Concordia University

project phase 1 report

COMP6721

AI Face Mask Detector

Author:	*Author ID:*	*Specialization:*
Somayeh Ghahary	40106359	Training
Mehrnoosh Amjadi	40091264	Evaluation
Bikash Jaiswal	40115186	Data

Team Name: FL-08

Submitted to: Dr. René Witte

March 24, 2021

Concordia University
Engineering and Computer Science

1 Dataset

In this project, we use two different image datasets: Face Mask Dataset and Cifar-10 dataset [3]. Face Mask Dataset consists of images with mask and non-mask and located in directories Mask and Non-Mask respectively. These directories act as two classes necessary for our project. We use a Cifar-10 dataset for third class i.e "Non-Human". Since, Cifar-10 dataset are grouped in 10 different class folders, we take 600 samples randomly from each class folder and store them in a single folder. We categorise the output folder as Non-Human class for our experiment. The statistics and structure about the final dataset use for our Experiment are mentioned in Table 1.

We perform pre-processing step in training and testing data before feeding into Convolutional Neural Networks. The pre-processing step include resizing, normalising, centrecrop. All these data is converted to tensor data. Then we normalize these tensor data with mean = [0.485, 0.456, 0.406] and standard deviation = [0.229, 0.224, 0.225]. We resize the each data to size of 256*256.

Info	**Face Mask Dataset**	**Cifar-10 Datasets**
Name	Face Mask ∼12K Images Dataset	CIFAR-10 PNGs in folders
Author	Ashish Jangra	Swaroop Kumar and [3]
Source	https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset	https://www.kaggle.com/swaroopkml/cifar10-pngs-in-folders
Licence	No license specified, the work may be protected by copyright.	No license specified, the work may be protected by copyright.
Total Size:	12000 images	60000 images
Image per class	With Mask: 6000 Images Without Mask: 6000 Images	10 classes, with 600 images per class
Training Size	With Mask: 5000 Images Without Mask: 5000 Images	NonHuman: 5000 Images
Testing Size	With Mask: 483 Images Without Mask: 509 Images	NonHuman: 500 Images

Table 1: Statistics and structure about the final dataset

2 CNN Architecture

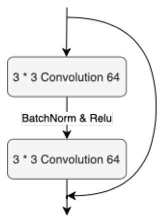

Figure 1: building blocks

In this project, we design a deep neural network to train, test the dataset and evaluate the results. To design our model we consider common issues in deep network such as overfitting and vanishing gradients and try to solve these problems efficiently. In this regard, to overcome the over-fitting problem which occurs in a network with lots of parameters, we decide to have a deeper network with more layers and we apply dropout strategy. For resolving vanishing gradient, we apply the Relu function as activation function. Also we use batch normalisation as another solution for vanishing gradients. Another efficient procedure is the idea of Residual network, so we supply the residual connections straight to earlier layers.

Generally we get the idea of our project from famous Resnet Algorithm [2] and [4]. According to [2], with Residual networks, deeper network has less complexity and shows improvement at analysing. This model consists of building blocks of convolution layers, which is shown in figure 1. These blocks are stacked together while the main idea is that "identity shortcut connection" changes the output of the block. The final architecture of the model, figure 2 has 15 layers. Although expanding the depth of this network is easy due to the property of the block, we limit our self to 15 layers. To speedup the process of testing and training we need GPU, so we decide to implement our model in the google collaboration [1], actually we access to limited hardware.

For training phase, the model iterates over 4 epochs and in each epoch, it trains model with all the batches in the train dataset. Besides we train famous AlexNet model [3], from torch library [4] with our dataset. Then we test the dataset on this trained models and compare the final results of our model and AlexNet.

3 Evaluation

In this phase, we evaluate the performance of our model from various aspects including accuracy, precision, recall, and f1 score. We also use the confusion matrix to visualize the performance of two models. To better understand the performance, we compare our model with one of the well-known models in the field of deep learning, namely the ALexNet model. By comparing these two models and the results obtained from them, we find that our model performs very close to the AlexNet model. This shows that the structure of our model is at least as good as the structure of the AlexNet model.

Metric	Our Defined Model	AlexNet Model
Accuracy	95.84	97.72
Precision Micro	95.84	97.72
Precision Macro	96.15	97.75
Recall Micro	95.84	97.72
Recall Macro	95.74	97.71
F1 Score Micro	95.84	97.72
F1 Score Macro	95.80	97.70

Table 2: Evaluation Table

As there are three different classes in the dataset, we consider micro and macro averaging for recall, precision, and f1 score. As shown in the table 2, we obtain very high accuracy, precision, recall, and f1 score from testing phase. Also, it can be seen in the confusion matrix that the model predicts the category of images well. However, our model has a little difficulty in recognizing "with masked" images. Hence, we can conclude that our data are well balanced and our model designed well.

In this project, we combined two datasets to have 3 different classes. The Cifar-10 dataset

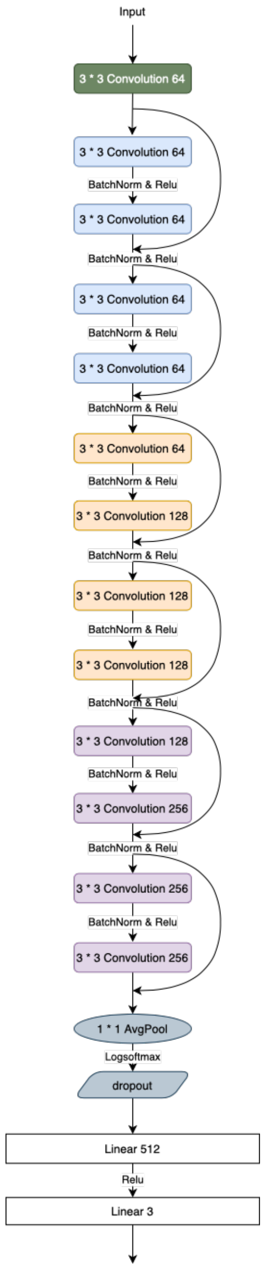

Figure 2: model architecture

contains non-human images with low resolution, but the other dataset includes high-resolution "with mask" and "without mask" images. Although our model predicts non-human classes well, we can further improve this part and find a dataset for this class with the high-resolution images in the next step to make sure some important features are not missing during training due to low resolution.

In the second phase of the project, we can increase the size of the dataset and find appropriate

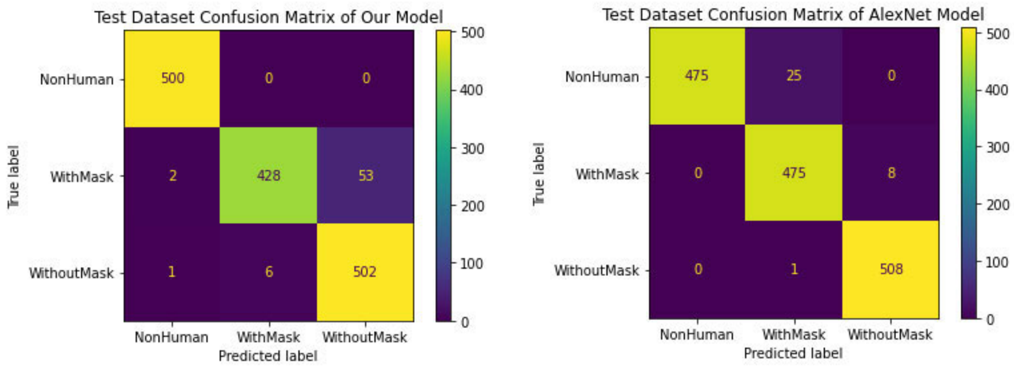

Figure 3: Confusion Matrix of two Models

dataset containing non-human images with high resolution. Also we can utilize cross validation technique to make sure that data is balanced across training and test sets. Currently, we consider 4 epochs in this phase. However, we can increase the number of epochs so the training set has more opportunity to update model parameters and minimises errors. Also, we investigate our time to find out if our model exposes any kind of biases, including age, race, etc., and remove them.

References

[1] Ekaba Bisong. *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA, 2019.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[4] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery.