Face Mask Detection Using Deep Learning and Machine Learning

ICS 381: Principles of Artificial Intelligence

Final Report

12/8/2020 - Term 201

For

Prof. EL-SAYED EL-ALFY

Group members:

Mohammed Alghamdi	*201670960*
Ahmed Alzahrani	*201642960*
Azzam Alkhaldi	*201694080*

Contents

1. Introduction....4

2. Problem Description. 4

3. Related Resources. 5

3.1 Face Recognition using Deep Learning.... 6

3.2 Face Classification using Machine Learning (Supervised). 7

4. Team Solution....8

4.1 Dataset. 8

4.2 Mask Detection using Deep Learning. 9

4.3 Mask Detection using Machine Learning.... 10

4.4 Deployment.... 11

5. Results....11

5.1 Mask Detection using Deep Learning. 12

5.2 Mask Detection using Machine Learning.... 13

6. Conclusion. 14

6.1 Problem.... 14

6.2 Our approach.... 14

6.2.1 Deep Learning approach 14

6.2.2 Machine Learning approach 14

6.3 Implementing the Telegram Bot 15

6.4 Issues faced.... 15

6.5 Improvements that can be made 15

6.6 Lessons learned 16

7. Resources. 17

8. Appendix. 18

Team roles and meetings

The team started working on the first phase project on the 23rd of October and meeting on Microsoft teams using group calls. For the phase, Mohammad Alghamdi had the idea of using Pytorch and Sklearn to be our 2 algorithms for the project and we worked on it as a team, Mohammad Alghamdi and Azzam Alkhaldi writing the code and Ahmad Alzahrani debugging and refactoring. We then decided to make the project into a Telegram bot for ease of access to anyone who wants to test the algorithm, the basic code for the bot was provided by Telegram and we implemented out code in it.

Distribution of work:

- Phase 1 and proposal:

  Idea: Mohammad Alghamdi.
  Writing report: All team members, but mostly Ahmad Alzahrani.
  Collecting resources and datasets: Mohammad Alghamdi & Azzam Alkhaldi.

- Final phase:

  Coding and bot implementation: Mohammad Alghamdi & Azzam Alkhaldi.
  Debugging: Ahmad Alzahrani.
  Writing the final report: All team members.
  Power point presentation: Ahmad Alzahrani.

Meetings

We met on our official project group multiple times, especially for the final phase. Here is the list our meeting we made:

- 1st of December, duration: 2hrs 34mins. Summary:
  - We have discussed work distribution between team members
  - We have discussed proposed solution in previous phases and how to work on them
  - The tools to be used for development (Colab, Jupyter)
  - How to do the report and the presentation

- 3rd of December, Duration: 9 minutes. Summary: Tasks update meeting.
- 5th of December, Duration: 2hrs. Summary: Model training and several bugs fixing.
- 6th of December, Duration: 1hrs 48mins. Summary: Telegram bot initial implementation and local machines setup.
- 7th of December, Duration: 3hrs. Summary: Telegram bot final implementation and report writing.

1. Introduction.

During these times we are going through unprecedented pandemic crisis that the whole world is affected by. The Corona virus has taken place everywhere affecting all different aspects of people's lives. There are people dying due to the dangerous infection of this virus. However, governments and authorities have taken actions in order to fight the spread of this disease and enforce new legislation on different levels to protect people from getting infected. Additionally, air flights have been banned regionally and globally, penalties have been applied to those who doesn't follow the new arrangements and laws protecting them from getting infected. By the time this document is written there are 67,185,262 registered cases of the corona virus, 1,539,454 of them are death cases (Coronavirus Cases, n.d). It is a serious matter and needs attention from people and care at the same time in order to prevent the increase in the number of casualties.

2. Problem Description.

One issue has been faced by authorities is that some people are not abiding by the rules and laws that has been offered by governments to protect them. For example, one of the new legislations introduced is wearing a face mask in order to minimize the probability of transferring the infection between people. However, there are some people ignoring these laws and don't realize the fact behind applying such laws. Although, penalties have been introduced for those not wearing a face mask. However, tracking people in public places like malls, restaurants, or workplaces is a challenge for authorities. Finally, we are trying to solve this issue by offering a face-mask detection system that can be used by governments and official authorities in order to detect people not wearing a face mask in public places.

The team has developed an Artificial Intelligence (AI) system that can determine if a person is wearing a mask or not. AI includes different fields for different applications like; Machine Learning (ML) and Deep Learning (DL). The two techniques will be presented in this report to produce a face mask detection system. Developing ML and DL system from scratch is a challenging task and requires prior and solid knowledge of the two fields. As a result, the team will use some python libraries to ease and speed the process of developing the system. The final product (models) will be developed and saved in a way that will increase portability. This will allow the team to use the same model in different environments in the future.

3. Related Resources.

Many creative solutions and ideas were invented by scientists to solve face detection and recognition problems. Many algorithms and machine learning techniques were developed and published throughout the years since the 1960s, according to ReadWrite (as cited in Dharaiya, 2020). Moreover, today facial recognition has many applications that ease people's life like face authentication for smartphones.

Face detection algorithms are various. Also, many scientific papers have been published to accomplish the best solution. However, two existing solutions will be discussed. First, face recognition using deep learning Convolutional Neural Network (CNN), or Multi-Task Cascaded Convolutional Neural Network (MTCNN). The first solution will provide a demonstration for Cascade Classifier using the MTCNN python library. Second, face classification using machine learning. The second solution is the most similar solution for the face mask detection problem because the problem is to classify not to recognize. In this solution, a machine learning model will be trained to classify different people based only on pixels. A feature extraction technique will be used to reduce the features (pixels size) and achieve better results.

3.1 Face Recognition using Deep Learning.

Deep learning neural networks are challenging to implement and required massive data and a powerful computer to train. This solution will demonstrate Multi-Task Cascaded Convolutional Neural Network (MTCCN) and how to recognize faces in an image. MTCCN consisted of three networks Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net) as shown in figure01 (Dwiyantoro, 2018). Fortunately, python has a library called MTCCN that implements the MTCCN network with a pre-trained model. MTCCN can predict a single face and multiple faces in the same picture. The pictures were used from a Kaggle dataset can be found at (DataTurks, 2018).

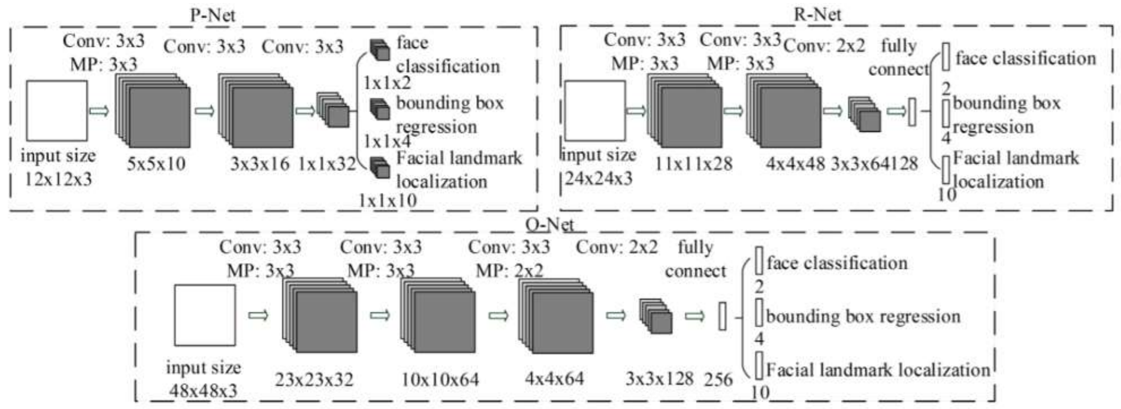
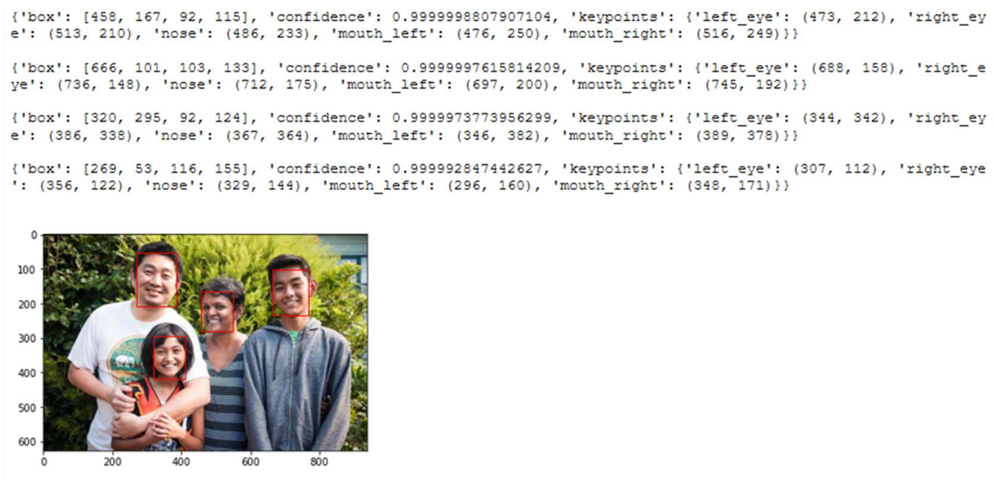


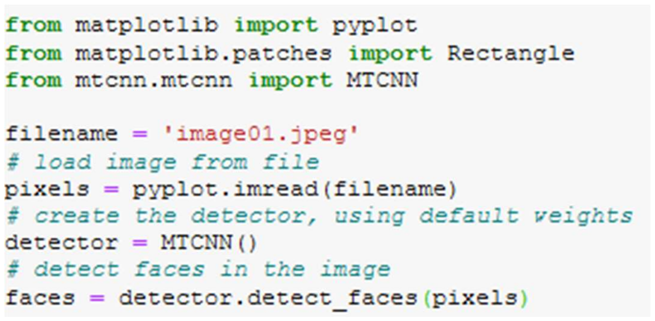
Figure01: MTCCN Architecture

MTCCN library returned multiple values for each detected face in the picture which is the box coordinate for the detected face, the confidence of the prediction, and the coordinates for the (left eye, right eye, nose, mouth left, and mouth right) as shown in figure02.



```
{'box': [458, 167, 92, 115], 'confidence': 0.9999998807907104, 'keypoints': {'left_eye': (473, 212), 'right_ey
e': (513, 210), 'nose': (486, 233), 'mouth_left': (476, 250), 'mouth_right': (516, 249)}}

{'box': [666, 101, 103, 133], 'confidence': 0.9999997615814209, 'keypoints': {'left_eye': (688, 158), 'right_e
ye': (736, 148), 'nose': (712, 175), 'mouth_left': (697, 200), 'mouth_right': (745, 192)}}

{'box': [320, 295, 92, 124], 'confidence': 0.9999973773956299, 'keypoints': {'left_eye': (344, 342), 'right_ey
e': (386, 338), 'nose': (367, 364), 'mouth_left': (346, 382), 'mouth_right': (389, 378)}}

{'box': [269, 53, 116, 155], 'confidence': 0.999992847442627, 'keypoints': {'left_eye': (307, 112), 'right_eye
': (356, 122), 'nose': (329, 144), 'mouth_left': (296, 160), 'mouth_right': (348, 171)}}
```

Figure02: MTCCN Output



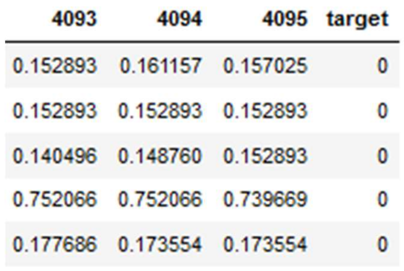
```
from matplotlib import pyplot
from matplotlib.patches import Rectangle
from mtcnn.mtcnn import MTCNN

filename = 'image01.jpeg'
# load image from file
pixels = pyplot.imread(filename)
# create the detector, using default weights
detector = MTCNN()
# detect faces in the image
faces = detector.detect_faces(pixels)
```

Figure03: Code sample for using MTCNN library.

As mentioned before the mask detection is a classification problem, not a detection problem. So, the team need to do more research to find a better and simple solution to classify a picture either with a mask or not.

3.2 Face Classification using Machine Learning (Supervised).

A simpler approach to classify images is by training a machine learning model with pixels values only (between 0 and 1). However, pixels might not provide any information for the machine learning model because pixels values are not related and relevant to each other. So, a process is known as Principal Component Analysis (PCA) is applied to produce a new dataset with small dimensionality (Radečić, 2020). PCA will generate a more helpful dataset to train a machine learning algorithm like Support Vector Machine (SVM).



4093	4094	4095	target
0.152893	0.161157	0.157025	0
0.152893	0.152893	0.152893	0
0.140496	0.148760	0.152893	0
0.752066	0.752066	0.739669	0
0.177686	0.173554	0.173554	0

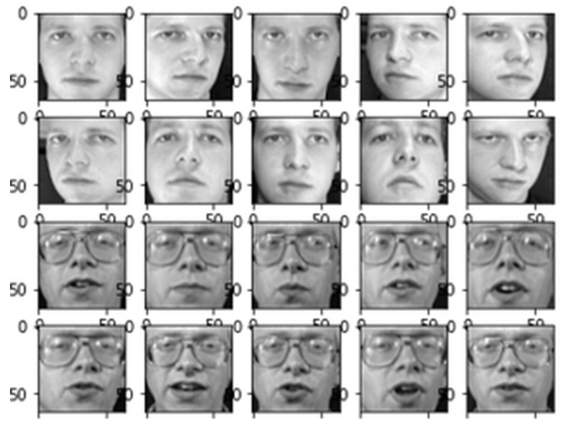


Figure04: Face Classification Using Machine Learning Dataset.

The dataset consists of multiple pictures of different people. The pictures are 64*64=4096 pixels which corresponded to the number of features (columns) in the dataset. The last feature (column) represents the person in the picture (target) as shown in figure04 (Radečić, 2020). Using the generated data from PCA a (SVM) model was trained to achieve an accuracy of around 90%. This approach is really useful for the mask detection problem because instead of classifying different people the machine learning algorithm is trained for two classes either with a mask or without a mask.

4. Team Solution.

4.1 Dataset.

The team set some criteria to accept a dataset. The dataset should be efficient and diverse. Efficient means the dataset should include a large number of images for training and validation, and diverse means the dataset should include different people, masks colors, image background, positions, etc. First dataset consists of 1376 images divided into two groups with a mask or without a mask. 690 images of people wearing masks and 686 images of people not wearing masks. The data was generated by simply adding a mask image on top of a person's image as shown in figure05. The data was created by Prajna Bhandary (Degila, 2020).

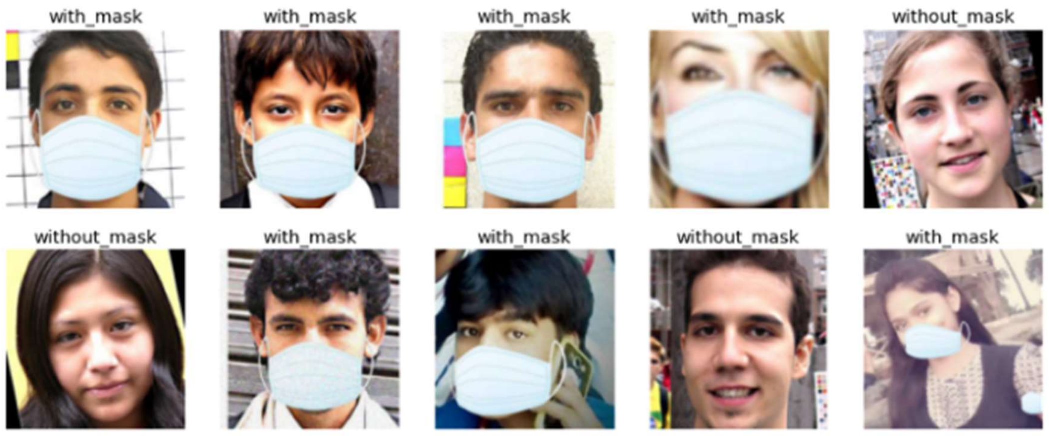


Figure05: The Face Mask Detection First Dataset.

Unfortunately, the first dataset failed in achieving good results, especially when changing mask color or give a picture with a noisy background; the models tended to give the wrong prediction. As a result, a new dataset was found that includes the first dataset with more diverse pictures with real masks as shown in figure06. The dataset consists of 7553 images. 3725 images of people wearing masks and 3828 images of people not wearing masks. The second dataset help in improving the models amazingly. The dataset was found in kaggle (Omkar, 2020).

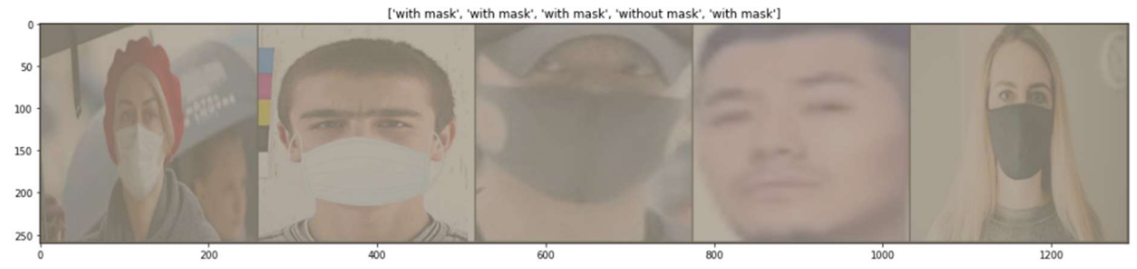

['with mask', 'with mask', 'with mask', 'without mask', 'with mask']
0
50
100
150
200
250
0
200
400
600
800
1000
1200

Figure06: The Face Mask Detection Second Dataset.

4.2 Mask Detection using Deep Learning.

Neural networks are fascinating and interesting to learn figure07. However, implementing a neural network and understand all the terminologies like loss function, optimization, backpropagation, feedforward, stochastic gradient descent, etc. required hard work and a lot of time. Fortunately, many libraries solve this problem like TensorFlow, Keras, and PyTorch by providing well-documented documentation and implement everything to build a simple neural network architecture or even complex architectures. PyTorch is a python library and will be used in this approach. As a result, PyTorch will be used to build, train, test, and use a simple neural network to classify a picture either with a mask or without a mask.

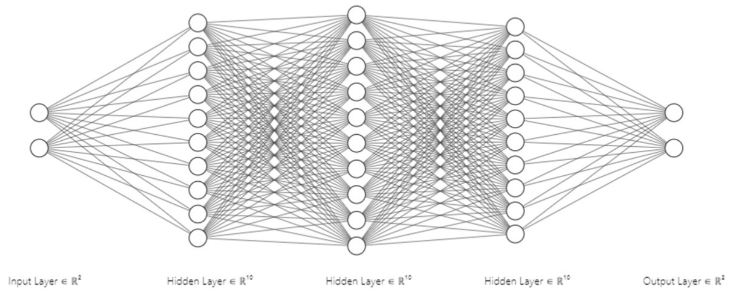


Figure07: A Neural Network Architecture with input, output, and 3 hidden layers.

Deciding the optimal architecture for a problem is an art, not a science. However, there are some known facts for a better neural network like a deeper network (more layers), and differentiable activation functions (like sigmoid or ReLU). However, PyTorch provides efficient pre-made neural network architectures that can be used immediately. As a result, a resnet18 network architecture will be used. Resnet18 consisted of convolutional, batch normalization, activation, pooling layers. A simplified resnet18 architecture can be shown in figure08.

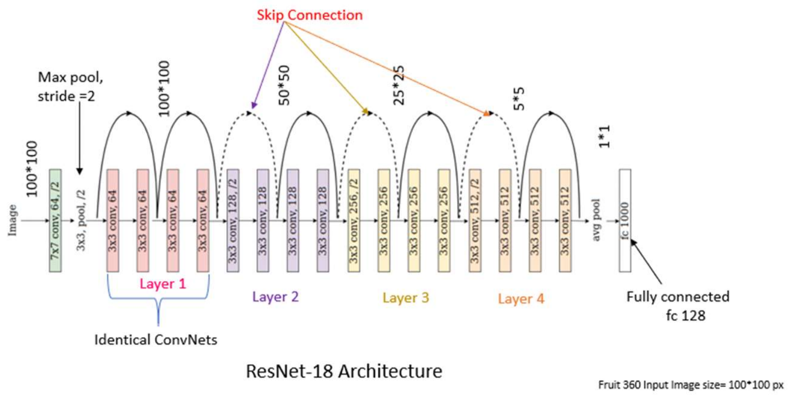


Figure08: ResNet-18 Architecture.

Finally, the cross-entropy loss divergence function and stochastic gradient descent optimizer will be used. Training a neural network required a powerful PC. However, google colab provides a service for using GPU which will reduce the training time significantly.

4.3 Mask Detection using Machine Learning.

Using only the pixels to train a machine learning model seems so attractive to use. The reason is that no data pre-processing or image transformation is required. However, using only pixels is not efficient neither in terms of accuracies nor in terms of calculations amount. As a result, a feature extraction technique will be used to generate more efficient features and reduce the number of features. Histogram of Oriented Gradients (HOG) descriptor will be used in the feature extraction step. HOG descriptor extract features by focusing on the structure or the shape of an object as shown in figure09 (Singh, 2020).

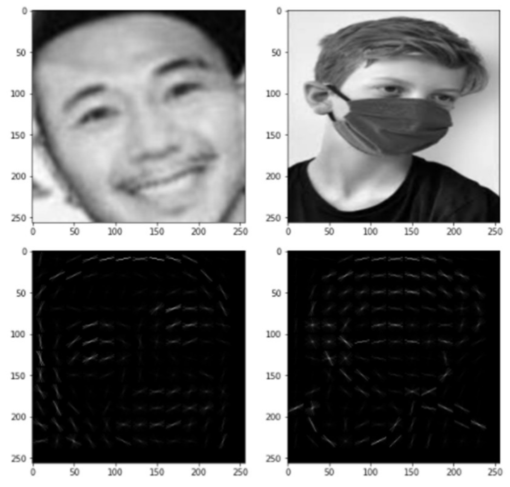


Figure09: Histogram of Oriented Gradients.

HOG helped to reduce the number of features from (256*265 = 65536) to 9000. Different machine learning algorithms will be used which are Support Vector Classifier (SVC), Gaussian Naive Bayes (GNB), Multi-Layer Perceptron (MLP), Decision Tree, and Random Forest. Using different machine learning algorithms with cross-validation will help to determine the best machine learning according to accuracy.

4.4 Deployment.

The two solutions will be deployed as a Telegram bot for portability. So, anyone will be able to try the two solutions using any device that has the Telegram App. The bot would work as follow, a user would send a picture of himself with a mask or without a mask, and the Telegram bot will pass the image to our program, then the program will classify the image as “with mask” or “with-out mask”. After that it will pass the result back to the bot and the bot will send the result to the user.

5. Results.

Amazingly both approaches achieved astonishing results in predicting if a person wearing a mask or not. The two models were validated and tested using the validation dataset and external images. However, the deep learning model achieved better results than the machine learning model. The deep learning model acts more rationally with external data. On the other hand, the machine learning model acts great with similar data to the dataset which might indicate that the model is overfitted to the dataset.

5.1 Mask Detection using Deep Learning.

Resnet18 was used as discussed before. The model was trained for 10 epochs which means the training process goes over the same training dataset 10 times. The model started with great results since the first epoch and achieved almost 100% training accuracy by the 10th epoch. As figure10 shows the rapid increase in training accuracy and validation accuracy which indicates good architecture for this application. Moreover, the continues increase in the validation accuracy indicates that the model is not overfitted to the training dataset.

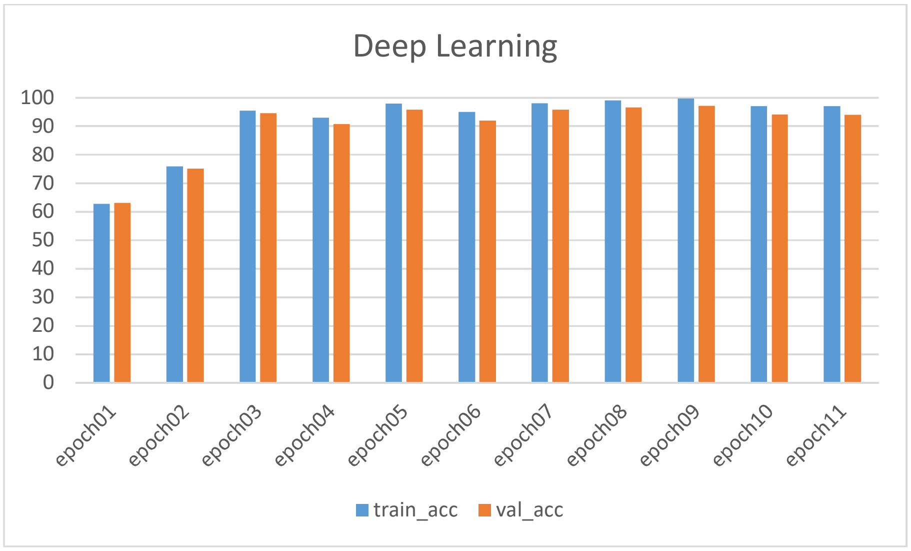


Figure10: Resnet18 Results

5.2 Mask Detection using Machine Learning.

Machine learning algorithms are variant, and some of them achieve better in some applications. So, to find the best algorithms, cross-validation with 5-fold was used with Support Vector Classifier (SVC), Gaussian Naive Bayes (GNB), Multi-Layer Perceptron (MLP), Decision Tree, and Random Forest. The best algorithm was SVC with an 85.5% accuracy. However, almost all the models achieved good results as shown in figure11. After deciding the best model, SVC was trained and achieved 86% validation accuracy.

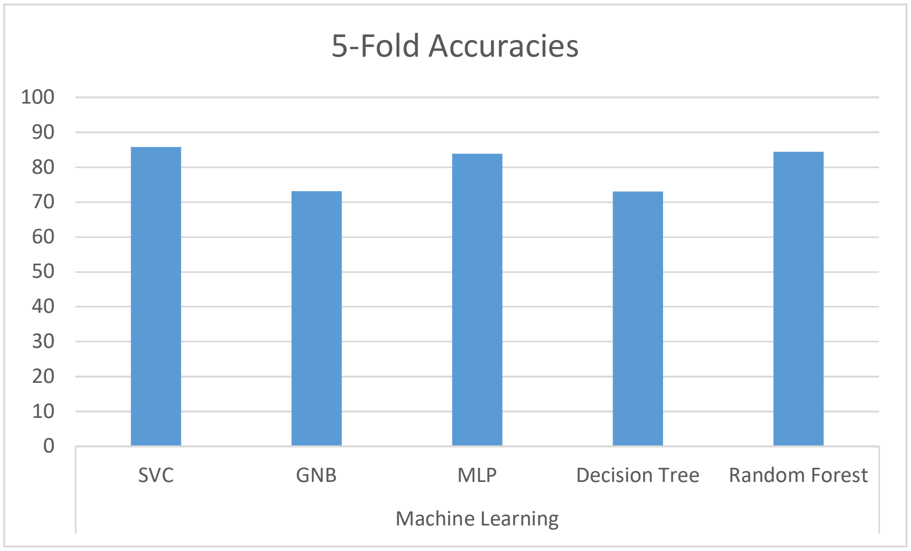


Figure09: 5-Fold Results.

6. Conclusion.

6.1 Problem

In this project we tackled one of the most traditional Artificial Intelligence problems, which is a face feature recognition problem. We needed to find out if a person is wearing a face mask or not to try and solve one of the challenges that we as the world are facing in this pandemic.

6.2 Our approach

We approached this problem with 2 AI strategies, Machine Learning and Deep Learning. We needed to test it over these approaches to finalize an optimal method for approaching such problem.

6.2.1 Deep Learning approach

Our Deep Learning approach consisted of using a large dataset of people’s images with and without face masks (7000+ images) and training our model on them over 10 epochs. We trained the model on Google Colab, and after almost a day of training we finally ended up with an accuracy close to 100%. The model achieved nearly perfect result in the testing phase after finalizing the model.

6.2.2 Machine Learning approach

Our Machine Learning approach wasn’t as successful as the DL approach but achieved a pleasing result, nonetheless. After using multiple algorithms to define the best one to implement (SVC, RF, GNB, MLP and DT), we finally ended up with the SVC algorithm with an accuracy of 86%. All training and processes were made using 5 folds cross-validation method.

6.3 Implementing the Telegram Bot

After having the model set and ready for use, we needed a fast and accessible way to display our program on. Using a Telegram bot was one of the best solutions that we found since its fairly simple to implement, and also you can use it online. All we needed to do was to add the bot to Telegram's api and use our token for communication between the code and Telegram's api. Telegram had a ready to use base code for fast implementation and had a really good documentation for all of their function regarding handling incoming messages.

6.4 Issues faced

Our first issue that we needed to solve, was determining what approaches we need to use to solve this problem, luckily our solution was acceptable in the end. Also, one of the largest challenges was optimizing and retesting the code and training over and over again, the training process takes substantial amount of time and needing to redo it 10 times just to make sure that it is optimal was the most time-consuming part of the project. Another issue we faced is Python on Windows, Windows does not support Python natively, so the amount of errors and debugging on our own systems that we had to do has wasted a lot of time and resources to fix.

6.5 Improvements that can be made

The first thing we need to do to improve our system is to stick with one of the approaches we have, the DL approach is the best candidate in this case. Next, since we have the trained model ready for business, we can modify our code to work with video input rather than image input. Doing that will enable the program to be used in malls and airports to detect violations in real time. Also, retraining the model with a larger more real life-accurate dataset possibly gathered from public cameras will definitely result in more accurate results.

6.6 Lessons learned

One of the biggest gains from doing this project is identifying the best approach for certain kinds of problems. In our case, we now know that the best approach for classifying people’s images based on face features is to use Deep Learning rather than Machine Learning. Although Machine Learning might achieve better results depending on how we present our dataset.

7. Resources.

Dwiyantoro, A. (2018, April 04). The Evolution of Computer Vision Techniques on Face Detection, Part 2. Retrieved October 8, 2020, from https://medium.com/nodeflux/the-evolution-of-computer-vision-techniques-on-face-detection-part-2-4af3b22df7c2

Dharaiya, D. (2020, March 10). Face Recognition Technology Past, Present, and Future. Retrieved October 10, 2020, from https://readwrite.com/2020/03/12/history-of-facial-recognition-technology-and-its-bright-future/

Radečić, D. (2020, May 27). Eigenfaces-Face Classification in Python. Retrieved October 12, 2020, from https://towardsdatascience.com/eigenfaces-face-classification-in-python-7b8d2af3d3ea

Degila, K. (2020, October 22). How to deploy Machine Learning models on Android and IOS with Telegram Bots. Retrieved October 12, 2020, from https://medium.com/analytics-vidhya/how-to-deploy-machine-learning-models-on-android-and-ios-with-telegram-bots-a6fb16922741

Singh, A. (2020, May 10). Feature Descriptor: Hog Descriptor Tutorial. Retrieved December 5, 2020, from https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/

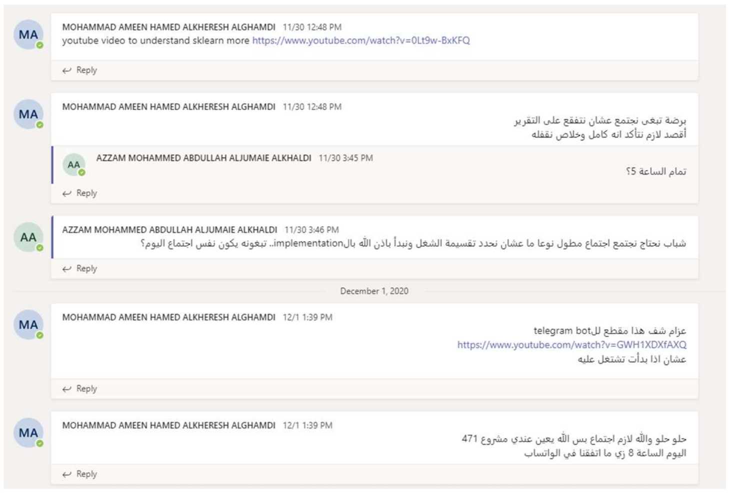
DataTurks. (2018). Face Detection in Images (V1.0) [Data file]. Retrieved from https://www.kaggle.com/dataturks/face-detection-in-images/metadata

Omkar Gurav. (2020). Face Mask Detection Dataset (V1.0) [Data file]. Retrieved from https://www.kaggle.com/omkargurav/face-mask-dataset

Coronavirus Cases:. (n.d.). Retrieved December 01, 2020, from https://www.worldometers.info/coronavirus/

8. Appendix.

Screenshots of team meetings in Microsoft Teams:



MA MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 11/30 12:48 PM
youtube video to understand sklearn more https://www.youtube.com/watch?v=0Lt9w-BxKFQ

Reply

MA MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 11/30 12:48 PM
برضة تبغى نجتمع عشان نتفقع على التقرير
أقصد لازم نتأكد انه كامل وخلاص نقفله

AA AZZAM MOHAMMED ABDULLAH ALJUMAIE ALKHALDI 11/30 3:45 PM
تمام الساعة 5؟

Reply

AA AZZAM MOHAMMED ABDULLAH ALJUMAIE ALKHALDI 11/30 3:46 PM
شباب نحتاج نجتمع اجتماع مطول نوعا ما عشان نحدد تقسيمة الشغل ونبدأ باذن الله بالimplementation.. تبغونه يكون نفس اجتماع اليوم؟

Reply

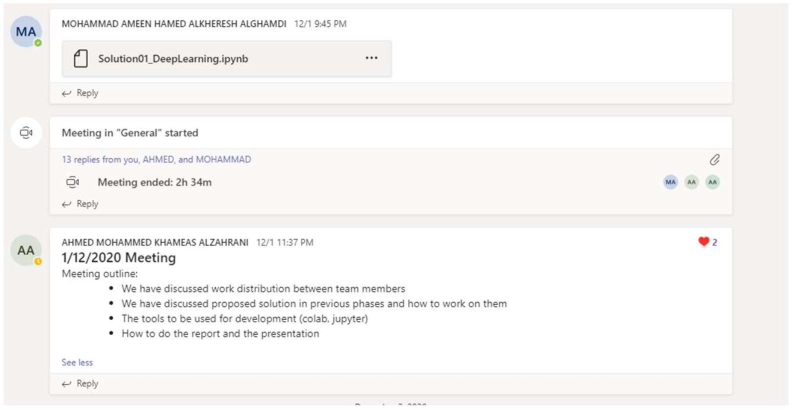
December 1, 2020

MA MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 12/1 1:39 PM
عزام شف هذا مقطع للtelegram bot
https://www.youtube.com/watch?v=GWH1XDXfAXQ
عشان اذا بدأت تشتغل عليه

Reply

MA MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 12/1 1:39 PM
حلو حلو والله لازم اجتماع بس الله يعين عندي مشروع 471
اليوم الساعة 8 زي ما اتفقنا في الواتساب

Reply



MA MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 12/1 9:45 PM
Solution01_DeepLearning.ipynb

Reply

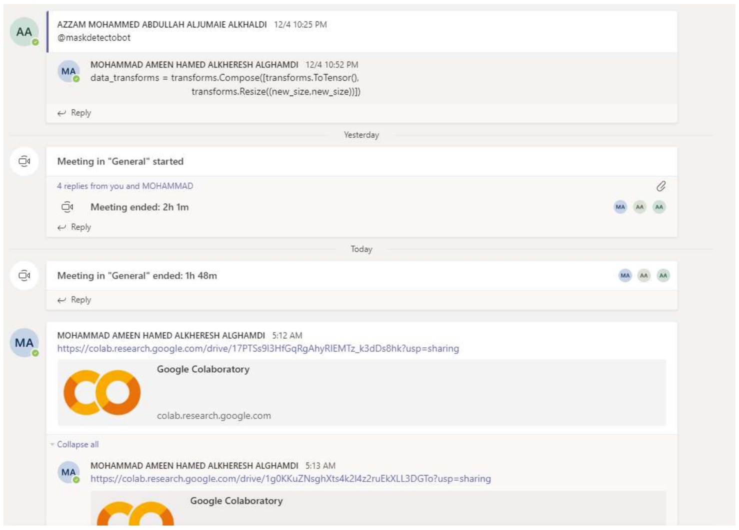
Meeting in "General" started
13 replies from you, AHMED, and MOHAMMAD
Meeting ended: 2h 34m

Reply

AA AHMED MOHAMMED KHAMEAS ALZAHRANI 12/1 11:37 PM
1/12/2020 Meeting
Meeting outline:
- We have discussed work distribution between team members
- We have discussed proposed solution in previous phases and how to work on them
- The tools to be used for development (colab, jupyter)
- How to do the report and the presentation

See less

Reply



AA AZZAM MOHAMMED ABDULLAH ALJUMAIE ALKHALDI 12/4 10:25 PM
@maskdetectobot

MA MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 12/4 10:52 PM
```
data_transforms = transforms.Compose([transforms.ToTensor(),
                        transforms.Resize((new_size,new_size))])
```

Reply

Yesterday

Meeting in "General" started
4 replies from you and MOHAMMAD
Meeting ended: 2h 1m

Reply

Today

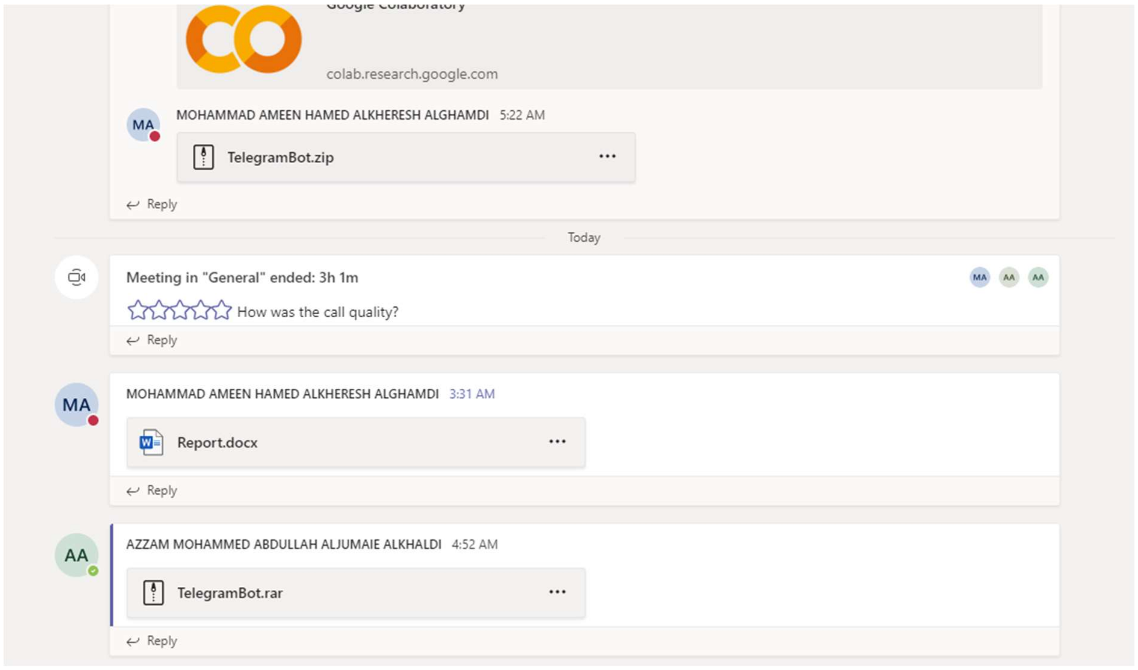
Meeting in "General" ended: 1h 48m

Reply

MA MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 5:12 AM
https://colab.research.google.com/drive/17PTSs9I3HfGqRgAhyRIEMTz_k3dDs8hk?usp=sharing
Google Colaboratory
colab.research.google.com

Collapse all

MA MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 5:13 AM
https://colab.research.google.com/drive/1g0KKuZNsghXts4k2l4z2ruEkXLL3DGTo?usp=sharing
Google Colaboratory



colab.research.google.com

MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 5:22 AM

TelegramBot.zip

Reply

Today

Meeting in "General" ended: 3h 1m

How was the call quality?

Reply

MOHAMMAD AMEEN HAMED ALKHERESH ALGHAMDI 3:31 AM

Report.docx

Reply

AZZAM MOHAMMED ABDULLAH ALJUMAIE ALKHALDI 4:52 AM

TelegramBot.rar

Reply