Face Mask Detector

CMPE 255 Data Mining

https://github.com/CharikaBansal/cmpe-255-project

Team : 11

Rahul Sunil Pillai (016068793)

Faizan Mustafa Shaikh (016063372)

Charika Bansal (016709030)

Ira Sharma (010766704)

Table of Contents

1. Introduction

1.1. Objective

During the pandemic, it became a necessity to wear masks and maintain social distancing to prevent any exposure to covid 19. The objective of this project is to detect if a person is wearing a face mask or not. Places such as hospitals, restaurants, malls or any crowded spaces can use this technology if the person in front of the camera is wearing a mask or not.

1.2. Motivation

As the pandemic progressed in 2020, schools and offices were closed down and people were forced to work and study from home. Even after the vaccinations were introduced, a mask mandate is still in place in public areas to prevent the spread of coronavirus.With mask mandates a part of our future and present it is very necessary to have a system which can identify if a person is wearing a mask or not. Additionally different types of masks have different applications which has made it necessary to differentiate between these types as well. Our project aims to achieve this by identifying different kinds of masks and classifying them.

1.3. Literature Market Review

The previous work done in this department was limited to classification of whether a person is wearing a mask or not. It was only during the covid 19 pandemic when the awareness among people regarding different types of masks and their uses started to grow. Now as the general public is aware of these types and can be seen wearing them, it has become very important that we build a system which can differentiate between these mask types and this information can further be used for various business needs.

2. System Design and Implementation Details

2.1. Algorithm(s) considered/selected

Various approaches like unsupervised image classification with 'K-mean' and 'ISODATA' and also supervised methods of image classification with algorithms like support vector machine, random forest were considered. But after doing considerable research and comparing results seen with various approaches Convolutional Neural Networks were discovered to be the ideal approach for the task at hand

2.2. Technologies & Tools used

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.
Pytorch tool was used to perform Neural Network related functions. In addition to that, various libraries like DataLoader and ImageFolder were used to fetch data from the dataset. Other libraries and tools which were used include numpy, pandas, matplot and more.

2.3. Architecture

2.3.1. Model

The CNN architecture consists of convolution layers and max pooling layers. The basic working of Convolution Neural Network is illustrated in the Figure 2.1 [3] Convolution layer filters 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]

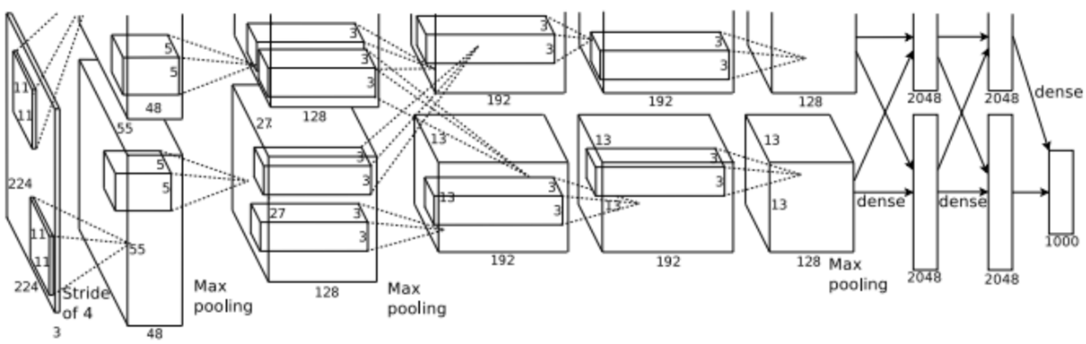

Figure 2.1 Architecture of CNN

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 layers. 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 layers 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 layer. Finally the fully connected layers will compute the final output results.

2.4. Data Flow

The dataset size is 830 MB of type .jpg and .png with minimum resolution of 64*64. It is subdivided into 5 folder structures according to classes and loaded using ImageFolder class in torchvision. Since all images are of different sizes, normalization needs to be 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. The data is finally then fed to the CNN for training where optimizer and loss functions are used for training. Finally the model is tested against testing data and results are evaluated.

3. Proof of concept evaluation

3.1. 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.

3.1.1. 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 labeling 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 labeled into the required categories.

For the class - no mask, Kaggle Human Faces by Ashwin Gupta[2] dataset was repurposed. For the class - masks 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].

3.1.2. 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 the 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	1277
N95 mask	560
Cloth Mask	588
No mask	939
Mask worn incorrectly	1015

Table 3.1: Data Class Distribution

3.2. 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 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.

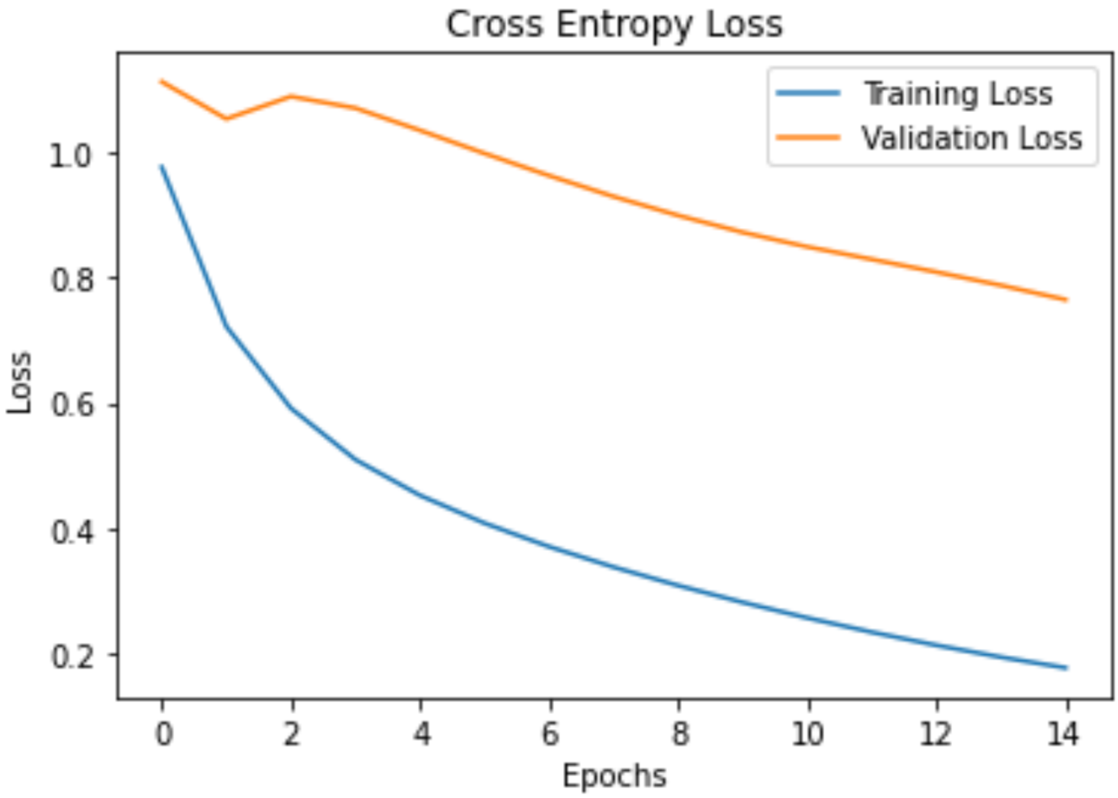

Figure 2.2 Graph of training loss

In addition to the above mentioned model, two of its 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 layers in total with each convolution layer followed by a max pooling layer.

3.3. Splitting Data

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

3.4. 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 is also tabulated. Our main module has an accuracy of 70 percent, we are planning to increase the accuracy of the model by further splitting the training data set for validation, augmenting images.

3.4.1. 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

3.4.2. 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

3.4.3. 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

4. Discussion & Conclusions

4.1. Decisions made and difficulties

Availability of data was the biggest challenge since it has not been long since the pandemic. Therefore the decision of generating our own dataset by collecting it from different sources was made. Given the disparateness in the data sources, pulling and merging them was another task. Multiple models were tried on the dataset and the model with best accuracy, that is model 2 was finalized as the model for classification. The model uses 3 convolution layers in total and 3 max pooling layers. 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 layers 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 layer. Finally the fully connected layers will compute the final output results.

4.2. Conclusion

The model was able to successfully classify the images into the five classes with an accuracy of around 70 percent. Different approaches towards image classification which ranged from supervised methods to unsupervised methods were considered. Different ways of building a Convolutional Neural Network model were explored before coming up with an optimal approach. Since a large dataset of different types of masks is not available at this time, In future when this is made available a better and more generalized model can be built.

5. Task Distribution

Faizan Mustafa Shaikh	Implemented the convolutional network model and experimented with 2 more variants with differing layers to maximize the accuracy, precision, recall and f1 scores
Rahul Sunil Pillai	Worked on Optimization function and loss function. Also worked on coming up with the right approach to train the data which involved deciding on the percentage of testing and training data. Also worked on finding the right number of epochs to train the model
Charika Bansal	Transformed the input data from the dataloader and computed the mean and standard deviation values required for normalization
Ira Sharma	Worked on repurposing the already available data over the internet to create a dataset suitable for our application. Implemented a web scraping tool to fetch google images for different kinds of masks and manually separated the images into various classes.

6. Citation

[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