Face Authentication from Masked Face Images Using Deep Learning on Periocular Biometrics

Jeffrey J. Hernandez V.[1], Rodney Dejournett[1], Udayasri Nannuri[1] , Tony Gwyn[1], Xiaohong Yuan[1], and Kaushik Roy[1]

[1] North Carolina Agricultural & Technical State University, Greensboro NC 27407, USA
jjhernandezvillarreal@aggies.ncat.edu, rldejournett@aggies.ncat.edu, unannuri@aggies.ncat.edu, tgwyn@aggies.ncat.edu, xhyuan@ncat.edu, kroy@ncat.edu

Abstract. Nowadays, identity theft is an alarming issue with the growth of e-commerce and online services. Moreover, due to the Covid-19 pandemic, society has been pushed towards the usage of masks for people to safely interact with one another. It is hard to recognize a person if the face is mostly covered, even more so to artificial intelligence who have more difficulty identifying a masked individual. To further protect personal information and to develop a secure information system, more comprehensive bio-metric approaches are required. The currently used facial recognition systems are using biometrics such as periocular regions, iris, face, skin tone and racial information etc. In this paper, we apply a deep learning-based authentication approach using periocular biometric information to enhance the performance of the facial recognition system. We used the Real-World Masked Face Dataset (RMFD) and other datasets to develop our system. We implemented some experiments using CNN model on the periocular region information of the images. Hence, we developed a system that can recognize a person from only using a small region of face, which in this case is the periocular information including both eyes and eyebrows region. There is only a focus on the periocular region with our model in the view of the fact that the periocular region of the face is the main reliable source of information we can get while a person is wearing a face mask.

Keywords: Biometrics, Periocular Recognition, Overfitting, Facial Landmarks, CNN Model, Augmentation, Authentication.

1 Introduction

Authentication has become a fundamental issue to any computing system. Moreover, it is also a crucial part in any security-based computing system. Authentication allows only legitimate users to access system resources. Hence implementation of the authentication system is difficult. There are many ways to authenticate a person and give them access into the system. However, memorizing passwords for multiple systems and managing several smart cards are inconvenient. There is always a chance that the means of self-authentication can be stolen or lost. This results in password-based and card-based authentication being less than a reliable way of securing a system or files. Furthermore, the subject of a biometrics-based authentication system is to be considered when trying to cover someone's identity or role. But, not a lot of focus is

brought upon this subject as a means of authenticating someone into the system. When it comes to these forms of authentication, biometric-based authentication is a more secure way of implementing into the system, but it does require a bit more set-up.

Biometrics mostly refer to a part of the human body being utilized for something, which can include the face, iris, fingerprint, and skin tone. One way of using these biometrics is to identify or verify a person. A biometric based authentication system consists of two phases: Feature extraction and verification. During the feature extraction phase, a set of biometric features are extracted from the image dataset that has been collected. From there, a collection of the features gathered from the biometric data is made and stored as a template. In the verification phase, the biometric feature data is applied in the algorithm to verify/ authenticate the label with the legitimate person. Biometric features can be different for the same person due to some factors such as variations in scale, pose, lighting and occlusions. So, more images of a subject/person are needed to prepare a sophisticated biometric feature dataset. A few images of a subject may not provide most biometric information. Due to this reason, a large image dataset is preferable in developing any biometric based authentication system.

With this all-in mind, this project aims to replicate facial recognition with the focus of only the periocular region of the face using a CNN model. The periocular region is the most reliable aspect of the face that can be viewed and identified especially in the case of the individual has their face covered up. After being detected, the periocular region will then be used to detect if the person is part of the database of authenticated people, giving a pass or fail to the tested image(s). The dataset will be split to fit the CNN model and made up entirely of masked individuals. The expected result should be the CNN model being able to present a high accuracy from the dataset being tested. As of now, the CNN model is still being tweaked to get the most accurate results possible.

2 Related Work

Biometric data is becoming increasingly used as a means of establishing a secure verification process. In [1], the author proposed a Deep Convolutional Neural Networks for the iris and face based Presentation Attack Mitigation by using machine learning techniques and implementing a feature extraction tool like the discrete wavelet transform (DWT). From there, the author developed a multiple CNN channel model like modified AlexNet, modified-SpoofNet, and modified-VGGNet, tested on the video dataset and get an accuracy for each channel: channel 1 being 99.90% accurate, Channel 2 being 99.83% accurate, and Channel 3 being 99.68%. In [2], Wang et al. described what type of dataset they had collected to test for the peculiar region with masked images, here they used a Masked Face Detection Dataset (MFDD), Real World Masked Face Recognition (RWMFR), and Simulated Masked Face Recognition Dataset (SMFRD). From there, they trained the dataset with a face eye-based multi granularity Recognition model. By testing these datasets, they got an accuracy of 95%. So, in our project we utilized a masked dataset.

Authors in [3] discussed whenever the periocular region is brought up, the report describes mainly facial features like eyelids, eye shape, eyebrows, eyelashes, the top of the nose, and skin texture. The paper provides a detailed survey of periocular

biometrics and a deep knowledge of various aspects, like ROI Extraction, and functionality of periocular region stand-alone methods, like LBP, LPQ, PIGP, and a combination with an eye. It applied itself to many applications, such as smartphone authentication, to discover the role of the periocular region in the soft biometric categorization of the facial region [3]. Thus, the importance of the periocular region of the face whenever it comes to our experiment.

Chandana et al [4] describes about every time they cannot capture an image or video within high resolution. So, to identify these high-definition images, they require these features, like irises, eyebrows, eyelids, and skin texture. These features can get the identification of a person within an image or video surveillance. By using an FGNET dataset and applying machine learning algorithms like logistic regression and naive bayes algorithms, the project was able to score a high accuracy of 96% by using only the periocular region.

With all these reports, we can take in their concepts of facial recognition and the periocular region and apply them into our working project. The result of this leaves us with our own CNN model, a baseline model to compare to, and a large periocular region image dataset made up of masked faces.

3 Methodology

In this research, we begin with gathering up and processing the dataset that is planned to be used the CNN model's testing and training sets. After data collection, the project will then focus on the application of the dataset to the CNN model and achieve the highest accuracy as possible when testing for a masked individual on whether they are an authenticated user or not.

We preprocessed the dataset to become a large dataset of clear images made of masked individuals' faces. Initially, we used the RFMD Dataset [5] [6] that was already separated into individual subjects with multiple images in each subject folder. The images in the dataset were then all adjusted to be the size of 400 pixels by 400 pixels to ensure that all the images were of the same size and prepared to grab the periocular region from each of the faces in the images. To accomplish this, an algorithm was developed, utilizing a combination of facial landmarks, OpenCV, and a library called lib. Once finished, this program would scan the image of the masked subject and grab a particular region of their face.

Using a CNN face detection model and shape predictor from within dlib, the program would detect facial landmarks within the masked images. We wanted to make sure that the new images obtained would contain both eyes and eyebrows, so we did not need to detect all the faces. But to make sure that the landmarks were accurate, we still went ahead and detected (or predicted as much as the program can) all 68 facial landmarks in each face. After that, we modified it in such a way that instead of completely displaying all 68 facial landmarks, the program would focus on just the leftmost part of the left eyebrow through the bottom part of the right eyeball (facial landmarks 18 through 27, landmarks 28-30 were added later to be able to capture the periocular region much better) [7]. Once all the necessary facial landmarks have been detected on the actual face within the image, the program would go through and confirm what parts of the face it managed to grab that relate to the periocular region. This would

allow the program to crop that part of the face in the images and save it as a separate image. Figure 1 showcases these facial landmarks and then the extraction process.

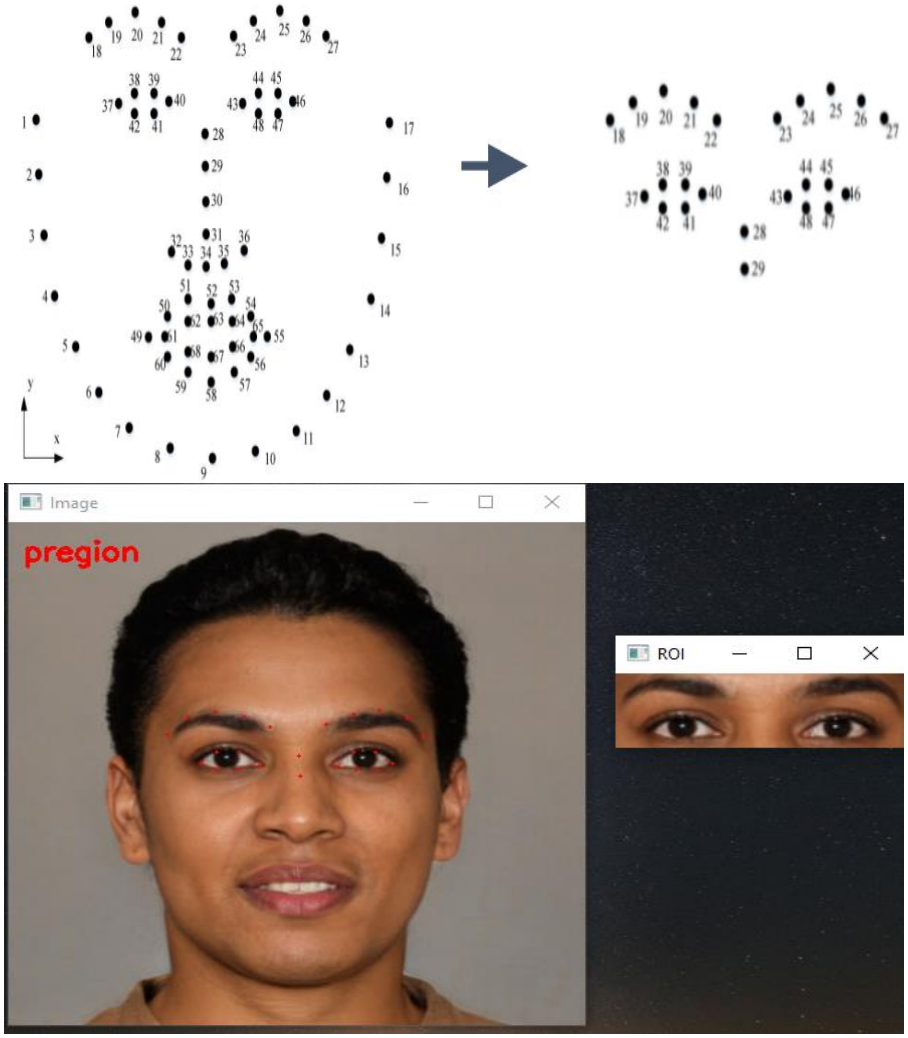


Fig. 1. Facial Landmarks being broken down to only needing landmarks located in the periocular region and the program extracts the region into a new image. Image is taken from [5][6].

Unfortunately, within the limits of the RMFD dataset, many of the subjects' faces were not detected and their periocular region could not be extracted. This would lead to adding more images of masked faces to the original masked dataset, resulting in new periocular masked faces being placed and more subjects to augment. With this in mind, we proceeded with a Real and Fake Face Detection dataset to add into our periocular dataset [5][6]. Despite the faces not being masked, the extracted periocular region images will be added into the training set to allow the CNN model to better distinguish and understand what parts of the face it should focus on.

With the facial landmarks extracted, we were able to create a new dataset with periocular images, made from the masked dataset we have. We then started to prepare our data to be used for deep learning to identify if the image matches an authenticated face within the database, all using a CNN binary image classifier. To further establish the dataset, the next focus would have to be image augmentation. Image augmentation is necessary within this project since it will provide training and exposure to our CNN model through allowing multiple different variations of the subject images into the training and testing sets. Unfortunately, a question came up of what style of the original dataset should we go for. For each subject in the masked dataset, there were multiple images of them. The question of style was if the placement of the subjects or where they are distributed really matter enough for the CNN model to operate successfully. From there, we only focused on one format of the dataset, but that same question would be brought into place later.

Then we augmented the images through another program we implemented that replicates the images into different versions and aspects, as shown in Figure 2. The large, combined dataset soon grew to have 2,005 subjects, with each subject getting 10 augmented images of them, resulting in a total of 20,045 images. We aimed to overcome the overfitting issue of our dataset to ensure that the CNN model would be validating it when calculating its accuracy. Once the augmentation has been completed, the dataset needed to be split into a training set and testing set. These sets would then be utilized by our CNN model. To achieve a high accuracy with our model, we decided to get an 80% to 20% relationship with the 80% of images going into the training set and 20% going into the testing set. Not only that, the 80% that the training set will get must also have an image from each subject to maintain consistency. Once the images have been split, the CNN model that we developed requires both sets to have authentic and unauthentic sets, so it can try to distinguish. We originally had it named as real and fake sets but, after discussion over the concept, we decided to rename them to authentic and unauthentic sets to better fit what their purpose within the model, For this case, we split each set into 90% of the images going into an unauthentic set and the rest (10%) would go into an authentic set within the folder. With the dataset being as prepared and fitted as possible, we bring out focus to our CNN model and its results.

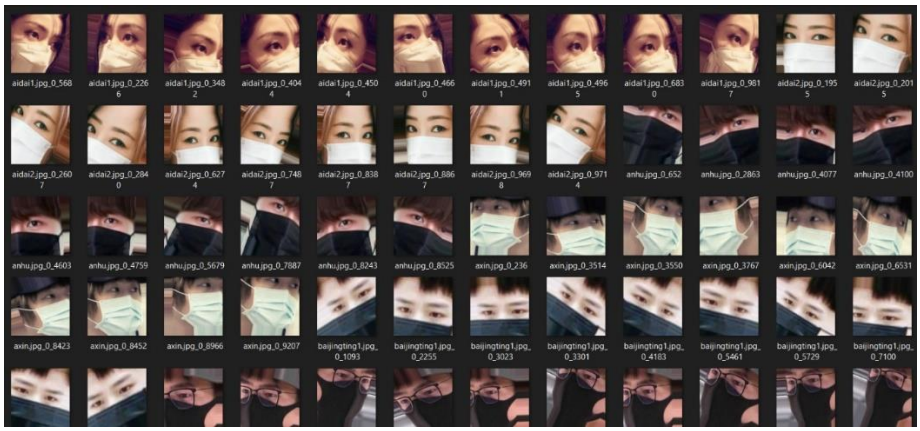


Fig. 2. A preview of how the augmented dataset looked before the images were extracted of their periocular region [5][6].

4 Results

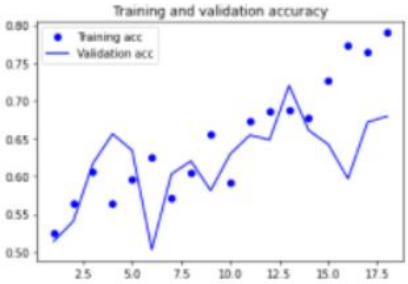
In this section, we provide our experimental results. The CNN model that was developed and used came to be from integrating with Keras Image Classification and following a base model of a classification matrix and its many layers [9]. The model itself is a binary image classifier, meaning that it will identify the images placed in the dataset as one class or the other. In this experiment, the model will take in the training set and testing set as its own classes and focus on it. It operates with layers and an optimizer that helps with the minimizing of classifying the images with the data. As a result, the CNN model produces a report showing the accuracy, precision, recall and two graphs that show the overall accuracy and loss as the model works with the dataset, as shown in Figure 3. Using the RFMD dataset we used images to fit for binary image classification for authenticated subjects vs unauthenticated subjects. We experimented with this data on our benchmark model to see if we could better results with subjects using their full face. We used the formula, steps_per_epoch * batch_size = total # of images to determine where the new number of epochs or batch sizes should be. On our first attempt to train the model on the data we ended up with 33% loss and 85% accuracy in training and 78% loss and 62% in validation. Total time taken was about 47 minutes for 15 epochs. Our confusion matrix resulted in 366 true positives, 70 false positives, 370 false negatives, and 68 true negatives. For the second run we lowered the batch size for the testing dataset and increased number of epochs to 18. The training for the model ran for 54 minutes and resulted in 39% loss and 81% accuracy for training and 58% loss and 70% accuracy for validation. The predictions from the confusion matrix were 333 true positives, 103 false positives, 314 false negatives, and 125 true negatives.

```
ROC Curve- .52
Confusion Matrix
[[333 103]
 [313 125]]

Classification Report
              precision    recall  f1-score   support

   authentic       0.52      0.76      0.62       436
 unauthentic       0.55      0.29      0.38       438

    accuracy                           0.52       874
   macro avg       0.53      0.52      0.50       874
weighted avg       0.53      0.52      0.50       874
```



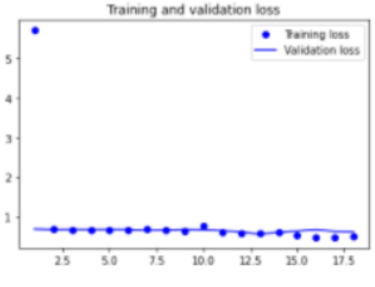


Fig. 3. The classification report that our CNN model provided with the datasets alongside two graphs that show the accuracy and loss of the data

With the results, we see that the accuracy when it comes to the CNN binary image classifier model ranges around 53%.

5 Conclusions

The project continues to develop as the focus of trying to achieve a high accuracy when it comes to the authentication of masked faces is still relevant. Theories have been made and new approaches are currently being taken to get the results we desire. For now, a baseline facial recognition model using binary image classifier has been taken into consideration to compare the results it gives to our own model [10]. Even though it utilized TensorFlow instead of what we used, the model still could be used as a baseline since it gives up accurate results with so little data needed. This model allows us to configure the parameters. This will allow us to break down what will work as a means for better results: an update to the periocular region dataset or reworking the CNN model. At the current iteration of the dataset, we switched into full, uncovered faces instead of the periocular region of masked faces. An idea was brought up to see if the CNN model's facial recognition would operate better with a full-face dataset rather than a dataset with a part of someone's face. Future tests will now experience the full face rather than the periocular region to see if better results appear. This will allow us to improve on the CNN model and promise better results in the future.

Acknowledgement

This research is supported by National Science Foundation (NSF). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

References

1. Chatterjee, P. (2020). (rep.). *Deep Convolutional Neural Networks for the iris and face based Presentation Attack Mitigation* (pp. 1–101). Greensboro, NC: The Graduate School of NCAT.
2. Wang, Z., Wang, G., Huang, B., et. al. (2020). (rep.). Masked Face Recognition Dataset and Application (pp. 1–3). From https://arxiv.org/abs/2003.09093
3. Kumari , P., & K.R., S. (2019). (rep.). *Periocular biometrics: A survey* (pp. 1–12). Delhi, India: Journal of King Saud University.
4. Chandana, C. S., Rao, K. D., & Sahoo, P. K. (2020). (rep.). *Face Recognition through Machine Learning of Periocular Region* (3rd ed., Vol. 9, pp. 1–5). Hyderabad, India: International Journal of Engineering Research & Technology (IJERT).
5. Huang, B. (2020, February 13). Real-World Masked Face Dataset (RMFD）. GitHub. From https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset#real-world-masked-face-datasetrmfd.

6. B. Huang et al., "Masked Face Recognition Datasets and Validation," 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 1487-1491, doi: 10.1109/ICCVW54120.2021.00172, From https://ieeexplore.ieee.org/document/9607619
7. Rosebrock, A. (2017, April 3). Facial Landmarks with dlib, opencv, and python. PyImageSearch. From https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/.
8. Computational Intelligence and Photography Lab, Yonsei University. (2019, January 14). Real and Fake Face Detection. Kaggle. From https://www.kaggle.com/ciplab/real-and-fake-face-detection.
9. Brownlee, J. (2019, June 3). *How to Perform Face Detection with Deep Learning*. Machine Learning Mastery. From https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/.
10. Phan, B. (2020, May 30). *10 Minutes to Building a Fully-Connected Binary Image Classifier in TensorFlow*. Towards Data Science. Retrieved December 13, 2021, from https://towardsdatascience.com/10-minutes-to-building-a-fully-connected-binary-image-classifier-in-tensorflow-d88062e1247f.