AMERICAN JOURNAL OF MULTIDISCIPLINARY RESEARCH AND INNOVATION (AJMRI)

ISSN: 2158-8155 (ONLINE), 2832-4854 (PRINT)

VOLUME 1 ISSUE 5 (2022)



PUBLISHED BY: E-PALLI, DELAWARE, USA

American Journal of Multidisciplinary Research and Innovation (AJMRI)

Volume 1 Issue 5, Year 2022
ISSN: 2158-8155 (Online), 2832-4854 (Print)
DOI: https://doi.org/10.54536/ajmri.v1i5.740
https://journals.e-palli.com/home/index.php/ajmri

Application of Three Convolutional Neural Network Algorithms for Occluded Face Identification and Recognition for System Security

Mamadou Diarra[1*], Ayikpa Kacoutchy Jean[2], Ballo Abou Bakary[1], Kouassi Brou Medard[1]

Article Information

Received: October 10, 2022

Accepted: October 21, 2022

Published: October 26, 2022

Keywords

Computer Vision, CNN Pre-Trained, Deep Learning, Face Recognition, Occlusion

ABSTRACT

Deep Learning techniques in computer vision have become indispensable elements in biometric systems, especially face recognition. Facial recognition can be reliably used as an identification and authentication tool for premises or network access security. The masks wearing, which is one of the problems of concealment, are nowadays part of our habits for preventing COVID-19 disease, which leads to an obstruction of facial recognition. Occulted face recognition is one of the most challenging problems biometrics deals with. This paper presents convolution neural network algorithms for occluded face recognition. Our study presents a robust method using algorithms such as ResNet-50, VGG-19, and DenseNet-201 to contribute to occluded face recognition. Various parameters are used for this experiment, such as the cross-entropy used as a loss function and optimization algorithms adapted to deep learning. These include the SGD, Adam, and RMSProp optimizers. The convolution neural network algorithms were evaluated on the AR database. This experiment gave results that ranged from 94.81 to 99.81% for SGD, from 0 to 96.92 for Adam, and finally from 0 to 96.92 for RMSProp. DenseNet-201 algorithm using the SGD optimizer obtained the best score with 99.81%, and all the performance metrics used such as accuracy, MSE, F-score, recall, and MCC were used to confirm this good performance.

INTRODUCTION

The significant increase in network security breaches, data breaches, and identity theft requires the design of robust security systems including biometrics. To circumvent biometric face authentication, some fraudsters are turning to face occlusion. Because biological and physical characteristics are unique to each individual, biometric security consists of measuring these characteristics before accessing an environment or a computer tool. Face recognition is a key research issue in computer vision. In recent years, researchers have proposed many algorithms; most previous biometric-based research exploiting physiological and behavioral characteristics, including human emotion signals and expression, has achieved satisfactory recognition performance under uniform lighting conditions with frontal face images (Jiang *et al.*, 2020). However, illumination, facial expression, pose, occlusions, and facial recognition methods are still affected. The development of research in facial recognition has led to a high level of performance in many applications. It is a field of study that remains very challenging because images of the same person seem to differ due to several phenomena, including occlusion (Wu *et al.*, 2019). In addition to existing security methods, face biometrics, especially with occlusion, can be used to protect cyberspace from hackers and malicious people among users of networks, the internet, connected devices, etc.

Among the various problems associated with a face recognition system, occlusion management is one of the most difficult problems to solve. Due to objects or elements such as sunglasses, scarves, or masks, the occlusion problem becomes eminent. One of the most recent problems is the wearing of the mask recommended by the health measures against the coronavirus disease (COVID-19). Cloaked face images mainly degrade the performance of face recognition systems, thus the need for a robust cloaked face system is necessary for real-world applications. In this perspective, we will use new approaches based on machine learning, more precisely on deep learning, which extracts hierarchical and semantic structures existing in images; these are convolutional neural networks used. Convolutional neural networks, which are multilayer perceptrons coupled with convolutional layers, are part of deep learning approaches and have become indispensable for detection and recognition in computer vision (Siegmund *et al.*, 2021). They can extract landmarks by themselves (Wang *et al.*, 2020). Pre-trained convolutional neural network algorithms allow for transfer learning, which transfers the skill learned on one dataset to adapt it to a new dataset it will be faced with (Arnia *et al.*, 2021). The algorithms used were trained on the image database, namely ImageNet.

The study contributes to recognizing hidden faces by comparing three deep learning algorithms, ResNet-50, VGG-19, and DenseNet-201, based on pre-trained convolutional neural network algorithms. These algorithms will be used for the face recognition of occluded faces from the reference AR face database, and it will be a question of comparing the performances by playing on the following parameters:

- Epochs
- Batch size
- Optimizers

[1] Laboratoire Mécanique et Informatique, Université Felix Houphouët-Boigny, Côte d'Ivoire
[2] Unité de Recherche et d'Expertise numérique, Université Virtuelle de Côte d'Ivoire, Côte d'Ivoire
[*] Corresponding author's e-mail: patoudiarra@gmail.com

All parameters will evaluate the performance of the previous algorithm and select the one with the best accuracy on the reference data used. The challenges of face occlusion and deep learning could bring innovation to the security of computer networks and cybersecurity.
This article is organized as follows: Section 2 presents some previous work. Section 3 offers convolution neural networks and details our proposed methodology and hardware. Section 4 presents the analysis of our experiments and results, evaluates and compares our used algorithms, also discusses the obtained results with other works, and finally, we will conclude.

LITERATURE REVIEW

In this section, we will review some face recognition work with occlusion. Methods addressing face recognition with occlusion include finding features or classifiers that tolerate corruption. For example, Aleix M. Martinez proposed a probabilistic approach that can compensate partially occluded faces (Martinez, 2002). Park *et al.* proposed a new feature-based face recognition algorithm where the face-ARG model represents a face. All geometric quantities and structural information are encoded into an attributed relational graph (ARG) structure. Then partial ARG matching is performed to match the face ARGs (Bo-Gun Park *et al.*, 2005); Min *et al.* propose an efficient approach that first analyzes the presence of a potential occlusion on a face and then performs face recognition on the unoccluded facial regions based on selective local Gabor binary models (Min *et al.*, 2014), Tsai *et Al.* proposed a deep convolution neural network architecture for efficient multi-person and multi-angle face recognition, this achieved the identity confidence by using a classifier for these features. The experimental results showed that the accuracy of identity recognition could reach 90.61% (Tsai *et al.*, 2018). Montera *et Al.* proposed a face recognition method performed using a hybrid process that combines Haar Cascades and Eigenface methods, which can detect multiple faces (55 faces) in a single detection process with an accuracy level of 91.67% (Mantoro *et al.*, 2018); Lu *et al.* proposed a partial occlusion face recognition algorithm based on a recurrent neural network that yields a result that ranges from 88.49 to 98.45% (Zhang *et al.*, 2020), Wu *et al.* proposed a method based on deep learning for occluded face recognition with the result that reaches 98.6% (Wu *et al.*, 2019).

MATERIALS AND METHOD

Machine learning is a branch of artificial intelligence (AI) that uses algorithms to enable computer systems to infer patterns from data. It has many applications, including bioinformatics, fraud detection, finance, human resource and risk management, market analysis, image recognition, and natural language processing (Praseetha *et al.*, 2019). New face recognition methods extract the best features from images and tend to learn these features using deep convolution neural network architectures (Idelette Kambi Beli & Guo, 2017). This has led to the extraordinary success of famous convolutional architectures such as VGGNet, GoogleNet, ResNet, etc.(Zhou *et al.*, 2018). Deep learning is one of the most widely used machine learning techniques that has been hugely successful in applications such as anomaly detection, image detection, pattern recognition, and natural language processing (Praseetha *et al.*, 2019). But training deeper neural networks is challenging due to the vanishing gradient and degradation problems (Reddy & Juliet, 2019).
However, there are four (4) significant families of deep learning algorithms, deep neural network, convolutional neural network, recurrent neural network, and deep belief network (Singh *et al.*, 2020). Our study will use convolutional neural networks such as ResNet-50, VGG-19, and Dense-Net-201.

ResNet-50 model

ResNet-50 model is a convolutional neural network 50 layers deep; Microsoft built and trained it in 2015 (He *et al.*, 2015). This model was trained on more than one million images from the ImageNet database, it can classify up to 1000 objects, and the network was trained on 224x224 pixel-colored images. It contains 33 623 012 parameters.

VGG-19 model

VGG-19 model is a convolutional neural network 19 layers deep; it was developed by the Visual Geometry Group of the Department of Engineering Sciences at Oxford University. This model has been trained on over a million images from the ImageNet database, it can classify up to 1000 objects, and the network was trained on 224x224 pixel-colored images. It contains 21 560 484 parameters.

DenseNet-201 model

DenseNet-201 model is a convolutional neural network of 201 layers of depth. It was implemented by Huang *et al.* (Siegmund *et al.*, 2021). This model was trained on more than one million images from the ImageNet database, it can classify up to 1000 objects, and the network was trained on 224x224 pixel-colored images. It contains 21 202 084 parameters.

AR faces database

The database used is the AR face database (AR Face Database Webpage, s. d.). It contains more than 4,000 colored faces of 126 persons, namely 70 men and 56 women. Frontal faces are characterized by different facial expressions, lighting conditions, and occlusions like sunglasses and scarf. There are 26 other images per person, taken in two sessions separated by two weeks, each consisting of 13 images. A dataset of 2600 images of 100 different subjects (50 males and 50 females) were used in our experiment; Martinez and Kak (Martinez & Kak, 2001) used the same data set. Each of the images in this dataset is 165x120x3 pixels in size.

- Faces with frontal view and lighting conditions
- Faces with facial expression
- Faces occluded with a foxhole
- Faces occulted with a lens and a pair of glasses
- Faces occulted with a pair of glasses
- Faces with a facial expression and a pair of glasses
- Faces occulted with a lens and a pair of glasses
- Faces occulted with a pair of glasses



Figure 1: Sample from the AR database

The architecture of our method

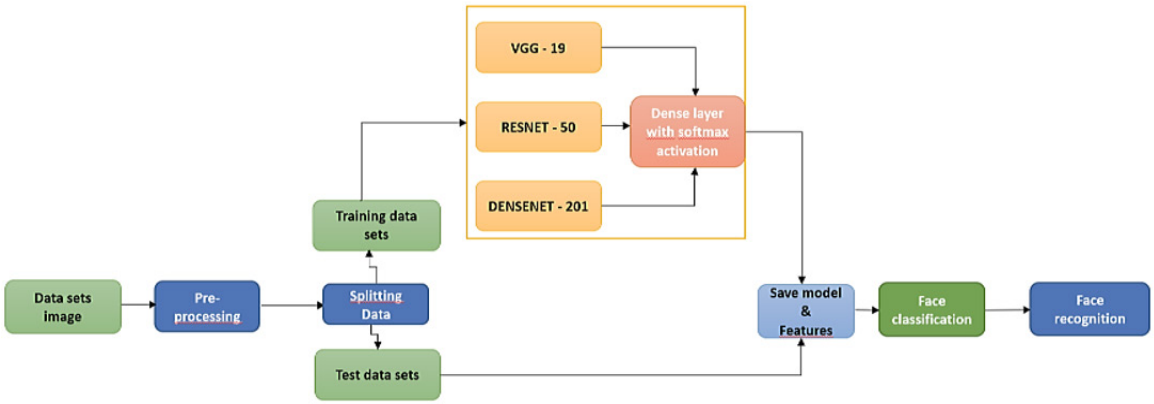


Figure 2: The architecture of our occluded face recognition method

Our method is as follows:

Step 1:

- Preprocessing
- Splitting data

Step 2:

- Loading of models
- Collecting extraction features
- Flatten data
- Activation function
- Model training

Step 3:

- Classification
- Face recognition

Preprocessing

• In this step, we will retrieve each image from our database to add it to a list and each label to another.

• Then, we will get the number of categories in our database to transform our list of labels into a matrix of size corresponding to the number of classes.

• Finally, we will normalize our images in the value interval [0;1].

Splitting data

• We will divide our dataset into two subsets:

• The first subset will be called the training dataset, which will be used to allow our model to do its learning.

• The second subset will be called the test dataset, which will be used to evaluate the learning of our model by testing the results obtained with the expected results

Loading models

We proceed to the loading of our model by passing the size of our images as a parameter without the fully connected layers of the model.

Collection of extraction features

We will use the convolution and pooling already trained in our model for the feature extraction of our images.

Flatten data

We will reduce the input dimensions of our data by adapting it to the input dimensions of the model.

Activation function

We used the softmax activation function because we have multiple categories, and softmax is efficient for multi-class classification. The mathematical representation of the softmax activation is :

$$\sigma(z)_j = \frac{e^{z_j}}{\dot{\Sigma}_{k=1}^{K} e^{z_k}} \qquad (1)$$

z is a vector such as z =(z1…,zk). .....(2)

K ∈ R+ and J ∈ {1…, K} .....(3)

Model training

We proceed to the training phase of the model in 15 epochs with different optimizers and batch sizes of 4,8, and 16.

Classification

We classify our test data according to the categories. Our data contains 100 categories that range from 0 to 99. The loss function used for our work is the cross-entropy to evaluate the loss during classification, its equation is as follows:

With test(x): vector containing the values of labels to be predicted and pred(x) is the vector containing values of labels provided by our softmax activation function. We will use the accuracy to evaluate the model. Its formula is as follows:

Face recognition

We proceed to recognize each image according to its classification in a category.

EXPERIMENTAL AND RESULTS

We trained the models on a Windows 10 system with an Intel(R) Core™ i7-8650U processor, 16 GB of random-access memory (RAM), and an NVIDIA GeForce MX150 graphics processing unit (GPU). The models are configured in Python using the Keras version 2.4 API with the TensorFlow version 2.4 backend and CUDA/CuDNN dependencies for GPU acceleration (Artificial Neural Networks. Pt. 3, 2010).

Setting

We used a batch size of 4,8, and 16 for 15 epochs for each method. Our study will use cross-entropy as a loss function and optimization algorithms suitable for deep learning to train the chosen models. These algorithms will directly affect the efficiency of the models in our study. The optimizers we will use are:

- SGD
- Adam
- RMSProp

SGD

SGD implements the stochastic gradient descent optimizer with a learning rate and momentum. The stochastic gradient algorithm is a gradient descent method that minimizes an objective function written as a sum of differentiable functions. The learning rate was set to 0.0001 with a momentum of 0.9 (Team, s. d.).

Adam

Adam is a stochastic gradient descent method based on adaptive estimation of first and second-order moments. Its implementation is quite simple and computationally efficient, and its memory usage is optimized and well adapted to significant data volume problems (Kingma & Ba, 2014).

RMSProp

Root Mean Squared Propagation, or RMSProp, is an extension of gradient descent using a decreasing average of partial gradients to adapt the step size for each parameter. Using a decreasing moving average allows

us to eliminate unnecessary gradients and keep the best partial gradients observed during the search progress, thus exceeding the limitations of AdaGrad (Papers with Code - RMSProp Explained, s. d.).

Summary of results

The results of the different models are shown in the table below:

SGD

Table 1: Results of the accuracy in % of our models with SGD.

Optimizer: SGD	**Batch size: 4**	**Batch size: 8**	**Batch size: 16**
Models	**Accuracy (%)**	**Accuracy (%)**	**Accuracy (%)**
VGG-19	98.65	98.27	94.81
ResNet-50	99.23	98.27	94.81
DenseNet-201	99.81	98.65	98.46

The results in Table 2 show the results obtained from our models on the different parameters; the DenseNet-201 model got the best results for optimizer SGD

Table 2: Results of the accuracy in % of our models with Adam

Optimizer: Adam	**Batch size: 4**	**Batch size: 8**	**Batch size: 16**
Models	**Accuracy (%)**	**Accuracy (%)**	**Accuracy (%)**
VGG-19	0.0	0.0	0.0
ResNet-50	96.92	95.00	81.35
DenseNet-201	92.50	92.50	91.35

The results in Table 2 show that the Res Net-50 model has a better score on batch sizes 4 and 8, while Dense Net-201 has the best result on batch size 16 for optimizer Adam

Table 3: Results of the accuracy in % of our models with RMSProp.

Optimizer: RMSProp	**Batch size: 4**	**Batch size: 8**	**Batch size: 16**
Models	**Accuracy (%)**	**Accuracy (%)**	**Accuracy (%)**
VGG-19	0.0019	0.0	0.0
ResNet-50	96.92	96.92	96.92
DenseNet-201	89.62	95.19	82.69

The results in Table 3 show the stability of the ResNet-50 model with a better score on all parameters used.

Evaluation Metrics

To validate the performance of the pre-trained models in our study, we will use the following metrics:

Precision is intuitively the ability of the classifier not to label as positive a sample that is negative

$$Precison = \frac{TP}{TP + FP} \qquad (4)$$

An estimator's mean square error (MSE) measures the average of the squared errors, i.e., the mean square difference between the estimated and actual values. It is a risk function corresponding to the expected value of the squared error loss. It is always non-negative, and values close to zero are better. The following equation defines it

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2 \qquad (5)$$

With Yi: the observed data and: the predicted values

Recall is the ability of a classifier to determine actual positive results

F1 score can be interpreted as a weighted average of precision and recall, where an F1 score reaches its best value at one and its worst score at 0.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad (7)$$

Matthews Correlation Coefficient (MCC) is used in machine learning to measure the quality of classifications. Its value is essentially between -1 and +1. A coefficient of (+1) represents a perfect prediction, 0 represents a random prediction, and (-1) represents an inverse prediction. The statistic is also known as the phi coefficient.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \qquad (8)$$

With:

- TP: True positive
- TN: True negative
- FP: False positive
- FN: False negative
-

Results of metrics evaluation

Metrics used to consolidate obtained results confirm that the ResNet-50 model had the best impact on batch size 4, while DenseNet-201 had the best effect on batch sizes 8 and 16 using RMSProp optimizer.

Table 4: Performance metrics of SGD optimizer models.

Setting – Optimizer: SGD - Batch size: 4					
Models	**Precision (%)**	**MSE (%)**	**F-score (%)**	**Recall (%)**	**MCC (%)**
VGG-19	99.02	0.019	98.70	98.64	98.64
ResNet-50	99.37	0.014	99.24	99.23	99.22
DenseNet-201	99.87	0.004	99.81	99.80	99.80
Setting – Optimizer: SGD - Batch size: 8					
Models	Precision (%)	MSE (%)	F-score (%)	Recall (%)	MCC (%)
VGG-19	98.58	0.025	98.28	98.26	98.25
ResNet-50	98.62	0.029	98.24	98.26	98.25
DenseNet-201	98.92	0.032	98.67	98.65	98.64
Setting – Optimizer: SGD - Batch size: 16					
Models	Precision (%)	MSE (%)	F-score (%)	Recall (%)	MCC (%)
VGG-19	96.01	0.070	94.88	94.80	94.75
ResNet-50	95.57	0.089	94.71	94.80	94.75
DenseNet-201	98.71	0.070	98.43	98.46	98.44

Metrics used to consolidate the obtained results confirm that the DenseNet-201 model had the best effects on all parameters using the SGD optimizer.

Table 5: Performance metrics of Adam optimizer models.

Setting – Optimizer: Adam - Batch size: 4					
Models	**Precision (%)**	**MSE (%)**	**F-score (%)**	**Recall (%)**	**MCC (%)**
VGG-19	0.0	0.99	0.0	0.0	0.0
ResNet-50	97.92	0.049	97.04	96.92	96.89
DenseNet-201	94.07	0.1268	92.39	92.50	92.43
Setting – Optimizer: Adam - Batch size: 8					
Models	Precision (%)	MSE (%)	F-score (%)	Recall (%)	MCC (%)
VGG-19	0.0	0.99	0.0	0.0	0.0
ResNet-50	96.12	0.078	94.92	95.00	94.95
DenseNet-201	94.30	0.1185	92.50	92.50	92.43
Setting – Optimizer: Adam - Batch size: 16					
Models	Precision (%)	MSE (%)	F-score (%)	Recall (%)	MCC (%)
VGG-19	0.0	0.99	0.0	0.0	0.0
ResNet-50	87.95	0.312	81.80	81.34	81.25
DenseNet-201	94.73	0.137	91.45	91.34	91.28

Metrics used to consolidate results confirm that the ResNet-50 model had the best results on batch sizes 4 and 8, while DenseNet-201 had the best effect at batch size 16 using the Adam optimizer

Table 6: Performance metrics of RMSProp optimizer models

Setting – Optimizer: RMSProp - Batch size: 4					
Models	**Precision (%)**	**MSE (%)**	**F-score (%)**	**Recall (%)**	**MCC (%)**
VGG-19	0.0	0.99	0.0	0.0	0.0
ResNet-50	97.92	0.049	97.04	96.92	96.89
DenseNet-201	94.07	0.1268	92.39	92.50	92.43
Setting – Optimizer: RMSProp - Batch size: 8					
Models	Precision (%)	MSE (%)	F-score (%)	Recall (%)	MCC (%)
VGG-19	0.0	0.99	0.0	0.0	0.0
ResNet-50	96.12	0.078	94.92	95.00	94.95
DenseNet-201	94.30	0.1185	92.50	92.50	92.43

Setting – Optimizer: RMSProp - Batch size: 16					
Models	Precision (%)	MSE (%)	F-score (%)	Recall (%)	MCC (%)
VGG-19	0.0	0.99	0.0	0.0	0.0
ResNet-50	87.95	0.312	81.80	81.34	81.25
DenseNet-201	94.73	0.137	91.45	91.34	91.28

Metrics used to consolidate obtained results confirm that the ResNet-50 model had the best impact on batch size 4, while DenseNet-201 had the best effect on batch sizes 8 and 16 using RMSProp optimizer

Comparative results of occluded and not occluded faces with the best model

We will use the model that provides the best results to observe the effect of occultation on the dataset images.

Table 7: Comparison of the best model on occluded and non-occluded images.

Types	Not occluded	Occluded
Accuracy (%)	**99.62**	**99.58**

We contact here that the occultation has impacted the result.

State-of-the-art comparison

The comparison of our study with the literature methods shows that we perform better using the SGD optimizer with epochs of 4. The table below displays this comparison.

Table 8: Comparison of literature methods with the best results of our study.

Methods	Accuracy (%)
WU *et Al.* (Wu *et al.*, 2019)	98.60
WAN and CHEN (Wang *et al.*, 2020)	93.80
VGG-19 with SGD	98.65
RESNET-50 with SGD	99.23
DENSENET-201 with SGD	99.81

DISCUSSION

Regarding results obtained in different experiments, we got:

For the RMSProp optimizer, we obtained the best results with the ResNet-50 model, which was able to stabilize at a value of 96.92% at all batch sizes, while the DenseNet-201 model saw its results vary between 82.69 to 95.19%;

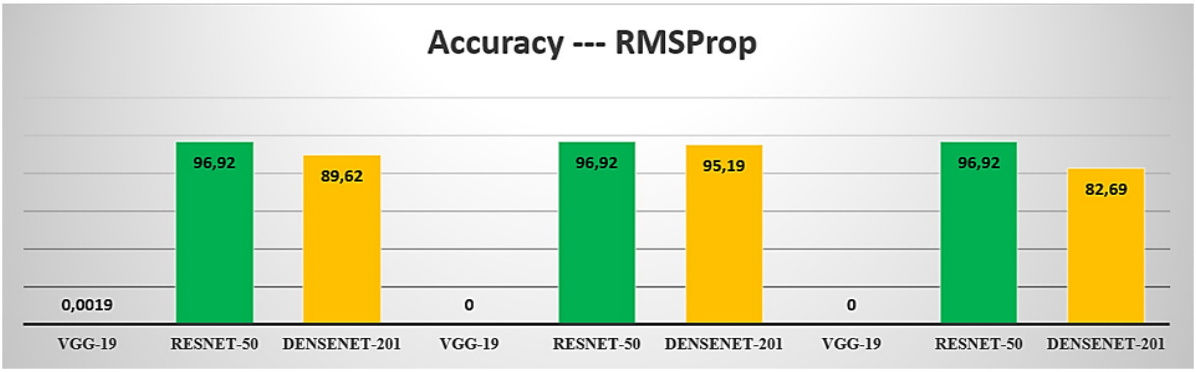


Figure 3: Face recognition accuracy used by RMSProp

finally, VGG-19 could not fit in our experiment with RMSProp optimizer and had a null result. These results are shown in the histogram.

For the Adam optimizer, the ResNet-50 model got the best scores with batch sizes 4 and 8 with respective values of 96.92 and 95%, and with batch size 16, it had a bad performance of 81.35%, while the DenseNet-201 model made a result of 92. 50% at batch sizes 4 and 8, which is less than the performance of ResNet-50 and got a score of 91.35, which is the best performance at batch size 16; finally, the VGG-19 did not fit yet in our experiment with the Adam optimizer and got a null result. The histogram Figure 3 illustrates our analysis

For the SGD optimizer, we observe results of more than 94% for each model, with ResNet-50 obtaining the second-best score with the values 99.23, 98.27, and 94.81 for the respective batch size of 4, 8, and 16; while the DenseNet-201 model outperformed all other models with scores of 99.61, 98.65, and 98.46 for the respective batch sizes of 4, 8, and 16; finally, the VGG-19 model could have good scores with SGD optimizer for the scores of 98.65, 98.27, and 94.81 for the batch sizes of 4, 8, and 16. Also, it could obtain the same results as ResNet-50 at batch sizes 8 and 16, as shown in the histogram Figure 4. SGD optimizer is the most optimal for our study. In the paper, Wu *et Al.* proposed a POOA (positioning the optimal occlusion area) algorithm for solving the occluded face detection problem with the use of a robust principal component analysis method to obtain a result of 98.60% [2], while WAN and CHEN proposed a MaskNet plan coupled with convolutional neural network to get a result of 93.8% on AR face database [29] used in our study. Our study used three convolutional neural network methods for face recognition with occluded. We found

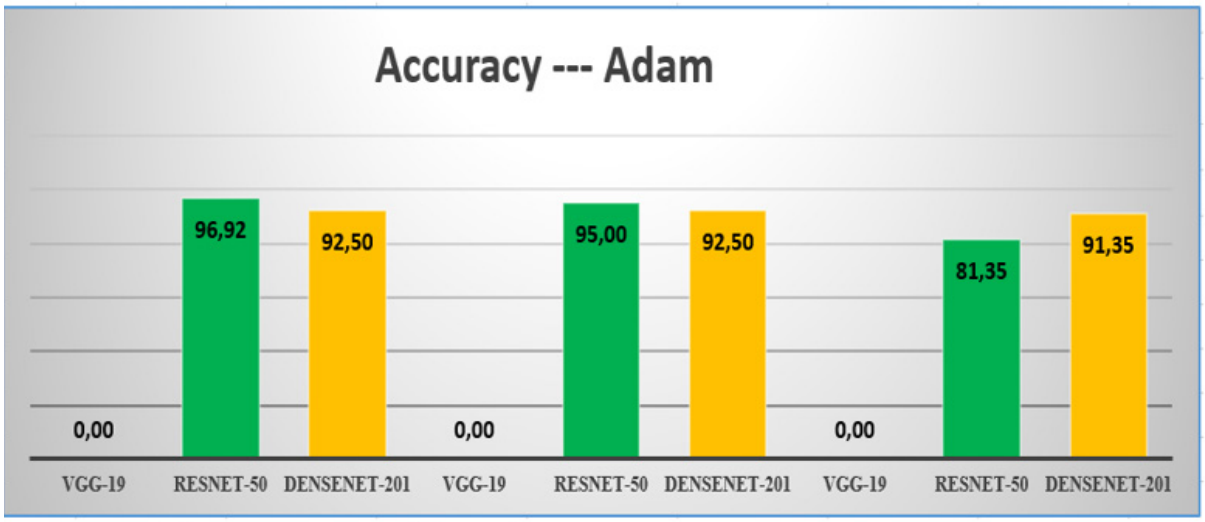


Figure 4: Face recognition accuracy used by Adam

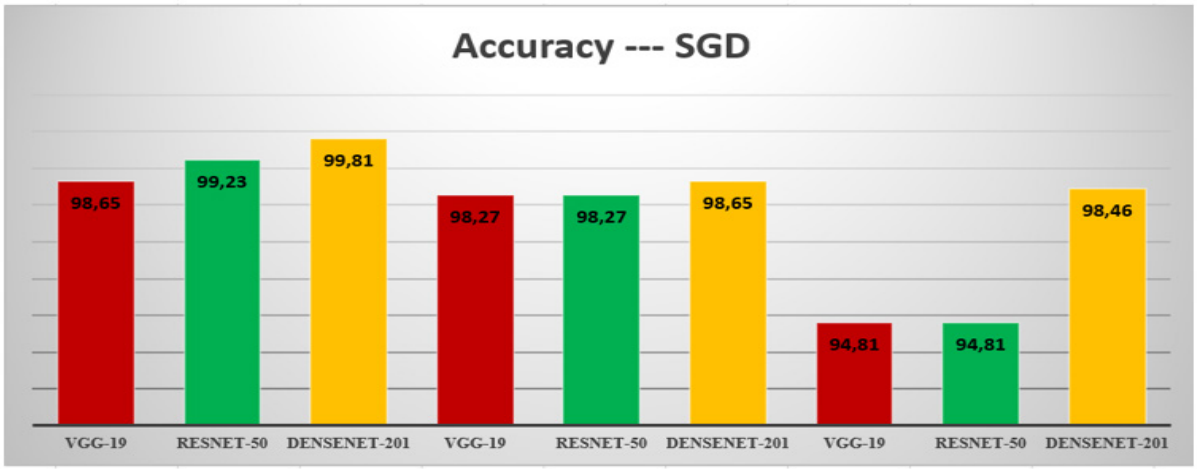


Figure 5: Face recognition accuracy used by Adam

the ResNet-50 and DenseNet-201 models using SGD optimizer and batch size 4 obtained very close results on the test datasets reaching 99.23% and 99.81%, respectively, outperforming the VGG-19 model. Based on the current literature survey results, our study proposes methods to improve the performance of cloaked face identification and recognition with a satisfactory result of 99.81%.

CONCLUSION

From the results obtained in this study, we can conclude: First, a comparison between different models was used to show that the most optimal result was obtained with DenseNet-201 using SGD optimizer with a batch size of 4. We find that VGG-19 failed to adapt to Adam and RMSProp optimizers with its poor results obtained during experiments.

Finally, as a robust security tool, occlusion face recognition can be improved by using pre-trained convolutional neural network models. We can see that results from our study produce better results than studies in the discussion.

Thus, in future studies, we can use other convolutional neural network models using techniques that will allow us to increase data for more accurate results.

REFERENCES

Arnia, F., Saddami, K., & Munadi, K. (2021). DCNet : Noise-Robust Convolutional Neural Networks for Degradation Classification on Ancient Documents. *Journal of Imaging, 7*(7), 114. https://doi.org/10.3390/jimaging7070114

AR Face Database Webpage. (2022). Consulté 10 octobre 2022, à l'adresse. https://www2.ece.ohio-state.edu/~aleix/ARdatabase.html

Artificial neural networks. Pt. 3. (2010). *Springer*.

Bo-Gun Park, Kyoung-Mu Lee, & Sang-Uk Lee. (2005). Face recognition using face-ARG matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(12), 1982-1988. https://doi.org/10.1109/TPAMI.2005.243

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. https://doi.org/10.48550/ARXIV.1512.03385

Idelette Kambi Beli, & Guo, C. (2017). Enhancing Face Identification Using Local Binary Patterns and K-Nearest Neighbors. *Journal of Imaging, 3*(3), 37. https://doi.org/10.3390/jimaging3030037

Jiang, R., Li, C.-T., Crookes, D., Meng, W., & Rosenberger,

C. (2020). Deep biometrics. *Springer*.

Kingma, D. P., & Ba, J. (2014). *Adam : A Method for Stochastic Optimization*. https://doi.org/10.48550/ARXIV.1412.6980

Mantoro, T., Ayu, M. A., & Suhendi. (2018). Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods. 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), 1-5. https://doi.org/10.1109/ICMCS.2018.8525935

Martinez, A. M. (2002). Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(6), 748-763. https://doi.org/10.1109/TPAMI.2002.1008382

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(2), 228-233. https://doi.org/10.1109/34.908974

Min, R., Hadid, A., & Dugelay, J.-L. (2014). Efficient Detection of Occlusion prior to Robust Face Recognition. *The Scientific World Journal*, 1-10. https://doi.org/10.1155/2014/519158

Papers with Code—RMSProp Explained. (s. d.). Retrieved 10 octobre 2022, à l'adresse https://paperswithcode.com/method/rmsprop

Praseetha, V. M., Bayezeed, S., & Vadivel, S. (2019). Secure Fingerprint Authentication Using Deep Learning and Minutiae Verification. *Journal of Intelligent Systems, 29*(1), 1379-1387. https://doi.org/10.1515/jisys-2018-0289

Reddy, A. S. B., & Juliet, D. S. (2019). Transfer Learning with ResNet-50 for Malaria Cell-Image Classification. 2019 *International Conference on Communication and Signal Processing (ICCSP)*, 0945-0949. https://doi.org/10.1109/ICCSP.2019.8697909

Siegmund, D., Fu, B., José-García, A., Salahuddin, A., & Kuijper, A. (2021). Detection of Fiber Defects Using Keypoints and Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence, 35*(05), 2150016.https://doi.org/10.1142/S0218001421500166

Singh, P. K., Kar, A. K., Singh, Y., Kolekar, M. H., & Tanwar, S. (Éds.). (2020). Proceedings of ICRIC 2019 : Recent innovations in computing. Springer.

Team, K. (s. d.). Keras documentation : SGD. Consulté 10 octobre 2022, à l'adresse https://keras.io/api/optimizers/sgd/

Tsai, A.-C., Ou, Y.-Y., Hsu, L.-Y.-C., & Wang, J.-F. (2018). Efficient and Effective Multi-person and Multi-angle Face Recognition based on Deep CNN Architecture. 2018 *International Conference on Orange Technologies (ICOT)*, 1-4. https://doi.org/10.1109/ICOT.2018.8705876

Wang, D., Wang, H., Sun, J., Xin, J., & Luo, Y. (2020). Face Recognition in Complex Unconstrained Environment with An Enhanced WWN Algorithm. *Journal of Intelligent Systems, 30*(1), 18-39. https://doi.org/10.1515/jisys-2019-0114

Wu, G., Tao, J., & Xu, X. (2019). Occluded Face Recognition Based on the Deep Learning. 2019 *Chinese Control And Decision Conference (CCDC)*, 793-797. https://doi.org/10.1109/CCDC.2019.8832330

Zhang, Y.-D., Wang, S.-H., & Liu, S. (Éds.). (2020). *Multimedia technology and enhanced learning : Second EAI international conference, ICMTEL* 2020, Leicester, UK, April 10-11, 2020, proceedings. Part I. Springer.

Zhou, J., Wang, Y., Sun, Z., Jia, Z., Feng, J., Shan, S., Ubul, K., & Guo, Z. (Éds.). (2018). *Biometric recognition : 13th Chinese Conference, CCBR 2018,* Urumchi, China, August 11-12, 2018: proceedings. Springer.