PROJECT REPORT

PREPARED FOR

Dr. Rene Witte

PREPARED BY

Shivani - 40092376

Karandeep - 40104845

DATASET

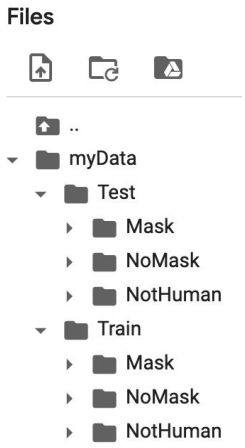
Introduction:

The dataset consists of 15153 images which are separated in 3 categories, human face with a mask, human face without a mask and a non human face. For the non human face category the images of cats have been used.

Size of Dataset:

	Human Face with Mask	Human Face without Mask	Non Human Face	Test Data
No. Of Images	3500	3500	3500	3000

Dataset Structure:



Source:

The dataset has been downloaded from Kaggle.com

Links to Datasets:

- Human Images - https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset
- Non Human Images - https://www.kaggle.com/andrewmvd/animal-faces
- Test Data - https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset https://www.kaggle.com/andrewmvd/animal-faces

License - https://creativecommons.org/publicdomain/zero/1.0/

CNN Architecture

Layers Implemented:

1. Convolution Layer: This layer transforms an image into a feature map using a filter. This system has **6 layers** (Convolution + pooling):
    a. First Layer : This is a **convolution layer** which converts a **3-channel image to a 100 channel image**. It has a kernel size of 3 and padding is set to 1. It is followed by a **ReLU layer** which sets all negative activation values to zero.
    b. Second Layer : The second **convolution layer** takes the 100 channel output of the previous layer and converts it into a **128 channel output**. The kernel and padding are the same as the last layer. An additional parameter for **strides is set to 1**. This layer is followed by another ReLU layer.
    c. Third Layer : This is the first **pooling layer** which **halves the activation map**.
    d. Fourth Layer : This **convolution layer** takes the 128 output of the previous layer and converts it into a **256 channel output**. Kernel, padding and stride are the same as the previous layer. This layer is also followed by a **ReLU layer**.
    e. Fifth Layer: This is a **pooling layer** which again halves the size of the activation map.
    f. Fully connected Layer : This layer is used to **flatten the activation map** obtained from the previous pipeline. A **series of linear function and ReLU layers** are applied to obtain a **final vector of size 10**.

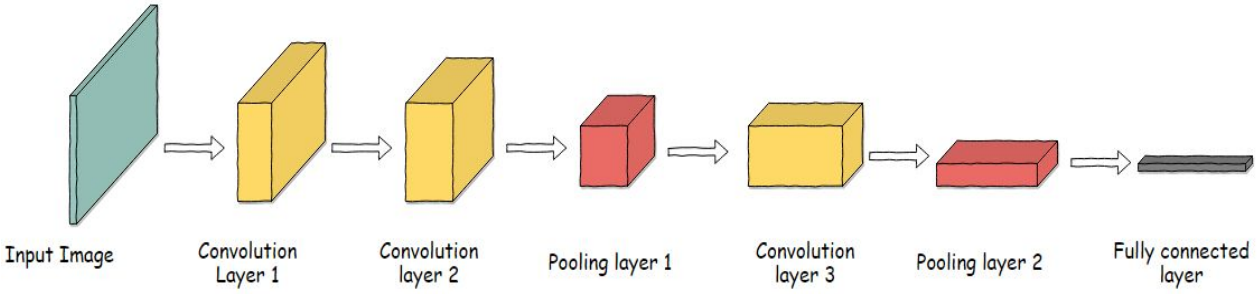


Fig 1. CNN Layers

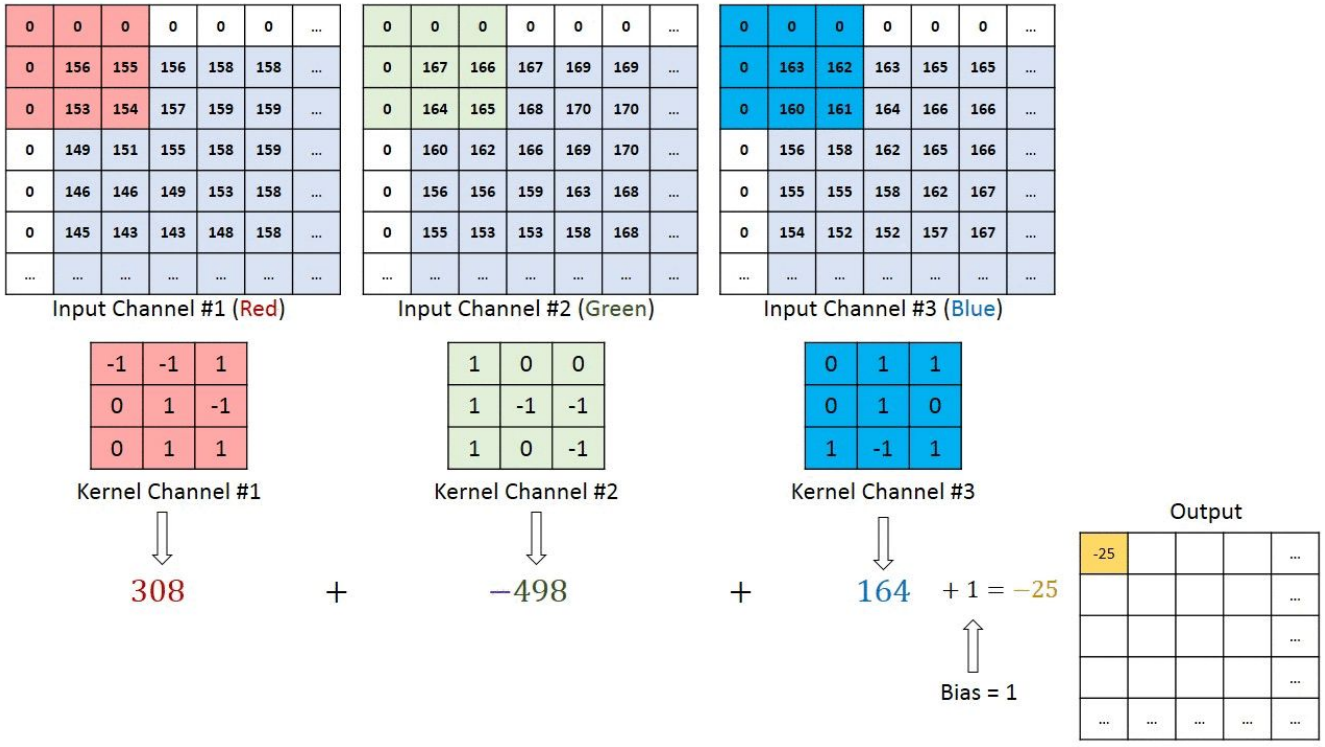


Fig 2. CNN Training

Evaluation :

1. Confusion Matrix

```
Confusion matrix, without normalization
[[317   3   5]
 [  6 330   0]
 [  0   0 339]]
```

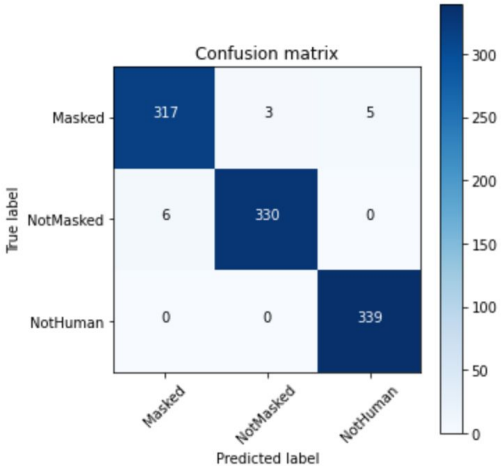


Fig. 2 Heat Map of Confusion Matrix.

2. Precision, recall, F1-measure

	precision	recall	f1-score	support
0	0.98	0.98	0.98	325
1	0.99	0.98	0.99	336
2	0.99	1.00	0.99	339
accuracy			0.99	1000
macro avg	0.99	0.99	0.99	1000
weighted avg	0.99	0.99	0.99	1000

Analysis of result:

- The obtained result shows high accuracy.
- The values of precision, recall and f-measure are lower for masked images as compared to the other 2 categories.
- The Non human data only contains images of cats and hence it shows high values for all the metrics. New images would be added to the dataset in the next phase which might lead to value changes.

Improvement:

Concern: The current Non human dataset only contains images of cats.
Improvement: To improve the model further we would incorporate other images in the next phase of the project and train our model for the new images and recalculate the evaluation metrics.

Reference Section

1. Image Classification and CNN: Convolutional Neural Network Pytorch | CNN Using Pytorch
2. Neural Networks Workshop : https://www.freecodecamp.org/news/how-to-build-a-neural-network-with-pytorch/
3. Confusion Matrix Guide: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html#sklearn.metrics.confusion_matrix
4. Confusion Matrix guide: https://deeplizard.com/learn/video/0LhiS6yu2qQ
5. Artificial Neural Networks with Pytorch: https://towardsdatascience.com/how-i-built-a-face-mask-detector-for-covid-19-using-pytorch-lightning-67eb3752fd6 1