Finger Type Classification with Deep Convolution Neural Networks

Yousif Ahmed Al-Wajih[1], Waleed M. Hamanah[2,*][a], Mohammad A. Abido[2,3,4,][b], Fouad AL-Sunni[1] and Fakhraddin Alwajih[5]

[1]*Control and Instrumentation Engineering Department at KFUPM, Dhahran, 31261, Saudi Arabia*
[2]*Interdisciplinary Research Center in Renewable Energy and Power Systems (IRC-REPS), KFUPM, Saudi Arabia*
[3]*Department of Electrical Engineering at KFUPM, Dhahran, 31261, Saudi Arabia*
[4]*K. A. CARE, Energy Research & Innovation Center (ERIC), KFUPM, Saudi Arabia*
[5]*Faculty of Computers and Artificial Intelligence Cairo University, Giza, Egypt*

Keywords: Artificial Intelligence, Deep Learning, Fingerprint Identification, Convolutional Neural Network.

Abstract: The Automated Fingerprint Identification System (AFIS) is a biometric identification methodology that uses digital imaging technology to obtain, store, and analyse fingerprint information. There has been an increased interest in fingerprint-based security systems with the rise in demand for collecting demographic data through security applications. Reliable and highly secure, these systems are used to identify people using the unique biometric information of fingerprints. In this work, a learning-based method of identifying fingerprints was investigated. Using deep learning tools, the performance of the AFIS in terms of search time and speed of matching between fingerprint databases was successfully enhanced. A convolutional neural network (CNN) model was proposed and developed to classify fingerprints and predict fingerprint types. The proposed classification system is a novel approach that classifies fingerprints based on figure type. Two public datasets were used to train and evaluate the proposed CNN model. The proposed model achieved high validation accuracy with both databases, with an overall accuracy in predicting fingerprint types at around 94%.

1 INTRODUCTION

Biometric information encompasses a set of unique and measurable physical characteristics, including a person's fingerprints and particular facial features, as well as one's voice and handwriting. Each person's fingerprints are formed of unique shapes and curves that remain unchanged during a person's lifetime. Hence, fingerprinting can quickly identify and authenticate a person efficiently. Due to its evident reliability in accurately identifying persons, biometric information has become the focus of researchers and companies specialized in protection technology. Fingerprints are now being extensively used as a simple means of authentication on smartphones and other mobile devices.

A fingerprint is a biometric method utilized to identify people and authenticate identities. Unique features are extractable from the surface of each fingerprint (Bian et al. 2019) and (Rani et al., 2019). Many biometric techniques have been devised for fingerprint recognition and identification using the ridges and greyscale images. This work emphasized using a deep learning algorithm and testing its ability to perform this task. Fingerprints identification methods have conventionally outperformed other biometrics methods, such as face and speech recognition, being well-established, reliable, and robust (Minaee et al., 2019) and (Chaitra et al., 2021). In this area, fingerprint orientation field estimation typically improves the performance of automated fingerprint identification systems.

Fingerprint identification systems typically encompass fingerprint imaging, acquisition, preprocessing, and feature extraction matching. A significant number of studies focus on various aspects of fingerprinting identification systems, including selection and extraction of optimized features as well as different proposed methods of matching (Valdes et al., 2019)

[a] https://orcid.org/0000-0002-5911-7364
[b] https://orcid.org/0000-0001-5292-6938
[*] Corresponding Author

Al-Wajih, Y., Hamanah, W., Abido, M., Al-Sunni, F. and Alwajih, F.
Finger Type Classification with Deep Convolution Neural Networks.
DOI: 10.5220/0011327100003271
In *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2022)*, pages 247-254
ISBN: 978-989-758-585-2; ISSN: 2184-2809

and (Srivastava et al., 2022). The old technique was utilized based on deep learning in order to distinguish four classes (arch, tented arch, left loop, right loop, and whorl), using the Galton-Henry classification in (Shea, 2009 and Srivastava et al., 2021).

Fingerprints and facial features are presently the most thoroughly studied biometric indicators, allowing for reliable recognition in various applications. There has been a growing need for more accurate and reliable biometric identification and authentication-based models from smartphones to border control. Recently, researchers have been able to enhance the robustness of recognition and identification models by incorporating deep learning (Ribeiro et al., 2018) and (Ayan et al., 202). In the following segments, we review a number of the most recent related studies.

The authors in (Stojanovic et al., 2017) reviewed recent methods in identifying latent fingerprints and compared the most recent minutia descriptors. They reported that selecting a good minutia would result in improved accuracy of the AFIS. Their work detailed the various minutia descriptors that could be used in automatic fingerprint feature extraction and compared them in terms of identification rates. They proposed that the minutia descriptor C&J - which relies on deep learning algorithms - be the new focus of research in the area of latent fingerprint identification. Furthermore, the authors recommended that new minutia descriptors based on deep learning be developed and the identification accuracy be studied. In particular, they recommended conducting studies to identify the best minutia descriptors to enhance the performance of AFISs.

In (Preetha and Sheela, 2018), the authors suggested that understanding the advantages and limitations of the fingerprint orientation field estimation methods is of fundamental importance to creating fingerprint identification. According to the authors, a common misunderstanding is that automatic fingerprint identification had not been appropriately addressed, despite AFIS being a subject of research for decades. They explained that fingerprint identification remains a significant pattern identification dilemma of interest to researchers due to the large intra-class mutability and inter-class relationships in fingerprint patterns. They stressed that automatic fingerprint identification systems typically attempt to ascertain reliable matching features from fingerprint images of inferior quality or latent images, ‘damaged’ and ‘defects’ such as scars, dirt, grease, and/or moisture on the surface of fingertips. In (Cao and Jain, 2015), the authors concluded that learning-based methods based on deep learning had significantly improved the performance of fingerprint orientation field estimation systems, especially when dealing with challenges that traditional methods had typically failed to tackle, including latent fingerprints (such as poor-quality fingerprints). They summarized the limitations of conventional techniques as follows: 1) the initial orientation fields are typically unreliable; 2) relying primarily on high-quality fingerprints, their algorithms may fail to handle latent and poor-quality fingerprints; 3) human intervention during the process of algorithm execution may be required; 4) high computational complexity of such approaches. In (Cao and Jain, 2019), the authors studied using a CNN in running the fingerprint estimation algorithm by modeling orientation field estimations of a poor-quality image patch as a classification mission. They classified the latent patch as one of a set of illustrative orientation patterns using a CNN. The CNN was able to learn the input images' characteristic features directly. The authors concluded that fingerprint identification estimated through a CNN would result in higher accuracy than dictionary-based methods. Schuch et al. 2017 trained CNNs as regression networks to assess a fingerprint orientation field. They called this proposed model a ConvNetOF. The most recent work done in fingerprint classification using the DL method is reported in (Michelsanti et al., 2017), (Peralta et al., 2018), and (Zia et al., 019). DL-based methods have been recognized as powerful tools in the classification field (Lecun et al0., 2015). Despite the fact that the wide use of DL approaches in image classification, there remains a research gap with regards to their use in fingerprint classification. In that regard, the early work on this field was started by authors in (Shea, 2009), (Wang et al., 2016), and (Kakadiaris et al. 2009).

The most recent works on fingerprint classification with new deep learning techniques are considered in (Michelsanti et al., 2017); two pre-trained CNN models (VGG) were evaluated using the National Institute of Standards and Technologies (NIST) SD4 dataset. The proposed models were compared in terms of fast feature extraction. The authors showed that DL-based methods outperformed other methods due to their learning ability from the row data. In addition, a deep CNN (DCNN) was used, and the reported accuracy stood between 88.9% and 90% with the same NIST SD4 dataset 0(Peralta et al., 2018). Further in (Zia et al., 2019), the authors proposed a baseline DCNN model, and the reported accuracy stood between 92.2% and 96.1% with the NIST SD4 dataset. In addition, the authors reported the high robustness of the proposed model. In (Blanco et al., 2020), basic and modified extreme learning machines (ELMs) were tested for their efficacies concerning fingerprint classification. The authors showed that the improved ELM had outperformed the other CNN models in terms of training speed and computational cost.

Furthermore, the authors reported that the enhanced ELM was able to handle data with the unbalanced class distribution. They said the accuracy of 95%. They concluded that the weighted ELM had achieved better results in terms of accuracy and penetration rate metrics. In (Iloanusi and Ejiogu, 2020), the work focused on classifying input fingerprint grayscale images according to the gender of the person being identified. The authors reported an overall accuracy rate of 91.3% in the classification. In this study, a 20-layer CNN model was used. The model used was built from the ground up. They employed both a Sokoto Coventry Fingerprint Dataset (SOCOFing) dataset and their dataset for training and testing. All previous work focused on typically studied categories of the old four classes (arch, tented arch, left loop, right loop, and whorl). The paper focuses on fingerprint type. Therefore, in this brief, labeling the datasets and utilizing the state-of-the-art deep learning technique with CNN structure is conducted to classify a fingerprint type. As presented in the literature review, and to the best of our knowledge, no work has previously tackled finger type classification, which marks this work's novelty.

In this study, the proposed new classifier was designed to identify fingerprints as either thumb or non-thumb. Such classification will improve the matching time and the accuracy of AFIS. The data have been labeled in the two-class. Then, training and validation of the data have been applied. Deep learning was used to classify the gray image of fingerprint. A model of CNN was applied using the benchmarked dataset. The proposed DL model is used to help the matching algorithm verify the input fingerprint more expediently, as it would require the matching algorithm to search on half of the database.

This paper is organized as follows: In Section 2, the data preparation with the proposed structure is involved. The CNN architecture model is presented and discussed in Section 3. Then, Section 4 presents the experimental results for Thumb CNN (TCNN) model. Finally, a conclusion is derived in Section 6.

2 PROPOSED SCHEME AND DATA PREPARATION

The proposed model in this work is used to classify the fingerprint image based on the finger type, as shown in Figure 1. First, the dataset has is prepared and labeled into the target classes. The labeled dataset is separated into training and validation sets. A part of the dataset is reserved for measuring the performance of the proposed model. Second, the deep learning model based on CNN structure is investigated. Then, the proposed model is trained and tuned in order to achieve high classification accuracy. Finally, the unseen dataset is used to test the model performance.

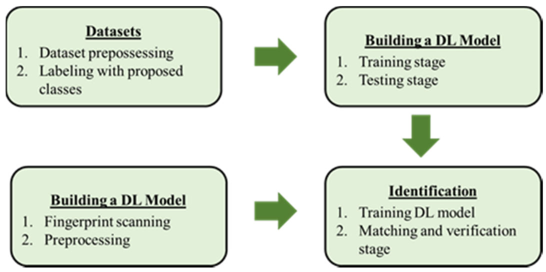


Figure 1: CNN Proposed scheme.

Two benchmark datasets were used to evaluate the performance of our proposed model. The first one was the NIST dataset (NIST, Biometric Special Databases and Software, 2022). This database featured 4000 8-bit grayscale 512x512 pixel fingerprint images at the time of the study. The dataset was collected randomly and stored in PNG format. This dataset has been widely used in testing and developing automated fingerprint classification systems. The original database was classified into five categories (L = left loop, W whirl, R = right loop, T = tented arch, and A = arch). Subsequently, the dataset is reorganized to match the newly proposed classes. The naming scheme of the PNG files was done such that the two numbers after the underscore indicate the finger type and from the hand from which the fingerprint image was taken. For example, in the file labeled 'f0001_01.png,' the '01' after the underscore indicates that the fingerprint belongs to a right thumb (Karu and Jain, 1996)0. The dataset was divided into five files, one for each finger type. For the thumb classifier, the same dataset was utilized. In this case, thumb data was used, and the non-thumb is collected randomly from the index, middle, ring, and little fingers. Thumb sample data included 700 samples, 600 of which were set for training and 100 for validation and testing. An equal number of 175 samples were randomly selected from each class (index, middle, ring, and little fingers). Data in this model was divided into 85% for training and 15% for validation. Samples from the datasets are illustrated in Figure 2.

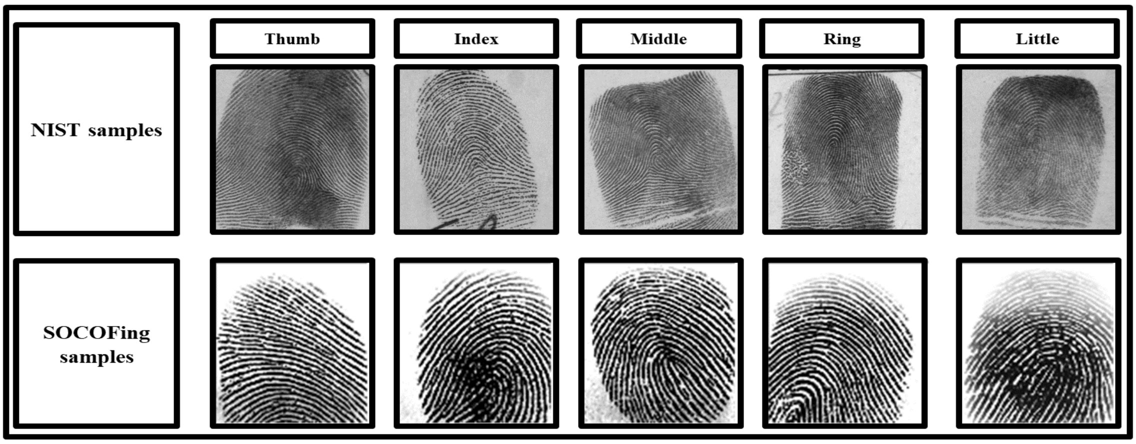


Figure 2: Samples from the Datasets.

The second dataset used to train and test our proposed models was the SOCOFing dataset (Shehu et al., 2019). The study consisted of actual fingerprints taken from 600 subjects at the time of the study. Images were labeled according to the exclusive attributes of gender, hand, and finger type. The real part of this dataset was used for the purposes of this study. The dataset was divided into subclasses in order to create both thumb and other finger-type models. For the thumb model, the dataset was clustered into two classes of a total of 1200 thumb fingerprint images of the BMP format. The non-thumb class consisted of 300 images from the index, middle, ring, and little fingers, for 1200 images. Of this dataset, 75% was used for training and 25% for validation.

3 CNN ARCHITECTURE

As discussed in the previous sections, the study was to classify fingerprint data into subclasses. Our approach in this work was to use a learning-based method with a supervised learning methodology. To this end, a deep learning technique was utilized to achieve the classification target. As a state-of-the-art model of deep learning and machine learning, CNN was determined as ideal for classification tasks of image-based data (Shyu et al., 2020). The architecture of the thumb CNN (TCNN) model used in this work is described in the coming subsection.

A CNN model was developed to train the classification model. The model consisted of four convolutional layers, each of which was followed by a max-pooling layer. Filters in the four convolutions numbered 256, 128, 64, and 32, respectively. Type 3X3 filters and a ReLU activation were used. The application of consecutive convolutional and max-pooling layers resulted in tensors of size (6, 6, 32), which were flattened to size (1,152). Two dense layers 128 and 20 neurons in size were then added. The fully connected layers were supplied with ReLU and Softmax activations consecutively. We used a dropout layer between the two fully connected layers with a drop rate of 30%. For training, we used cross-entropy as the loss and an Adam optimizer in (Shyu et al., 2020) for the backpropagation algorithm. Keras with TensorFlow backend was used to create and train the CNN model. The model summary is shown in Figure 3.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 126, 126, 256)     7168
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 63, 63, 256)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 63, 63, 256)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 61, 61, 128)       295040
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 30, 30, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 30, 30, 128)       0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 28, 28, 64)        73792
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
dropout_3 (Dropout)          (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 12, 12, 32)        18464
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 6, 6, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1152)              0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               147584
_________________________________________________________________
dense_4 (Dense)              (None, 128)               16512
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 129
=================================================================
Total params: 558,689
Trainable params: 558,689
Non-trainable params: 0
_________________________________________________________________
```

Figure 3: TCNN Model Summery.

4 EXPERIMENTS AND RESULTS

The proposed technique has been tested based on the preparation data, and all experiments were conducted using Python with TensorFlow and Keras libraries. Training the models was conducted using Google Collaboratory GPU resources. The results of the training and testing TCNN model will be detailed in the subsequent sections.

4.1 Training TCNN Model

Input data of both datasets used for the purposes of training and testing are summarized in Table 1. The NIST D4 dataset was randomly distributed into 70% for training, 15% for validation, and 15% for testing. At the same time, SOCOFig datasets were randomly distributed into 75% for training, 15% for verification, and 10% for testing. Before training the model, data augmentation was employed and tuned in order to increase accuracy and prevent overfitting. Images used in training rotated within 20°, shifted right and left within 10%, with image shearing and zoom within 10%, and with horizontal flipping—the use of the aforementioned augmentation technique allowed for the enhancement of all models. The best accuracy and loss metrics results stood at a 97% validation accuracy, a 0.13 validation loss with the NIST D4 dataset, a 96% validation accuracy, and a 0.1 validation loss with the SOCOFing dataset. Results are detailed in Table 1; also, Figure 4 and Figure 5 illustrate the training performed on the mentioned dataset.

Table 1: Summary of The Two Datasets.

Dataset	Training	Validation	Test
NIST SD4	498 thumbs 498 not-thumb • 125 indexes • 125 middles • 124 rings • 124 little	102 thumbs 102 not-thumb • 26 indexes • 26 middles • 25 rings • 25 little	100 thumbs 100 not-thumb • 25 indexes • 25 middles • 25 rings • 25 little
SOCOFing	900 thumbs 900 not-thumb • 225 indexes • 225 middles • 225 rings • 125 little	180 thumbs 180 not-thumb • 45 indexes • 45 middles • 45 rings • 45 little	120 thumbs 120 not-thumb • 30 indexes • 30 middles • 30 rings • 30 little

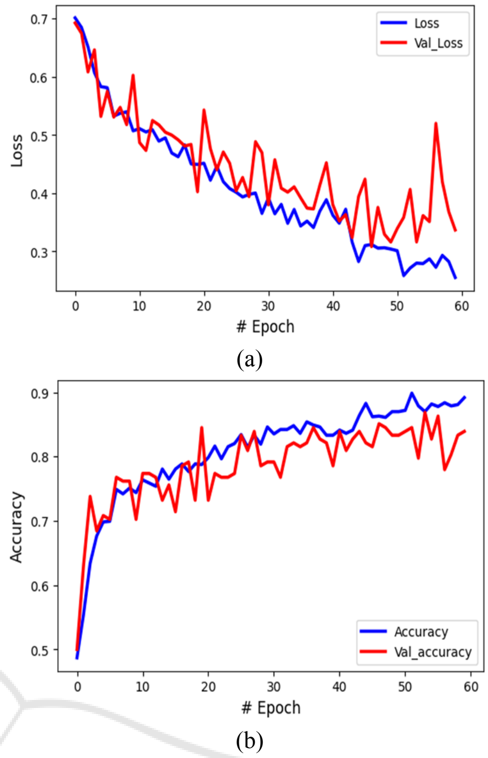


Figure 4: Results of the TCNN Model based on NIST SD4: (a) model loss and (b) model accuracy.

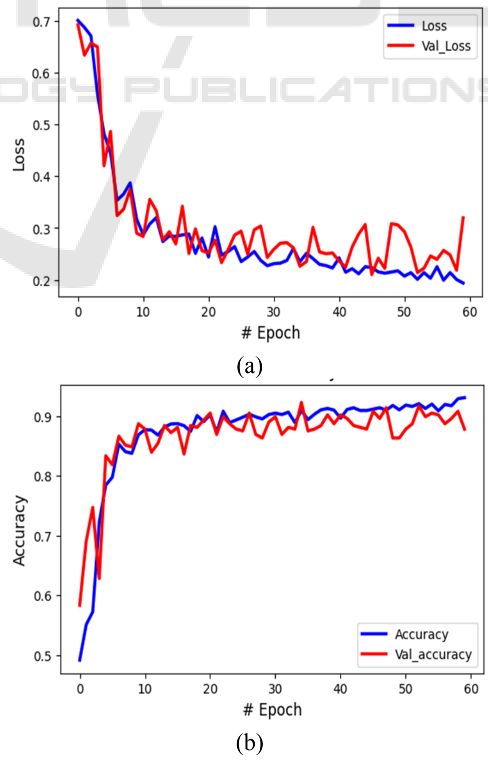


Figure 5: Results of the TCNN Model based on SOCOFing: (a) model loss and (b) model accuracy.

Table 2: Training of TCNN Model.

Dataset	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
NIST D4	89.43	0.23	86.90	0.21
SOCOFing	91.97	0.18	92.26	0.21

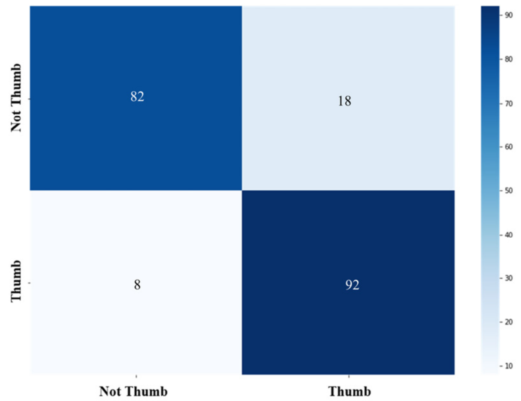
Table 3: Matric Results for Test Set.

Metric	NIST D4	SOCOFing
Accuracy	90.00%	89.00%
Precision	95.28%	83.63%
Recall	84.16%	92.00%
F1-score	89.38%	87.61%

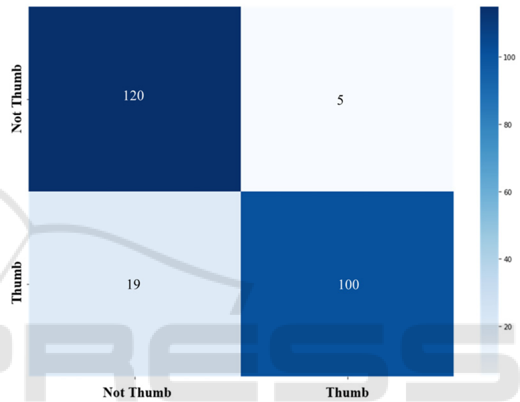
4.2 Testing CNN Model

Loss and accuracy results of both the training and validation sets are illustrated in Figure 4 and Figure 5. Accuracy results of the validation set are summarized in Table 2. Regularization induced by the dropout layer allowed for more extended training of the model and reduced the possibility of overfitting. Table 3 shows the unseen test dataset's accuracy, precision, recall, and F1 scores. Notably, accuracy is decent for a classification problem. Other metrics indicated that predictions were somewhat uniform across the different classes. In order to check how our model had performed concerning individual classes, we used confusion matrices. Each matrix showed the correctly classified samples in the diagonal, according to class; it also gave an insight into what classes are confused by the model. Our model performed superiorly in terms of differentiation between classes, as shown in the diagonal of the confusion matrix in Figure 6.

To the best of our knowledge, there is no work in the literature tackling the classification of the fingerprint image to the finger type (thumb or not thumb). However, there are some works have been done on the same dataset with different problems which are not comparable with our proposed work. Table 4 summarizes the best result achieved in the literature of three different field and features. The best accuracy achieved in classifying fingerprints to Galton-Henry classification (arch, tented arch, left loop, right loop, and whorl) is 95.05 % (Michelsanti et al. 2017). The best accuracy reached in assigning gender (Male or Female) from fingerprints image is 91.3% (Iloanusi and Ejiogu, 2020). The accuracy achieved in classifying that fingerprint is for a right hand or left hand is 96.80% (Kim et al., 2020) where this accuracy is a validation accuracy during the training process not a test accuracy on an unseen dataset.



(a)



(b)

Figure 6: Confusion Matric (a) SOCOFing test set. (b) NIST D4 test set.

Table 4: Tackled PROBLEMS IN Literature.

Tackled Problem	Accuracy (%)
classifying fingerprints into arch, tented arch, left loop, right loop, and whorl (Michelsanti et al. 2017).	95.05
Gender classification (Iloanusi and Ejiogu, 2020)	91.30
Left- or Right-Hand Classification (Kim et al., 2020)	96.80
Finger Type Classification	90.00

5 CONCLUSION

A novel approach for classifying fingerprints based on the finger type was introduced through this work. Results of implementation and experimentation indicated that the TCNN model performed superiorly, with a high accuracy rate of fingerprint type classification. The deep learning technique evidently aided in the proper extraction and classification of fingerprints. The developed model was trained and evaluated using two datasets, NIST and SOCOFing. The main metrics considered in this work, commonly considered in studies of DL/CNN architecture, were chosen to best reflect the level of performance in terms of classification and features extraction. The proposed model was able to classify the type of the fingerprint with the accuracies of 90% and 89% with the NIST D4 and SOCOFing datasets, respectively.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support provided by King Fahd University of Petroleum & Minerals. The authors also acknowledge the support by KACARE Energy Research & Innovation Center (ERIC) at KFUPM.

REFERENCES

Bian W., Xu D., Li Q., Cheng Y., Jie B., and Ding X., (2019). A Survey of the methods on fingerprint orientation field estimation, IEEE Access, vol. 7, pp. 32644–32663.

Rani S., Kumar P., (2019). Deep Learning Based Sentiment Analysis Using Convolution Neural Network, *Arab J Sci Eng* 44, 3305–3314. https://doi.org/10.1007/s13369-018-3500-z

Minaee S., Abdolrashidi A., Su H., Bennamoun M., and Zhang D., (2019). Biometric Recognition Using Deep Learning: A Survey, no. December.

Chaitra Y. L., Dinesh R., Gopalakrishna M. T., Prakash B. V., (2021). Deep-CNNTL: Text Localization from Natural Scene Images Using Deep Convolution Neural Network with Transfer Learning, *Arab J Sci Eng*. https://doi.org/10.1007/s13369-021-06309-9.

Valdes-Ramirez D. et al., (2019). A Review of Fingerprint Feature Representations and Their Applications for Latent Fingerprint Identification: Trends and Evaluation, IEEE Access, vol. 7, pp. 48484–48499.

Srivastava, Hari M., Khaled M. Saad, and Walid M. Hamanah, (2022), Certain New Models of the Multi-Space Fractal-Fractional Kuramoto-Sivashinsky and Korteweg-de Vries Equations" Mathematics 10, no. 7: 1089. https://doi.org/10.3390/math10071089.

Shea J. J., (2009). Handbook of fingerprint recognition [Book Review], vol. 20, no. 5.

Srivastava, Hari M., Abedel-Karrem N. Alomari, Khaled M. Saad, and Waleed M. Hamanah, (2021), Some Dynamical Models Involving Fractional-Order Derivatives with the Mittag-Leffler Type Kernels and Their Applications Based upon the Legendre Spectral Collocation Method" Fractal and Fractional 5, no. 3: 131. https://doi.org/10.3390/fractalfract5030131.

Ribeiro J. Pinto, Cardoso J. S., and Lourenco A., (2018). Evolution, current challenges, and future possibilities in ECG Biometrics," IEEE Access, vol. 6, pp. 34746–34776.

Ayan E., Karabulut B., Ünver H. M., (2021). Diagnosis of Pediatric Pneumonia with Ensemble of Deep Convolutional Neural Networks in Chest X-Ray Images, *Arab J Sci Eng*. https://doi.org/10.1007/s13369-021-06127-z.

Stojanovic B., Marques O., and Neskovic A., (2017). Latent overlapped fingerprint separation: a review, Multimed. Tools Appl., vol. 76, no. 15, pp. 16263–16290.

Preetha S. and Sheela S. V., (2018). Selection and extraction of optimized feature set from fingerprint biometrics-a review, Proc. 2nd Int. Conf. Green Comput. Internet Things, ICGCIoT 2018, pp. 500–503.

Cao K. and Jain A. K., (2015). Latent orientation field estimation via convolutional neural network, Proc. 2015 Int. Conf. Biometrics, ICB 2015, pp. 349–356.

Cao K. and Jain A. K., (2019). Automated Latent Fingerprint Recognition, IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 4, pp. 788–800.

Schuch P., Schulz S. D., and Busch C., (2017) Convnet regression for fingerprint orientations, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10269 LNCS, pp. 325–336.

Michelsanti D., Ene A. D., Guichi Y., Stef R., Nasrollahi K., and Moeslund T. B., 92107). Fast fingerprint classification with deep neural networks, VISIGRAPP 2017 - Proc. 12th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl., vol. 5, no. Visigrapp, pp. 202–209.

Peralta D., Triguero I., Garcia S., Saeys Y., Benitez J. M., and Herrera F., (2018). On the use of convolutional neural networks for robust classification of multiple fingerprint captures, Int. J. Intell. Syst., vol. 33, no. 1, pp. 213–230.

Zia T., Ghafoor M., Tariq S. A., and Taj I. A., (2019). Robust fingerprint classification with Bayesian convolutional networks, IET Image Process., vol. 13, no. 8, pp. 1280–1288.

Lecun Y., Bengio Y., and Hinton G., (2015). Deep learning, Nature, vol. 521, no. 7553, pp. 436–444.

Wang R., Han C., and Guo T., (2016) A novel fingerprint classification method based on deep learning, Proc. - Int. Conf. Pattern Recognit., vol. 0, pp. 931–936.

Kakadiaris I. A., Passalis G., Toderici G., Perakis T., and Theoharis T., (2009). Face Recognition, 3D-Based, Encycl. Biometrics, pp. 329–338.

Blanco D., Mora M., Barrientos R. J., Hernandez-García R., and Naranjo-Torres J., (2020). Fingerprint classification through standard and weighted extreme learning machines, Appl. Sci., vol. 10, no. 12.

Iloanusi O. N. and Ejiogu U. C., (2020). Gender classification from fused multi-fingerprint types, Inf. Secur. J., vol. 29, no. 5, pp. 209–219.

NIST, (2022). Biometric Special Databases and Software | NIST," [Online]. Available: https://www.nist.gov/itl/iad/image-group/resources/biometric-special-databases-and-software.

Karu K. and Jain A. K., (1996). Fingerprint classification, Pattern Recognit., vol. 29, no. 3, pp. 389–404.

Shehu Y. I., Ruiz-Garcia A., Palade V., and James A., (2019). Detailed Identification of Fingerprints Using Convolutional Neural Networks, Proc. - 17th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2018, pp. 1161–1165.

Shyu M., Chen S., and Iyengar S. S., (2020). a Survey on Deep Learning Techniques, Strad Res., vol. 7, no. 8.

Kim, Junseob, Beanbonyka Rim, Nak-Jun Sung, and Min Hong. (2020). Left or right hand classification from fingerprint images using a deep neural network, CMC-Computers Materials & Continua, v.63, no.1, pp.17 – 30.