Facial recognition

- Tim. E. Doughery: Undergraduate Senior in Comp. Sci.
- Chenhao Lu: Undergraduate Senior in Comp. Sci.
- Seho Chung: Undergraduate Senior in Comp. Sci.

Introduction:

Facial recognition is not a new field of study. There have been many implementations of the concept. Perhaps the most famous implementation, and the one most relevant to our project, is Google's facenet. There are various reasons why one might want to implement facial recognition. The most innocent being the simple facial recognition found in many flagship smartphones. Facial recognition, however, may be used for controversial uses, especially by governments for law enforcement purposes, such as identifying and tracking criminals or even civilians by using camera footage. Despite its controversial use, it remains a popular enough topic of research, and as technology advances, the accuracy of facial recognition models will only increase (last year, a certain algorithm was able to achieve an error rate of only 0.08%). However, accuracy does see a decrease when tested on images of non-static models or especially when the face is obscured by a mask; this is what our project hoped to test.

For this project, what we intended on doing was to develop a model capable of recognizing and classifying given faces, and then further develop it for the use of recognizing masked faces. However, our goal was not to create an accurate and reliable model for facial recognition, even when obscured by masks, but to simply develop a usable model for such uses, and to then learn from it for future reference. Thus, our motivation was not one of improvement, but rather understanding how one might develop a model.

Method:

For the purposes of this project, the central dataset we decided to use was deep-funneled LFW, a collection of faces originally derived from the Labelled Faces in the Wild (LFW) dataset from the University of Massachusetts Amherst. The advantage of using deep-funneled LFW over the base dataset is that the faces are pre-aligned, which was incredibly useful for the preprocessing step. Secondly, we also used the LFW Simulated masked face dataset for the second and third part of the project to test the model's ability to recognize and classify masked faces. Lastly, we also used a custom-made collection of data that was a mix of both masked and unmasked faces so that we can train and later test the model on both.

For preprocessing, what we had to do was both facial detection and feature extraction. As the masked dataset was already cropped to show only the faces, we mainly used preprocessing on the deep-funneled set. For facial detection, what we did was use an imported haar-cascade classifier to detect the faces, and then crop around the boundaries of what was detected. The haar-cascade classifier, however, was not perfect, and often failed to detect a face, instead returning nothing. To get around this issue, we cropped the approximate location of where the face should be since we were using a deep-funneled set that was pre-aligned. Lastly, we resized the output of this preprocessing to (160, 160) since that was the input size to facenet. To create the mixed dataset, we simply saved the results of detecting and cropping the faces within the deep-funneled set, and then merged it together with the masked set, and then set aside a collection for the purposes of testing.

For feature extraction, we used Google's facenet compiled with triplet loss and adam optimizer to generate a 128-dimensional vector given an input image representing the features or embeddings of the face. First, however, we had to standardize the data contained with a given face array since that's what the particular implementation of facenet that we used requires. The last step before extraction is to expand the dimensions of the face array to transform it into one sample. Then for feature extraction, we used facenet on the face array to convert them into embedding vectors. Finally, before training and then testing the classifier on these embeddings, we had to normalize them using L2-norm and then convert the true labels of the faces, which had turned into one-hot encodings by ImageDataGenerator to integers.

To classify these faces and their features, we used a linear support vector classifier. The parameters we used were squared-hinge loss, a tolerance of 1e^-4, and a max one-thousand iterations.

After initially training the classifier on the deep-funneled set, we then tested the model on the same set until we were able to achieve a satisfactory accuracy; for the purposes of this project, the accuracy criteria to finalize training was at least 80%. Once we achieved an acceptable accuracy on the deep-funneled set, we then experimented on the masked set without training on masked images, as was inherent to the objective of this project. After that, to improve accuracy on the masked set, we retrained the model on a mixed set containing both masked and unmasked faces. To evaluate the accuracy of this model, we once again tested it on the masked set. There was no essential criteria to finalize training for this step beyond it being better than only half probability of correct classification.

Results:

After being trained on the deep-funneled set, we tested the model's accuracy and were able to achieve 96.269% probability of correct classification on the testing set. For convenience, we visualized images and predictions by iterating through the testing set using the same index for iteration to get the predicted and actual label.

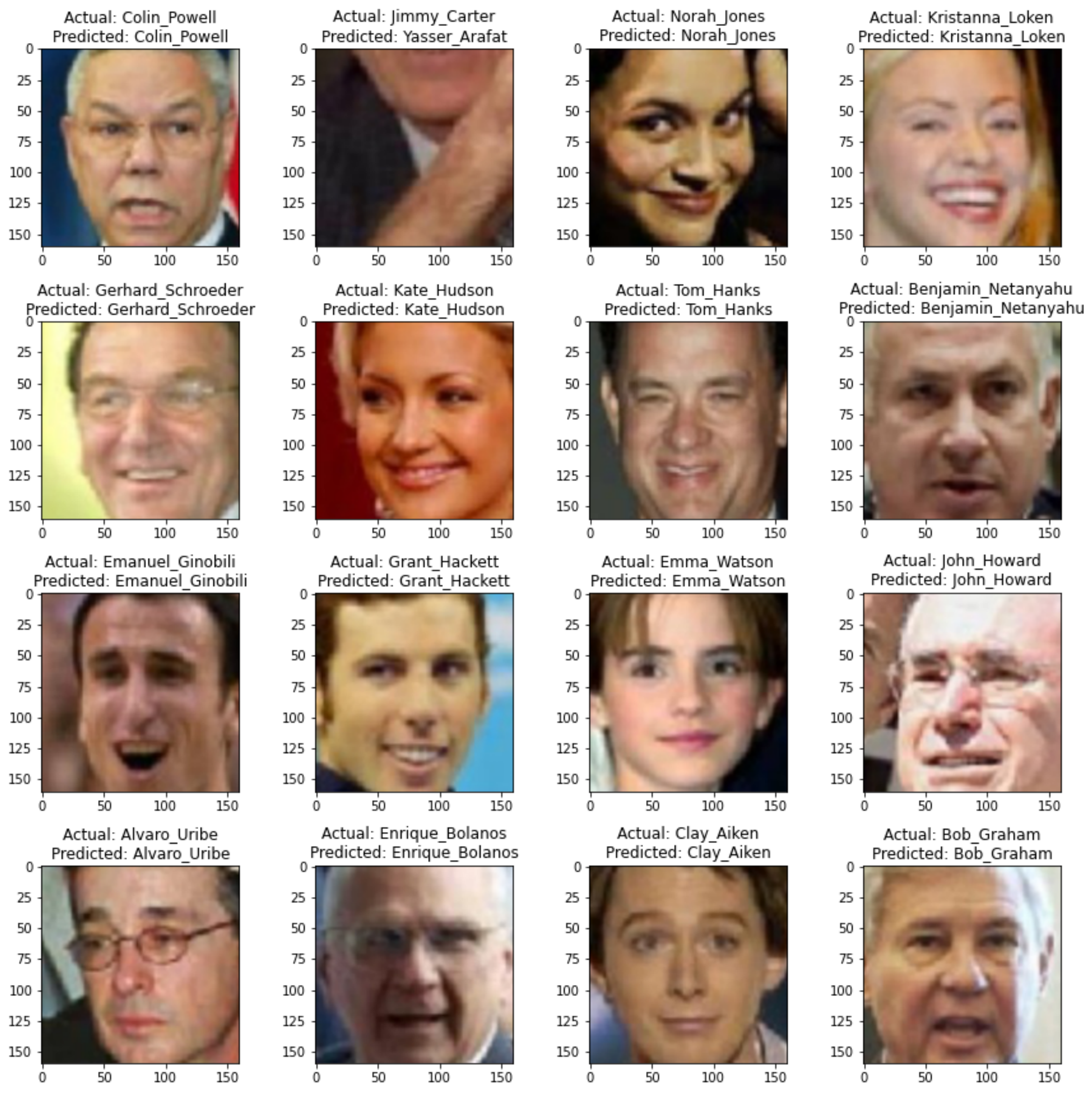

Fig. 1: Predictions on a subset of the unmasked dataset

The initial results for the masked set are less impressive, however, achieving only 55.6% accuracy.

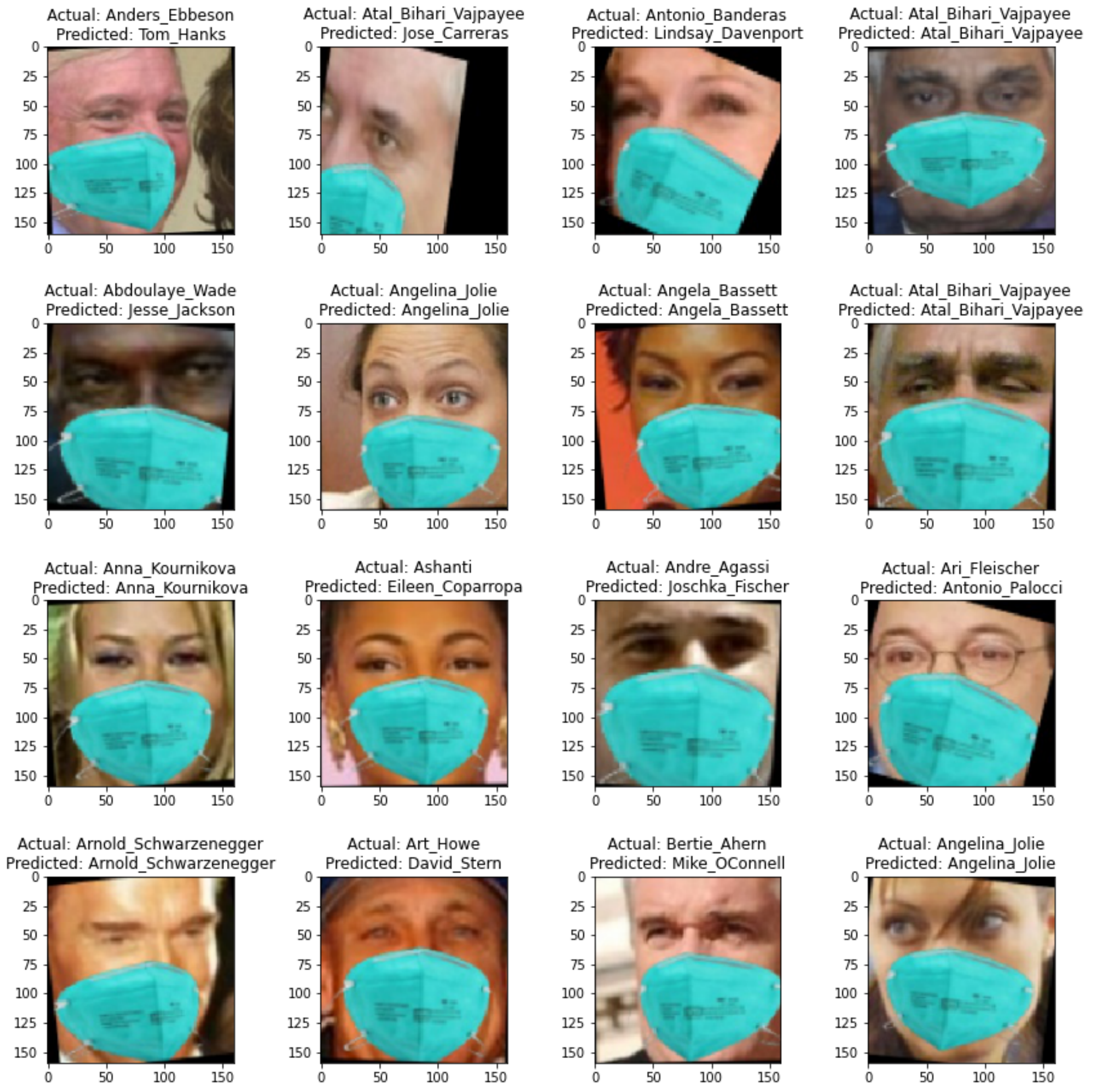
Actual: Anders_Ebbeson
Predicted: Tom_Hanks
Actual: Atal_Bihari_Vajpayee
Predicted: Jose_Carreras
Actual: Antonio_Banderas
Predicted: Lindsay_Davenport
Actual: Atal_Bihari_Vajpayee
Predicted: Atal_Bihari_Vajpayee
Actual: Abdoulaye_Wade
Predicted: Jesse_Jackson
Actual: Angelina_Jolie
Predicted: Angelina_Jolie
Actual: Angela_Bassett
Predicted: Angela_Bassett
Actual: Atal_Bihari_Vajpayee
Predicted: Atal_Bihari_Vajpayee
Actual: Anna_Kournikova
Predicted: Anna_Kournikova
Actual: Ashanti
Predicted: Eileen_Coparropa
Actual: Andre_Agassi
Predicted: Joschka_Fischer
Actual: Ari_Fleischer
Predicted: Antonio_Palocci
Actual: Arnold_Schwarzenegger
Predicted: Arnold_Schwarzenegger
Actual: Art_Howe
Predicted: David_Stern
Actual: Bertie_Ahern
Predicted: Mike_OConnell
Actual: Angelina_Jolie
Predicted: Angelina_Jolie

Fig. 2: Predictions on a subset of the masked dataset without retraining

To improve accuracy on the masked set, we then trained it on the mixed set, and then retested it. We were able to achieve an accuracy of 71.1% on the testing set. However, the accuracy on the training set decreased from an initial 99.326% to 94.577%.

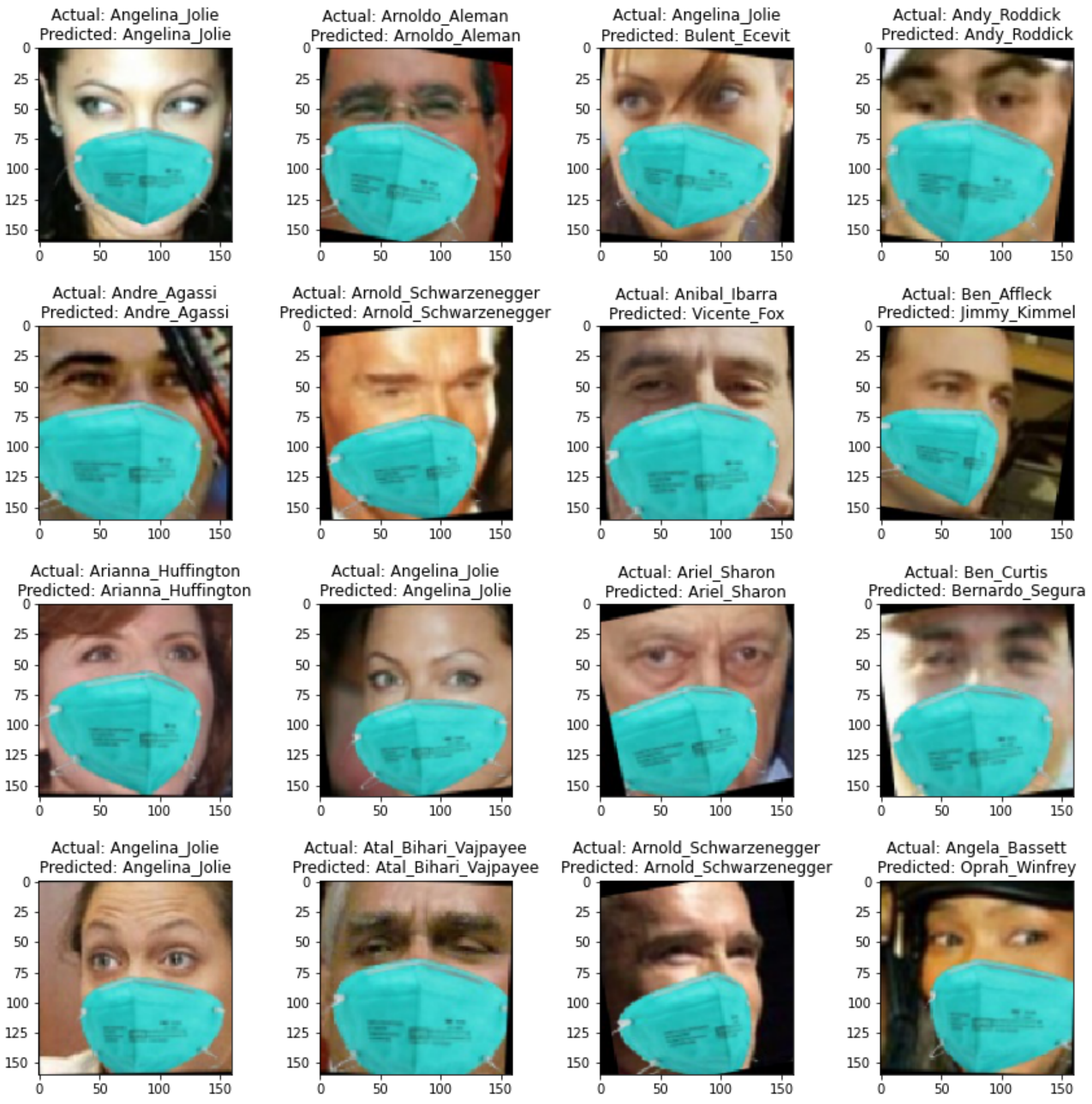

Fig. 3: Predictions on a subset of the masked dataset after retraining

Discussions and Conclusions:

When trained on the deep-funneled dataset, our model achieved a 99.326% training accuracy and 96.269% testing accuracy, which we considered a success given what we initially had in mind at the beginning of the semester. There are areas we can improve on, one being that our preprocessing method was not the best, shown in Fig. 1 where the second image was not cropped properly to contain the face of the person. We could also incorporate data augmentation to give the model more samples to learn form. On the mixed dataset, our model achieved 94.577% training accuracy and 71.1% testing accuracy. Even though it's not as reliable, the model is definitely usable and good enough for a project of our scale. We considered some of the limitations to be that 1) masks are covering too much of the face and its features, and 2) the masks have identical color. We could have improved the performance if we had a better masked dataset, but overall we considered the project to be a successful attempt at facial recognition.

References

Gary B. Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller. **Learning to Align from Scratch.** *Advances in Neural Information Processing Systems (NIPS)*, 2012.

Dalkiran, M. (2020, September 7). LFW simulated masked face dataset. Kaggle. Retrieved December 21, 2021, from https://www.kaggle.com/muhammeddalkran/lfw-simulated-masked-face-dataset

Nyoki-Mtl. (n.d.). *Facenet implementation by Keras2*. GitHub. Retrieved December 21, 2021, from https://github.com/nyoki-mtl/keras-facenet

Skúli, S. (2018, January 3). *Making your own face recognition system*. freeCodeCamp.org. Retrieved December 21, 2021, from https://www.freecodecamp.org/news/making-your-own-face-recognition-system-29a8e728107c/