COMP 6721 Applied ArtificialIntelligence Project, Part II

Face Mask Detection using CNN

Instructor: Dr. René Witte

Team: OB_12

Payal Chaudhary (40194488): Data Specialist

Jai Sahni (40195587): Training Specialist

Abhishek Mittal (40185375): Evaluation Specialist

1. Dataset

The collection of datasets is a vital part of each machine learning project. According to proven research, the state-of-art dataset performs well on image classification tasks and is widely used in several categorization tasks. For training a model an extensive large pool of data is required for the layers to learn the features well.

The dataset for our given project wasn't readily available and it required to be taken from **Kaggle,** a data modeling and analytics platform, and a few classes were scraped from **Google Search.**

Earlier, our dissertation consisted of images trained into 5 classes - 334 for Cloth Mask, 366 N95 Mask, 450 Mask Worn Incorrectly, 433 Surgical Mask and 472 for Without Masks. Now, in addition to it, in our balanced dataset, we have added around 500 images in each category(male, female, white, and black) for bias testing. The main motivation for selecting images in *.png format is due to its lossless format. Whenever the images undergo a transformation, they may lose some of their information during this process.

1.1 Dataset Transformations

The input images need to be set in proper format before being input to our Convolutional model. For this, a series of transformations need to be applied to the input image as described below:

- The input size of each image is different as they are taken from different sources
- Resizing the image changes the pixel information. We resize and cropped our input image.
- These are now converted to tensors and normalized.

1.2 Splitting Dataset

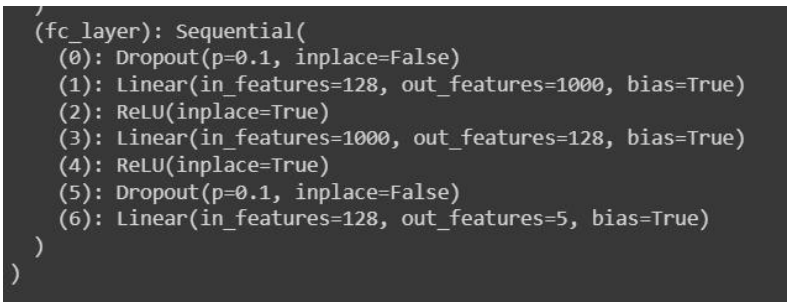
The entire data was split into 2 subsets named "training set" and "test set", each accounting for 75% and 25% respectively. Since we are working on a huge number of images, such a large dataset will be both time and memory-consuming. Pytorch, a machine learning framework

provides us pragmatic solution to such a problem. We assign a DataLoader to each of our splits – train_loader and test_loader. Our Dataloader takes three parameters as an input – dataset,batch size which indicates the number of samples passed through each generated batch and shuffle set to “TRUE” which indicates that at each epoch data is reshuffled.

2. CNN Architecture

- In total, we created 9 CNNs layers connected in sequential order, where 3 max pooling filters applied, each with kernel size set to 2, where the first max pooling filter is applied between CNN_3 and CNN_4, the second max pooling filter applied between CNN_5 and CNN_6, and the third max pooling filter applied between CNN_7 and CNN_8. Also, we have one average pooling filter with kernel size set to 4 applied after applying the last CNN layer CNN_9.

  We have also added dropout layers (P=0.1) to take into account, the overfitting of the model.

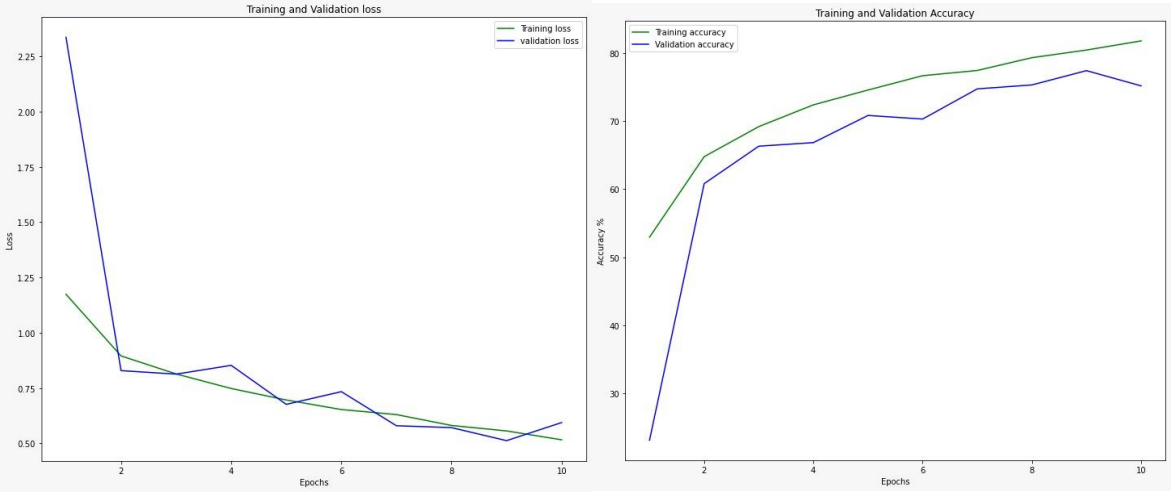


3. K-fold Cross Validation

Instead of splitting the entire dataset between training samples and testing samples, we applied the K-fold Cross Validation. We divided the original dataset into 10 folds with enabled random shuffling, and a random state seed set to 42. Each fold executed 10 epochs with a batch size set to 96 samples while performing training and validation.

Figure below shows aggregated training and validation loss curve of the model; (left side), also it shows aggregated training and validation accuracy curve of the model;(right side). Both curves measure the performance of the model across the ten cross-validation folds applied by the model.

In the first 3 epochs, both, the training loss and validation loss show smooth decreasing, meanwhile, we can see a sudden irregular loss variation in the validation part till it reaches the minimum loss which is equal to around (0.62) at epoch 10 of the 10^{th} fold. On other hand, the training loss shows a smooth decrease till it hits the minimum value, which is around (0.49) at epoch 10 of the 10^{th} fold. After performing the k-fold validation, the averaged k-fold's performance during 10 epochs is: Averaged Accuracy- 75.75%, Averaged Precision- 76.7%, & Averaged F1- Score- 73.96%.



Comparing the averaged testing accuracy (71.34%) when applying K-fold cross validation with the testing accuracy scored in the testing stage of the model without the k-fold cross validation

(75.68%), we can tell that the performance of the K-fold cross validation is a bit less than the model performance when dividing the dataset into training samples and testing samples. The same thing follows when comparing the precision, the recall, and the f1-score values, where they achieved better results when applying the model of the second phase on the dataset divided into training samples and testing samples with no k-fold cross validation.

4. Bias Detection & Analysis

For detecting bias in our dataset source of images, we trained our model on the full original dataset we have. The full dataset contains all the images for different genders and races of people for each category, where it contains approx. 200 images for each.

After training the models, we tested the trained model against 4 different biased datasets, where each category and include only images of specific category of people. The biases we decide to detect and analyze in our models are based on gender (male, female) and race (black people, and white people).

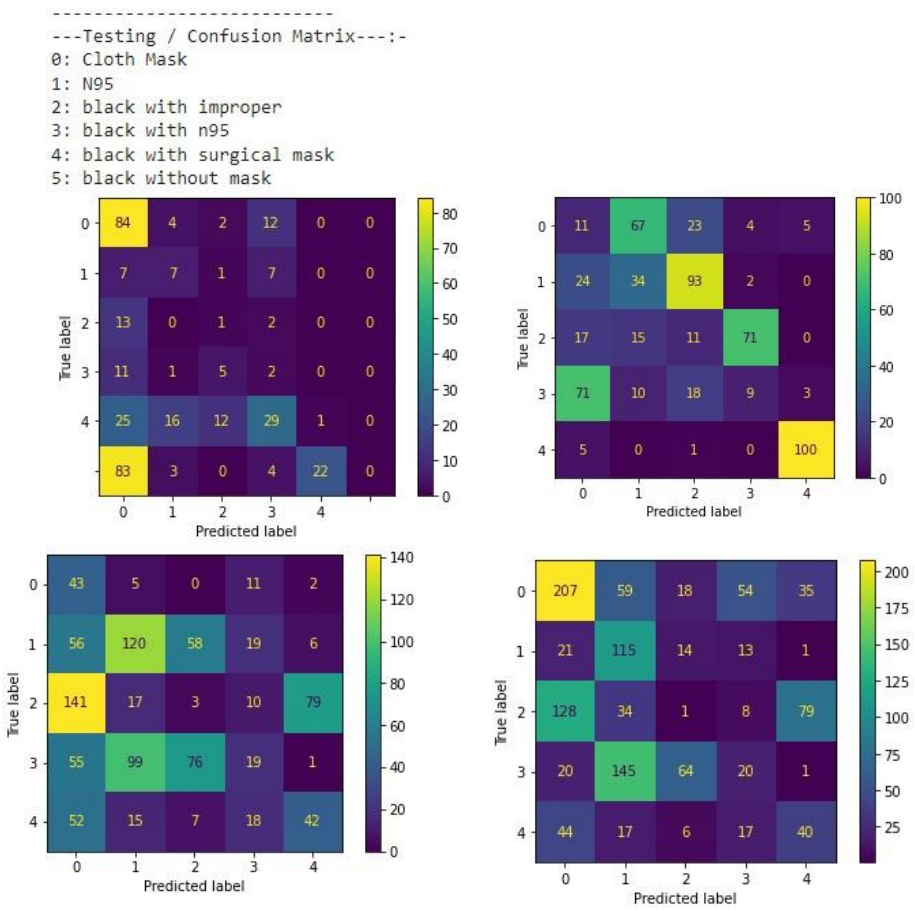
According to the performance metrics of accuracy, precision, recall, and f1-score values it seems the tested model successfully classified the images of the datasets of the female people with high performance metrics (accuracy, precision, recall, f1-score) compared to other tested biased categories. This gives clear indication that the majority of the images in the full datasets used to train the models include more images of female people compared to male, black people, and white people biased categories.

For fixing the bias we detected in the original models, we created a balanced dataset which includes equal number of images for each type of race or gender analyzed in our bias categories we mentioned before. This balanced training dataset should enhance the interception of the models in the training phase against the biased tested datasets. New performance measures obtained when testing the model against the biased testing datasets after training the models using the balanced dataset.

In the balanced dataset, we made sure each type of face masks category has equal number of images compared to other types of face masks categories, we included in total around 500 images for each face masks type category, where each category includes around 120 images out

of the 500 images per each biased category (gender (male, female), race (black, white)) to make sure we have equal number of images in the biased datasets while training the model. In total we have around 2000 images as a balanced dataset used to train the model.

According to the performance metrics of accuracy, precision, recall, and f1_score values, it seems the tested models successfully improved the performance of classification for white race, and male gender biased datasets, the classification performance of black race dataset almost kept the same compared to the results obtained when training the model using the unbalanced dataset, meanwhile the performance of the classification for the female biased dataset decreased compared to the results obtained from the unbalanced dataset, this decrease of classification performance in the female bias dataset is expected because the number of images of the this biased category is not the same as the one used in the unbalanced dataset.



Conclusion:

For eliminating or decrease the bias detected in running models and enhance the performance of classification, we suggest to increase the number of images in the biased datasets and the quality of resolutions captured on these images. Also, we need to make sure to include equal number of images in each biased category used in the balanced dataset used to train the model, and the biased tested datasets by the models after the training.

References

[1] https://www.kaggle.com/prasunroy/natural-images

[2] https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

[3] https://pytorch.org/docs/stable/data.html

[4] https://towardsdatascience.com/understanding-pytorch-with-an-example-a-step-by-step-tutorial-81fc5f8c4e8e

[5] https://www.kaggle.com/datasets/vijaykumar1799/face-mask-detection

[6] Marceddu A. C., Montrucchio B.: Recognizing the Type of Mask or Respirator Worn Through a CNN Trained with a Novel Database. 2021 IEEE 45th Annual Computer Software and Applications Conference (COMPSAC). (2021).

[7] Marceddu A. C., Montrucchio B.: Facial Masks and Respirators Database (FMR-DB). IEEE DataPort (2020). http://dx.doi.org/10.21227/wg71-v415.