: Loss: 0.009, Accuracy: 0.997 // Validation: Loss: 0.067, Accuracy: 0.978 // lr: 0.0025
Epoch 15 -> Train: Loss: 0.007, Accuracy: 0.998 // Validation: Loss: 0.085, Accuracy: 0.975 // lr: 0.0025
Epoch 16 -> Train: Loss: 0.006, Accuracy: 0.999 // Validation: Loss: 0.075, Accuracy: 0.974 // lr: 0.0025
Epoch 17 -> Train: Loss: 0.005, Accuracy: 0.999 // Validation: Loss: 0.071, Accuracy: 0.979 // lr: 0.0025
Epoch 18 -> Train: Loss: 0.005, Accuracy: 0.999 // Validation: Loss: 0.065, Accuracy: 0.979 // lr: 0.0025
Epoch 19 -> Train: Loss: 0.004, Accuracy: 0.999 // Validation: Loss: 0.070, Accuracy: 0.977 // lr: 0.0025
Epoch 20 -> Train: Loss: 0.003, Accuracy: 1.000 // Validation: Loss: 0.063, Accuracy: 0.980 // lr: 0.0025
Finished Training
```

Training Log



Loss and Accuracy plots

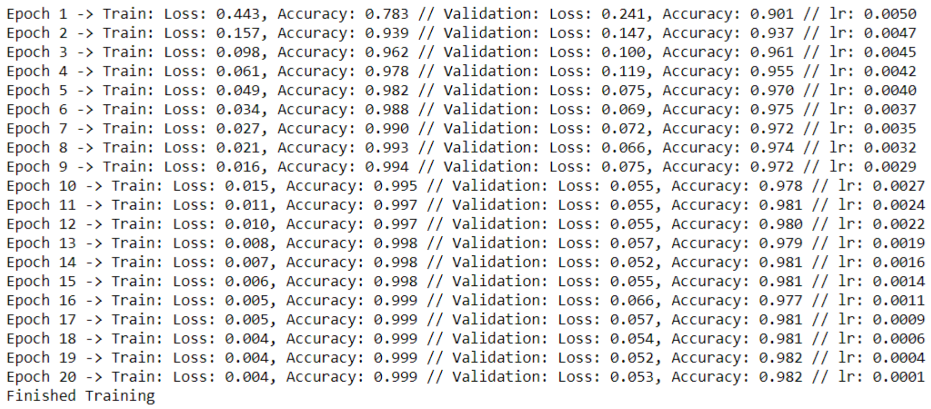


Confusion Matrix Test Samples

4. Without Xavier Initialization

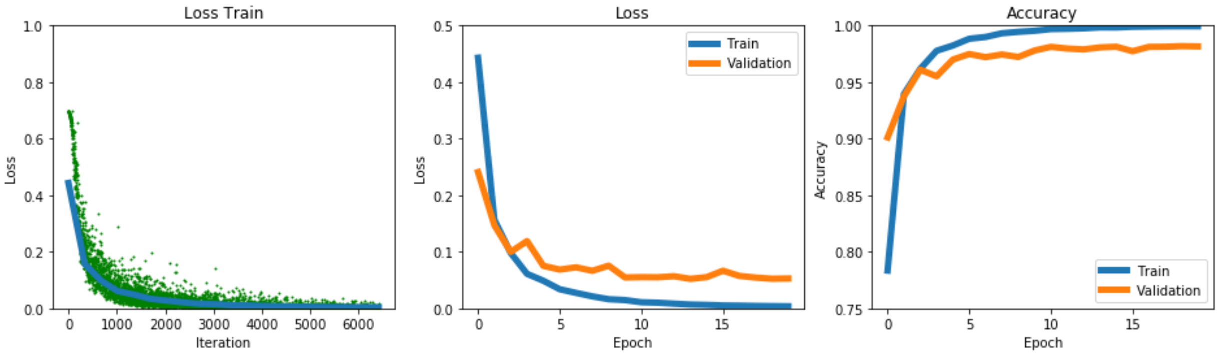
Configuration	Used	Configuration	Used	Configuration	Used
Data Normalization	NO	Linear Decay	YES	Image Size	60x60
Batch Norm	NO	Xavier Init	NO	Number CNN	2

Configurations used

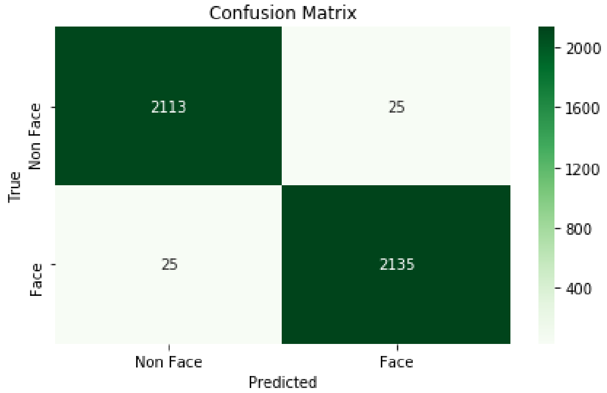


```
Epoch 1 -> Train: Loss: 0.443, Accuracy: 0.783 // Validation: Loss: 0.241, Accuracy: 0.901 // lr: 0.0050
Epoch 2 -> Train: Loss: 0.157, Accuracy: 0.939 // Validation: Loss: 0.147, Accuracy: 0.937 // lr: 0.0047
Epoch 3 -> Train: Loss: 0.098, Accuracy: 0.962 // Validation: Loss: 0.100, Accuracy: 0.961 // lr: 0.0045
Epoch 4 -> Train: Loss: 0.061, Accuracy: 0.978 // Validation: Loss: 0.119, Accuracy: 0.955 // lr: 0.0042
Epoch 5 -> Train: Loss: 0.049, Accuracy: 0.982 // Validation: Loss: 0.075, Accuracy: 0.970 // lr: 0.0040
Epoch 6 -> Train: Loss: 0.034, Accuracy: 0.988 // Validation: Loss: 0.069, Accuracy: 0.975 // lr: 0.0037
Epoch 7 -> Train: Loss: 0.027, Accuracy: 0.990 // Validation: Loss: 0.072, Accuracy: 0.972 // lr: 0.0035
Epoch 8 -> Train: Loss: 0.021, Accuracy: 0.993 // Validation: Loss: 0.066, Accuracy: 0.974 // lr: 0.0032
Epoch 9 -> Train: Loss: 0.016, Accuracy: 0.994 // Validation: Loss: 0.075, Accuracy: 0.972 // lr: 0.0029
Epoch 10 -> Train: Loss: 0.015, Accuracy: 0.995 // Validation: Loss: 0.055, Accuracy: 0.978 // lr: 0.0027
Epoch 11 -> Train: Loss: 0.011, Accuracy: 0.997 // Validation: Loss: 0.055, Accuracy: 0.981 // lr: 0.0024
Epoch 12 -> Train: Loss: 0.010, Accuracy: 0.997 // Validation: Loss: 0.055, Accuracy: 0.980 // lr: 0.0022
Epoch 13 -> Train: Loss: 0.008, Accuracy: 0.998 // Validation: Loss: 0.057, Accuracy: 0.979 // lr: 0.0019
Epoch 14 -> Train: Loss: 0.007, Accuracy: 0.998 // Validation: Loss: 0.052, Accuracy: 0.981 // lr: 0.0016
Epoch 15 -> Train: Loss: 0.006, Accuracy: 0.998 // Validation: Loss: 0.055, Accuracy: 0.981 // lr: 0.0014
Epoch 16 -> Train: Loss: 0.005, Accuracy: 0.999 // Validation: Loss: 0.066, Accuracy: 0.977 // lr: 0.0011
Epoch 17 -> Train: Loss: 0.005, Accuracy: 0.999 // Validation: Loss: 0.057, Accuracy: 0.981 // lr: 0.0009
Epoch 18 -> Train: Loss: 0.004, Accuracy: 0.999 // Validation: Loss: 0.054, Accuracy: 0.981 // lr: 0.0006
Epoch 19 -> Train: Loss: 0.004, Accuracy: 0.999 // Validation: Loss: 0.052, Accuracy: 0.982 // lr: 0.0004
Epoch 20 -> Train: Loss: 0.004, Accuracy: 0.999 // Validation: Loss: 0.053, Accuracy: 0.982 // lr: 0.0001
Finished Training
```

Training Log



Loss and Accuracy plots

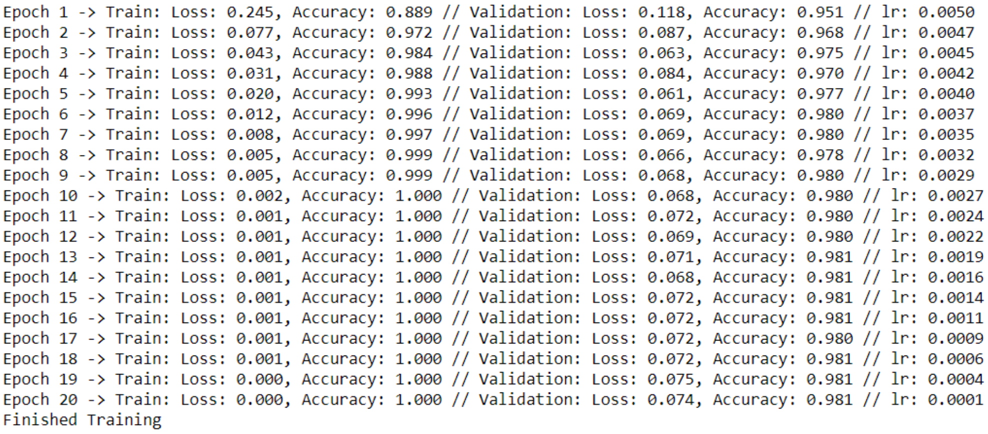


Confusion Matrix Test Samples

5. Image Size 30x30

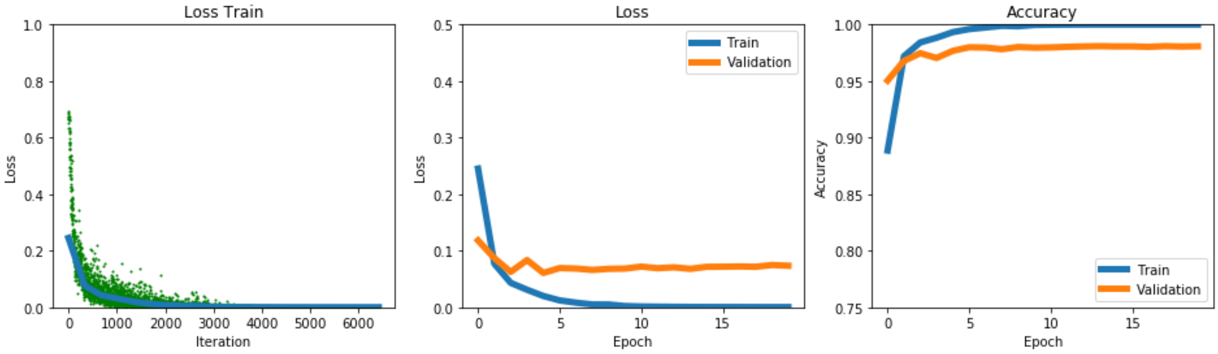
Configuration	Used	Configuration	Used	Configuration	Used
Data Normalization	NO	Linear Decay	YES	Image Size	30x30
Batch Norm	NO	Xavier Init	YES	Number CNN	2

Configurations used

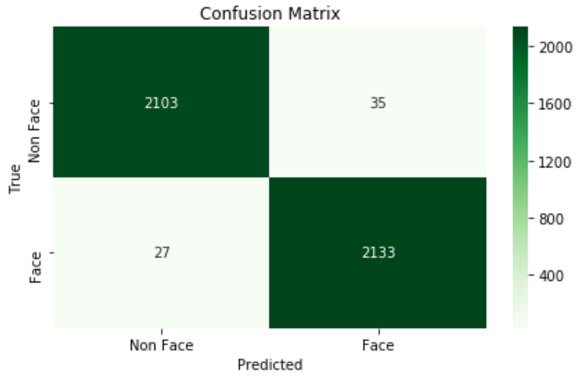


```
Epoch 1 -> Train: Loss: 0.245, Accuracy: 0.889 // Validation: Loss: 0.118, Accuracy: 0.951 // lr: 0.0050
Epoch 2 -> Train: Loss: 0.077, Accuracy: 0.972 // Validation: Loss: 0.087, Accuracy: 0.968 // lr: 0.0047
Epoch 3 -> Train: Loss: 0.043, Accuracy: 0.984 // Validation: Loss: 0.063, Accuracy: 0.975 // lr: 0.0045
Epoch 4 -> Train: Loss: 0.031, Accuracy: 0.988 // Validation: Loss: 0.084, Accuracy: 0.970 // lr: 0.0042
Epoch 5 -> Train: Loss: 0.020, Accuracy: 0.993 // Validation: Loss: 0.061, Accuracy: 0.977 // lr: 0.0040
Epoch 6 -> Train: Loss: 0.012, Accuracy: 0.996 // Validation: Loss: 0.069, Accuracy: 0.980 // lr: 0.0037
Epoch 7 -> Train: Loss: 0.008, Accuracy: 0.997 // Validation: Loss: 0.069, Accuracy: 0.980 // lr: 0.0035
Epoch 8 -> Train: Loss: 0.005, Accuracy: 0.999 // Validation: Loss: 0.066, Accuracy: 0.978 // lr: 0.0032
Epoch 9 -> Train: Loss: 0.005, Accuracy: 0.999 // Validation: Loss: 0.068, Accuracy: 0.980 // lr: 0.0029
Epoch 10 -> Train: Loss: 0.002, Accuracy: 1.000 // Validation: Loss: 0.068, Accuracy: 0.980 // lr: 0.0027
Epoch 11 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.072, Accuracy: 0.980 // lr: 0.0024
Epoch 12 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.069, Accuracy: 0.980 // lr: 0.0022
Epoch 13 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.071, Accuracy: 0.981 // lr: 0.0019
Epoch 14 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.068, Accuracy: 0.981 // lr: 0.0016
Epoch 15 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.072, Accuracy: 0.981 // lr: 0.0014
Epoch 16 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.072, Accuracy: 0.981 // lr: 0.0011
Epoch 17 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.072, Accuracy: 0.980 // lr: 0.0009
Epoch 18 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.072, Accuracy: 0.981 // lr: 0.0006
Epoch 19 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.075, Accuracy: 0.981 // lr: 0.0004
Epoch 20 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.074, Accuracy: 0.981 // lr: 0.0001
Finished Training
```

Training Log



Loss and Accuracy plots

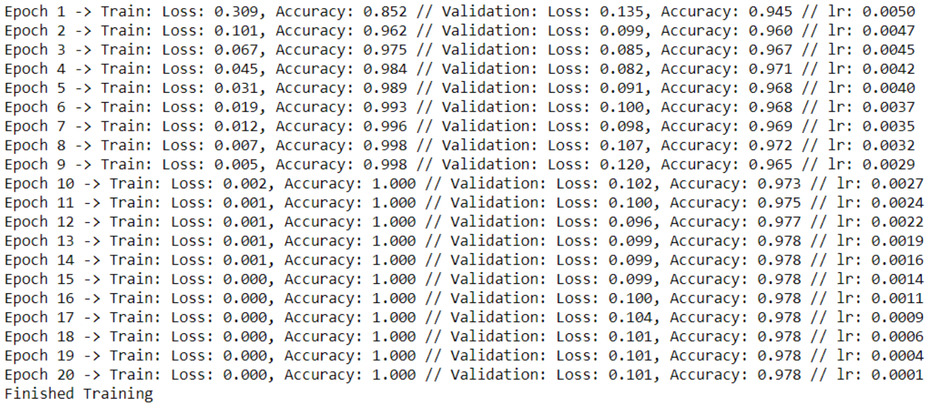


Confusion Matrix Test Samples

6. Adding 01 convolutional layer

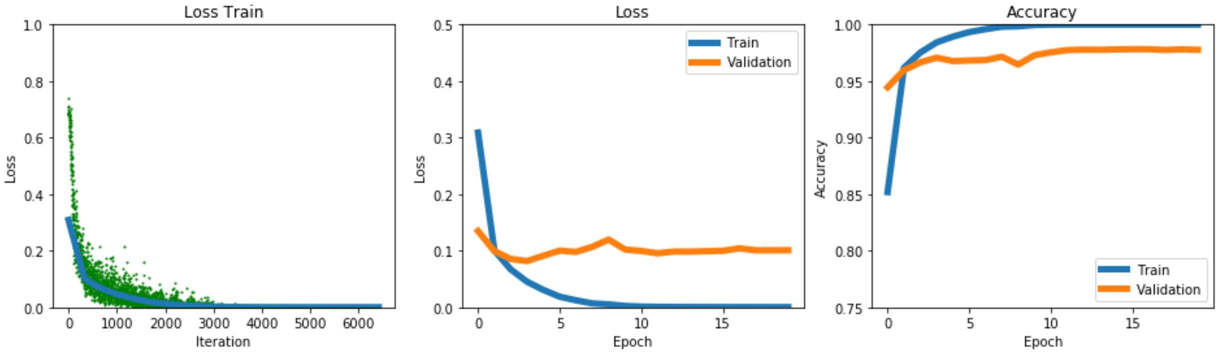
Configuration	Used	Configuration	Used	Configuration	Used
Data Normalization	NO	Linear Decay	YES	Image Size	60x60
Batch Norm	NO	Xavier Init	YES	Number CNN	3

Configurations used

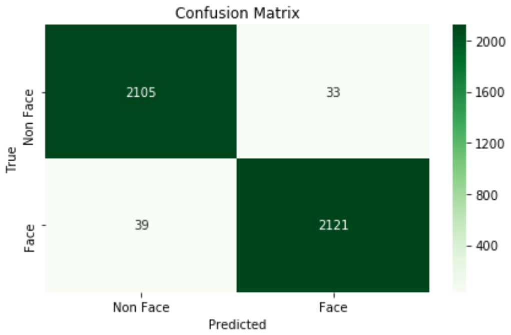


```
Epoch 1 -> Train: Loss: 0.309, Accuracy: 0.852 // Validation: Loss: 0.135, Accuracy: 0.945 // lr: 0.0050
Epoch 2 -> Train: Loss: 0.101, Accuracy: 0.962 // Validation: Loss: 0.099, Accuracy: 0.960 // lr: 0.0047
Epoch 3 -> Train: Loss: 0.067, Accuracy: 0.975 // Validation: Loss: 0.085, Accuracy: 0.967 // lr: 0.0045
Epoch 4 -> Train: Loss: 0.045, Accuracy: 0.984 // Validation: Loss: 0.082, Accuracy: 0.971 // lr: 0.0042
Epoch 5 -> Train: Loss: 0.031, Accuracy: 0.989 // Validation: Loss: 0.091, Accuracy: 0.968 // lr: 0.0040
Epoch 6 -> Train: Loss: 0.019, Accuracy: 0.993 // Validation: Loss: 0.100, Accuracy: 0.968 // lr: 0.0037
Epoch 7 -> Train: Loss: 0.012, Accuracy: 0.996 // Validation: Loss: 0.098, Accuracy: 0.969 // lr: 0.0035
Epoch 8 -> Train: Loss: 0.007, Accuracy: 0.998 // Validation: Loss: 0.107, Accuracy: 0.972 // lr: 0.0032
Epoch 9 -> Train: Loss: 0.005, Accuracy: 0.998 // Validation: Loss: 0.120, Accuracy: 0.965 // lr: 0.0029
Epoch 10 -> Train: Loss: 0.002, Accuracy: 1.000 // Validation: Loss: 0.102, Accuracy: 0.973 // lr: 0.0027
Epoch 11 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.100, Accuracy: 0.975 // lr: 0.0024
Epoch 12 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.096, Accuracy: 0.977 // lr: 0.0022
Epoch 13 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.099, Accuracy: 0.978 // lr: 0.0019
Epoch 14 -> Train: Loss: 0.001, Accuracy: 1.000 // Validation: Loss: 0.099, Accuracy: 0.978 // lr: 0.0016
Epoch 15 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.099, Accuracy: 0.978 // lr: 0.0014
Epoch 16 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.100, Accuracy: 0.978 // lr: 0.0011
Epoch 17 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.104, Accuracy: 0.978 // lr: 0.0009
Epoch 18 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.101, Accuracy: 0.978 // lr: 0.0006
Epoch 19 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.101, Accuracy: 0.978 // lr: 0.0004
Epoch 20 -> Train: Loss: 0.000, Accuracy: 1.000 // Validation: Loss: 0.101, Accuracy: 0.978 // lr: 0.0001
Finished Training
```

Training Log



Loss and Accuracy plots



Confusion Matrix Test Samples

7. Summary Chart

Below is presented a chart that summarizes the results from the configurations tested.

	Train		Validation		Test
Configuration	Loss (*)	Acc (*)	Loss (*)	Acc (*)	Acc
Original	0.000	1.000	0.051	0.984	0.988
Data Normalization	0.000	1.000	0.032	0.989	0.986
Batch Norm	0.000	1.000	0.071	0.985	0.976
No Linear Decay	0.003	1.000	0.063	0.980	0.986
No Xavier Init	0.004	0.999	0.053	0.982	0.988
Image Size 30x30	0.000	1.000	0.074	0.981	0.985
Number CNN to 03	0.000	1.000	0.101	0.978	0.983

Summary results

(*) Last epoch

F. Comments

In all possible configuration tested the more relevant component for accuracy it is the network size, it was realized that 2 convolutional layers and 3 dense layers were by far enough to reach 90% accuracy for test samples.

From the configurations tested, accuracy is almost the same. Some changes were observed in speed of convergence and stabilization of learning process.

Most of the effort is used in data preprocessing, convert the data into tensors for the learning step. Pytorch library makes easy the implementation, parametrization and training of neural networks.