AI Face Mask Detector

COMP 6721 Applied Artificial Intelligence (Summer 2022) Project Assignment, Part I

https://github.com/amritsinghcse/MaskNet
Team : OB_9

Amrit Singh (40195950)
Data Specialist

Sayali Kulkarni(40191368)
Training Specialist

Nipun Hedaoo(40165942)
Evaluation Specialist

June 9, 2022

Contents

List of Figures

List of Tables

1 Dataset

The data has been sourced from multiple sources including Kaggle, HITL and some sparse sources over the Internet using Google Images. Since no single dataset was readily available, other datasets were repurposed for the task.

1.1 Publicly available datasets

Many face-mask datasets are available publicly related to Covid-19 pandemic, but either they are incomplete for the task at hand, or the labelling information is not provided. For the project, we needed data divided into separate classes - surgical mask, N95/PPF2 mask, cloth mask, no mask, and mask worn incorrectly. The Kaggle Face Mask dataset by Larxel[1] provides 853 images but there is no separation between types of masks. Using the annotation files present within the dataset, around 4000 face images were extracted and then manually separated and labelled into the required categories.
For the class - no mask, Kaggle Human Faces by Ashwin Gupta[2] dataset was repurposed. For the class - mask worn incorrectly, the real data points found were far too less, so an already generated synthetic dataset was used for this purpose given by Cabani et al, called Masked-FaceNet[4].

1.2 Other Data

After repurposing the above datasets, there was a class imbalance where the images for the class N95 were far too less compared to the other classes. To resolve this, more data points were collected using Mask dataset provided by Humans in the Loop[5] and some other websites[6]. A total of 1928 images were gathered from across all the sources which will be divided into 80% training set and 20% test set. The class distribution for each of the classes is as follows:

Class	*Count*
surgical mask	412
N95 mask	306
cloth mask	410
no mask	400
mask worn incorrectly	400

Table 1.1: Data Class Distribution

The dataset size is 830 MB of type *.jpg* and *.png* with minimum resolution of 64*64. It is subdivided into 5 folder structure according to classes and loaded using ImageFolder class in torchvision. Since all images are of different sizes, normalization needs to performed, which is done by calculating mean and standard deviation over RGB channels. The images are then resized and converted into tensors. Finally the dataset is split into 80% training and 20% test set using *random_split*.

2 Architecture

The Model used for detecting the type of mask and whether the mask is worn correctly, is implemented using Convolution Neural Network. The convolution neural network makes use of perceptrons to create layers.

2.1 Model

The CNN architecture is comprised of convolution layers and max pooling layers. The basic working of Convolution Neural Network is illustrated in the Figure 2.1 [3] Convolution layer filter the image followed by the activation layer that applies an activation function inside the network architecture. The pooling layer reduces the size of the input volume.[7]

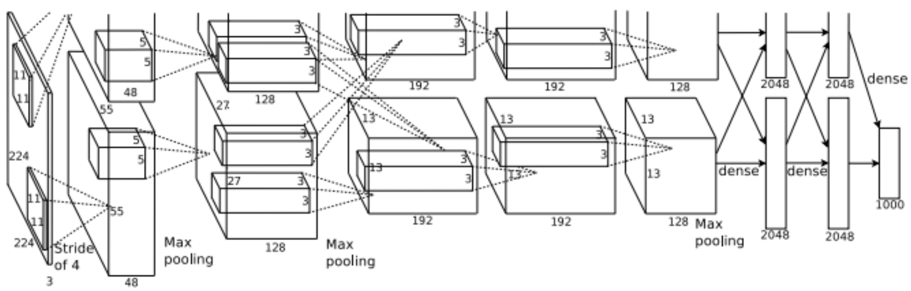

Figure 2.1: Architecture of CNN

Multiple models were tried on the dataset and the model with best accuracy was finalized as the model for classification. The model uses 3 convolution layers in total and 3 max pooling layer. A max pooling layer is followed after each convolution layer. Each convolution layer uses a padding of size 1 in order to maintain the resolution. Each of the max pooling layer has a filter of size 2 and stride of size 2. The network is composed such that a max pooling layer follows after every convolution layers. Finally the fully connected layers will compute the final output results.

2.2 Training

The training on the dataset is done using the Adam optimizer and Cross Entropy Loss. Adam optimizer is efficient with large volumes of data. It is used as a replacement for

Gradient Descent.[8] The training losses are depicted in the Figure 2.2. The training loss is helpful in knowing if the model is apt for the dataset used for training. The training loss has a declining form approaching zero. The training is done with 12 epochs.

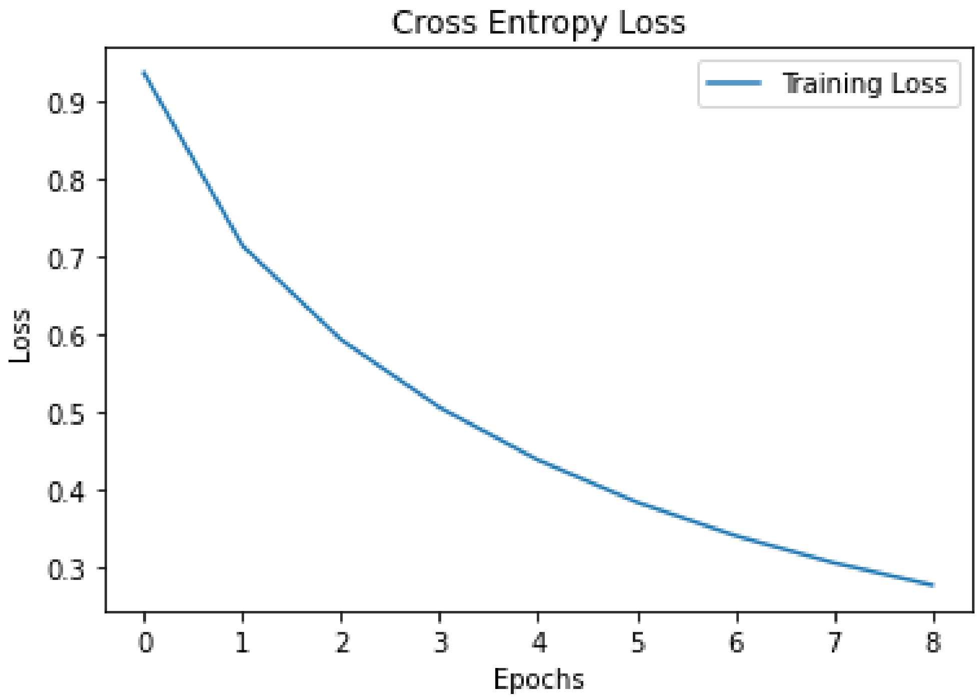

Figure 2.2: Graph for Training loss

In addition to the above mentioned model, two of it's variants were tested. The first variant consisted of 9 convolution layers and 3 max pooling layers. A max pooling layer followed after 3 convolution layers. The second variant has 2 convolution layer in total with each convolution layer followed by a max pooling layer.

3 Evaluation

3.1 Splitting Data

The data set is divided into two subsets i.e. training and testing. The eighty percent data is used for training and twenty percent for testing. We have automated the data splitting process and assigned Data Loader to instance of train and test split.

3.2 Comparison and Improvements

We created three different models by altering Pooling layers and Convolution layers. All metrics i.e Accuracy, Precision, Recall and F1 are represented below. The confusion matrix for the three models are also tabulated. Our main module has an accuracy of 70 percent, we are planning to increase the accuracy of model by further splitting the training data set for validation, augmenting images.

3.2.1 Model 1

Metric	*Training Set*
Accuracy	68.9
Precision	38.4
Recall	68.9
f1	68.8

Table 3.1: All Metric for testing data, Model 1

Predicted	*Cloth Mask*	*Mask Worn Inc*	*N95 Mask*	*No Mask*	*Surgical Mask*
Cloth Mask	58	0	3	8	22
Mask Worn Inc	0	62	0	0	1
N95 Mask	15	0	4	2	32
No Mask	4	0	0	76	4
Surgical Mask	22	4	0	4	69

Table 3.2: All Metric for testing data, Model 1

3.2.2 Model 2

Metric	*Training Set*
Accuracy	70.26
Precision	70.24
Recall	70.25
f1	67.45

Table 3.3: All Metric for testing data, Model 2

Predicted	*Cloth Mask*	*Mask Worn Inc*	*N95 Mask*	*No Mask*	*Surgical Mask*
Cloth Mask	51	1	3	4	32
Mask Worn Inc	0	62	0	0	1
N95 Mask	9	0	5	2	37
No Mask	6	0	0	75	3
Surgical Mask	11	2	2	3	81

Table 3.4: Confusion Matrix for testing data, Model 2

3.2.3 Model 3

Metric	*Training Set*
Accuracy	61.54
Precision	60.47
Recall	61.53
f1	60.84

Table 3.5: All Metric for testing data, Model 2

Predicted	*Cloth Mask*	*Mask Worn Inc*	*N95 Mask*	*No Mask*	*Surgical Mask*
Cloth Mask	53	2	13	7	16
Mask Worn Inc	0	63	0	0	0
N95 Mask	11	1	17	7	17
No Mask	10	0	3	63	8
Surgical Mask	24	6	17	8	44

Table 3.6: Confusion Matrix for testing data, Model 2

Bibliography

[1] Face Mask Detection. https://www.kaggle.com/datasets/andrewmvd/face-mask-detection

[2] Human Faces. https://www.kaggle.com/datasets/ashwingupta3012/human-faces

[3] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton *ImageNet Classification with Deep Convolutional Neural Networks*

[4] Cabani, A., Hammoudi, K., Benhabiles, H., and Melkemi, M., “MaskedFace-Net – A Dataset of Correctly/Incorrectly Masked Face Images in the Context of COVID-19”, ¡i¿arXiv e-prints¡/i¿, 2020.

[5] Humans in the Loop. https://humansintheloop.org/mask-dataset-download/?submissionGuid=13738a93-af75-4289-9a72-1ed2a32073e5

[6] https://googleimages.com/. Refer References.txt file.

[7] Convolutional Neural Networks (CNNs) and Layer Types https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/

[8] Adam optimizer PyTorch https://pythonguides.com/adam-optimizer-pytorch/

[9] How to Calculate Mean and Standard Deviation. https://towardsdatascience.com/how-to-calculate-the-mean-and-standard-deviation-normalizing-datasets-in-pytorch-704bd7d05f4c

[10] GeeksForGeeks. https://www.geeksforgeeks.org/how-to-normalize-images-in-pytorch/