University of Zawia Journal of Educational and Psychological Sciences (UZJEPS)
Volume 13, Issue 1, (2024), pp279-293

INVESTIGATION OF THE DIFFERENCE BETWEEN USING THE SHALLOW CONVOLUTIONAL NEURAL NETWORK MODEL IN THE CASE OF PYTHON AND MATLAB FOR FACE RECOGNITION.

Ahmed Salem Salamh, Assaid Othman Sharoun
Department of Computer - Education of faculty - University of Zawia
EMAIL: a.salamh@zu.edu.ly, asyed.sharoon@zu.edu.ly

Received: *01/04/2024* **Accepted:** *16/04/2024* **Available online:** *30/06/2024* ***DOI***

ABSTRACT

Convolutional neural networks have become strong tools for different computer vision tasks like image classification, object detection, and image segmentation. We carried out a practical study in Matlab and Python to assess the dependability of a shallow model under diverse conditions while considering different dataset sizes and facial recognition challenges. The model was tested and evaluated using the VGG Face model which showed similar performance on all datasets. We utilise the deep learning toolbox in the MATLAB implementation, which offers a user-friendly setting for creating, training, and assessing CNN models. The experiments show how pre-processing, data augmentation and transfer learning techniques can be smoothly incorporated to improve CNN's performance in image classification tasks. We investigate widely used pre-trained models from libraries such as TensorFlow and Torch Vision to model training and enhance classification accuracy. The comparative analysis is implemented based on the capability of use and flexibility of the implementations. In addition, we assess how the CNN models perform on standard datasets, pointing out similarities and discrepancies in training time, accuracy, and resource consumption. This paper introduces good knowledge on researchers, students, and practitioners guidance on the practical aspects of implementing CNNs in MATLAB and Python, assisting in selecting the ideal platform for their requirements in the fields of deep learning and computer vision.

Keywords- Face recognition, deep learning, feature extraction, Shallow CNN, Matlab, Python.

دراسة الفرق بين استخدام نموذج الشبكة العصبية التلافيفية الضحلة في حالة بايثون وماتلاب للتعرف على الوجوه.

أحمد سالم سلامة1، السيد عثمان شعرون2

1، 2 قسم الحاسب الآلي – كلية التربية – جامعة الزاوية

EMAIL: a.salamh@zu.edu.ly, asyed.sharoon@zu.edu.ly

تاريخ الاستلام: 2024/04/01م   تاريخ القبول: 2024/04/16م   تاريخ النشر: 2024/06/30م

الملخص:

أصبحت الشبكات العصبية التلافيفية أدوات قوية لمهام رؤية الحاسوب أوالكمبيوتر المختلفة مثل تصنيف الصور واكتشاف الكائنات وتجزئة الصورة. لقد أجرينا دراسة عملية في Matlab و Python لتقييم موثوقية النموذج الخفيف في ظل ظروف متنوعة مع الأخذ في الاعتبار أحجام مجموعات البيانات المختلفة وتحديات التعرف على الوجه. تم اختبار النموذج وتقييمه باستخدام نموذج VGG Face الذي أظهر أداءً مماثلاً في جميع مجموعات البيانات. نحن نستخدم مجموعة أدوات التعلم العميق في تطبيق MATLAB، والتي توفر إعدادًا سهل الاستخدام لإنشاء نماذجCNN وتدريبها وتقييمها. تُظهر التجارب كيف يمكن دمج تقنيات المعالجة المسبقة وزيادة البيانات ونقل التعلم بسلاسة لتحسين أداء CNN في مهام تصنيف الصور. نحن نتحقق من النماذج المدربة مسبقًا المستخدمة على نطاق واسع من المكتبات مثل TensorFlow و Torch Vision لنمذجة التدريب وتعزيز دقة التصنيف. يتم تنفيذ التحليل المقارن على أساس القدرة على الاستخدام ومرونة التطبيقات. بالإضافة إلى ذلك، نقوم بتقييم كيفية أداء نماذج الشبكات العصبية التلافيفية على مجموعات البيانات القياسية، مع الإشارة إلى أوجه التشابه والاختلاف في وقت التدريب والدقة واستهلاك الموارد. تقدم هذه الورقة معرفة جيدة لتوجيهات الباحثين والطلاب والممارسين حول الجوانب العملية لتطبيق الشبكات العصبية التلافيفية في MATLAB و Python، مما يساعد في اختيار النظام الأساسي المثالي لمتطلباتهم في مجالات التعلم العميق والرؤية الحاسوبية.

الكلمات المفتاحية – التعرف على الوجه، التعلم العميق، استخراج الميزات، ، ماتلاب، بايثون.

1–Introduction

The implementation of MATLAB and Python are powerful environments in which scientists, engineers, and data scientists prefer to test machine learning. MATLAB, made by MathWorks, is perfect at operations such as math, analyzing data, and creating algorithms. MATLAB is easy to use and has lots of powerful tools for image processing, controlling systems, and teaching machines. Python is known for being easy to read and being able to achieve lots of different objects. Python is implemented for websites, data sciences, and artificial intelligence field. Python has special libraries like NumPy, pandas, sci-kit-learn, and TensorFlow that help to process data.

While advancements have been made in the area of recognition, there are still challenges to address in tasks involving detecting and recognizing faces [1]. The limitation includes head position, size, lighting changes, and background separation. Recognition of faces is a popular method of verifying identity in computer vision applications. [2]. After the face detection and alignment the recognition takes place by using different methods for feature extraction. The most famous technique comes under traditional and deep learning techniques. The traditional methods well known as handcraft feature extraction, such as Local binary pattern [3][4], Histogram of oriented gradients [5], Gabor filter [6] and Scale Invariant Feature Transform [7] used for extracting discriminative features. The techniques were being robust and efficient for facial recognition. These techniques were modified and developed many times [8] for example, LBP modified for more efficient in case of face recognition [9].

Convolution neural networks (CNN) are among the most common techni ques for solving

computer vision problems throughout the last decade[10]. When it comes to image classification, Convolutional Neural Networks are the most crucial aspect of machine vision. CNN has achieved significant advancements in classification, segmentation, and recognition[11][12].Deep neural networkstake more resources for computing, wi th the exception of the great achievement of the prior deep learning techniques [13].

In many image-based applications, e.g., image classification, face recognition, fine-grained image categorization and depth estimation, (CNNs) have achieved state-of-the-art efficiency. CNNs have the benefit of automatically extracting almost all low-level and high level deeper features rather than creating handcrafted features descriptor compared to conventional visual recognition techniques. CNNs have transformed the computer vision community due to these strong characteristics [14].

Usually, enormous volumes of convolutions and parameters consume high computing costs and more processing space, limiting their implementations to resource-constrained real-world devices [15]. there are many CNN architectures, deep model such as GoogLeNet [16], ResNet[17] and SqueezeNet[18] and light. The deep learning model requires computational cost which is disadvantage of these models. The researchers in advanced in this field were interested to introduce CNN structure that less deep than others which available with less computational resources.

The model based on deep convolutional networks for large-scale image recognition [19] and also relying on recently study and experiments to investigate the effect of filter size and number of filters in CNN[20]. The VGG designed the deep network and achieved comparable state of the art results different datasets. [21] the deep face architecture which consisted of 13 convolution layers with filter size 3x3 and each layer, followed by a rectifier linear unit (ReLu). 5 max pooling with a stride of two and 3 fully connected layers. The goal of this paper is to create a simple CNN design that can be used with small, medium, or large datasets, building off of the existing VGG16 architecture.

The upcoming sections will detail the suggested CNN model: next sections discusses the proposed Shallow -CNN architectures, covers results, discussion, performance evaluation, and comparison, and concludes on CNN architectures.

1. Shallow -CNN Model for Face Recognition

This Shallow CNN model based on VGG16 is suitable for different datasets while considering the constraints of computational resources. The proposed architecture relies on the VGG16 structure as is shown in Table 1 as we tested the technique [22] in previous work. We retained the initial layer with dimensions 224X224x3 pixels modified the filter size to 7X7 as opposed to 3X3 with a stride of 4 and eliminated the second layer consisting of 64 filters. This results in a reduction in the size of the vector in the subsequent layer, enabling the CNN to identify important features using 64 filters. The subsequent two layers operate with 55X55 input and 3X3 filters, each having 256 filters. Afterward, perform max pooling with a 3X3 size and a stride of 2. The fourth convolution layer takes in a 27X27 input with a 3X3 filter and 512 filters, then performs max pooling with a 3X3 size and stride of 2. The final result has dimensions of 6 by 6. In the end, we retain the three final fully connected layers featuring 4096 neurons and two dropouts, identical to those in VGG Face. Each dropout was set to 0.1 and every convolution layer had batch normalization and a rectifier linear unit (ReLU) to speed up processing and enhance stability[23][24] for the model.

Table 1. Convolutional neural network model for face Recognition[22].

Layer name	Input size	Filter size	# Filters	Window size for pooling	Stride	Padding	Output size	# Feature maps
Conv1	224 × 224	7 × 7	64	−	4	0	55	64
Conv2	55*55	3*3	256	−	1	1	55	256
Conv3	55*55	3*3	256	−	1	1	55	256
Pooling1	55*55	−	−	3*3	2	0	27	256
Conv4	27*27	3*3	512	−	2	1	14	512
Pooling2	14*14	−	−	3*3	2	0	6	512
Fully_Connected1	4096−neurons							
Dropout	0.2							
Fully_Connected1	4096−neurons							
Dropout	0.2							
Fully_Connected3	N classes							
Softmax								

2. Performance Evaluation

The performance evaluation and compare the effectiveness of the proposed CNN, the following metrics were utilized, Original 'P' for the number of positives, while 'N' denotes the number of negatives. 'T' indicates true, and 'F' indicates false. In our analysis, Recall, Precision, and F1_score are seen as different facets of accuracy. Accuracy (Acc).

$$\mathrm{Acc} = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})} \qquad (1)$$

1. Recall

$$\mathrm{Recall} = \frac{(\mathrm{TP})}{(\mathrm{TP} + \mathrm{FN})} \qquad (2)$$

1. Precision

$$\mathrm{TNR} = \frac{(\mathrm{TP})}{(\mathrm{FP} + \mathrm{TP})} \qquad (3)$$

2. F1 score:

$$\mathrm{F1}_{\mathrm{score}} = \frac{(2\mathrm{TP})}{(2\mathrm{TP}+\mathrm{FP}+\mathrm{FN})} \qquad (4)$$

3. Results And Discussion

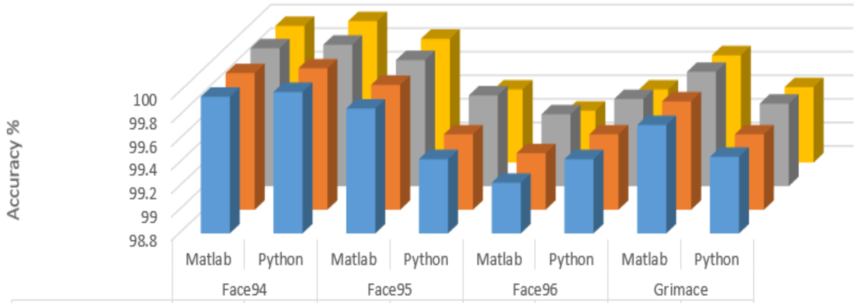
3.1. Face recognition datasets

a well-chosen dataset reflects the diversity and complexity of real-world scenarios in which the face recognition model is expected to operate. real datasets assist CNN learn features and patterns that are relevant to various environmental conditions, lighting conditions, facial expressions, and demographics. The dataset serves as the training ground for CNN to generalize its learning beyond the specific examples it has seen. A diverse dataset helps the model become robust to variations not present in the training set.

CNN The ability of the model to recognize faces in various locations depends on generalization that is crucial for the evaluation of unseen faces. larger datasets frequently result in greater model performance and better generalization. Researchers and developers can evaluate how their CNN model works compared with current state-of-the-art models by using appropriate datasets. The face recognition systems is evaluate by the careful selection of datasets, which improves the model's capacity to generalise, minimises biases, and guarantees its applicability to a wide range of real-world scenarios. The sources of the datasets were [25] There are four sets of datasets: faces94, faces95, faces96 and grimace. All the databases have a combined total population of 395 people including 20 women and men aged between 18 and 20. Faces94 is composed of 152 individuals with twenty female ones, one hundred and thirteen male ones, and twenty others that are males showing slight differences in their head orientation against a green background. Face95 consists of seventy-two subjects whose head sizes differ significantly while located at the red curtain backdrop. The face database Face96 includes one hundred and fifty-one people with diverse lighting conditions, as well as backgrounds; Grimace has eighteen males alongside eighteen females all under varying lighting conditions.

3.2. Performance Evaluation and Comparison

Data sets are composed of various subjects, each having 20 samples representing a specific label. The dataset was divided into two parts, training and testing set by 5 fold cross validation. We had chosen five pictures of each person to test and another set of images for training and validation. This is the standard way of evaluating the performance of two models; namely, the proposed model and VGG16 model. It was trained on Stochastic Gradient Descent (SGD) optimizer with momentum 0.9 over both models having run it for 22 epochs. Different datasets have different learning rates depending on the batch sizes used in them. We tested this model in different settings with changing sizes of datasets to observe its behavior in such cases. Prior to proceeding with advanced experiments, it has been realized that there is not much need for data preparation here. Python has given a perfect accuracy to Face94, 99.43% for Face95, Face96 at 99.43%, Grimace at 99.45% from test sets shown in figure 1 respectively as seen in table III below. Clearly recall, precision and F1 score were excellent Matlab attained overall accuracies of 99.96%, 99.85%, 99.23% and 99.72%. Our experiments showed a small difference between Matlab and Python because each environment has unique characteristics.



	Face94 Matlab	Face94 Python	Face95 Matlab	Face95 Python	Face96 Matlab	Face96 Python	Grimace Matlab	Grimace Python
Accuracy	99.96	100	99.86	99.43	99.23	99.43	99.72	99.45
Recall	99.96	100	99.86	99.44	99.28	99.44	99.72	99.44
Precision	99.97	100	99.87	99.57	99.41	99.54	99.77	99.5
F1_score	99.96	100	99.85	99.42	99.24	99.42	99.71	99.44

Figure 1. Assessment of matrices in facial recognition

The VGG face model was used to assess the accuracy of facial recognition at the end. 5-fold cross validation was applied in carrying out an experimentation assessment. The method involved pre-trained VGG face model and transfer learning that used similar training and testing set division protocol for all datasets. Figure 2 shows that the proposed CNN model achieved almost similar accuracy across all datasets, despite less complexity than VGG face. Most of the results show no major variations.

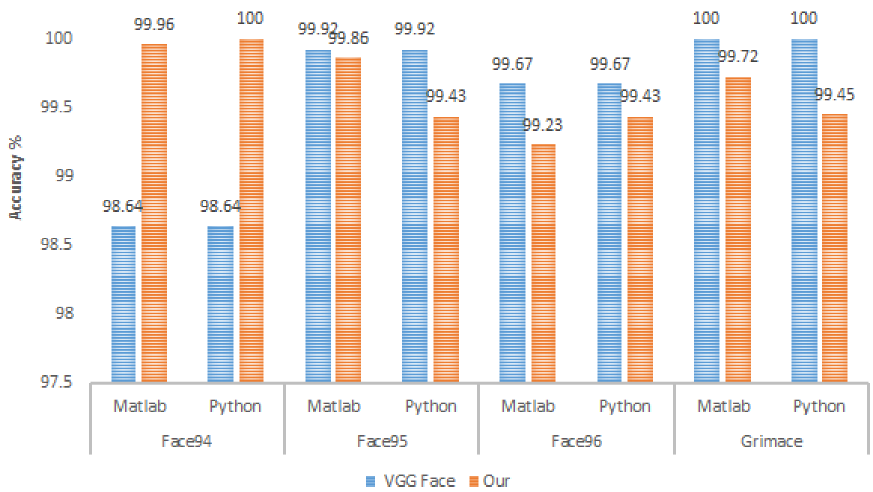


Figure 2. Comparison of performance between proposed and VGG face models on four datasets.

To continue with our experiment based on the previous section, we had a thorough evaluation of accuracy versus filters' number and size. The lastly achieved accuracy is determined by filter sizes as well as the number of filters in the first layer. This explains why the size 7X7 was chosen for its initial filtering stage as seen in table 2.

Table 2. Demonstrates how the accuracy of the miniature-CNN model varies with varying numbers of filters and filter sizes in the initial convolution layer.

Proposed CNN	Filter size	Number of filters	Accuracy %
CNN 1	5x5	32/64/128	99.48/ 98.84 /99.29
CNN 2	7x7	32/64/128	99.36/ 99.04/ 99.23
CNN 3	9x9	32/64/128	98.78/ 98.91 /98.97
CNN 4	11x11	32/64/128	98.72/ 99.17 /99.29

4. Conclusion

MATLAB and Python stand as two versatile and powerful programming languages, each bringing unique strengths to the machine learning field. MATLAB, with its concept in numerical computing, offers an intuitive and user-friendly environment, making it exceptionally well-suited for engineering, scientific research, and algorithm development. The extensive set of built-in functions, toolboxes, and Simulink for dynamic system modeling contribute to its prominence in academia and industry. On the other hand, Python's readability, flexibility, and expansive ecosystem have propelled it to the forefront of diverse domains, including web development, data science, machine learning, and artificial intelligence. The vast array of libraries such as NumPy, Pandas, Scikit-Learn, and TensorFlow empowers Python developers to tackle a wide range of tasks with efficiency and scalability. Both languages have played pivotal roles in advancing technology, with MATLAB excelling in specialised domains and Python serving as a general-purpose language with broad applicability. The choice between MATLAB and Python often hinges on the specific needs of a project, the preferences of the user, and the collaborative nature of the development environment. Ultimately, the synergy between MATLAB and Python is increasingly evident, as seen in interoperability tools and cross-language integration. As technology continues to evolve, the coexistence of MATLAB and Python provides a diverse and powerful toolkit for researchers, engineers, and developers, fostering innovation and progress across many disciplines. Whether it's MATLAB's matrix-centric approach or Python's readability and versatility, both languages contribute significantly to the ever-expanding landscape of computational tools and continue to shape the future of scientific and technological advancements.

It is clear that shallow CNN and data preparation lead to significant results rather than training a very deep network, which needs more resources and computational time as well as transfer learning. The model was developed and tested using Matlab and Python to evaluate its performance under various dataset conditions. From the experiments, Matlab and Python give approximately good results. There is a significant difference in the Bach normalization layer between two environments related to differences in configurations that should be fine-tuned.

References

[1] A. B. S. Salamh and H. I. Akyüz, “A Novel Feature Extraction Descriptor for Face Recognition,” Eng. Technol. Appl. Sci. Res., vol. 12, no. 1, pp. 8033–8038, Feb. 2022, doi: 10.48084/etasr.4624.

[2] Z. Pei, H. Xu, Y. Zhang, M. Guo, and Y. Yee-Hong, “Face recognition via deep learning using data augmentation based on orthogonal experiments,” Electron., vol. 8, no. 10, pp. 1–16, 2019, doi: 10.3390/electronics8101088.

[3] L. Zhou, H. Wang, S. Lin, S. Hao, and Z.-M. Lu, “Face recognition based on local binary pattern and improved Pairwise-constrained Multiple Metric Learning,” Multimed. Tools Appl., vol. 79, no. 1–2, pp. 675–691, Jan. 2020, doi: 10.1007/s11042-019-08157-0.

[4] T. Lindahl, “Study of Local Binary Patterns,” Sci. Technol., p. 3, 2007.

[5] C. H. Hima Bindu and K. Manjunatha Chari, “Face Recognition Using Improved Co-HOG Features,” 2020, pp. 647–655.

[6] T. M. Abhishree, J. Latha, K. Manikantan, and S. Ramachandran, “Face Recognition Using Gabor Filter Based Feature Extraction with Anisotropic Diffusion as a Pre-processing Technique,” Procedia Comput. Sci., vol. 45, pp. 312–321, 2015, doi: 10.1016/j.procs.2015.03.149.

[7] T. Lindeberg, “Scale Invariant Feature Transform,” Scholarpedia, vol. 7, no. 5, p. 10491, 2012, doi: 10.4249/scholarpedia.10491.

[8] M. Ghorbani, A. T. Targhi, and M. M. Dehshibi, “HOG and LBP: Towards a robust face recognition system,” 10th Int. Conf. Digit. Inf. Manag. ICDIM 2015, no. Icdim, pp. 138–141, 2016, doi: 10.1109/ICDIM.2015.7381860.

[9] B. Ahuja and V. P. Vishwakarma, “Local Binary Pattern Based Feature Extraction with KELM for Face Identification,” in 2020 6th International Conference on Signal Processing and Communication (ICSC), Mar. 2020, pp. 91–95, doi: 10.1109/ICSC48311.2020.9182760.

[10] X. Wu, R. He, Z. Sun, and T. Tan, “A light CNN for deep face representation with noisy labels,” IEEE Trans. Inf. Forensics Secur., vol. 13, no. 11, pp. 2884–2896, 2018, doi: 10.1109/TIFS.2018.2833032.

[11] A. B. S. Salamh, A. A. Salamah, and Halil Ibrahim Akyüz, “A Study of a New Technique of the CT Scan View and Disease Classification Protocol Based on Level Challenges in Cases of Coronavirus Disease,” Radiol. Res. Pract., vol. 2021, pp. 1–9, 2021, doi: 10.1155/2021/5554408.

[12] A. B. S. Salamh and H. Akyüz, "New Deep Learning Model for Face Recognition and Registration in Distance Learning," Int. J. Emerg. Technol. Learn., vol. 17, no. 12, pp. 29–41, Jun. 2022, doi: 10.3991/ijet.v17i12.30377.
[13] A. B. Perdana and A. Prahara, "Face Recognition Using Light-Convolutional Neural Networks Based on Modified Vgg16 Model," 2019 Int. Conf. Comput. Sci. Inf. Technol. ICoSNIKOM 2019, pp. 14–17, 2019, doi: 10.1109/ICoSNIKOM48755.2019.9111481.
[14] M. Zhang, S. Khan, and H. Yan, "Deep eigen-filters for face recognition: Feature representation via unsupervised multi-structure filter learning," Pattern Recognit., vol. 100, p. 107176, 2020, doi: 10.1016/j.patcog.2019.107176.
[15] C. Tian et al., "Lightweight image super-resolution with enhanced CNN," Knowledge-Based Syst., vol. 205, p. 106235, Oct. 2020, doi: 10.1016/j.knosys.2020.106235.
[16] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, "Going Deeper with Convolutions Christian," J. Chem. Technol. Biotechnol., vol. 91, no. 8, pp. 2322–2330, 2016, doi: 10.1002/jctb.4820.
[17] V. Sangeetha and K. J. R. Prasad, "Deep Residual Learning for Image Recognition," Indian J. Chem. - Sect. B Org. Med. Chem., vol. 45, no. 8, pp. 1951–1954, 2006, doi: 10.1002/chin.200650130.
[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," Feb. 2016, [Online]. Available: http://arxiv.org/abs/1602.07360.
[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1-14, 2015.
[20] W. S. Ahmed and A. amir A. Karim, "The Impact of Filter Size and Number of Filters on Classification Accuracy in CNN," in 2020 International Conference on Computer Science and Software Engineering (CSASE), Apr. 2020, pp. 88–93, doi: 10.1109/CSASE48920.2020.9142089.

[21] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep Face Recognition,” no. Section 3, pp. 41.1-41.12, 2015, doi: 10.5244/c.29.41.

[22] A. B. S. SALAMH and H. İ. AKYÜZ, “FEN BİLİMLERİ ENST İ TÜSÜ,” KASTAMONU ÜNİVERSİTESİ, 2023.

[23] S. Ioffe and C. Szegedy, “Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift.”

[24] S. Santurkar, D. Tsipras, and A. Ilyas, “How Does Batch Normalization Help Optimization ?,” no. NeurIPS, 2018.

[25] U. University of Essex, “Description of the Collection of Facial Images.” .