

ROBUSTNESS OF FACIAL RECOGNITION TO NOISE

Tan Ern Min¹, Daniel Lee Jek Han², Andrew Tan²

¹Dunman High School, 10 Tg Rhu Rd, Singapore 436895

²Defence Science and Technology Agency, 1 Depot Road, Singapore 109679

ABSTRACT

In light of the increasing use of facial recognition in Singapore, this research aims to compare the robustness of facial recognition to Gaussian noise. Noise in images is the greatest common reason why models inaccurately identify subjects, produced due to low lighting conditions which causes the camera sensor to not capture the information properly during the shot. Hence to simulate the local environment, an asian dataset was put together and ImageMagick software was imported to add Gaussian noise filters of different noise levels: 0%, 25%, 50%, 75% and 100%. Open-source facial recognition models were used through deepFace. The final shortlisted models that were compared were ArcFace, VGG-Face, Facenet, SFace and Dlib. The models' consistency and accuracy with noise were calculated using standard deviation and mean of the scores respectively. VGG-Face was the model that was most robust to noise, with the standard deviation and mean accuracy score of 2.9% and 81.6% respectively.

INTRODUCTION

In Singapore, facial recognition has made great progress for various applications. From the SingHealth facial recognition system for hospital visitors [1], facial recognition check-in systems in Changi airport [2] to contact tracing during COVID-19 [3]. Facial recognition technology is increasingly embedded in our lives as not only does it reduce manpower needed, it is faster and sometimes, more accurate than human recognising faces [4].

However, facial recognition is currently challenged with uncontrolled conditions, such as varying illumination, poses, facial expressions, and noise. This ultimately affects its capability to accurately recognise faces. Extensive research has been carried out towards the illumination, pose, and expression problems [5][6][7]. But when it comes to the noisy images, the recognition accuracy of most approaches would drop significantly. Noise is a random variation of image density, visible as grain in film and pixel level variations in digital images. It is a key image quality factor; nearly as important as sharpness. They are produced due to low lighting conditions which causes the camera sensor to not capture the information properly during the shot and the camera processor has to make its own interpretation of the image [8]. 4 common types of noise are, Gaussian, Salt and Pepper, Poison and Speckle.

Currently there has been research conducted to compare the robustness of noise of various open-source facial recognition softwares [9], however they are mainly tested on datasets that majority is made up of white faces. In a National Institute of Standards and Technology report, researchers studied 189 facial recognition algorithms and they found that most facial recognition algorithms exhibit bias. According to the researchers, facial recognition technologies falsely identified Black and Asian faces 10 to 100 times more often than they did white faces [10]. This is worrying as it can lead to false identification and widen pre-existing inequalities between races [11].

Hence, the aim of this experiment is to test the robustness of facial recognition to noise among several state-of-the-art models. Robustness to noise refers to how capable a model is to withstand noisy images and how unaffected its results are when faced with noisy images. In addition, asian datasets were used, to identify which model would have the highest accuracy when used in the local context. To ensure the reliability of the test, each algorithm is trained and tested with the identical data sets. The test results are then compared and evaluated, allowing us to pinpoint the best ones.

MATERIALS AND METHODS

1) Background research

Research was conducted on open-source models to select the best model to carry out the experiment. Since there were many available open-source models, two criterias were used to shortlist the models:

1. Accuracy
2. Convenience

Firstly, the accuracy of the model was determined by its performance using the LFW(Labelled Faces in the Wild), a public benchmark for facial recognition models. Secondly, convenience of the model was evaluated by how accessible the code is and whether further processing on the training and testing datasets were required. For example, resizing and conversion of colour image to grayscale image.

2) Making a dataset



Fig. 1: Test images

A dataset of 120 images of 20 subjects was put together with the help of the google extension tool to download the images with the subject's name correctly labelled. Additionally, images that had at least 80% of the space occupied by the subject's face were selected. This was to ensure that unnecessary surrounding material is reduced from the image, and improve the accuracy of the facial recognition model. To simulate local context, asian subjects varying in ethnicity were used in this dataset. Some such examples are Sundar Pichai (Indian), Michelle Yeoh (Chinese) and Shahrizal Salleh (Malay). This is to ensure that the model would be able to accurately reflect its facial recognition capabilities when used on the local people. After the images were downloaded, 20 images were set aside for testing and 100 were used for training the model. Images were then zipped into "test asian dataset" and "asian dataset" files respectively and were ready to be used.

The dataset was put together from scratch to control the quality and sizes of the images, so as to reduce the amount of changed variables. Furthermore, with little light shed on

experimenting with asian dataset on facial recognition models, scarce open-source asian dataset was available.

3) Developing noisy images



Fig. 2 and 3: Akshata Murthy before (left) and after (right) 25% Gaussian noise filter

```
# Import Image from wand.image module
from wand.image import Image
# Read image using Image() function
for path in directory_path:
    with Image(filename = path) as img:

        # Generate noise image using spread() function
        img.noise("gaussian", attenuate = 1)
        img.save(filename = path)
```

Fig. 4: Code for noise addition

Gaussian filter was applied on the test dataset using noise() function from Wand.image by Imagemagick. Wand.image was selected as not only does it require fewer lines of code, fewer softwares needed to be imported. Lastly, it was easy to adjust the amount of noise in the image. The amount of noise was adjusted by passing an attenuate where the value can be between 0.0 and 1.0, this enables the amount of noise to be controlled from 0-100%. A loop was created to add the filter to all images in the test dataset. Five sets of Gaussian noise test data sets were created. They were 0%, 25%, 50%, 75% and 100% noise respectively.

4) Building on to the selected model

index	identity
0	/content/asian dataset/content/deepface/tests/asian dataset/jay chou03.jpg
1	/content/asian dataset/content/deepface/tests/asian dataset/jay chou10.jpg
2	/content/asian dataset/content/deepface/tests/asian dataset/jay chou02.jpg
3	/content/asian dataset/content/deepface/tests/asian dataset/jay chou09.jpg
4	/content/asian dataset/content/deepface/tests/asian dataset/jay chou11.jpg
5	/content/asian dataset/content/deepface/tests/asian dataset/jay chou08.jpg

Fig. 5: Model Facenet giving a list of matching images when Jay Chou's image was inputted

The models identify the subject in the test image and give a rank of matching images from the training images. (Fig. 5) The smaller the index, the higher the ranking. This means that the model has greater confidence that the subject in the image is the subject in the test image. For example, Model Facenet has determined that facial alignments of subject in "jaychou03.jpg" are most similar to the facial alignments of subject in the input test image.

```

dflist = []
for img in directory_list:
    dflist.append(DeepFace.find(img_path = img, db_path = "/content/asian dataset/

```

Fig. 6: Looping DeepFace

Since the original model was constructed to process one test image at a time, it was necessary to construct a loop so that the model outputs a list of matching images for all test images individually. (Fig. 6)

```

score = [0 for i in range(len(directory_list))]
for i,img in enumerate(directory_list):
    top5 = dflist[i].head(5)
    for row in top5.iterrows():
        top_img = row[1]['identity']
        img= re.sub('\d', '', img)
        top_img= re.sub('\d', '', top_img)
        if img in top_img:
            score[i]+=1

```

Fig. 7: Setting up point system

Afterwards, a point system was set up. (Fig. 7) First the top five rows of the results were selected and digits present in their file names were removed using Regex. The same was done for the input test image. Only the first five rows of the results were selected as the number of rows of results were inconsistent among the different models and five was the minimum number of rows for each set of results. This was done to ensure that the number of results were kept constant for all test images across all models.

Subsequently, a point is added when the string of file names of the test image is present in one of the file names of the top five results.

```

[4, 5, 5, 4, 5, 5, 5, 4, 1, 5, 5, 5, 5, 5, 1, 0, 5, 5, 1, 5]
80

```

Fig. 8: Counting and addition of scores

Finally scores were printed out. Fig. 8 shows the individual scores for each test image and the total score for the model. Scores were then recorded. In order to compare robustness and accuracy, standard deviation and mean scores were calculated for each model:

$$\text{Standard deviation score} = \sqrt{\frac{\sum (\text{each score} - \text{mean score})^2}{\text{no.of scores}}}$$

$$\text{Mean score} = \frac{\sum \text{scores from 0-100\% noise}}{\text{no.of scores}}$$

RESULTS AND DISCUSSION

1) Shortlisted models

Table 1: Models and their LFW score

Model name:	LFW score (%)
Facenet512	99.65
SFace	99.60
ArcFace	99.41
Dlib	99.38
Facenet	99.20
VGG-Face	98.78
OpenFace	93.80
DeepID	99.15

The primary shortlisted models were Facenet512, SFace, ArcFace, Dlib, Facenet, VGG-Face, OpenFace and DeepID. They were selected as they showcased the highest few LFW scores and they could be easily accessed through DeepFace, a hybrid face recognition framework that wraps multiple state-of-the-art models and could access each model with a line of code[12]. Furthermore, no additional processing on the images were required.

Table 2: Models and their asian dataset score (0% noise)

Model name:	Asian dataset score (%)
Facenet512	63.00
SFace	77.00
ArcFace	86.00
Dlib	79.00
Facenet	74.00
VGG-Face	86.00
OpenFace	5.00
DeepID	9.00

A second round of shortlisting was carried out to choose the top five performing models using the 0% noise asian dataset. They were ArcFace, Facenet, VGG-Face, SFace and Dlib. It is shown that models with high performance with the LFW dataset are not necessarily as robust when the dataset has been substituted with the asian dataset.

2) Models with noisy data

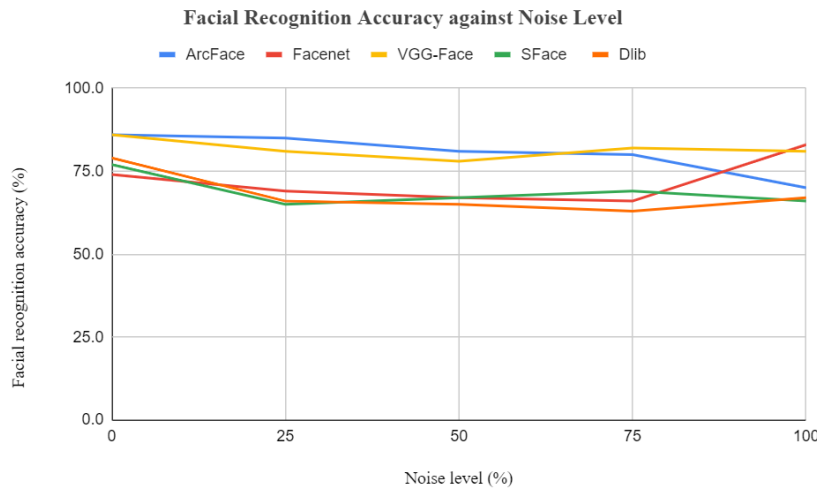


Fig. 9: Facial recognition accuracy against noise level

It can be seen from Fig. 9 that there was no observable drastic decrease when more noisy data was introduced into the model. Models were relatively robust against noise. However it could be observed that the trend of facial recognition accuracy decreasing as the noise level increased was not applicable for all models. This is seen for Facenet and Dlib as a sharp increase in accuracy could be seen when the noise level increased from 75% to 100%. The increase was from 66.0% to 83.0% and 63.0% to 67.0% respectively. Visually, it can also be observed that facial recognition accuracy of VGG-Face remains the least affected as the noise level increases.

It can also be observed that despite images of different sizes being used (Fig. 1) most of the models were still able to recognise the subjects in the image accurately.

Overall, it is intriguing that the increasing noise levels did not cause a significant decrease in the accuracy of the models. This may be due to insufficient data to test and train the model such that a significant difference in accuracy can be observed. Another reason may be due to insufficient noise being added to the image. The module ImageMagick that was used to add noise to the images has noise attenuation fixed in a constant range from attenuate 0.0-1.0. The noise that was added at attenuate 1.0 was maybe too little to cause a significant accuracy decrease in facial recognition. Hence, the model was not largely impacted by the noise and were able to calculate similar euclidean distances between facial features for test images at different noise levels.



Fig. 9, 10, 11, 12, 13: Akshata Murthy with increasing noise levels, 0%, 25%, 50%, 75% and 100% respectively

A reason why the models appear to have close accuracy results when the noise level was increased was because the images appeared similar even after doing so. Since the noise level range was controlled by the module ImageMagick, the range of noise from 0% to 100% may have been too small. Hence similar results were obtained due to similar images.

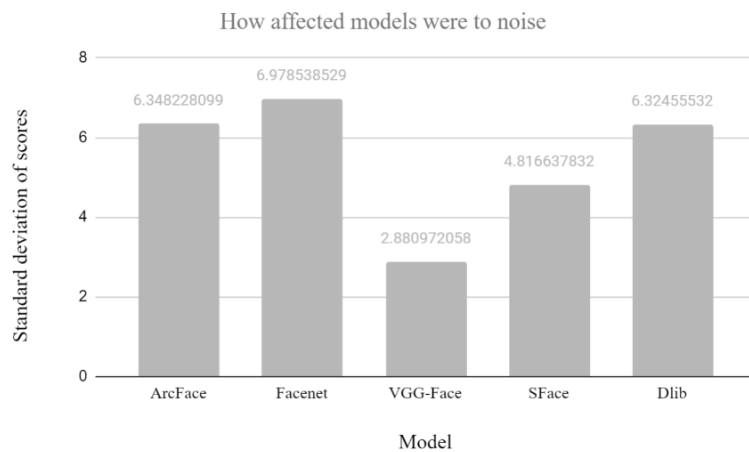


Fig. 10: Standard deviation of scores against the models

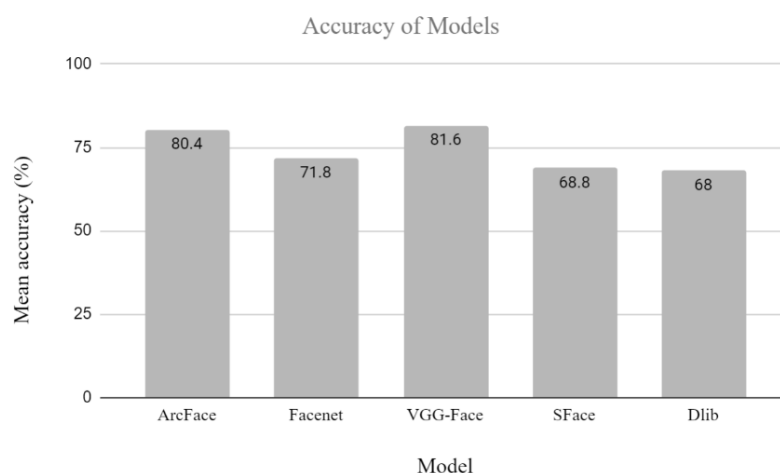


Fig. 11: Mean accuracy against the models

Moving on, standard deviation was used to calculate how consistent the results were, enabling us to compare the robustness to noise more precisely.

From the graph above (Fig. 10) it can be seen that VGG-Face is the least affected by noise when carrying out facial recognition as it has the smallest standard deviation of 2.9%, implying that it has the most consistent results when the noise of the data was increased from 0-100%. This is followed by SFace, Dlib, ArcFace and Facenet.

Afterwards, in order to ensure that the model is robust against noise without having its overall facial recognition accuracy being compromised, mean accuracy was calculated and compared among the models.

From Fig. 11 it can also be seen that VGG-Face has the greatest mean accuracy among the other models. This shows that it is the model most suitable to use when Asian subjects take up the majority of the dataset. Furthermore, it can be observed that VGG-Face is the least biased when subjects of different races are used. As the model was the closest to its LFW score where 83.5% of the subjects used were white [13].

```
ValueError: Face could not be detected. Please confirm that the picture is a face photo or consider to set enforce_detection param to False.
```

Fig. 12: Model unable to detect face

There were however some limitations in this experiment. When running the model, the error (Fig. 12) occurred, in which the could not detect faces in the images. Hence it was necessary to add the code:

```
enforce_detection=False
```

According to the original author of deepFace, this Multi-Task Cascaded Convolutional Neural Networks (MTCNN) does not detect any face in the picture and the library will consider the whole input image as a face and compute its embedding. This eventually leads to a decrease in accuracy when the model is identifying the subject in the image. Hence a true reflection of the models' performance cannot be achieved and the final results may be limited in their accuracy.

Even so, since the code was applied to all models, it can still be considered a fair test as the addition was consistent among all models and all of them were affected.

CONCLUSION

In conclusion, VGG-Face was the best performing model against noisy asian data images with the standard deviation of scores being the lowest of 2.9% and highest accuracy score of 81.6%. As VGG-Face is an open-source software, it can be easily duplicated and developed in agencies. This can provide more transparency and privacy as compared to private company's facial recognition models, where they would have access to confidential information and one's privacy may be compromised.

The experiment could be further improved by adding more noise to the images. It can be done so by exploring other modules to add noise or adding noise to the image multiple times using the current module, ImageMagick. Future research can also be done to test the models' robustness to different forms of noises, not limiting to only Gaussian noise. This is to investigate how unaffected the model is to other forms of noise. More can be done as well to increase the amount of training and test data used in the experiment to observe more precise results.

ACKNOWLEDGEMENTS

I would like to sincerely thank Defence Science and Technology Agency (DSTA) for giving me such an opportunity to embark on this project. I would also like to thank my mentor, Daniel Lee Jek Han and co-mentor Andrew Tan who both gave invaluable guidance and support throughout the entire course of the project.

REFERENCES

- [1] *SingHealth testing facial recognition system for hospital visitors*. (2022, November 1). The Straits Times
- [2] Bhunia, P. (2017, November 1). *Applications of facial recognition technology being explored by Singapore Government - OpenGov Asia*. OpenGov Asia
- [3] CHAN , G.K.Y. (2020, September) *Reflections on the use of facial recognition technology during COVID-19*, *Ink.library.smu.edu.sg*. Singapore Management University.
- [4] ALICE J. O'TOOLE, XIAOBO AN, JOSEPH DUNLOP, VAIDEHI NATU (no date) *Comparing face recognition algorithms to humans on challenging tasks*. National Institute of Standards and Technology.
- [5] H Roy, D Bhattacharjee, *Local-gravity-face (LG-face) for illumination-invariant and heterogeneous face recognition*. *Info. Forensics Secur. IEEE Trans.* 11(7), 1–1 (2016)
- [6] X Wang, Q Ruan, Jin, et al., *Three-dimensional face recognition under expression variation*. *EURASIP J. Image Video Process.* 54(1): 1–11 (2014)
- [7] MH Siddiqi et al., *Human facial expression recognition using curvelet feature extraction and normalized mutual information feature selection*. *Multimedia Tools Appl.* 75(2), 935–959 (2016)
- [8] *Noise in photographic images* (no date) *Imatest*. iamtest.
- [9] Wei Hu¹ Yangyu Huang^{2*} Fan Zhang¹ Ruirui Li¹ (no date) *Noise-tolerant paradigm for training face recognition cnns*. Beijing University of Chemical Technology, China ²Yunshitu Corporation, China.
- [10] Wei Hu¹ Yangyu Huang^{2*} Fan Zhang¹ Ruirui Li¹ (2019, December) *Noise-tolerant paradigm for training face recognition cnns*. Beijing University of Chemical Technology, China ²Yunshitu Corporation, China.
- [11] SITNFlash (2020) *Racial discrimination in face recognition technology*, *Science in the News*. Science in the News.
- [12] Serengil, S.I.S. (no date) *Serengil/deepface: A lightweight face recognition and facial attribute analysis (age, gender, emotion and race) library for python*, *GitHub*. GitHub.
- [13] Beth Findley. (2020, November 4). *Why racial bias is prevalent in facial recognition technology*. Harvard Journal of Law & Technology.