Journal of Applied Artificial Intelligence
https://jaai.sabapub.com
ISSN: 2709-5908

2024 Volume 5, Issue 2: 34-49
DOI: https://doi.org/10.48185/jaai.v5i2.1237

Comparative Analysis of FaceNet, VGGFace, and GhostFaceNets Face Recognition Algorithms For Potential Criminal Suspect Identification

Muhammad Indra Ardiawan[1, *], Gede Putra Kusuma Negarara[2]

[1, 2] Computer Science Department, BINUS Online Learning, Bina Nusantara University, Jakarta, Indonesia 11480

Received: 08.07.2024 • Accepted: 03.09.2024 • Published: 10.09.2024 • Final Version: 27.09.2024

Abstract: The escalating concerns surrounding criminal activities underscore the imperative for bolstered security measures to safeguard public welfare. Despite concerted efforts, the identification of suspects remains fraught with limitations, hindering the attainment of comprehensive individual profiles. Leveraging advancements in facial detection and identification technologies, this study assesses the efficacy of three prominent deep learning models—FaceNet, VGGFace, and GhostFaceNets—in the domain of facial recognition for suspect identification. Drawing upon data collected in 2023, the investigation scrutinizes FaceNet's intricate methodologies, including triplet loss optimization and Euclidean space mappings, yielding exceptional accuracy rates of 97.05% during validation and 97.4% during testing. Conversely, VGGFace, while displaying commendable accuracies, registers marginally lower accuracy metrics, standing at 97.05% and 96.1% during validation and testing, respectively. GhostFaceNets, integrating novel architectural components, exhibit diminished accuracy rates, signaling avenues for refinement. These empirical insights underscore FaceNet's prowess in furnishing robust and reliable facial recognition outcomes, while delineating the imperative for iterative enhancements in GhostFaceNets to foster their pragmatic applicability in security domains.

Keywords: Artificial Intelegence (AI), Machine Learning, Deep Learning, FaceNets, VarGFaceNet, GhostFaceNets, Facial Recognition.

1. Introduction

One of the issues in the world in general and in the surrounding environment in particular is regarding criminal activity. To reduce the crime rate and enhance public safety, it is necessary to synergize efforts between the community and security personnel [1]. However, when determining suspects, security personnel have limitations in identifying individuals. The identification of suspects also cannot yet display the profile of each individual.

Facial detection and identification technology can be used to identify potential threats from individuals, thereby enhancing security for the community against criminal threats posed by certain individuals who may have a criminal history, allowing security personnel to respond quickly to such threats. Face recognition enables the recognition and verification of a person's identity based on unique facial characteristics [2]. This can be achieved by matching the individual's face with a database[3] containing information on the criminal history of that individual.

* Corresponding Author: kulomasindra2@gmail.com

This research aims to develop a face recognition method for identifying criminal suspects and evaluating the accuracies of Deep Learning models: FaceNet, VGG-Net, and VarGFacenet. The limitation of this study is the use of FaceNet, VGG-Net, and VarGFacenet methods to generate facial vector representations and measure similarities between detected faces and data registered in the database. The data used in this research was collected in 2023, involving personal documentation and cooperation with relevant agencies.

The structure of this article is as follows. Section 2 discusses the definition and workings of each model. Section 3 contains the evaluation method. Section 4 discusses the implementation, section 5 contains testing and validation, section 6 contains the conclusion of this project, and the last section contains all the sources used.

2. Literature Review

2.1. FaceNet

FaceNet is a system that learns mappings from facial images to a solid Euclidean space that is well suited for directly measuring facial similarity[4]. FaceNet produces 128 feature vectors that can be used for facial recognition, facial verification, and facial clustering [5].

The architecture of FaceNet can be described in Figure 1, where facial images are fed into a deep architecture, which is then normalized with L2 and produces embeddings followed by triplet loss during the training process[6].

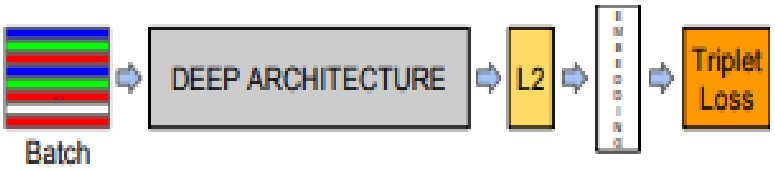

Figure 1. FaceNet Model

The Triplet Loss minimizes the distance when two images have the same identity and maximizes the distance when two images have different identities[7]. The Triplet Loss diagram is depicted in Figure 2.

Figure 2. Triplet Loss

In Figure 3, the architecture of FaceNet is displayed[8].

type	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj (p)	params	FLOPS
conv1 (7×7×3, 2)	112×112×64	1							9K	119M
max pool + norm	56×56×64	0						m 3×3, 2		
inception (2)	56×56×192	2		64	192				115K	360M
norm + max pool	28×28×192	0						m 3×3, 2		
inception (3a)	28×28×256	2	64	96	128	16	32	m, 32p	164K	128M
inception (3b)	28×28×320	2	64	96	128	32	64	L_2, 64p	228K	179M
inception (3c)	14×14×640	2	0	128	256,2	32	64,2	m 3×3,2	398K	108M
inception (4a)	14×14×640	2	256	96	192	32	64	L_2, 128p	545K	107M
inception (4b)	14×14×640	2	224	112	224	32	64	L_2, 128p	595K	117M
inception (4c)	14×14×640	2	192	128	256	32	64	L_2, 128p	654K	128M
inception (4d)	14×14×640	2	160	144	288	32	64	L_2, 128p	722K	142M
inception (4e)	7×7×1024	2	0	160	256,2	64	128,2	m 3×3,2	717K	56M
inception (5a)	7×7×1024	2	384	192	384	48	128	L_2, 128p	1.6M	78M
inception (5b)	7×7×1024	2	384	192	384	48	128	m, 128p	1.6M	78M
avg pool	1×1×1024	0								
fully conn	1×1×128	1							131K	0.1M
L2 normalization	1×1×128	0								
total									7.5M	1.6B

Figure 3. FaceNet Architecture

The model will be implemented with the following steps: images will be normalized and transformed into vector representations using FaceNet. Then, a convolutional process and processing will be built for facial recognition and data training[9].

FaceNet also has an accuracy of 94.04% and only has 0.018s inference time when using it on mobile computing device [10], this indicate that FaceNet is a lightweight model that can be used in a CCTV based environment.

2.2. GhostFaceNets

GhostFaceNets leverage Ghost modules, a novel architectural component, to enhance the efficiency and effectiveness of lightweight face recognition models[11]. By employing Ghost modules, GhostFaceNets extract additional feature maps from intrinsic features through cost-effective linear transformations, thereby mitigating feature map redundancy and facilitating a more comprehensive representation of underlying facial information.

Built upon GhostNetV1 and GhostNetV2 architectures, GhostFaceNets integrate Ghost modules to achieve lightweight yet robust face recognition models. GhostNetV2 further refines GhostNetV1 by incorporating an attention mechanism. GhostFaceNets are best implemented in FPGA based device because of their highly paralel nature. [12]

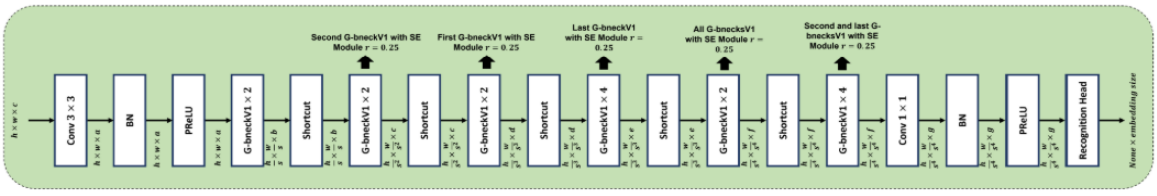

Figure 4. GhostFaceNets Architecture

GhostFaceNets introduce three major modifications. Firstly, they incorporate a different output head setting. Secondly, they replace ReLU with PReLU as the activation function[13]. Thirdly, they adapt the SE module to enhance the discriminative power of GhostFaceNets [2]

2.3. VGGFace

VGGFace is a method used for image recognition[14]. It has an input size of 224x224 pixels and utilizes small 3x3 convolutional filters. The padding layer contains 1 pixel for every 3 convolutional layers. Max-pooling is performed with a 2x2 layer. There are 3 fully-connected layers. The first two layers have 4096 channels each, and the last layer has 100 channels. The final layer includes a softmax function [9]. The VGGFace model is illustrated in Figure 5.

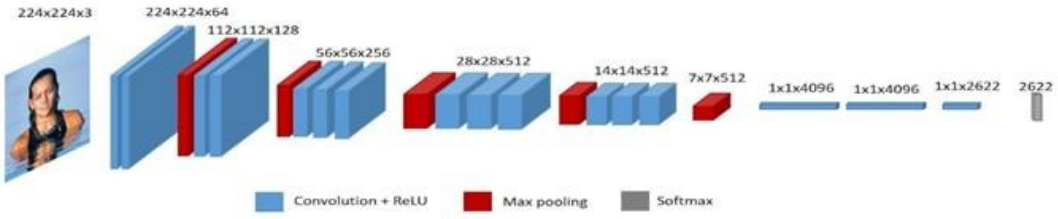

Figure 5. VGGFace Model

The model will be implemented by performing data preprocessing, which includes image normalization and adjustment to meet the input requirements of VGGFace. The structure of this model involves building an encoder, decoder, and convolutional layers. The final step is model training[15].

VGGFace is very sensitive to their training data, not only it affect the performance, it can also affect the accuracy of the model resulted from that training data [16], but on the right training dataset, it also can achieve up to 95% accuracy[17].

3. Methodology

3.1. Confusion Matrix and Accuration

Confusion Matrix is a tool used in classification analysis to evaluate the accuracy of a model or algorithm. The variables used are the "predicted name" of the face owner compared to the "actual name" as the Ground Truth. With these two variables, we obtain the counts of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) [18]. After obtaining the value from confusion matrix, then it will be used to calculate the accuracy.

Accuracy will be calculated using the following formulas[19]:

$$\text{Accuration} = \frac{TP + TN}{TP + TN + FP + FN}$$

3.2. Cumulative Match Characteristic (CMC)

The Cumulative Match Characteristic (CMC) is an evaluation method used to measure the effectiveness of face recognition in correctly identifying individuals from a given list of people and when additional data is added to the list[20]. This evaluation is conducted by sorting the given list of people based on the likelihood of data matching and recording the rank indicating how many

subjects were successfully identified. CMC is presented in the form of a graph illustrating the accuracy of identification against various sizes of different lists of people.

4. Implementation

4.1. Datasets Explanation

Dataset collection was performed using a Xiaomi Mi 10 smartphone camera. Ten photos were captured for each individual, taken from various angles including right profile, frontal view, and left profile.

The collected dataset was then divided into training, validation, and testing data sets with a ratio of 60% for training data, 20% for validation data, and 20% for testing data.

4.2. Feature Extraction

Face Detection.

The face detection process will utilize the Multi-Task Cascaded Convolutional Neural (MTCNN), which will produce facial locations in the form of x and y coordinate points. These coordinate points are utilized to crop the photo, retaining only the portions containing the faces.

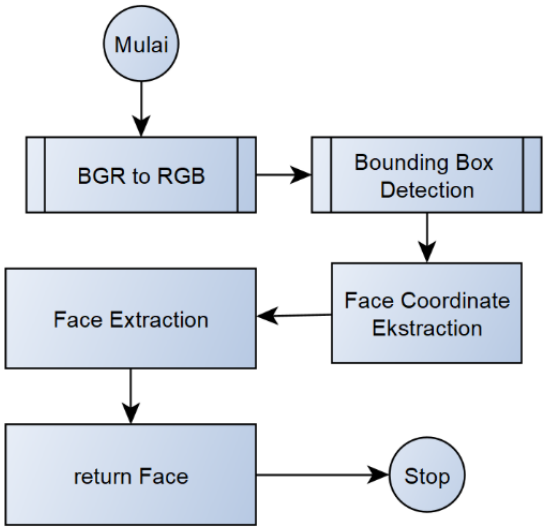

Figure 6. Face Detection Flowchart

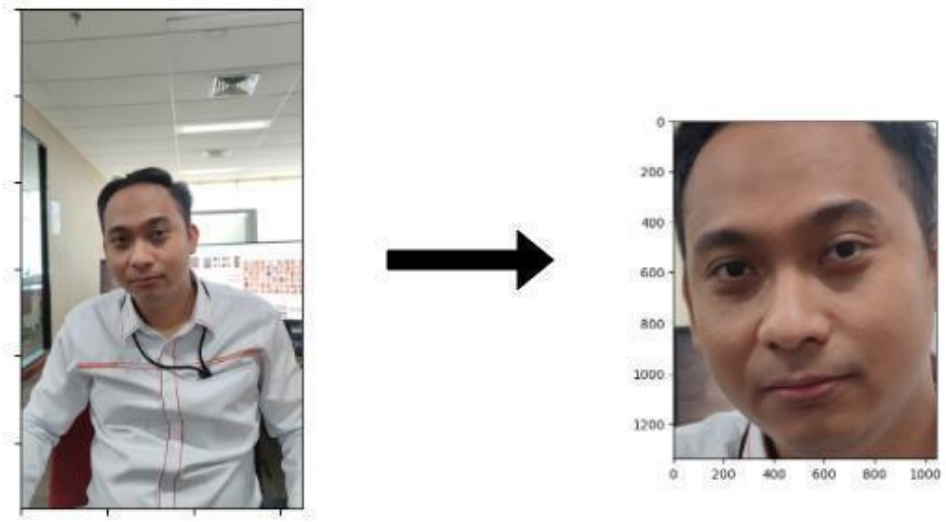

Figure 7. Detected Face Images

Image Size Adjustment.

The sizes resulting from the face detection process vary for each photo, necessitating adjustments in image size to ensure smooth processing. Initially, the process involves identifying the disparity between the size of the detected face and the desired image size. Subsequently, this difference is utilized to scale the detected face to match the target image size. If the image size after adjustment still does not align with the desired dimensions, the final step involves adding a black background to the image until it reaches the desired size.

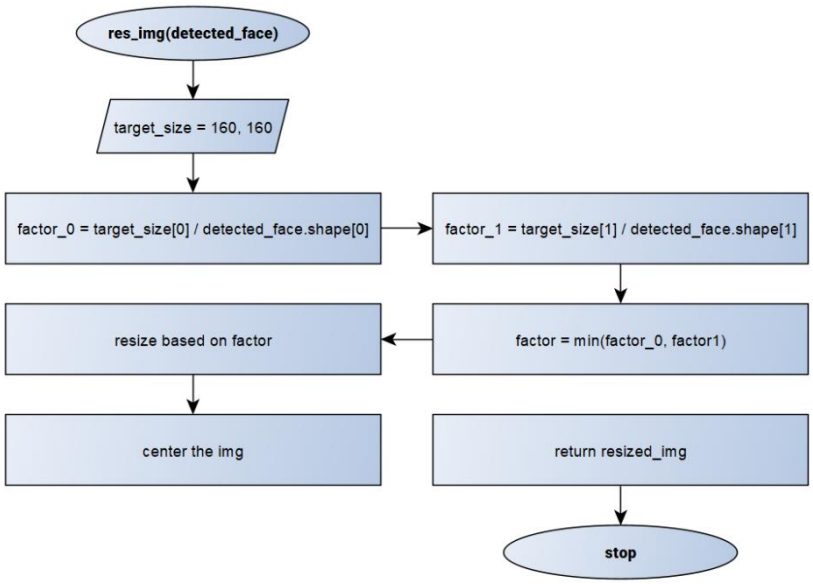

Figure 8. Image Resizing Flowchart

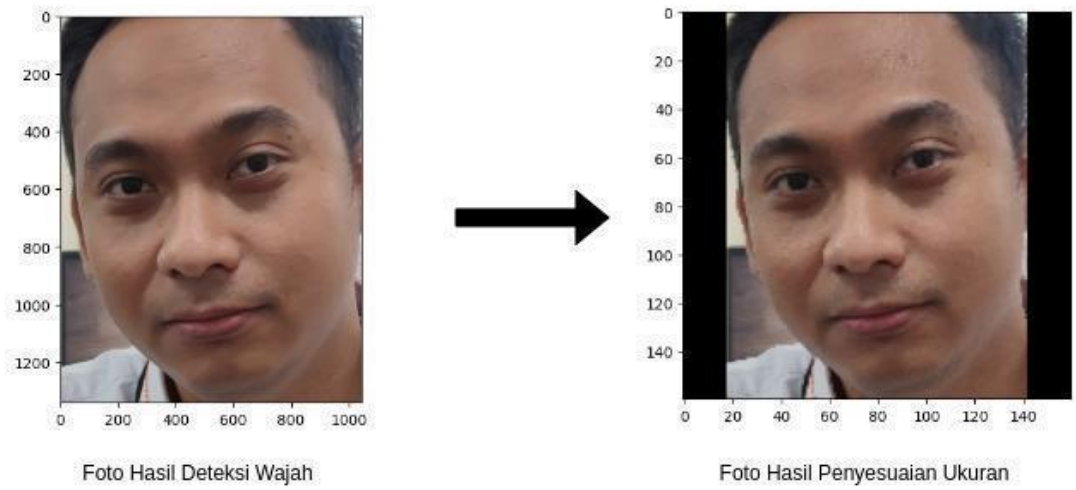

Figure 9. Image Before and After Image Resizing

Face Embedding.

Face embedding in FaceNet will yield a 128-dimensional vector data. To generate this vector, the images resulting from the face detection process are first converted into numerical data with values ranging from 0 to 1.

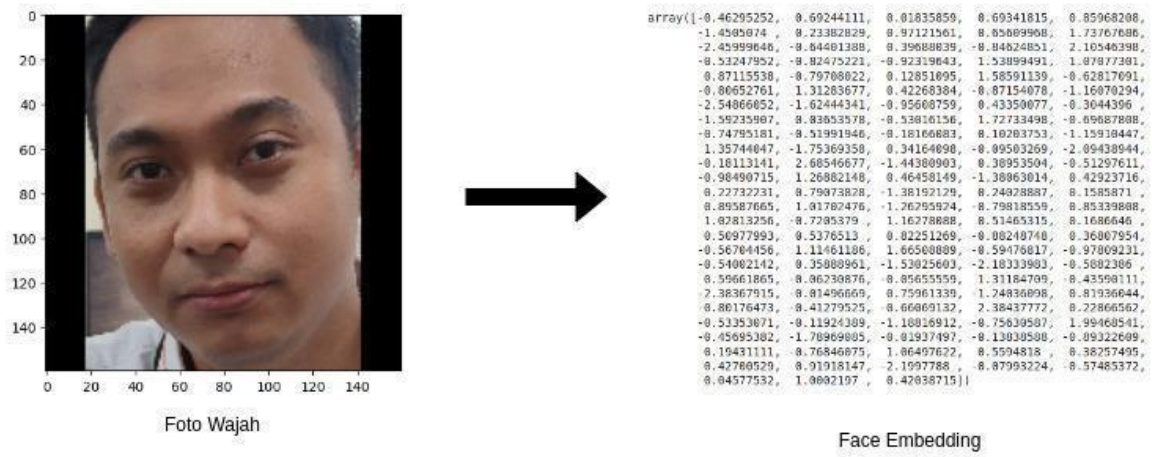

Figure 10. Face Embedding

Face Detection Training.

The training data for face recognition is conducted by extracting face embedding data from each facial image in the training data for each individual. Subsequently, the face embedding data is aggregated into a single dataset by computing the average face embedding from each individual's facial images. The resulting averages are then saved in files with the (.txt) extension.

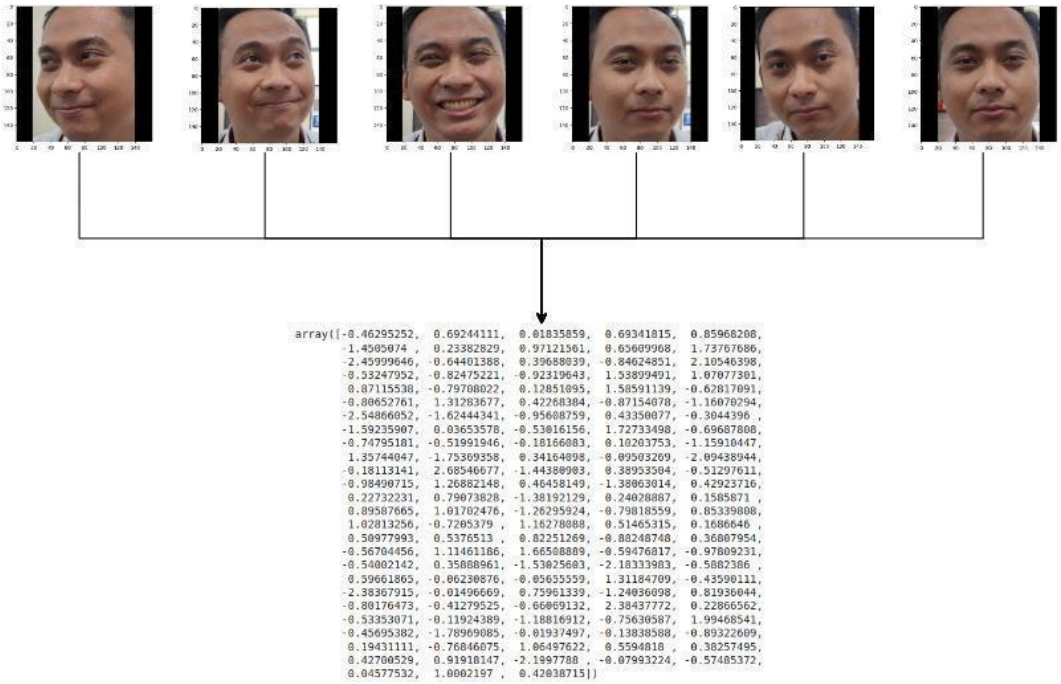

Figure 11. Image Before and After Resizing

4.3. Validation

Validation data are used to tune the model by feeding it with image that are has not been trained. The model will be currated based on the accuracy of each validation data.

```
def validation(classifier):
  y_true = list()
  y_pred = list()

  for folder in data_validatitons:
    filenames = os.listdir(folder)

    face_embedding_temp = list()
    for filename in filenames:
      if filename.endswith(".txt"):
        continue

      image_path = f"{folder}/{filename}"
      img = cv2.imread(image_path)
      detected_face = face_detection(img)
      if detected_face is None:
        print("Face Not Detected : ", image_path)
        continue

      current_img = resize_image(detected_face, (112, 112))

      face_embedding = face_embeddings(current_img)
      face_embedding = np.array(face_embedding)
      closest_distances = classifier.predict([face_embedding])
      label = image_path.split('/')[2]

      y_true.append(1)
      if label in closest_distances:
        y_pred.append(1)
      else:
        y_pred.append(0)

  accuracy = accuracy_score(y_true, y_pred) * 100
```

4.4. Testing

Facial recognition initiates with the retrieval of face embedding data stored in (.txt) files, resultant from the facial data training process. Subsequently, facial recognition entails the computation of distances from the face embedding data of each image employing the Euclidean distance formula. The distance values for each facial image obtained are then subjected to a process to identify facial images that meet the predetermined threshold distance value.

Following this, testing is conducted to verify the accuracy of the facial recognition process. This entails identifying the smallest distance value among the entire set of distances, comparing them to facial data with distance values meeting predefined thresholds. A correct testing outcome indicates successful facial data recognition, while an incorrect outcome signifies a failure in facial data recognition.

5. Results and Discussion

Validation and testing procedures are conducted by aligning the identities of individuals in facial images with the facial recognition results. It will compare the distance between testing data and trained data. If the distance is lower than the threshold value, it will mark that as true. Subsequently, the outcomes are utilized to construct accuracy data and confusion matrices to quantify the accuracies of each method.

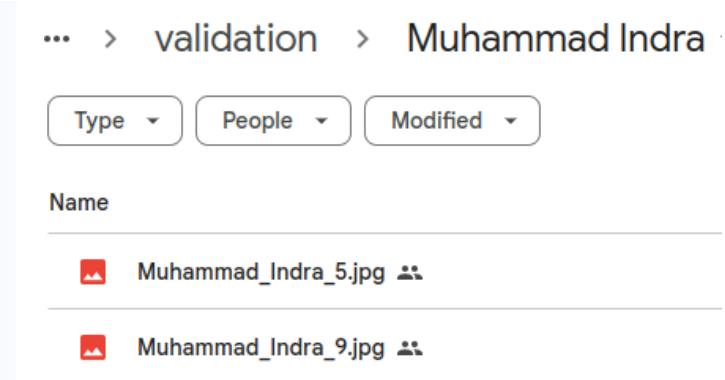

Figure 12. Validation Data

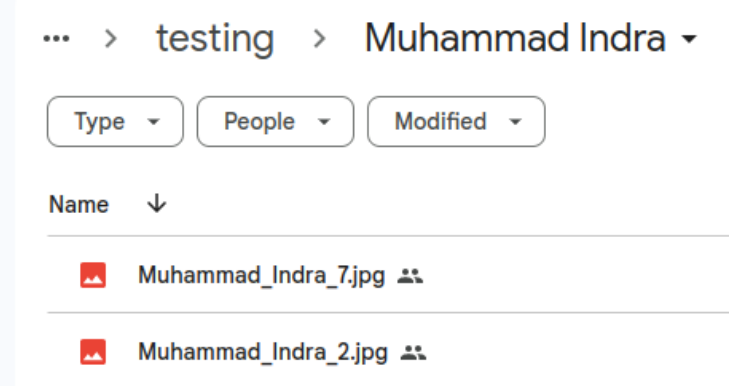

Figure 13. Testing Data

5.1. Comparison Table

The validation and testing procedures employ a dataset comprising two facial images for validation and two for testing per individual. Initially, the algorithm computes the embedding value of the current test image and subsequently measures its Euclidean distance from the embeddings of the trained data. The algorithm selects the embedding with the minimum distance as the final prediction. A predetermined threshold of 10 units is applied to distinguish between valid and invalid matches during validation and testing.

Figure 14. Minimum Distance Searching

The validation outcomes reveal the accuracy of each method. Notably, the FaceNet method attained an accuracy of 97.05% during validation and 97.4% during testing, showcasing its robust accuracy. Conversely, the VGGFace method exhibited similar accuracy levels, with 97.05% during validation and 96.1% during testing. These results indicate that the FaceNet method demonstrates robust accuracy, characterized by high and consistent accuracy levels, making it the preferred choice for facial recognition applications. However, the emergence of GhostFaceNets presents an intriguing addition to the landscape, albeit with significantly lower accuracy rates at 79.41% during validation and 75.32% during testing. Further exploration and refinement are necessary to unlock its full potential for practical deployment. Detailed validation and testing outcomes are provided in Table 1.

Table 1. Accuration Table Comparison

No	Model	Accuration	
		Validation	Testing
1	FaceNet	97.05%	97.4 %
2	VGGFace	97.05%	96.1%
3	GhostFaceNets	79.41%	75.32%

5.2. Cumulative Match Characteristic (CMC)

In the Cumulative Match Characteristic (CMC) calculation, the dataset of 68 facial images from validation folder is partitioned into 17 groups, each containing four photos. Subsequently, the algorithm computes the percentage of successful matches achieved at various ranks within the CMC curve. Here, rank 1 signifies that all photos within a group are correctly identified, while rank 4 indicates that only one photo is correctly identified within the group.

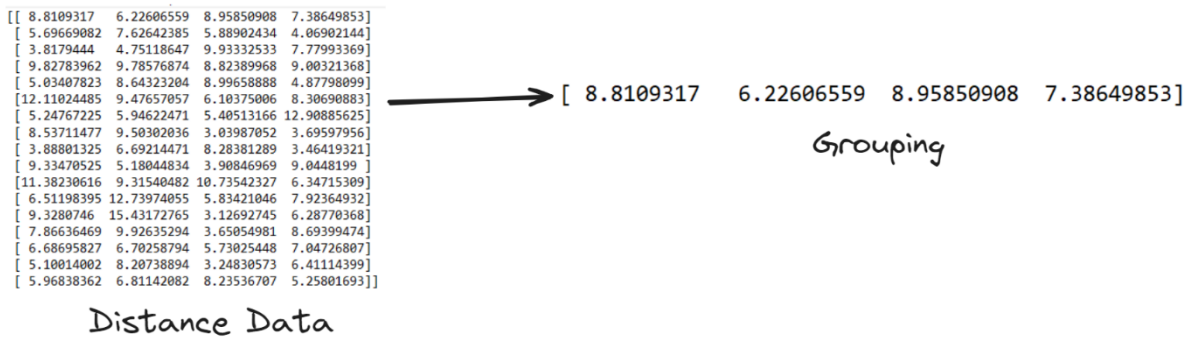

Figure 15. Data Grouping

FaceNet

During the assessment of the FaceNet model, variations in recognition efficacy were observed corresponding to the quantity of accurately identified images within distinct sets. Notably, within any given set, the FaceNet model displayed a 70.5% likelihood of correctly identifying all four images. Additionally, the model exhibited a recognition accuracy of 95.1% in identifying three out of the four images within a set. Furthermore, when tasked with recognizing two out of the four images, the FaceNet model achieved a flawless recognition rate of 100%. Noteworthy is the model's consistent accuracy, achieving a perfect recognition rate of 100% when identifying at least one image from a set of four.

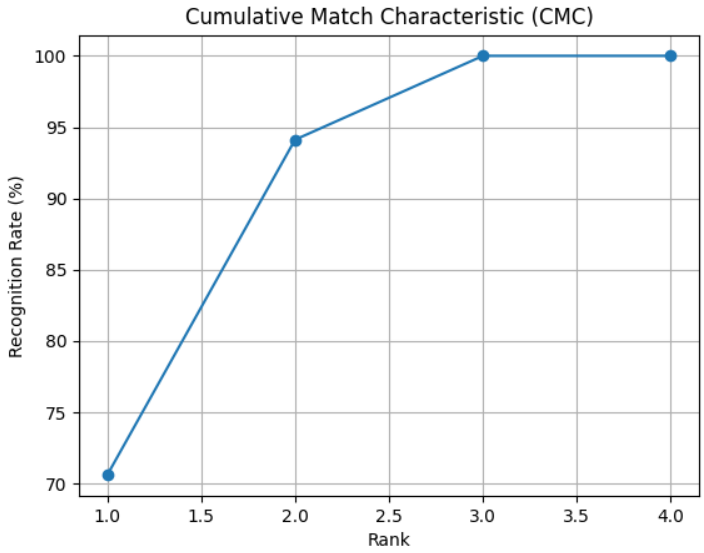

Figure 16. FaceNet CMC Calculation

VGGFace

In the evaluation of the VGGFace model, variations in recognition accuracy were observed corresponding to the number of accurately identified images within different sets. Notably, within each set, the FaceNet model demonstrated a 64.7% likelihood of correctly identifying all four images. Furthermore, the model exhibited a recognition accuracy of 94.1% in identifying three out of the four images within a set. Additionally, when tasked with recognizing two out of the four images, the FaceNet model achieved a flawless recognition rate of 100%. It is noteworthy that the model consistently performed well, achieving a perfect recognition rate of 100% when identifying at least one image from a set of four.

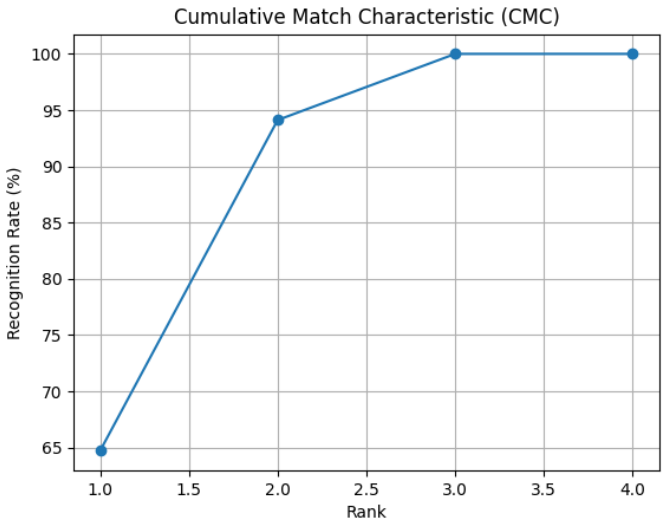

Figure 17. VGGFace CMC Calculation

GhostFaceNets

The assessment of the GhostFaceNets model revealed varying recognition accuracy based on the number of images correctly identified within different sets. Within each set, the FaceNet model showed a 23.5% chance of accurately identifying all four images. Additionally, it achieved a recognition accuracy of 58.8% for identifying three out of the four images and a flawless recognition rate of 94.1% for recognizing two out of the four images. Notably, the model consistently excelled, achieving a perfect recognition rate of 100% when identifying at least one image from a set of four.

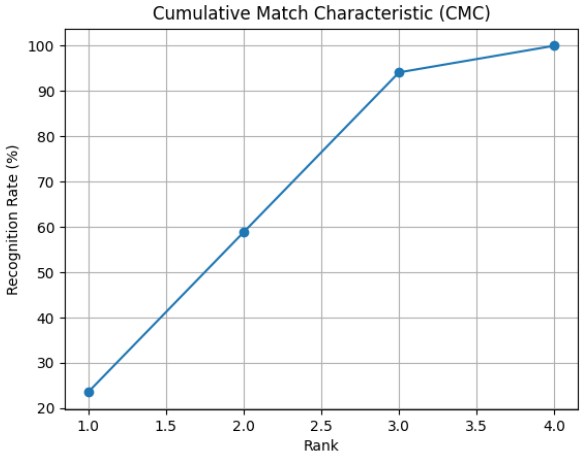

Figure 18. GhostFaceNets CMC Calculation

6. Conclusion

In conclusion, the comparative analysis of FaceNet, VGGFace, and GhostFaceNets sheds light on their respective accuracy in facial recognition for suspect identification. FaceNet emerges as the frontrunner, showcasing consistent and superior accuracy rates of 97.05% during validation and 97.4% during testing. Conversely, while VGGFace demonstrates commendable accuracy, its slightly lower accuracy, with rates of 97.05% and 96.1% during validation and testing respectively, positions it behind FaceNet. Notably, GhostFaceNets, though promising, exhibits notably lower accuracy rates at 79.41% during validation and 75.32% during testing, signaling areas for further refinement. Therefore, FaceNet stands as the method of choice for robust and reliable facial recognition applications, with GhostFaceNets warranting further investigation and optimization for practical deployment.

Acknowledgment

The main prospect and the scope for this research was conducted and idea perspective investigations with the manuscript writing was done by the authors themselves. All the datasets, data tools, data models, data sources which have been retrieved and used for the conduction of this research are mentioned and referenced where appropriate

References

[1] A. D. Putra, G. Stevi Martha, M. Fikram, R. J. Yuhan, and P. S. Stis, “Faktor-Faktor yang

Memengaruhi Tingkat Kriminalitas di Indonesia Tahun 2018," *jurnal.uns.ac.idAD Putra, GS Martha, M Fikram, RJ YuhanIndonesian J. Appl. Stat. 2021•jurnal.uns.ac.id*, 2020, Accessed: May 20, 2024. [Online]. Available: https://jurnal.uns.ac.id/ijas/article/view/41917

[2] D. M. Hemil Shah, Saakshi Mishra, Bhumi Dubey, "Criminal Investigation with the Help of Face Recognition," *Interantional J. Sci. Res. Eng. Manag.*, vol. 08, no. 02, pp. 1–13, 2024, doi: 10.55041/ijsrem28671.

[3] V. C R, V. Asha, B. Saju, S. N, T. Reddy, and S. M, "Face Recognition and Identification Using Deep Learning," 2023, pp. 1–5. doi: 10.1109/ICAECT57570.2023.10118154.

[4] J. A. Mensah, J. K. Appati, E. K. A. Boateng, E. Ocran, and L. Asiedu, "FaceNet recognition algorithm subject to multiple constraints: Assessment of the performance," *Sci. African*, vol. 23, p. e02007, 2024, doi: https://doi.org/10.1016/j.sciaf.2023.e02007.

[5] F. Schroff, D. Kalenichenko, J. P.-P. of the IEEE, and undefined 2015, "Facenet: A unified embedding for face recognition and clustering," *cv-foundation.orgF Schroff, D Kalenichenko, J PhilbinProceedings IEEE Conf. Comput. Vis. pattern, 2015•cv-foundation.org*, Accessed: May 20, 2024. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Schroff_FaceNet_A_Unified_2015_CVPR_paper.html

[6] H.-C. Li, Z.-Y. Deng, and H.-H. Chiang, "Lightweight and Resource-Constrained Learning Network for Face Recognition with Performance Optimization," *Sensors*, vol. 20, 2020, doi: 10.3390/s20216114.

[7] A. Alnissany and Y. Dayoub, "Modified centroid triplet loss for person re-identification," *J. Big Data*, vol. 10, no. 1, p. 74, 2023, doi: 10.1186/s40537-023-00753-0.

[8] A. Raju, T. Saravanan, and J. Arul, "AI based face recognition system using FaceNet deep learning architecture," in *AIP Conference Proceedings*, 2022, vol. 2640, p. 20031. doi: 10.1063/5.0118073.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, 2015.

[10] M. A. Munim and M. Kohinoor, "Performance Evaluation of Deep Learning-Based Facial Recognition Models on Mobile Computing Environments," 2023, pp. 13–18. doi: 10.1109/R10-HTC57504.2023.10461876.

[11] M. Alansari, O. A. Hay, S. Javed, A. Shoufan, Y. Zweiri, and N. Werghi, "GhostFaceNets: Lightweight Face Recognition Model From Cheap Operations," *IEEE Access*, vol. 11, no. April, pp. 35429–35446, 2023, doi: 10.1109/ACCESS.2023.3266068.

[12] F. Zhao, P. Zhang, R. Zhang, and M. Li, "UnifiedFace: A Uniform Margin Loss Function for Face Recognition," *Appl. Sci.*, vol. 13, no. 4, 2023, doi: 10.3390/app13042350.

[13] A. Bukti Djufrie, M. Amri Akbar, B. Armansyah, M. Fadhil Bin Bahrunnida, and N. Azqalani, "Face Recognition on Low Resolution CCTV Video using GhostFaceNets," *journal.uob.edu.bhA Bukti Djufrie, M Amri Akbar, B Arman. M Fadhil Bin Bahrunnida, N AzqalaniInternational J. Comput. Digit. Syst. 2024•journal.uob.edu.bh*, Accessed: Jul. 08, 2024. [Online]. Available: https://journal.uob.edu.bh/handle/123456789/5741

[14] K. Balamurali, S. Chandru, M. Razvi, and S. Kumar, "Face Spoof Detection Using VGG-Face Architecture," *J. Phys. Conf. Ser.*, vol. 1917, p. 12010, 2021, doi: 10.1088/1742-6596/1917/1/012010.

[15] J. Sen, M. Hena, Rahman, and B. Sarkar, "Face Recognition Using Deep Convolutional Network and One-shot Learning," *Int. J. Comput. Sci. Eng.*, vol. 7, pp. 23–29, 2020, doi: 10.14445/23488387/IJCSE-V7I4P107.

[16] Z. Qawaqneh, A. A. Mallouh, and B. D. Barkana, "Deep Convolutional Neural Network for Age Estimation based on VGG-Face Model," Sep. 2017, Accessed: Jul. 08, 2024. [Online]. Available: http://arxiv.org/abs/1709.01664

[17] H. Sunarko, R. Hidayat, and R. Hartanto, "Comparative Analysis of Masked and Unmasked for Face Recognition Using VGG Face and MTCNN," 2023.

[18] P. Diez, "Chapter 1 - Introduction," in *Smart Wheelchairs and Brain-Computer Interfaces*, P. Diez, Ed. Academic Press, 2018, pp. 1–21. doi: https://doi.org/10.1016/B978-0-12-812892-3.00001-7.

[19] T. Hernández-Del-Toro, F. Martínez-Santiago, and A. Montejo-Ráez, "Chapter 7 - Assessing

classifier's performance," in *Biosignal Processing and Classification Using Computational Learning and Intelligence*, A. A. Torres-García, C. A. Reyes-García, L. Villaseñor-Pineda, and O. Mendoza-Montoya, Eds. Academic Press, 2022, pp. 131–149. doi: https://doi.org/10.1016/B978-0-12-820125-1.00018-X.

[20] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, "The relation between the ROC curve and the CMC," in *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*, 2005, pp. 15–20. doi: 10.1109/AUTOID.2005.48.