Neural Computing and Applications (2023) 35:6471–6497
https://doi.org/10.1007/s00521-022-07894-y

ORIGINAL ARTICLE

Improving automated latent fingerprint detection and segmentation using deep convolutional neural network

Megha Chhabra[1] · Kiran Kumar Ravulakollu[2] · Manoj Kumar[3] · Abhay Sharma[4] · Anand Nayyar[5]

Received: 6 April 2022 / Accepted: 23 September 2022 / Published online: 22 November 2022

Abstract

Latent fingerprint segmentation is a complex process of separating relevant areas called fingerprints from an irrelevant background in the latent fingerprint image which is of poor quality. A breakthrough in the field can be used to segment fingerprints accurately from the background by using optimal resources. Processing of unwanted background of the entire image can lead to false and missed detection of fingerprints. An early fingerprint distinction technique based on colour and saliency masks is proposed to detect potentially relevant areas out of the entire image area for further processing, using a non-learning approach. Later, the patches of early detected fingermarks are fed to a stacked convolutional autoencoder for separating imposters of fingerprint(s) region from relevant fingerprint(s) regions, using a deep learning approach. The inspiration to use the convolutional neural network in this hybrid approach is to effectively capture feature distinction from potential features similar to that of object detection and classification. The inspiration to use autoencoder in a stack is to provide better feature engineering for CNN. The use of the pre-trained convolutional neural network with a stack of autoencoders for image classification and segmentation produces better results than a naive convolutional neural network. The experiments are conducted on the IIIT-D database. The efficiency and effectiveness of the model over good quality images is evaluated by experimenting over different patch sizes, with and without the use of dropout in CNN, with and without use of Autoencoder with CNN. The early detection of contours along with patch-based classification-cum-segmentation using SCAE on good quality images produces 98.45% segmentation accuracy.

Keywords Convolutional neural network · Segmentation · Classification · Latent fingerprints · Autoencoder

1 Introduction

Fingerprint matching has provided a measure of verification and authentication for centuries. The practice started with document authentication first in China and is still in practice worldwide. The identification and verification of fingerprints are established on the fact that these are unique for an individual [15]. The ridge and pore patterns can be transferred to surfaces. The contact residue in the form of oils, skin, moisture, etc., can be transformed on surfaces of various kinds. The captured data is stored for an indefinite time as an image form. This facilitates manual annotation-

✉ Anand Nayyar
anandnayyar@duytan.edu.vn

Megha Chhabra
Megha.chhbr@gmail.com

Kiran Kumar Ravulakollu
kiran.r@woxsen.edu.in

Manoj Kumar
wss.manojkumar@gmail.com

Abhay Sharma
abhaysharma2004@gmail.com

[1] Department of Computer Science & Engineering, School of Engineering & Technology, Sharda University, Gr. Noida, UP 201310, India

[2] School of Technology, Woxsen University, Hyderabad, Telangana, India

[3] Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, Dubai Knowledge Park, Dubai, UAE

[4] Department of Computer science and engineering, Amity University Rajasthan, Jaipur 303002, India

[5] Graduate School, Faculty of Information Technology, Duy Tan University, Da Nang 550000, Viet Nam

based fingerprint matching or automated fingerprint matching against fingerprints in the archives. The representation is captured via ink methods or on optical scanners, etc. The process of evaluating the match observes multitudes of applications. The state-of-the-art applications include biometric fingerprint spoofing [45], fingerprint-based forensic document identification [66], gender identification based on fingerprint information [25] and gender, height and position identification based on fingerprint information [22], identification of fingerprint information alteration [51], etc.

Fingerprints are used for a person's identification. Personal identification based on computer-based biometric authentication has opened a wide area of research and development related to fingerprint processing [21]. The need for lesser human expert intervention and "lights-out" automation is the urge of the time. It is only fair to establish the fact that fingerprint processing and verification are used as a biometric modality for accurate and good-quality results. For that reason, the quality of fingerprints captured significantly impacts the outcome [43]. The fingerprint captured using offline or online live scanning methods can be used in government ID proofs, passports, border crossing patrol, banking, etc., followed by well-established protocols that ensure resulting captured fingerprints are of good quality for future use in fingerprint verification [60]. In the scenario where crime is involved, the matching of latent fingerprints (latents, poor-quality prints, not visible to the naked eye), requires a good match score with database prints[4]. If the quality of stored data is good, the outcome is reliable; otherwise, poor matching can impact the quality of forensics in a bad light. The poor quality of images captured has overlapping of foreground relevant ridge pattern and noisy irrelevant background, making the process of segmentation and detection (subsequently for matching) a difficult task.

Fingerprint identification, a task required to match a suspect with the fingerprints stored in the police database, has received mass attention in research and Automated Fingerprint Identification Service (AFIS) [46] has received overwhelming accuracy rate in the recent decade. The nature of fingerprints found on crime scenes is latent. The process needs to be accurately performed by improving tasks involved in processing if captured fingerprints are of poor quality. The latent fingerprint segmentation lies around image segmentation [9, 36, 39] and image classification [55, 62].

1.1 Latent fingerprint segmentation

Accuracy of segmentation is essential as it affects the reliable extraction of minutiae in the following processing steps. The prominent issue with the latent fingerprint image (LFI) is the capturing of the image with probe correctly from the crime scene under the challenges of the presence of structured noise, poor-quality ridge structure, varieties of lighting, and poor visibility adding to poor visual saliency, etc. [32]. Figure 1 demonstrates the causes of poor-quality images of latent fingerprint databases. Here, the images highlighted are extracted from the IIIT-D latent database.

With huge amount of databases with law enforcement agencies, the scrutiny of data for suspect identification is becoming a large-scale research industry. With such a high level of the task, it is sane to use a fully automated latent fingerprint segmentation and detection system as part of the identification system. As a step in that direction, the proposed work introduces Stacked Convolutional Auto-Encoder (SCAE)-based efficient model to separate latent fingerprints from background complexities. The database used in the process is IIIT-D CLF [53]. The state-of-the-art literature has seen supervised as well as unsupervised techniques with pre-feature and post-feature extraction tasks involved to achieve maximum accuracy with an overall efficient model. A few use patch size as a standardised parameter as a pre-feature extraction task, whereas a few have used full image-based classification and segmentation techniques. Patch-based technique ignores the neighbourhood relationships and is subject to experimentation when patch size is to be decided.

In addition to that, full image-based techniques have established a ground of discussion that patch-based

Fig. 1 Sample images from IIIT-D database displaying various causes of the poor quality of the images

techniques consume more time to process each patch into the model and due to its rigid patch policy, are unaware of what is being processed, consequently, do not process multiple instances efficiently. On the other hand side, full-image-based segmentation techniques do not guarantee the extraction of useful features without pre-processing of images and mostly lead to a relatively larger amount of resource utilization.

The challenges addressed in the proposed work and the contributions of the paper are:

(1) The understanding and comparison of state-of-the-art latent fingerprint image (LFI) segmentation techniques.
(2) The detection-cum-segmentation of LFI using Stack of Convolutional Autoencoder. The detection is performed by generating contours based on colour and saliency masks, thereby reducing the amount of irrelevant information for wholesome processing and detection of single, multiple and partial fingerprints. The classification-cum-segmentation is provided by feeding these CoIs to SCAE.
(3) Parameter stabilization and improvements in CNN architecture using dropout (absent vs present with values 0.25 and 0.1), with Autoencoder and Without Autoencoder in stack and repeatability of results using cross-validation.
(4) The evaluation of results of SCAE and CNN using performance metrics Segmentation Accuracy (SA), Missed Detection Rate (MDR) and False Detection Rate (FDR) [49].

Further, the paper is structured as follows: Sect. 2 highlights the literature review. Section 3 elaborates proposed work. Section 4 elaborates the experimental results and analysis along with the comparative evaluation of the proposed work with state-of-the-art published work. The final section, sect. 5.

2 Related work

The latest trend in fingerprint biometric identification has taken a major leap globally as well as in the Indian market. The fingerprint market provides an open field for research to improve security and authentication-based applications. The fingerprint market is divided into two types based on the type of fingerprints, i.e. Patent or Latent. From historians to astrologers, from biometrics security systems to criminal investigators, the vast majority of the market applications perform scientific and systematic study and experimentation on fingerprints. The identification of sculptures in the temples and monuments, and reading one's future are major contributions of early applications of fingerprint analysis. The modern era uses this biometric with or without other biometric models for access providing to secure systems for instance our offices, fingerprinted authenticated biometric payment cards, banking systems for avoiding online fraud and efficient audit trails, identification of unknown deceased or disaster victim, identification of cold case unidentified postmortem fingerprint cases, identifying and matching fingerprints from crime scenes, use of fingerprint probabilities in courtroom [10, 14, 26].

Despite the advantages, biometric systems raise several issues and social dilemmas. Biometrics is unique, but it is not a secret once fed to systems online. This cannot be cancelled; hence, it always exists. This promotes cross-matching by anyone without the owner's consent; hence, anyone can track individuals without their consent and the privacy and security of biometrics-based data are questionable[15]. Nonetheless, the applications are based on the quality [27] of the fingerprint captured. Based on capture type, fingerprints are classified into four categories (a) indented or moulded fingerprints, (b) patent fingerprints, (c) live-scan fingerprints, (d) latent fingerprints [11, 35].

A latent fingerprint is a category found as unintentional fingerprint on surfaces not visible to the naked eye. These are found on contaminated hard, curved, etc., surfaces. These can be captured using alternate light sources such as UV sensors, special powders, chemical reagents, etc. With the advancement of technology, harder challenges are advancing in the identification of fingerprints in this uncontrolled scan such as latent fingerprints. The desired requirement of researching enhancing algorithms for latent fingerprint forensics leads to the in-depth analysis of processes of the concerned forensics and improves the outcome.

Crime scene investigation captures the live scene fingerprint images and processes these images to match with the fingerprints available in the database for criminal or victim identification. The entire process may not be effectively attempted manually. The matching [46] a) is prone to human error, and b) time-consuming due to the exemplar database. However, AFIS is a solution. Active research is invited in the context. To move to build such as system, the requirement is to understand the underlying tasks required to produce the outcome and later understand the challenges or scope to improve the design, consequently putting a proposal in the perspective.

2.1 Latent fingerprint segmentation challenges

Latent fingerprints, based on the amount of noise, are classified as good, bad and ugly with (a) single and noisy fingerprints, (b) multiple and clean fingerprints, (c) no fingerprints and (d) single and clean fingerprints with the

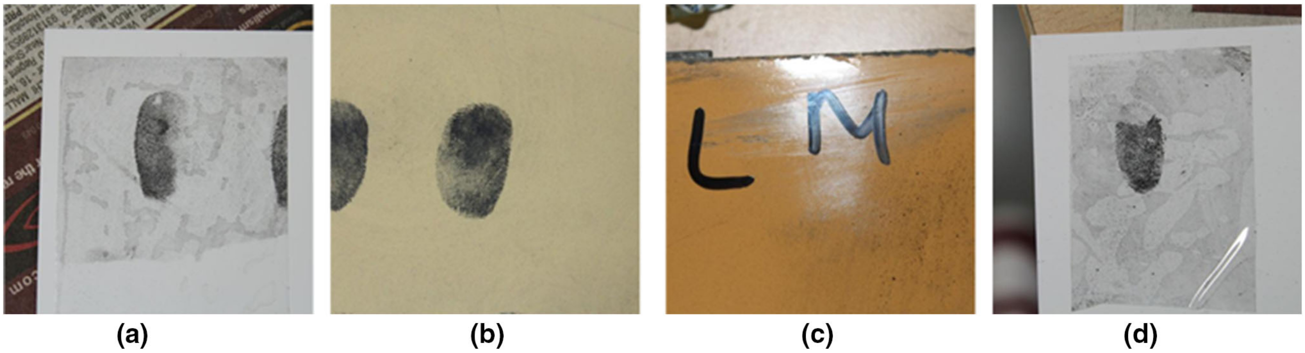


Fig. 2 Sample images from IIIT-D database divided into different categories, **a** single and noisy fingerprint image, **b** multiple and clean fingerprints image, **c** no fingerprints, only noise, **d** single and clean fingerprint image

partial presence of fingerprints, respectively, in Fig. 2. Segmentation is the process applied to extract ridge-like structures out of the noisy background. Level 1, level 2 and level 3 along with other extended features [1] are extracted, and the outcome is the unique feature map for identification. This map, if extracted with a lesser error rate than manual effort, would match the candidate in the available database with administration agencies for crime analysis. In general, manual experts annotate the fingermarks based on the feature map extracted. An Integrated Automated Fingerprint Identification System (IAFIS) [35] matches the annotated LFI with the background database of ten prints and finds top k probable matches which involves human intervention. The ultimate goal is to develop a fully automated system that is intended to reduce the difficulties of time consumption, automation of feature extraction and quality of fingerprint segmented for feature map extraction.

The challenges in latent fingerprint segmentation are majorly due to the quality of the image. The image is noisy; hence, the presence of structured noise, unstructured noise, smudging of the print, incomplete or partial print, and overlapping [54] of the prints may occur. The other issues may include ageing of the prints, defect in the lifting device or mechanism, etc. The challenges are addressed by various authors and multiple algorithms are designed to overcome or suppress the problems arising due to various challenges and enhance performance ultimately.

2.2 Colour-based segmentation

A common influencing factor on fingerprint segmentation from LFI is background interference. Diverse background information is irrelevant to the matching system and hence can be segmented out [11]. Therefore, more robust features can be designed to extract discriminating visual representation. The discriminating behaviour to make the system look at the relevant area of information is called saliency detection. The saliency in the image can be captured with low-level features for instance colour [38] or ridge texture, orientation or higher-level features learned via deep learning techniques [13]. Considering a similar principle in LFIs, colour maps can help in the early distinction of the salient region of interest from the irrelevant background.

Salience region detection using salience masks is a popular area of research lately [50]. Saliency map detection is used in applications such as person re-identification [69], computer games , object enhancement [38]. LFIs usually contain multiple instances of fingerprints, thus shifting attention to more than one fingerprint could not be put at risk. Saliency detection using colour information is another popular research area, [33] used as a patch-based approach with deep belief networks with sliding windows and colour matching to classify pedestrian patches. Saliency-based reliable patch identification is performed on colour-based outcome classified patches in the first stage. Other methods such as Visual Attention Retargeting [38] proposed optimal colour management of the foreground. The gap was to reproduce the original colour post-saliency detection. The proposed solution is inspired by the study of these mentioned approaching, thereby suggesting colour-based saliency detection to classify foreground and reserve original colour.

Deep Learning is termed as mapping of input to output based on how the model maps it and how well it learns to map without any interference. The advancement of deep learning and its variants such as autoencoder [3, 23] performs better using colour-based information and saliency detection [2] in terms of reducing MDR and FDR [12]. Hence, deep learning-based relevant feature learning is aligned along with colour-based visual distinction for better LFI segmentation and detection. With the applications of deep learning in consideration to improve performance in the field of latent fingerprint segmentation, the proposed solution is an SCAE to classify and segment patches of CoIs extracted using two mask techniques on LFI.

Table 1 State-of-the-art review of existing techniques of latent fingerprint forensics

Year	References	Method	Dataset	System type	Image handling	Performance	Review
2015	[7]	Coarse and fine structure ridge dictionary learning.	1. NIST SD 27 2. WVU	Learning-based	Full image	Accuracy	1. Local ridge or valley structure does not optimize the task. 2. Poor performance with noisy data.
2017	[47]	Salient feature extraction and random forest classifier based classification.	1.NIST SD 27 2.IIIT-D CLF	Supervised Learning	Local Patches	Accuracy, BSA, FSA, SA	1. Reason to choose a patch size was missing. 2. Choice of classifier not justified. 3. The generic approach of feature selection for different databases to extract optimal features is missing.
2017	[18]	Deep learning using RBM based patch classification	1.NIST SD 27 2.IIIT-D CLF	Deep-Learning	Local Patches	Accuracy, MDR and FDR	1.Classifier stability dependency. 2.The reason to choose a patch size was missing.3. The reason to choose DNN with RBM was missing.
2018	[42]	Non-patch, FCN and object detection based fusion	1.NIST SD 27 2. WVU	Deep-Learning	Full image	Accuracy, MDR, FDR and IoU	1. Lower accuracy with full image-based segmentation over the need of patch-based approach is not addressed.
2019	[28]	Patch-based segmentation using CNN	1. NIST SD 27 2. WVU 3. IIIT-D (good images only)	Deep-Learning	Local Patches	Accuracy, MDR and FDR	1. The reason to choose patch size, classifier, the architecture of classifier is missing.

2.3 Latent fingerprint segmentation approaches

Table 1 summarises the identified detection and segmentation techniques for latent fingerprint segmentation. Pixel-wise features alone are better with rolled prints in comparison with latent fingerprints. The comparison is based on non-learning and non-learning systems.

Cao and Jain [7] used coarse and fine structure ridge dictionary learning for sparse representation of the ridge structures. Unlike previous methods to find features from the input image, the learning system was used to learn patch classification based on the learned ridge. Based on the learning system classification, the local feature approach obtained better accuracy. The performance was measured using NIST SD 27 and WVU databases [53], and experimental results observed segmentation accuracy of 61.24% and 70.16%, respectively. The approach was lacking flexibility in terms of the quality of images as it relies on learned dictionary quality and convex hull to get a smooth mask. The application of the approach can be generalized more likely in image representation and less likely in fingerprint segmentation of latent fingerprints.

Sankaran et al. [47] used a set of features, prominently salient features [32, 61, 69]. The features were extracted based on saliency, image intensity, gradient, ridge and quality of the image from local patches. The method used a supervised learning technique along with feature selection. Improved RELIEF algorithm is used for feature selection. Experiments were performed on all features and optimally selected features and salient features. The proposed method used Random Decision Forest (RDF)[44] as a classifier for labelled features. The database used was NIST SD 27 and IIIT-D CLF. The performance metrics were segmentation accuracy, foreground accuracy and background accuracy, Rank-50 accuracy of 83% and 93.23%, respectively, on NIST SD and IIIT-D CLF. The process includes addressing the issue of feature selection and identifying the best set of features. Since images are used to divide the image into patches, not all patches are substantial; hence, the accuracy is impacted due to the use of irrelevant and relevant information both for classification with the major class as noise in some images.

The pre-deep network era in latent fingerprint segmentation focused on local feature-based non-learning techniques for segmentation or learning techniques-based pixel classification-cum-segmentation. The common features used for segmentation of latent prints are ridge orientation or frequency from an image or learned dictionary or Image mean, variance and in some cases gradient-based features, saliency-based features, image quality-based features, etc., of an image or block/patch of image fed to the supervised classifiers such as RDF, SVM, AdaBoost [68], etc., to predict the class labels. The CNNs [16, 32, 69] in the literature are used for various tasks including object detection, segmentation and object labelling, etc. The success behind DCNNs [9, 30, 63, 67] is the reason for researchers to explore deeper into feature engineering. Deep neural networks are used for various applications [23, 63]. In the present application area, DNN is recommended for latent fingerprint feature extraction and classification of patches into binary classes.

Deep neural network such as RNN is preferred in sequential data in comparison with hierarchical data since the ability to define boundaries is weak in such data. Up-to-date deep architectures used specifically for segmentation [48, 58] perform learning of mapping low-resolution image representation to pixel-level prediction. The basic hierarchy is to input images to the input layer, followed by training using an encoder. Later, the decoder may be used to represent feature extraction, learning and selection to achieve classification and feedback for cost adjustment. Multiple authors used reinforcement schemes for classification. A learning method such as RBM is used to learn features from training data. In comparison with RNN, RBM performs better feature extraction from training data. Although RBM can be trained with generative learning, hence boosting unlabeled data sample feature learning easier, the feature selection is not optimal. Ezeobiejesi and Bhanu [18] divides the image into patches and feeded the patches into an RBM-based learning model. Once the feature extraction, learning and selection are performed, RBM was fine-tuned with a single-layer perceptron.

The performance was measured using MDR and FDR. The RBM-based classification system provided 1.25% MDR and 0.04% FDR on NIST SD 27 along with 1.35% MDR and 0.54% FDR on the IIIT D CLF database. The value was reported for WVU as well which is almost similar and 0.6% FDR along with 1.6% MDR. No segmentation accuracy was specifically mentioned for individual databases. Also, the use of RBM over available resources is missing.

The deep neural networks are dependent and sensitive to the number of training iterations used. The entire dataset is fed to the model. Based on the train-to-test ratio, the training data is randomly chosen and fed to the network. The amount of information in latent images is imbalanced. If the data of the entire image is fed to the model, maybe as patches or full image the accuracy may reduce due to noise. The time taken to process an entire noisy image or patches of the same image is huge in large volume datasets. Nguyen et al. [42] used a new approach of using the entire image instead of merely patches to explore ROI by directing attention to important regions of the image. The mechanism also used a voting-based choice of relevant regions out of all ROIs. Later the technique combines an FCN and object detection approaches to segment the voted regions based on a learning mechanism to detect better regions of the entire image. The FCN addresses issues of patch-size dependency and problems that accompany it. Overall the approach was too complex, cumbersome to understand, and very difficult to implement.

Faster RCNN was used to fetch feature maps which consequently provide ROIs out of the entire image. Later these ROIs pass through FCM to find out visual salient region based on voting. The results were fused and the final segmented region and related probabilities are produced. The result were measured using MDR, FDR and IU. The MDR of 2.57% and FDR of 16.36% were obtained on NIST SD27. The higher values were justified due to the use of a full image. Another confirmation is obtained using WVU as MDR and FDR obtained over WVU was 13.15% and 5.3%, respectively. Another popular approach was proposed by Kahn and Wani [28] where again the patch-based technique is followed and CNN was used as a classifier. The approach was simple to understand and architecture performed classification on IIIT-D CLF and obtained an MDR and FDR of 10.5% and 4.5%, respectively. The lacking point was the use of architecture without justifying the architectural stability, depth and patch size.

Murshed et al. [41] used mask-RCNN architecture to segment fingerprint slaps of adults and juvenile. The authors created a database and segmented manually labelled exemplar images of left and right hands. The object identification was performed using CNN as a backbone, and the region proposed network was used for proposing the candidate object bounding box where a candidate is a fingermark here. The ground truth box was compared against the observed bounding box created by the mask-RCNN. The distance between the four boundaries of the boxes was the measure of the error. The performance is evaluated in terms of mean absolute error (MAE) and compared to NFSEG and proposed CFSEG for both categories of adults and children and fingerprint segmented with CFSEG performs better. A true positive rate of 0.9986 is observed for CFSEG at 0.1% FPR. The algorithm is performed with manually annotated class labels and is rotation-dependent. Although it does not directly address latent fingerprint segmentation, the technique of object detection with a bounding box was a potential candidate in latent fingerprint segmentation using architecture as CNN as the backbone to further improve error rate with automated class labelling network-cum-mechanism.

2.4 Gap analysis

The gaps are analysed out of the already published work. It is observed that the work before [47] lacks exploited set of features and resources such as supervised techniques and deep network; hence, full image-based segmentation was lacking better accuracy rates. Later, techniques using deep architectures and generic hand-holding are about architectural stability, use of patches over entire images vs region of interest identification or bounding box identification. Apart from the techniques, common performance metrics are identified. The techniques quite vigorously now use deep architectures to approach effective feature engineering. Also, each idea of patch-based [17] or full-image segmentation [42] has its pros. The need for multiple fingerprint segmentation, effective feature selection and hyperparameter stabilization has led to the proposed work.

The proposed work uses full images from the IIIT-D latent fingerprints database to detect colour-based salient and prominent regions [2, 40]. The use of colour-based segmentation techniques with deep networks is effectively seen as producing good results. The correlation between colour and saliency as a joint framework of context-based saliency targeting [61] and de-emphasis of the distracting region of the image regions can be explored which can help generalize the process of coloured image database captured in any light conditions. Further, the salient regions of interest are divided into patches, influenced by the advantages of patch-based techniques. The patches are fed to the SCAE for patch-based classification, thereby segmentation of latent fingerprints. The procedure adopted has not been explored with the given database and methodology in the past.

2.5 Convolutional neural network

Convolutional neural networks (CNN) are neural networks inspired by reduced connectivity among neurons in-between layers to add benefits to an artificial network's training. CNN's achieve this reduced connectivity with the convolutional layer, max-pooling layer and classification layer. These layers alternate and stack up to form a fully connected CNN.

Deep neural network (DNN) shows a success rate in various machine learning tasks such as person identification, visual recognition, speech recognition. CNN has been so far a successful model of DNN that is used to classify the images. The improvements in CNN over deep learning techniques such as batch normalization [29, 64], LeakyReLU [34], regularisation [59] outperform previous machine learning techniques in computer vision-based applications.

With this inspiration, Nguyen et al. [42] used CNN for orientation field estimation in latent fingerprints. A classification problem was proposed based on the latent fingerprint orientation field using a CNN-based approach to estimate the orientation. Khan and Wani [28] used a conventional patch-based approach with CNN-based classification-cum-segmentation approach for latent fingerprint segmentation. The proposed algorithm also fine-tunes CNN with various improvements and augments it with an autoencoder, where an autoencoder is used to extract, learn and represent features efficiently and CNN acts as the classifier for classifying and hence segmenting the image patches into the foreground or background data.

CNN is a deep learning algorithm that can take input as an image and assign ranks to different objects in the image area. The ranks are learnable weights and biases which enables CNN to differentiate objects. The pre-processing in CNN, unlike other supervised classifiers, is much lower. Also, unlike multiple primitive methods where filters are hand-crafted with training, CNN learns these filters which gives an advantage of human independent or no-subjective thinking in the model. The connectivity of perceptron in the CNN model resembles human brain connectivity of neurons.

A CNN uses relevant filters to capture the spatial and temporal dependencies in an image. The re-usability of the weights and reduction in the number of learnable parameters involves enabling the architecture to perform a better fitting to the image dataset. CNN enables ease of processing the high-resolution image data without compensating for feature quality required for accurate prediction. So, the ultimate goal is to design an architecture not only good at feature learning but also scalable to large-scale data sets. A CNN, the foundation of most computer vision technologies, unlike traditional multi-layer perceptron designs, uses two operations called convolution and pooling. These operations help reduce images into essential features, eventually, classifying images into fixed labels.

2.5.1 Convolutional layer

The element involved in carrying out the convolutional operation in the first area of the convolutional layer is called the filter or kernel say k. The size can be any, usually taken as $3x3or5x5or11x11$ but should be less than the size of the image. The filter hovers on the image of the same size at a time. The shifts or hovering count depends on stride length. The hovering or shift enables multiplication of matrix operation between K-sized filter and the same size i^{th} portion of the image. The filter to the next portion with a stride value till it parses the image width-wise. It starts again from the left beginning of the image and parses with the same operation and width. In the case of the image with channel = 3, the same action happens with the same size kernels on each colour matrix. The results are summed with the bias to give a one-depth channel outcome. This way the convolutional operation extracts high-level features. This can be achieved by a combination of multiple convolutional layers with the computational complexity trade-off of the architecture.

Conventionally, the initial layer extracts low-level features such as colour, gradient information ridge. The added layers enable further learning and help to learn high-level features. With increasing layers, better distinguishable features are learned by the network and a better understanding of the image is formed.

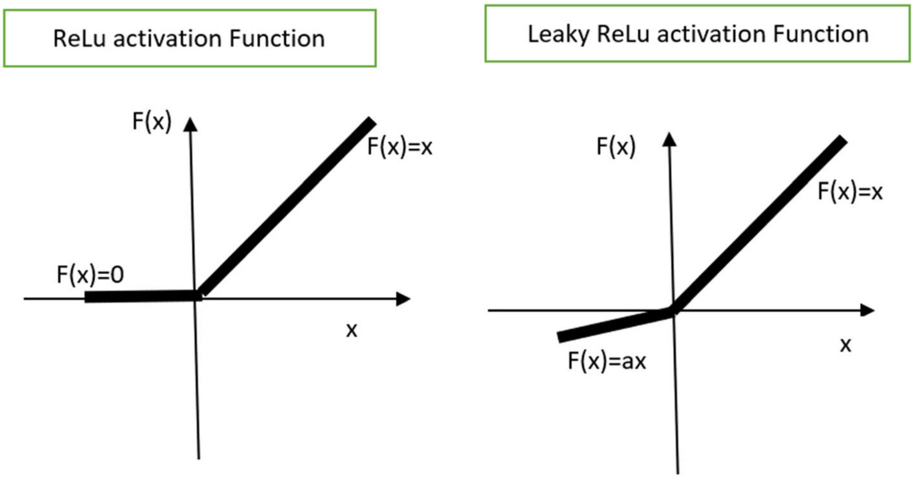


Fig. 3 ReLU vs LeakyReLU function

2.5.2 Improvement in nonlinearity using LeakyReLU

The additional operation with every convolutional operation is adding nonlinearity to the operation. This is performed by ReLU [59]. ReLU stands for Rectified Linear Unit, hence clear from the name, adds nonlinearity, when applied per pixel, replaces negative pixel values to zero in the feature map as mentioned in eq (1). The reason is to make linear convolutional operations to learn real-world resembling, nonlinearity in the data. The disadvantage is by converting the values to zero without understanding the need of it; it might result in a dead end. This makes a model a lot sparser, but there are cases where this is a lost cause. The data without normalization or standardisation of hyperparameters, when fed to a network, impact the weight change during the initial phases of the training. Some weights might become too negative and the importance is lost in zero conversion of ReLU. This makes neurons inactive hence the dead network. The solution is found in LeakyReLU as shown in Fig. 3. In comparison with other existing solutions, the simplicity of the LeakyReLU blends the cause of use with the network. Equation (1) elaborates the use of alpha, as the leverage given over zero replacement to include a portion of the weight in the decision making. So, now instead of replacing values with zero, it replaces pixel values as follows:

$$f(x) = \begin{cases} \alpha * x & \text{if } x < 0, \alpha = 0.25; \\ x & \text{if } x \geq 0. \end{cases} \tag{1}$$

2.5.3 Pooling layer

This layer is spatial pooling which reduces the dimensionality of the feature maps, reduces the count of features and retains only the most informative features. This can be performed using operations such as Max, Sum. The famously used Max pooling is defined by taking spatial neighbourhoods of a certain size say yXy and taking the max value only from that size of the rectified feature map, thus reducing y^2 to just 1 value. Now instead of max, the Average value can be taken, called average pooling, etc. This reduces the dimensionality of the feature map. To reduce it further, the stride values can be increased. If the channel is more than 1, the operation is applied separately on all feature maps generated due to different convolutional layers of channels.

2.5.4 Fully connected (FC) layer-classification layer

The nonlinear combinations of the high-level features are learned by adding an FC layer to the network. These high-level features are the outcome of the Conv-pooling layer. The multi-level representation of the image is now flattened into a column vector. This flattened outcome is fed a feed-forward network, and back-propagation is applied to every iteration of the training. These iterations or epochs enable the model in distinguishing important and non-important features in the image, and the learning is used to classify the objects using the Soft-Max classification technique.

2.6 Autoencoders

An autoencoder (AE) is a simple architecture neural network family. These architectures are trained to set target values equal to input values. [63]. The hidden layer between the input layer and output layer compresses the original data. The reconstruction of data can be provided using compressed representation to regain the original data. The hidden layer data represent the compressed data in a latent space with latent variables. Later the latents are used to reconstruct the original data. The relation between the data is explored while extracting features which can be later used in reconstructing the original form.

To achieve the reconstruction, the autoencoder is divided into two segments, Encoder and Decoder [5, 23, 37] such that

$$\begin{aligned} &\psi : X \rightarrow Y, \quad \rho : Y \rightarrow X' \\ &\psi, \rho = \arg\min_{\psi,\rho} \|X' - (\rho o \psi)X\|^2 \end{aligned} \tag{2}$$

Encoder function, ψ maps the original input X to the latent space at the bottle neck, Y. The decoder function, ρ maps the latent Y to the reconstructing data X' as shown in eq (2). The latent construction using encoder can be represented using neural network function constructing latent, such that eq (3) represents Y, with weights W and bias b, as:

$$y = \sigma(Wx + b) \tag{3}$$

Similarly, the decoder reconstructs the original data from the latent space as shown in eq (4):

$$x' = \sigma(W'y + b') \tag{4}$$

Hence the loss function C(x,x') is represented using eq (5):

$$\begin{aligned} C(x, x') &= \|x - x'\|^2 \\ &= \|x - (\sigma(W'y + b'))\|^2 \\ &= \|x - (\sigma(W'(\sigma(Wx + b)) + b'))\|^2 \end{aligned} \tag{5}$$

The basic requirement is reconstructing the similar, not the same outcome, to learn and difference generating features, hence prioritising the information. Meanwhile, the loss is minimised. The model, thus, is effective in feature selection and dimensionality reduction. While the output usually is of the same dimensionality as the input, an autoencoder's hidden layers are of different dimensions. The encoder maps the input to a new representation called the bottleneck. The decoder, on the contrary, maps the bottleneck representation back to output dimensionality as a sample architecture shown in Fig. 4.

As observed in the background of latent fingerprint segmentation, Ezeobiejesi and Bhani [18] used RBM and Khan and Wani [28] used CNN for the classification task of the segmentation procedure. Now RBM compresses the input data to "fit" into a smaller representation and attempts to reconstruct it back. This training is attempted to minimize an error and to find the most efficient compact representation for input data. The stochastic approach uses several steps of Gibbs sampling using joint probability. CNN on the other hand, instead of adjusting the global weight matrix, impresses on finding locally connected neurons. The kernels used to learn features are learned along with the network. The function is ultimately the same as compressed representation. The task of classification is specific, where features are learned which are spatially closely interrelated.

RBM translates m-dimensional data into the n-dimensional vector using dimensionality reduction property with keeping dominant features as an outcome along with noise reduction. The nonlinearity of dimensionality reduction helps in learning complex relations. But the model is not specific to the classification task but is capable of pretraining.

Now same procedure can be performed by a more suitable model, i.e., Autoencoder. The same features as that of RBM are available with auto-encoder, learning with an encoder and reconstruction with a decoder. These capabilities make auto-encoder a model for pre-training as well. The reason it is preferred over RBM is that, RBM is designed to find the joint probabilities of data, difficult to train and understand, whereas Autoencoder is easy to understand, easy to implement and easy to train to learn a more compact representation of the input data, help in extracting multiple layers of useful information.

The Autoencoders acts as noise suppressants; hence, stacked convolutional autoencoder (SCAE) can work even better for latent fingerprints. The amount of noise in the LFIs is significant due to the source of images. Hence, a model which can provide; a) ease of training, b) noise suppression and c) dimensionality reduction along with the advantages of CNN as a classifier, is proposed here.

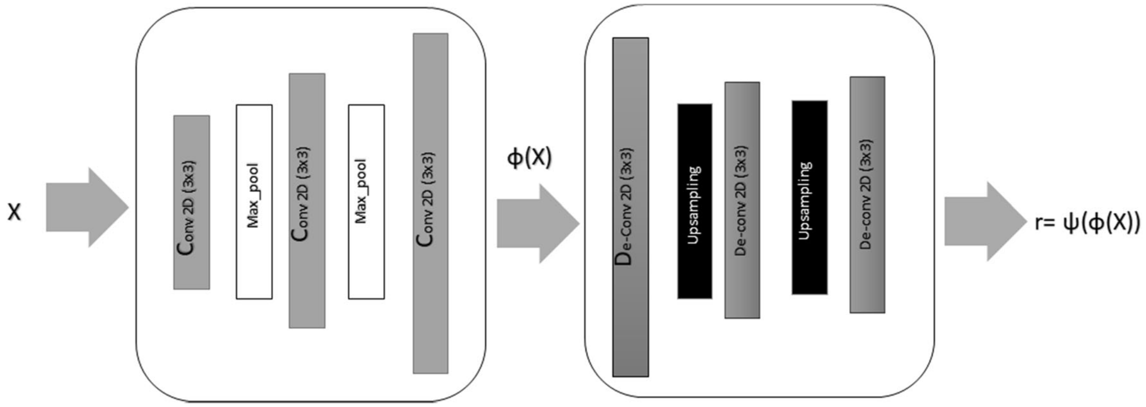


Fig. 4 Structure of an Autoencoder with encoder and decoder functionality

2.7 Stacked convolutional autoencoders

Where CNN is trained end-to-end to learn filters and combine features to classify their input, CAE helps in learning useful features by extracting filters, thus reducing reconstruction error and computational time.

Several AEs can be stacked to form a deeply layered structure. Each layer receives input from the latent representation of the previous layer. A greedy layer-wise unsupervised pre-training can be performed. Later, weights can be fine-tuned using back-propagation, or top-level activation can be used as feature vectors for some classifiers. With this fashion, an SCAE is used to pre-train a CNN with an identical encoder and the feature vector is forwarded to CNN for classifying foreground from background [3, 31, 37].

2.8 Hyperparameter tuning

As elaborated in Table 2, the curve of performance moves from underfitting to generalized state to overfitting as the parameters of the model update with increasing training time and iterations.

As the network is trained iteratively, the weightage of the already powerful connection relatively increases in comparison with poor connections. Hence, only a fraction of these connections is trained adequately. Dropout learns fractions of weights to resolve the problem of making strong connections stronger and vice-versa [59]. For the measurement of repeatability [57] of the model and to be able to reproduce the results, cross-validation [56] is performed. The algorithm proposed, is tested on the dataset to find the limitation to find the limit to train the model and to tune and standardize the parameters. In other words, to ensure the successful outcome of the model every time the training and the parameters are fine tuned. This improvement is performed with CNN FC layers to stronghold the classifier.

3 Proposed method

Colour-Maps are used to early distinguish salient contours of interest (CoIs)out of the entire image. The detection of the salient region not only reduces irrelevant background, saving resource utilization, but the fact is, the salient region detected based on color, helps to generalize the salient region is detected based on colour helps generalize the process for coloured images captured in normal light. The Colour mask and saliency map based on colour adjustment are fused to extract multiple CoIs within the same image. These CoIs can be a) single fingerprint, b) imposter background noise identified as fingerprint or c) partial print. Unlike the voting mechanism, the merging of maps provides a proportional chance to all significant fingermarks, small or large-sized, thereby addressing multiple fingermarks The equal size patches of the CoIs are provided as input to SCAE. The outcome is the class probabilities for foreground and background.

The proposed stack-based classification system addresses the efficiency and effectiveness of the system as follows:

(1) Deciding optimal patch size for better catering of features,
(2) Using saliency and colour-based information,
(3) Using the stack of classifiers in comparison with the single classifier, for the next step intelligent optimization of the segmentation-detection system.

The features are the representation of the LFI. The higher the complexity of the image w.r.t signal to noise ratio (SNR) [34], the higher the complexity of the feature

Table 2 Hyperparameter tuning with over-fitting, under-fitting and cross-validation

Over-fitting	Under-fitting	Cross-validation
The model that has been trained so much so that noise is learned as data.	Model is under-trained.	Performed on a dataset to find the limit to train the model, and to tune and standardize the parameters.
The model performing highly accurately might perform poorly in training data.	The model performs poorly on both training and testing data.	Measuring repeatability of the model.
The problem of overlearning can be solved by regularization.	The problem of underfitting can be solved by giving proper time and attention to training the model.	The solution is a K-fold cross-validation algorithm.
Dropout learns a fraction of total weights in the network in each training iteration.		Usually for a small dataset, K=3, 5 can be chosen, whereas, for a large dataset, K=10 is the suitable choice.

computation. The increased mathematical computation increases the cost of the analytical cost of the overall algorithm. Well, with this awareness the information captured in the latent images is a) good with high SNR, b) poor and c) ugly with critically low SNR.

Due to the poor-quality ridge and/ or valley patterns, the accuracy of the segmentation task is hampered. The attributes required to distinguish noisy data from required relevant data are also critical computations. Hence, the attributes or features to be extracted must be sufficient in the count. The global and local features are handcrafted to cater for the need of the nature of the data in the images. The features are representations of the source. If the higher the complexity of the source, the lower the correlation factor of features. The choice of the classifier is, consequently dependent on the quantity and quality of the features computed. With the limitation of small sample size leading to higher intra-class variance and binary nature of the classification, a stack-based classifier is expected to perform better than traditional classifiers due to its advantages of better prediction and a more stable model.

The proposed method addresses all such challenges and proposes the following points:

(1) The early distinction of CoIs using saliency and colour-oriented masks, to reduce irrelevant background area for segmentation, thereby increasing model prediction accuracy and reducing the amount of erroneous information for further processing.
(2) Contour-based CoIs generation subsequently detects and segments multiple instances of the fingerprints.
(3) Stable patch size for an adequate amount of information for feature extraction from the patch being processed, resulting in increased accuracy.
(4) Parameter standardization for effective and efficient automatic identification of fingermarks.
(5) Establishing repeatability and reproducibility of the model developed using cross-validation.

The proposed model is designed based on the outcome of the previous models taken in the order as follows:

(1) Patch-based system: This pre-feature extraction task is found to be effective in the domain of resource utilization. It has been observed that close relation of pixels with neighbours can find better features. Therefore, a patch-based system can be effective in this application domain.
(2) Hybrid System: Image is resized to 512 X 512 size. The entire image has two elements, irrelevant background and relevant foreground. The supervised technique used so far suggested the high MDR and FDR of the best of the classifiers due to the presence of irrelevant data. Instead of dividing the entire image into patches, if an entire image system can be initially used to early distinct CoIs which are potential fingermarks, MDR and FDR can be further reduced. Further to that, these CoIs can be divided into patches for further process.
(3) Stack of classifier: The suggested approach used provides the evidence to support the use of a stack of classifiers for the classification of patches. The labelled data when used come with the burden of identifying results before testing. As per the nature of the application, the noisy patches can result in erroneous labelling. Therefore, it is only advantageous to use deep neural networks to self-train the data and fine-tune the model classifier with a small amount of data while the testing phase.

3.1 Hybrid approach algorithm

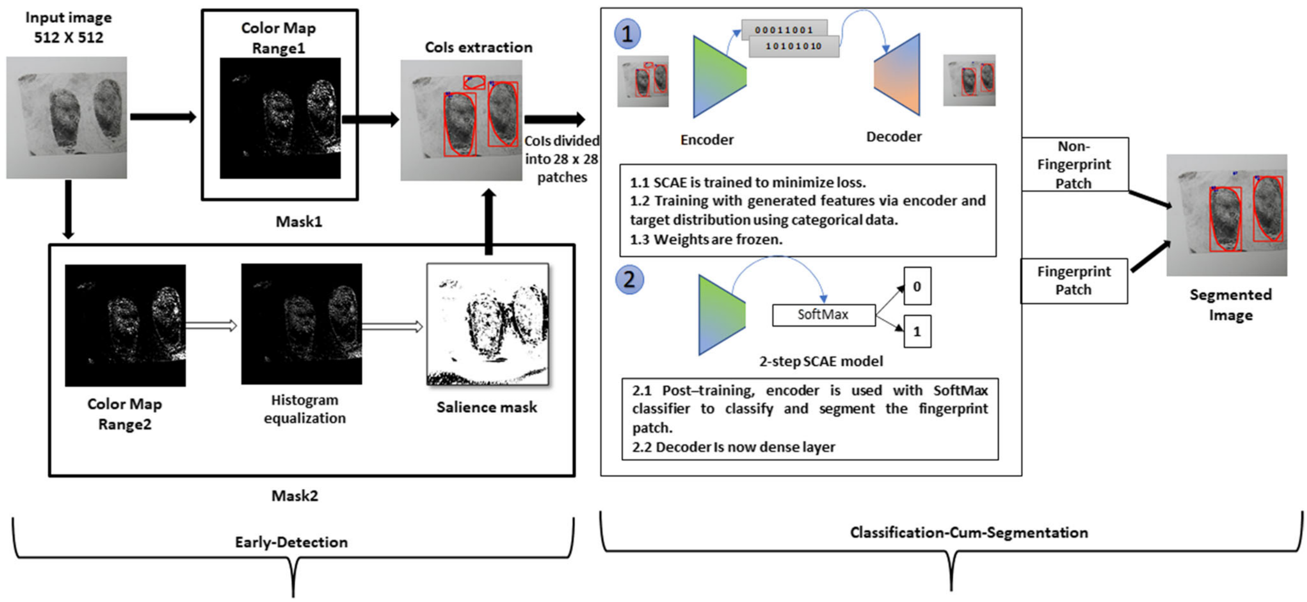
The proposed method performs the segmentation of latent fingerprints using the IIIT-D latent images database by introducing an early distinction technique using entire image information followed by a patch-based classification and segmentation technique using SCAE. As shown in Fig. 2, the images are categorised into two categories of fingermark instances and the quality of the image based on the amount of noise in it.

The task is divided into two parts. The first part is contour extraction based on colour and salient region masks which acts as initial CoIs. However, colour-based alone segmented CoIs will not guarantee generic segregation of CoIs as fingermarks. Consequently, the second subsequent aim is to feed patches of these CoIs to a staked CAE to classify these patches into fingermark or background of the image for better SA and effectiveness of the system.

3.1.1 Contours of interest extraction

Due to the diversity in the type of LFIs, the masks required to segment the ridge patterns from the image are different as well. The common element in all images is the colour-based identification of the object. The fingermark is significantly dark in colour w.r.t. its nearest surrounding background. The colour map addresses the need for segmenting the ridge pattern darker in this database. As shown in Fig. 5, the integration of the colour map with and without the salient mask and convex hull fitting produces the CoIs.

With LFIs, the colour of the image is confined to a colour range. To segment a fingerprint from an LFI, let the input image is I is used to produce an image with CoIs, say I_c. Before starting image enhancement, a check on B, G, R range of I is placed. Upon experimentation, it is optimally



1.1 SCAE is trained to minimize loss.
1.2 Training with generated features via encoder and target distribution using categorical data.
1.3 Weights are frozen.

2.1 Post–training, encoder is used with SoftMax classifier to classify and segment the fingerprint patch.
2.2 Decoder Is now dense layer

Fig. 5 Pictorial representation of proposed latent fingerprint(s) segmentation

found that if G component of the colour of I ranges between (+/- 4) of B as well as R component of I, then I is a small histogram image H_s else a large histogram image H_l as shown in eq (6).

$$I = \begin{cases} H_s & \text{if } G \in \{B-4, B+4\} \& \ G \in \{R-4, R+4\}; \\ H_l, & \qquad\qquad\qquad \textit{otherwise}. \end{cases} \tag{6}$$

A transformation function is required to change or stretch the range of input pixels to the entire full range of the image. This is called Histogram equalization. Histogram equalization performs global equalization and hence can lead to loss of information due to over brightness. This method divides the image into small windows and performs global equalization on these windows. In the case of the noisy window, the contrast limiting threshold is applied by clipping the pixels of the window above this threshold as they are probable results of noise enhancement.

Limit contrast enhancement-based adaptive histogram equalization (CLAHE)-based image enhancement helps in discrimination of nearest background neighbour of fingerprint, thus allowing the colour range to discard background colour and retain fingerprint colour intact for segmentation. This effective enhancement technique is applied in many areas in recent years [19, 52]. This technique is not explored in collaboration with full image-based latent fingerprint segmentation. Histogram equalisation performs better when the image intensity range is confined in a smaller region, for instance, Fig. 2(a) and 2(d), whereas the same shall not be effective alone in Fig. 2(b) and 2(c).

The colour-based contour model produces the output as a result of integrated two masks. The masks are differentiated based on the colour range of the input image. Two masks, Mask1 and Mask2, are applied on I. The Mask1 is applied on the image with H_s and Mask2 with H_l. As shown in Fig. 6 which describes Proposed algorithm of latent fingerprint(s) segmentation, contours out of Mask1 and Mask2 applied on I are combined to find CoIs.

Mask1 is the combination of a colour map and a salience map. The order of the pre-processing is dependent on a range of LFIs, and the subsequent masks are impacted by the amount of enhancement provided in the image. When the histogram of the image is broad, CLAHE is applied with $cliplimit = 0.2$ i.e. if the histogram is above 20 % contrast limit, the pixels of the window are clipped. CLAHE applied image is passed through colour-based thresholding. Later, that mask obtained is passed through a salience map. This feature provides a mean score of a pixel that is prominent in the neighbourhood. Saliency residuals are computed and mapped back to salient locations in the corresponding spatial domain. Consequently, a binary map using threshold selection is used to calculate thresh count which is used to extract salient regions from an image. More the salience mean, better are the chances of ridge area surrounded by structurally disturbed background [6]. These saliency-based closed regions called convex hulls are clustered. These convex hulls are classified as CoIs or irrelevant hulls.

Mask2 is purely colour-based thresholding but only on images with large area histograms, so CLAHE application is absent. The output mask from the colour map has now a small region of the histogram. Now due to this confined histogram, histogram equalization along with morphological operation *CLOSE* is performed to get rid of small noisy

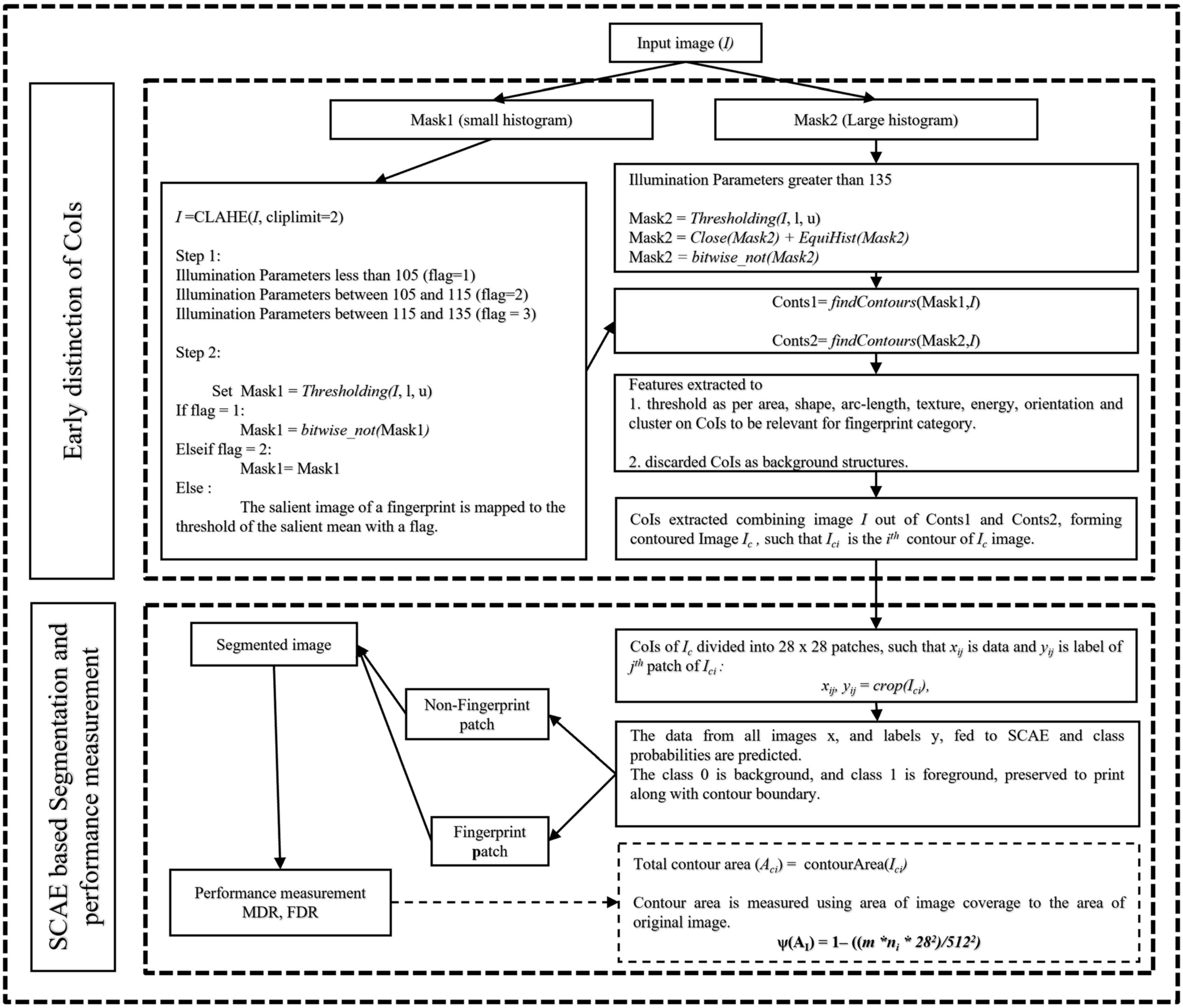

Input image (I)
Early distinction of CoIs
Mask1 (small histogram)
Mask2 (Large histogram)
I =CLAHE(I, cliplimit=2)
Step 1:
Illumination Parameters less than 105 (flag=1)
Illumination Parameters between 105 and 115 (flag=2)
Illumination Parameters between 115 and 135 (flag = 3)
Step 2:
Set Mask1 = Thresholding(I, l, u)
If flag = 1:
Mask1 = bitwise_not(Mask1)
Elseif flag = 2:
Mask1= Mask1
Else :
The salient image of a fingerprint is mapped to the threshold of the salient mean with a flag.
Illumination Parameters greater than 135
Mask2 = Thresholding(I, l, u)
Mask2 = Close(Mask2) + EquiHist(Mask2)
Mask2 = bitwise_not(Mask2)
Conts1= findContours(Mask1,I)
Conts2= findContours(Mask2,I)
Features extracted to
1. threshold as per area, shape, arc-length, texture, energy, orientation and cluster on CoIs to be relevant for fingerprint category.
2. discarded CoIs as background structures.
CoIs extracted combining image I out of Conts1 and Conts2, forming contoured Image I_c , such that I_ci is the i^th contour of I_c image.
SCAE based Segmentation and performance measurement
Segmented image
Non-Fingerprint patch
Fingerprint patch
Performance measurement MDR, FDR
CoIs of I_c divided into 28 x 28 patches, such that x_ij is data and y_ij is label of j^th patch of I_ci :
x_ij, y_ij = crop(I_ci),
The data from all images x, and labels y, fed to SCAE and class probabilities are predicted.
The class 0 is background, and class 1 is foreground, preserved to print along with contour boundary.
Total contour area (A_ci) = contourArea(I_ci)
Contour area is measured using area of image coverage to the area of original image.
ψ(A_1) = 1– ((m *n_i * 28²)/512²)

Fig. 6 Proposed algorithm of latent fingerprint(s) segmentation

colour detected unwanted regions resulting in enhanced LFI. The final classification of hulls out of contours, into fingerprint and non-fingerprint, is performed using SCAE.

The contour is a curve drawn by joining continuous points along the boundary with the same colour or intensity. Since the out-of mask is a binary image, the contour generation is easy and effective. In LFI, the contours are labelled as ridge patterns and non-ridge patterns based on a feature vector computed from the detected contours. The convex hull is drawn on the contours. Given all points in the Euclidean space of the given contour, the convex hull is the smallest possible convex set that contains all these points. Figure 5 shows the contour and the corresponding convex hull around the ridge pattern in the output image of masks.

3.1.2 Feature extraction

The contours are extracted using colour information. The features of the contours along with ridge information help in removing irrelevant contours and passing on relevant contours. Let the image I_c has i^{th} contour c_i and corresponding hull as h_i. The thresholding is applied on extracted following features from contours and/or convex hulls in I_c:

(1) Solidity: The relative amount of area used by the convex hull in comparison with contour area. The solidity is defined as in eq (7), if Δc_i is Contour area of i^{th} contour and Δh_i is Hull area of i^{th} hull corresponding to c_i :

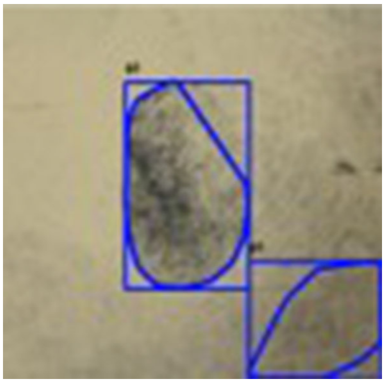


Fig. 7 Effect of solidity

$$Solidity = \frac{\Delta c_i}{\Delta_{h_i}} \tag{7}$$

If the solidity is high, it suggests the presence of an elliptical shape convex object such as a fingerprint. If the solidity is low, it suggests a noisy, irregular-shaped, background patch [65]. As shown in Fig. 7, with convex hull no.2, even if ridge orientation and energy are not effective, solidity can identify the relevant area. Also, convex hull no.5 shows lower solidity from 1, lesser relevant that convex hull becomes, therefore labelled as irrelevant.

(2) Extent: The type of shape in the bounding rectangle of the contour. The extent is given as in eq (8), when w and h are dimensions of the bounding rectangle of c_i contour:

$$Extent = \frac{\Delta_{c_i}}{(w * h)} \tag{8}$$

The lesser the extent, the probability of a noisy, irregular, background patch is higher. The higher the value of extent means the contour has a more regular, ridge patch. Figure 5 contours show the cases where the extent of contour no. 181 is low and therefore is considered irrelevant and contour no.4 is high hence relevant, respectively.

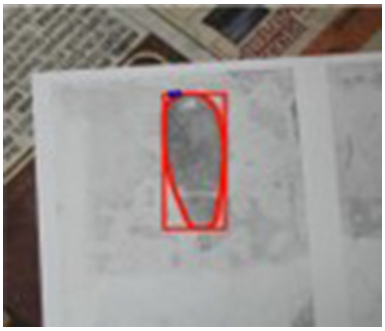


Fig. 8 Effect of perimeter

93) Contour Parameter: The perimeter of a contour defines its arc length. Lesser values perimeter contours are either overlapped on already existing contours due to another mask or are obtained contours with the irrelevant background. So, to avoid background or no boundary overlapped contours, higher values are preferred for ridge value presence. The parameter is given in eq (9).

$$Parameter = ArcLength(c_i) \tag{9}$$

As shown in Fig. 8, the contour parameter of contour no. 181, with overlapping boundaries, is lesser than contour no.4 and hence is labelled irrelevant.

(4) Ridge value texture: This is the measure of the local homogeneity. The measured value determines the relation with the presence of the ridge field. The lesser the homogeneity, the more ridge field presence [8]. Let p is the normalised grey-level co-occurrence matrix [20], if G_{max}, is the maximum possible quantized value by [20] in grey-level co-occurrence matrix, then eq (10) gives invert difference homogeneity or ridge value texture as :

$$RidgeTexture = \sum_{i=1}^{G_{max}} \sum_{j=1}^{G_{max}} \frac{1}{(1 + (i - j)^2)} * p_{i,j} \tag{10}$$

(5) Ridge value Energy: This is the measure of uniformity and organised structure in the image. The lesser the value, the lesser the uniformity and more chances of the presence of ridge or contour. [20]. The ridge value energy is given in eq (11):

$$RidgeEnergy = \sum_{i=1}^{G_{max}} \sum_{j=1}^{G_{max}} (p_{i,j})^2 \tag{11}$$

.

The biggest contribution or advantage of CoIs extraction is to avoid time consumption on an irrelevant portion of the image where the data are not available, whereas now relatively lesser time to learn via a deep network which was otherwise a time-consuming task.

3.1.3 Stacked convolutional autoencoder network for segmentation

The resultant CoIs from the first phase are now prepared and fed to SCAE. The contoured image is divided into patches. Apart from being a popular size, patch size $28x28$ experiments with size $56X56$.

Hence, when patches are fed to SCAE, to classify the greyscale patches into fingerprint or non-fingerprint area, the dimensions of the matrix are of size $28x28$. Now, Conv2d used takes input as the 2d structure of the input

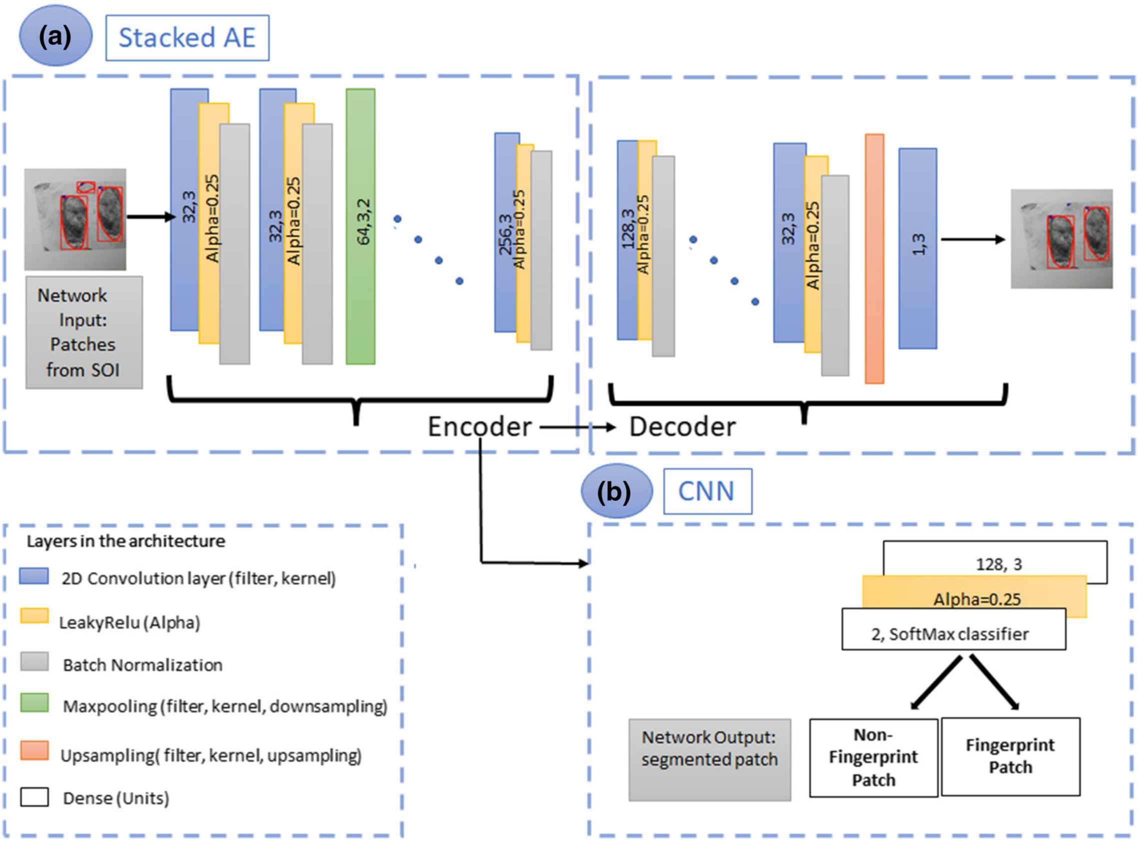


Fig. 9 Proposed structure of SCAE for classification and segmentation process: **a** stack of CAE used CNN initialization, **b** classification using pre-trained CNN

image, batch size and channel value of the input image; hence, the input is the form of [*BatchSize*, *ImageWidth*, *ImageHeight*, *ChannelInformation*]. Since it is a greyscale patch, therefore channel information is set to 1. The batch size is 64; hence, the input is ultimately, [64, 28, 28, 1]. When the database is processed, to train the model which can generalize better, partition the database into a training and validation set, here, the partition is the 8 : 2 ratio of the database. This step helps in reducing overfitting. Figure 9 elaborates the proposed structure of SCAE for classification and performance evaluation.

SCAE comprises encoder and decoder functions. The encoder has four convolutional blocks; each block has a convolutional layer and a batch normalization layer. Two Max pooling layers were added after the second and third blocks. The first block contains 32 filters of size (3, 3), followed by the max-pooling layer. The second block contains 64 filters followed by the max-pooling layer. The third block contains 128, and the fourth contains 256 filters of each size (3, 3). These layers are not followed with down-sampling of max-pooling. The decoder has three convolutional blocks; each block has a convolutional layer and a batch normalization layer. Up-sampling is carried out after the second and third layers. The architecture is chosen after trails of different set-ups of layers. The first block contains 128 filters of size (3, 3). The second block is similar except it contains 64 filters. This is followed by the up-sampling layer. The third block contains 32 filters followed by another up-sampling layer, and final block contains only 1 filter of size (3, 3). This is a reconstruction of the input back having only a single channel.

The max-pooling layer will downsample the input by two times every time included in layers, while the upsampling layer will upsample the input by two times each time it is used. The model is compiled using optimizer RMSProp. The training and validation loss plotted using the fit() function shows sync and is decreasing, hence showing good generalisation capability. The weights of the autoencoder trained in the previous step are loaded but only in the encoder part of the model. The encoder architecture is the same as used in the AE phase. Along with this phase, the fully connected layers are stacked with an encoder. Now, the model is trained with various epochs such as 50,100 and 400, batch size 64, in the absence and presence of dropout. The performance metrics of SA, MDR and FDR are measured while predicting labels.

The outcome is the classification of patches into fingermark or non-fingermark patches. The performance is measured with performance metrics MDR, FDR and SA with K-fold cross-validation to introduce repeatability and reproducibility of the model. The efficiency of the algorithm is observed using the performance metrics, whereas effectiveness is observed using reduction in the processing area, $\varphi(A_I)$ for optimized segmentation. The experimentation is performed with improvements in CNN, optimal patch size and regularization parameters. The comparison of pre-trained CNN and naive CNN with the same architecture is performed as well. In addition to it, the outcome is compared with other proposed approaches.

4 Experimentation and result analysis

4.1 Experimental setup

The experiments are performed using an open-source and available IIIT-D CLF database published by Indraprastha Institute of Information Technology, Delhi (IIIT-D). There are 150 classes of latent fingerprints with categories mixed with single, partial and multiple fingerprints along with clear and noisy fingerprints.

The original images are large. These are resized to 512 × 512. The masks are applied to images; features are extracted from masked images. Compartmentalization of images is done into single and multiple fingermark count presence in images. The resultant early detected contours are input images with the presence of convex hull as segment boundary on latent fingermark(s) as per categories, respectively.

Further, all the extracted contours are divided into equal-size patches of size 28 × 28. The total patches formed are further divided into 8 : 2 training and testing samples ratio, respectively. The ratio is experimented with and provides better results than 6 : 4 ratio. Also, due to the nature of the latent images, the images with SNR below 2.5 are ignored. The presence of such images may result in higher MDR and FDR, affecting the effectiveness and thereby efficiency of the model.

4.2 Performance metrics

The performance metrics help in validating the quality of the results. The metrics of any technique must be relevant to the measuring feature. The proposed system uses the following performance metrics used from the literature on latent fingerprint segmentation:

(1) Segmentation Accuracy (SA in %): It is a measure of the classifier to correctly predicted outcomes. In this case, it correctly predicted fingerprint patch count. The all correct fingerprint and background patch predicted count FBP, w.r.t total predicted patches TP, made, correct or incorrect. Let there are n patches to be predicted, such that p_i is i^{th} class probability, and $p_{i,j}$ is the predicted probability of class i predicted as class j, then SA is given in eq (12) as:

$$\begin{aligned} SA &= \frac{FBP}{TP} \\ &= \frac{p_i + p_j}{p_i + p_j + p_{i,j} + p_{j,i}} \end{aligned} \tag{12}$$

(2) Missed fingerprint detection rate (MDR in %): This is the average percentage measure of foreground pixels misclassified as background noise. As shown in eq (13), MDR is the count of patches predicted background w.r.t. to total foreground predictions. If total Missed Foreground Patches (MFP) i.e. foreground patch considered background, w.r.t Total Foreground Predicted patches (TFP), correct or incorrect.

$$\begin{aligned} MDR &= \frac{MFP}{TFP} \\ &= \frac{p_{i,j}}{p_i + p_{i,j}} \end{aligned} \tag{13}$$

(3) False fingerprint detection rate (FDR in %): This is the measure of background noise misclassified as foreground pixels. As shown in eq (14), FDR is calculated as the ratio of the Falsely Foreground Patch predicted (FFP) score out to the Total Correctly Patch (TCP) predicted score.

$$\begin{aligned} FDR &= \frac{FFP}{TCP} \\ &= \frac{p_{j,i}}{p_i + p_{j,i}} \end{aligned} \tag{14}$$

where j is class 0, or background region and i is class 1, or fingerprint region.

4.3 Latent fingerprint segmentation results

The entire process is shown in Fig. 5 using a suitable example. The results are divided into the following categories:

(1) Algorithm outcome: The outcome of the algorithm with each phase is discussed, and the impact of different phases on sample images is displayed.

(2) Effectiveness and efficiency of the model: The model's performance as per different patch sizes is measured and discussed along with the benefits of

the use of improvements such as the dropout layer in CNN. The performance outcome is measured using performance metrics SA, MDR and FDR using different epochs and dropout parameters.

(3) Repeatability and reproducibility of the model: The use of cross-validation for SCAE is obtained to verify the behaviour of the model over multiple folds.

(4) Comparative evaluation: Comparison of performance of SCAE using pre-trained CNN is performed with alternative CNN. Also, a comparison of performance with existing techniques is performed.

4.3.1 Algorithm outcome

The outcome of the algorithm is produced in two categories. The first outcome category shows a sample of overall outcome images. This combination consists of sample outcomes with single, multiple and partial fingerprints. Figure 10 shows a sample of successful cases. Figure 10a describes a clean image with structured noise and a small scale of the histogram, thereby producing a salient structure as a ridge pattern convex hull as shown. Figure 10b shows a case along with the additional effect of light, making it difficult to set a range of colourmap. Colourmap is adjusted based on the light, luminous and brightness of the image.

Figure 10c and d includes a clean and noisy image with a major single fingerprint and a small partial existing fingerprint with a small histogram, respectively, and Fig. 10e and Fig. 10f with a large histogram range. When the colourmap range is not appropriately falling in close range, light-coloured fingerprints become difficult to segment. With the help of the clip limit of CLAHE, light-coloured fingermarks can be optimally enhanced to fall in the range of colourmap. The same can be observed in Fig. 10g and Fig. 10h.

Figure 11 shows the bad output as (a) false detection due to the effect of light along with a single fingerprint hit, (b) misdetection due to the light-coloured range for colourmap, (c) large-sized convex hull due to the blended nearest background, (d) misclassification due to noise in the image with no result for segmentation.

The second is the category where intermediate results of different phases are shown in Figs.12a–e. Figure 12a is the sample image 1 which passes through mask1 Fig. 12b and mask 2 which produces the outcome Fig. 12c. Similarly, Fig. 13a is the sample image 2 which passes through mask1 Fig. 13b and mask 2 which produces the outcome Fig. 13c. The integration of both masks outcomes in the set of contours on the original image. Set of thresholding is applied on features of contours mentioned in sec:3.1.2. The thresholding reduces the count of contours to only potential candidates of fingerprints as shown in Fig. 12d and Fig. 13d. Finally, Fig. 12e and Fig. 13e are the outcome of classification-cum-segmentation using SCAE for a good sample and poor sample, respectively.

4.3.2 Effectiveness and efficiency of the model

The SCAE has been experimented on MIRC machine with specifications: 2 Intel Xeon processors, 256GB RAM and 32GB Tesla v100 for 27 minutes.

The efficiency of the work is measured in terms of the accuracy of the results. The accurate results are accompanied by erroneous results as well. For measuring the efficiency, the results are checked for a) accurate results using SA, b) false segmentation rate and c) missed segmentation rate measures. The proposed system produces improved segmentation and reduced FDR and MDR in comparison with state-of-the-art deep learning-based published results by [28] and [18], hence, the proposed system is efficient in

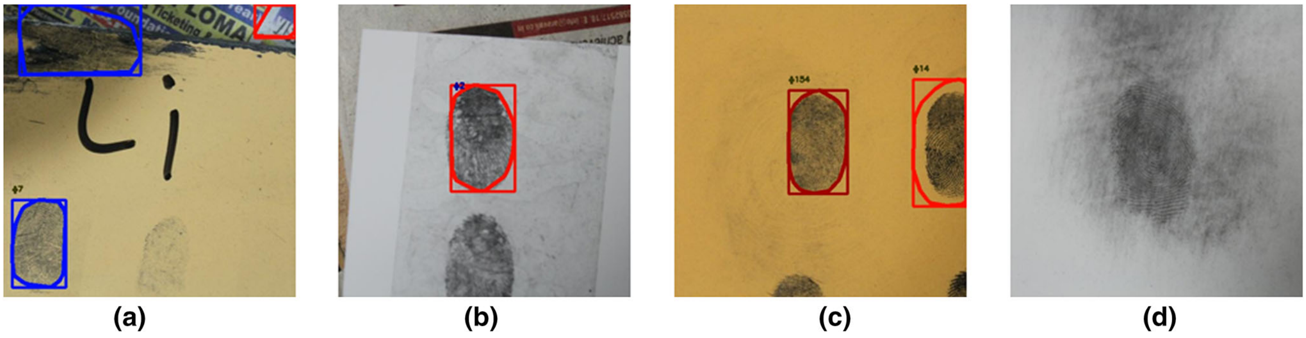


Fig. 10 Good segmentation with single and multiple fingerprint presence

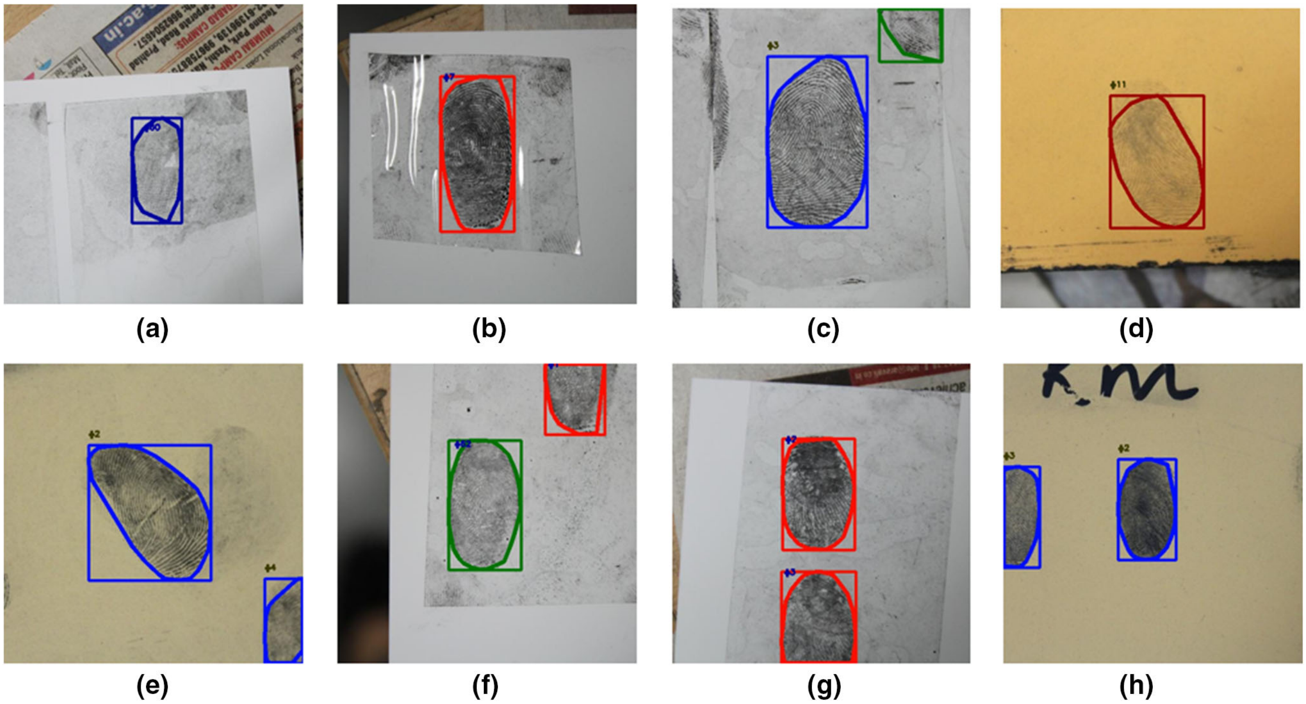


Fig. 11 Poor segmentation with single and multiple fingerprint presence

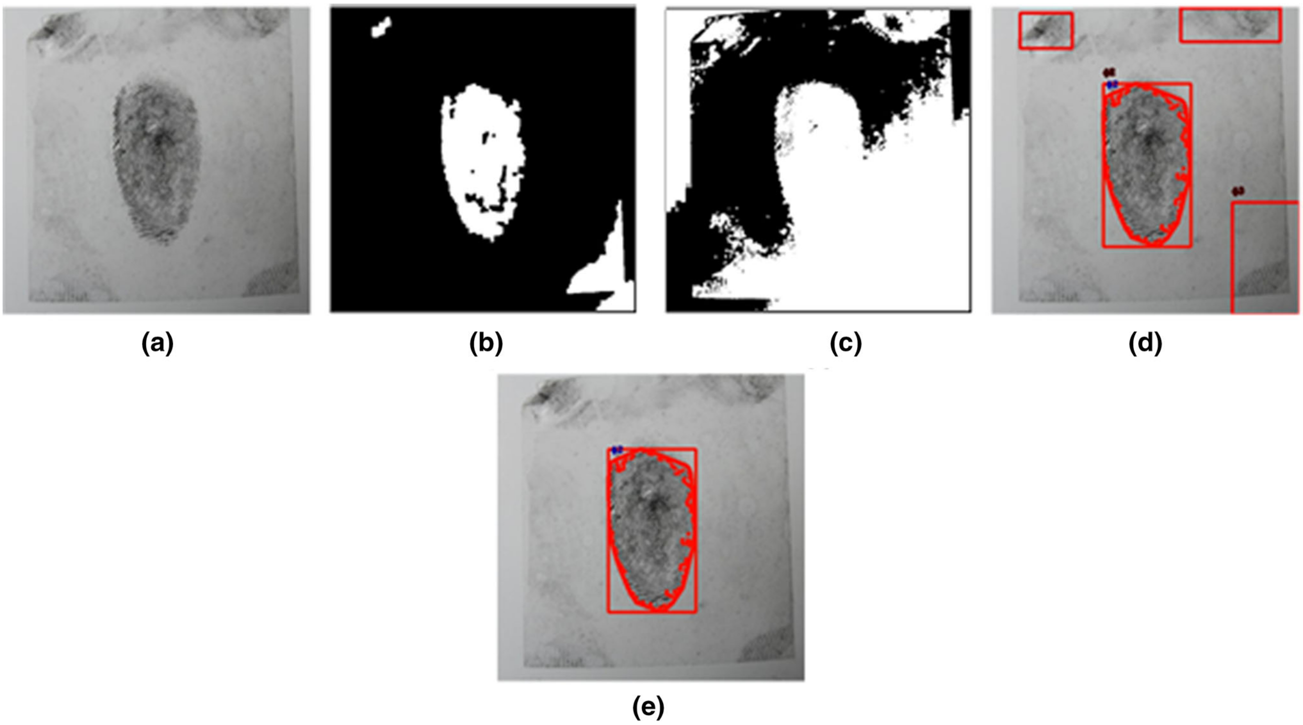


Fig. 12 Sample image 1 from IIIT-D CLF showcasing successful outcome of application of mask in producing contoured images and final outcome of SCAE

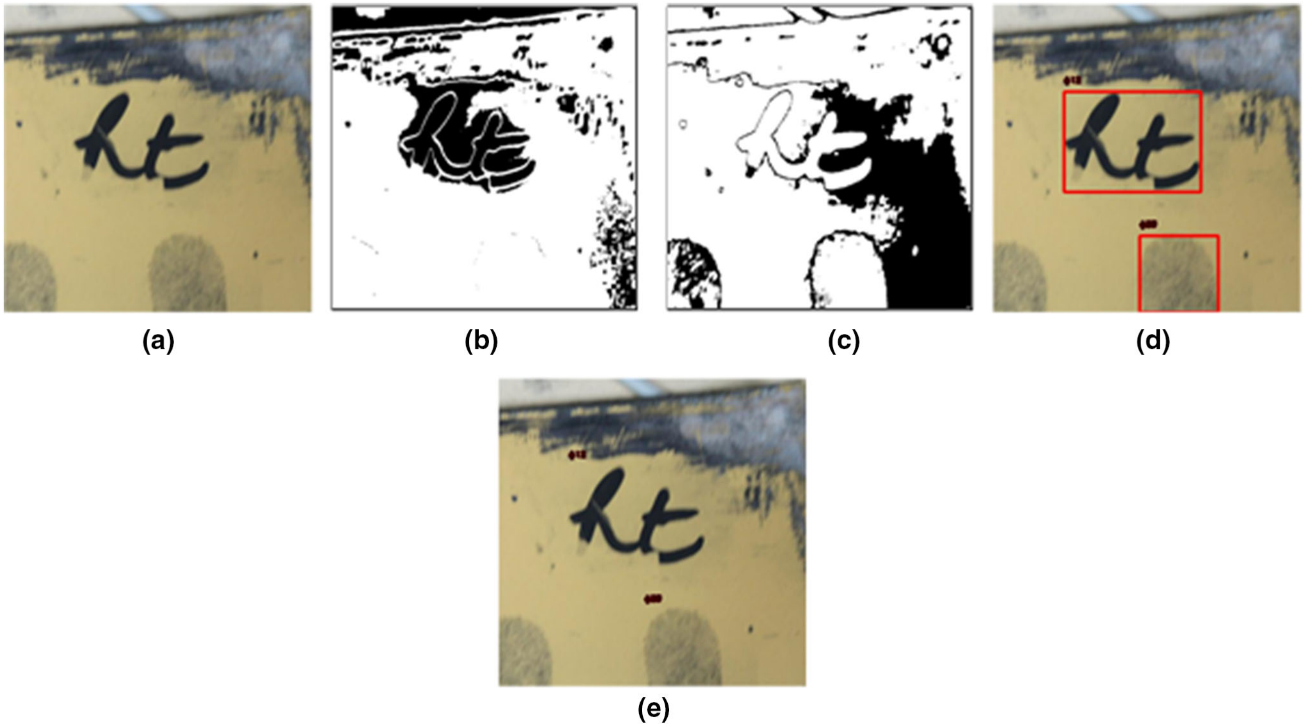


Fig. 13 Sample image 2 from IIIT-D CLF showcasing failed outcome of application of mask in producing contoured images and final outcome of SCAE

Table 3 Performance evaluation of different patch sizes

Patch size	Class distribution		SA	MDR	FDR
	class 1	class 0	%	%	%
28	11338	7288	98.16	2	2
56	55	6730	92	0.05	0.04

performance. The results are observed with experiments involving a) different patch sizes and b) use of regularization i.e. dropout layer during classification. Here the patch size considered is 28X28 and 56X56.

In the discussion about **Performance with different patch sizes**, post-contour detection, each contour is divided into equal-sized patches and fed to SCAE. Upon experimentation, a layered structure of SCAE is finalised and fed with patches of sizes 28X28 and 56X56. Table 3 elaborates on the performance outcome using each patch size. If C represents the Conv2D layer, U represents the up-sampling layer and P represents the Max-pooling layer, then using x in xC, xU and xP reflecting the number of Conv2D layers and number of max-pooling layers in the sequence of architectural layers, the architecture is represented as SCAE_19: 2C-P-2C-P-4C (encoder) and 4C-U-2C-U-C (decoder), where 19 is the count of layers in the architecture AE. The CNN classifier is pre-trained with AE. Different networks can experiment in future work. Hence, the architecture, patch size combinations used are [$SCAE$_19,28] and [$SCAE$_19,56]. The SA (in %) obtained using [$SCAE$_19,28] is 96.62 and [$SCAE$_19,56] is 92 with epochs =50.

The decision making is performed not solely on SA obtained but the class distribution as well. The reason to choose another stable parameter is due to the inclination of change of SA with an optimal set of epochs. The trade-off between SA and MDR-FDR is disturbing in the 28 vs 56 scheme. Hence class-distribution post-classification is considered, and clearly, 28 patch size provides a more stable distribution than patch size 56. The uneven distribution depicts that there exists imbalanced samples to learn from; hence, the model is biased toward training, thereby, to testing. The imbalance of slight nature is shown with patch 28 and severe with patch size 56. Hence, it is an optimal decision to choose patch size 28 with image size 512 X 512. If the original image experiments, in future, the patch size can experiment for a better balance of class distribution.

In the discussion about, **Performance evaluation with drop-out layer**, Table 4 summarizes the impact of the absence or presence of dropout in the architecture using noisy data for training and classification. Here, the patch size used on SCAE_19 is 28, with the train-to-test ratio as 8:2. The comparison of results is also performed using

Table 4 Performance of SCAE using 50,100 and 400 epochs in absence vs presence of dropout

Regularization		Without dropout			With dropout		
Epochs		50	100	400	50	100	400
Performance metrics (in %)	SA	97.49	95.79	97.65	98.16	98.61	98.45
	MDR	10	3	2.3	2	3	1
	FDR	12	3.8	2.6	2	3	1

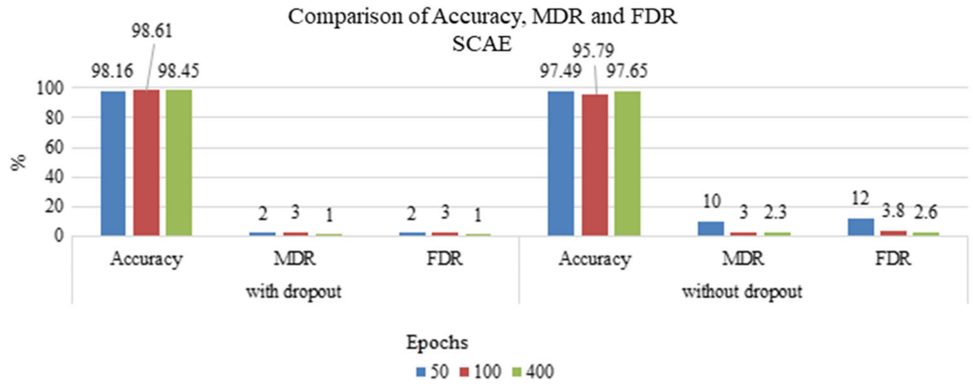


Fig. 14 Performance evaluation of SCAE using epochs = 50,100,400 in absence vs presence of dropout

Table 5 Performance comparison of SCAE and CNN, in absence, presence of dropout(0.25) and dropout(0.1)

Regularization		Without dropout		With dropout (0.25)		With dropout(0.1)	
Technique		SCAE	CNN	SCAE	CNN	SCAE	CNN
Performance metrics (in %) (Epochs (50))	SA	97.49	97.31	98.16	98.4	98.21	98.55
	MDR	10	10	2	1	2	1
	FDR	12	3	2	1	2	1

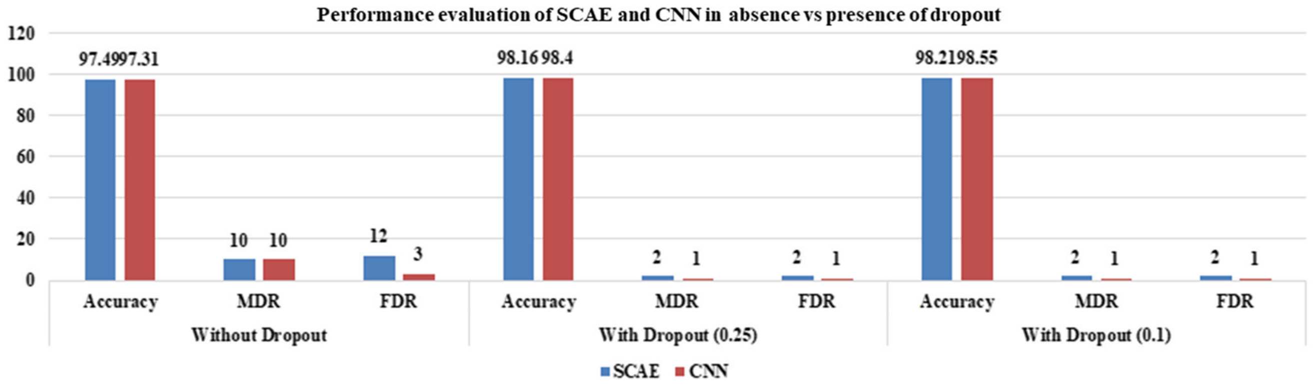
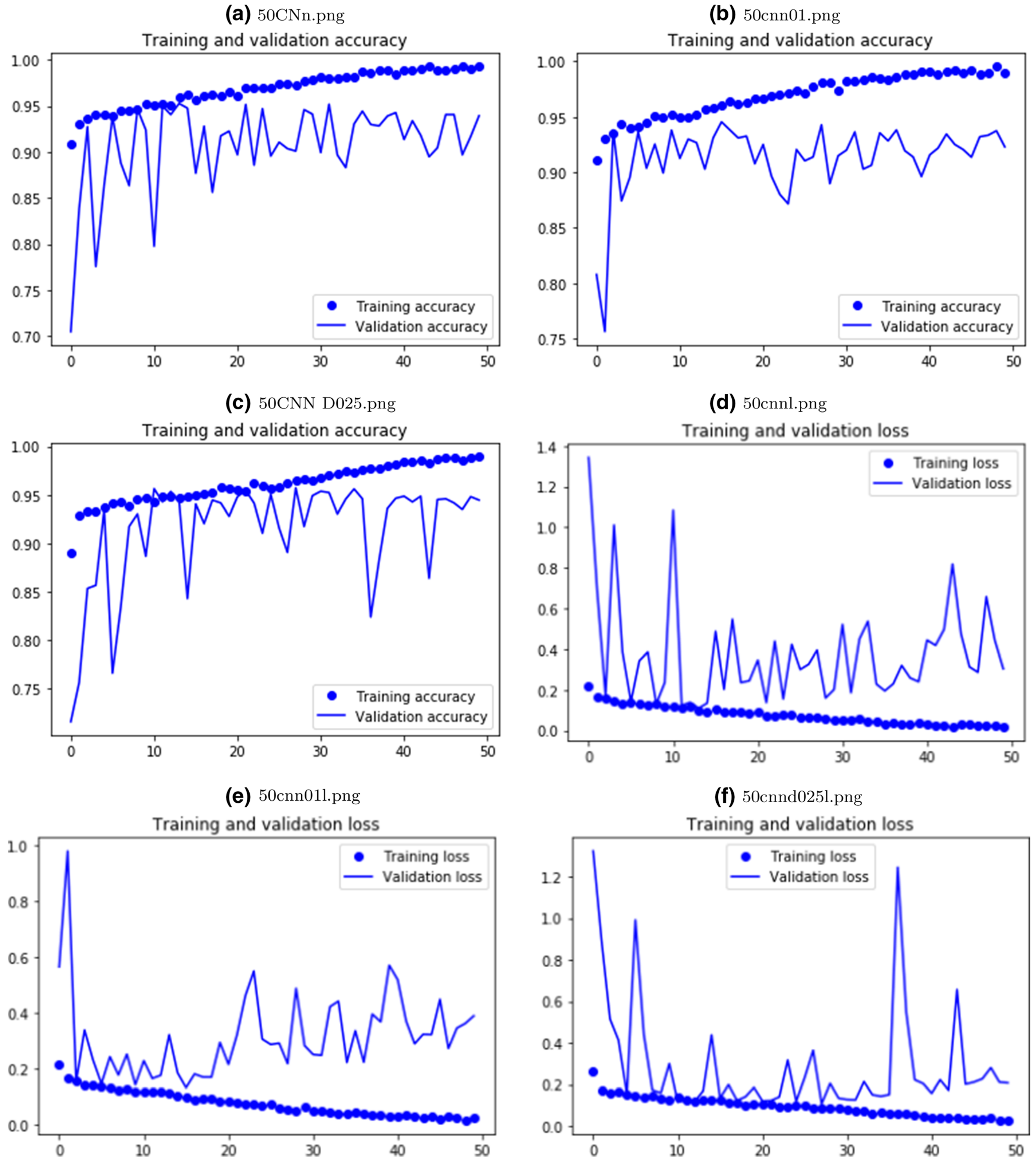


Fig. 15 Performance evaluation comparison of SCAE and CNN with epochs = 50, dropout = NA, 0.1, 0.25

CNN alone. The architecture is the same as SCAE_19 except the decoder(AE) is removed. Now CNN is not pre-trained. The SA along with MDR and FDR is compared on a)epochs= 50, b) without dropout, c) with dropout= 0.25 and 0.1. Figure 14 is the graphical representation of the results.

Table 5 shows an SA of 97.49% in absence of dropout. The SA is improved using SCAE with dropout 0.25 to 98.16% and 98.21% using dropout 0.1. The same measure using CNN is lesser than SCAE in absence of dropout, but CNN responds better when dropout is used. The SA is 98.4% and 98.55% using dropout 0.25 and 0.1, respectively. The improved result can be observed in MDR and FDR. Certainly, the use of dropout improves the MDR from 10% to 2% and FDR from 12% to 2%. The improvement is observed better using CNN where MDR is



(a) 50CNn.png

(b) 50cnn01.png

(c) 50CNN D025.png

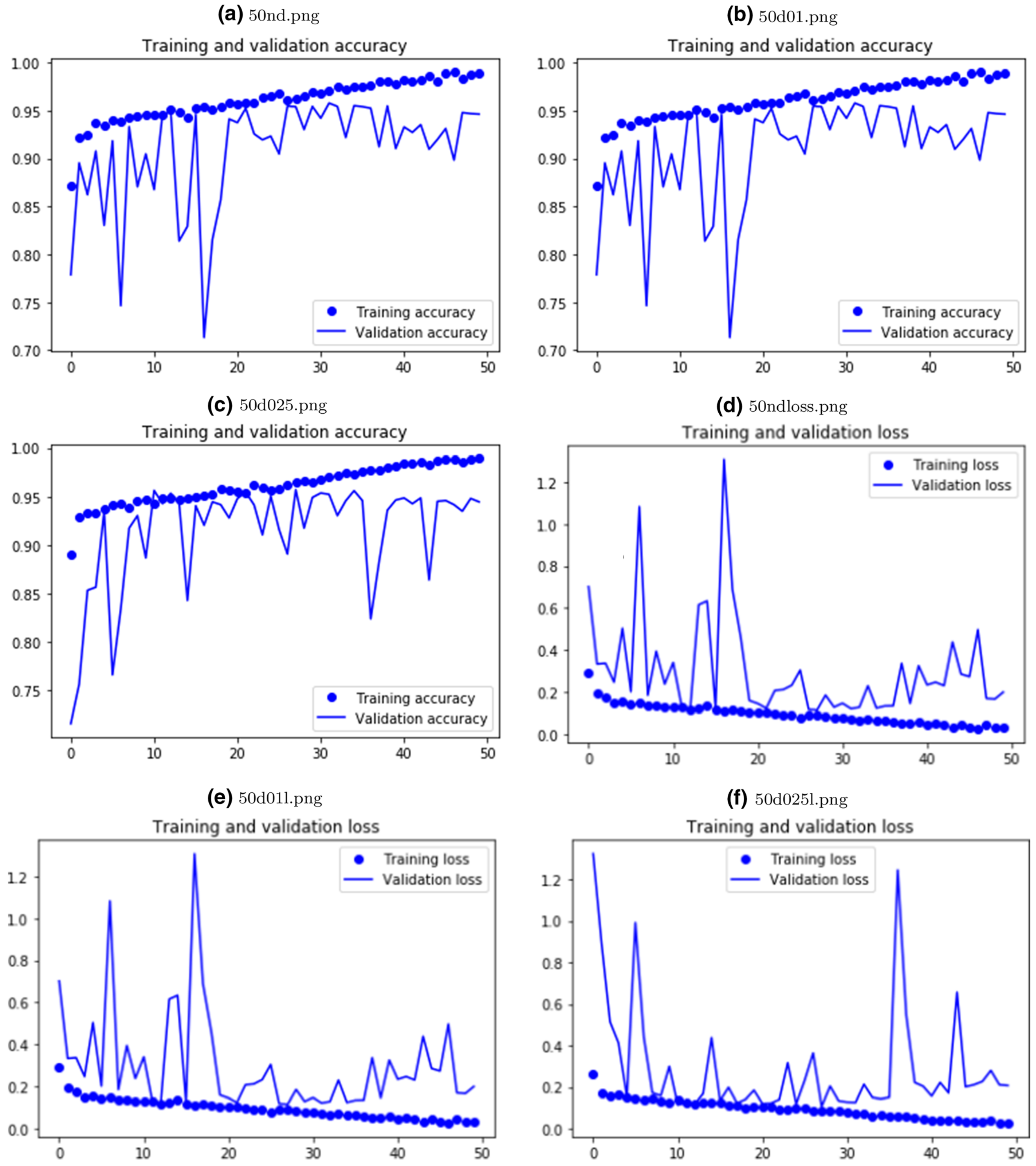
(d) 50cnnl.png

(e) 50cnn01l.png

(f) 50cnnd025l.png

Fig. 16 Training and validation accuracy and loss graph of CNN at epochs =50 such that **a** accuracy graph,dropout=NA, **b** accuracy graph, dropout=0.25, **c** accuracy graph, dropout=0.1, **d** loss graph, dropout=NA, **e** loss graph, dropout=0.1, **f** loss graph, dropout=0.25

1% after 50 epochs in presence of dropout. The results are visually displayed in Fig. 15. The training accuracy and loss graphs are compared for CNN and SCAE. Figure 16 is the comparison of the graphical outcome of training accuracy and loss with epoch 50, and dropout NA, 0.25 and 0.1, respectively, whereas Fig. 17 is the comparison of the graphical outcome of training accuracy and loss with epoch 50, and dropout NA, 0.25 and 0.1, respectively. The graphs in Fig. 17 show better stability.



(a) 50nd.png

(b) 50d01.png

(c) 50d025.png

(d) 50ndloss.png

(e) 50d01l.png

(f) 50d025l.png

Fig. 17 Training and validation accuracy and loss graph of SCAE at epochs = 50 such that **a** accuracy graph,dropout = NA, **b** accuracy graph, dropout = 0.1, **c** accuracy graph, dropout = 0.25, **d** loss graph, dropout = NA, **e** loss graph, dropout = 0.1, **f** loss graph, dropout = 0.25

The impact of epochs on SCAE is observed in the form of segmentation SA using epochs = 50,100,1000 and dropout 0.25. The SA is improved from epoch 50 to 100. But the results are better at epoch count 100 in comparison with 1000. This establishes the fact that epoch count upper limit is necessary. Over-learning will not produce better results.

On comparing SCAE response of segmentation SA, MDR and FDR over different epochs = 50, 100 and 400, it is observed in Table 4 that the results are certainly better in

Table 6 Segmentation accuracy comparison using CNN (epoch = 50, dropout = 0.25) and SCAE (epoch = 50,100,1000, dropout = NA, 0.1 and 0.25)

SA (in %) Patch_Size= 28, K=10, Test_size = 20%						
Technique	SCAE					CNN
Epochs	50			100	1000	50
Fold Number	Without Dropout	With Dropout (0.1)	With Dropout (0.25)	With Dropout (0.25)	With Dropout (0.25)	With Dropout (0.25)
1	98.2175	98.6041	96.3814	98.1262	90.2018	97.7611
2	97.4605	95.3505	98.4914	98.8457	98.6738	98.4484
3	98.7329	98.5289	97.9759	98.1209	98.6094	98.5933
4	98.7973	98.7651	98.502	98.8027	98.577	98.7598
5	98.7598	98.2336	98.49135	98.6685	98.4752	98.749
6	98.6738	98.5504	98.137	98.5665	98.3732	98.6685
7	98.7222	98.6846	98.2336	98.8349	98.4537	98.2361
8	98.4645	98.2497	98.3625	98.7061	98.4162	98.2235
9	98.6577	98.647	98.5987	98.7114	98.4376	98.2738
10	88.4569	98.5128	98.4591	98.7383	98.3839	98.6953
Average (%)	97.49	98.21	98.16	98.61	97.66	98.4
Std Dev (+/-)	3.04	0.97	0.62	0.2564	2.49	0.28

presence of dropout. But the new observation is that SCAE attains comparative results to CNN with a better epoch count of 400 and dropout of 0.1. The SA, MDR and FDR at 400 epochs are 98.45%, 1% and 1%, respectively.

There is a significant change with controlled learning of parameters due to dropout. Hence, in conclusion, the controlled passage of information to the classifier produces better SA and reduced MDR and FDR.

4.3.3 Repeatability and reproducibility of the model

The model shows a pattern of reduced MDR and FDR using CNN or SCAE. The model when observed over different epochs, shows fluctuating behaviour. The use of K-fold cross-validation helps in deciding the parameters with the stable model. Table 6 is the outcome of SA using 10-fold cross-validation. Table 6 shows the comparison of SCAE performance with epoch count 50 in the absence vs presence of dropout values 0.25 and 0.1 and shows a decent jump in SA from the absence of dropout to the presence of dropout with a significant standard deviation of 3.04 in absence of dropout to 0.62(min) in presence of 0.25 dropout. The overall table shows a reading of 88% SA at k=10 with SCAE, with no dropout. Such a behaviour can be expected out of the models and hence in a large amount of data along with noisy data, k=10 is suitable.

Table 6 also displays the comparison of the model performance of SCAE and CNN. SCAE over epoch count 50,100 and 1000 in presence of dropout (0.25) is compared. The comparison can show that the 1000 epoch count shows a poor start and high standard deviation, whereas epoch 100 shows a stable and repeatable SA count. Figure 18 is the graphical view of the comparison.

Table 6 and Fig. 18 also show the comparison of SCAE with CNN cross-validation behaviour with epoch 50. As stable as SCAE, CNN shows stable results but lesser SA than SCAE 100 epoch count and a comparable standard deviation. This can be seen in future experiments on how a naive CNN will behave in comparison with pre-trained CNN in SCAE with comparable stable epochs.

Now, although SA is best attained at epoch 100 SCAE, a better MDR and FDR are obtained with epochs 400. The results are better than CNN at epoch 50. The outcome with SCAE is comparable with CNN with higher epochs due to pre-training provided. The outcome is obtained by SCAE with SA 98.45% and MDR and FDR of 1% each with dropout 0.25, but the final results are better with CNN alone even at epochs 50, dropout 0.1 with SA 98.55%, MDR and FDR of 1% each. The value for real-valued data such as image patches, the value inclination should be close to ideal 0.5; hence, the choice of dropout 0.25 is considered with SCAE producing SA 98.45%.

4.3.4 Comparative evaluation

The comparison of the performance outcome of past published techniques using learning and non-learning-based systems is shown in Table 7. The table also contains the outcome of the proposed work in comparison. The performance is better than previously published work with

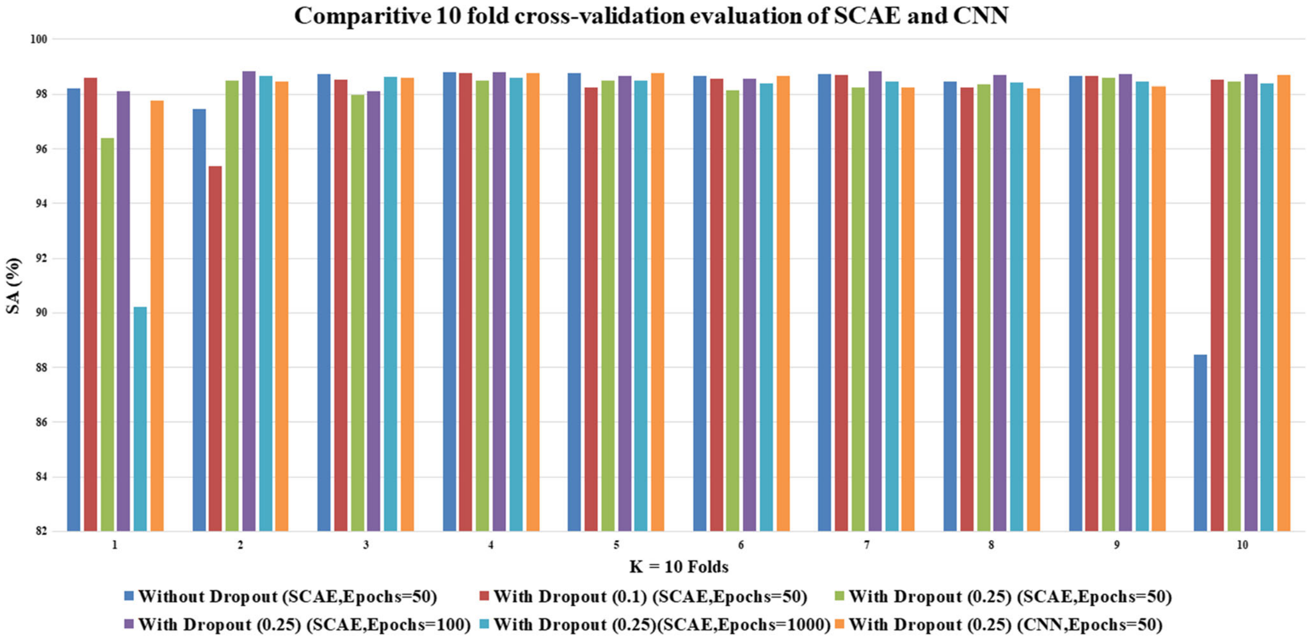


Fig. 18 Graphical representation of segmentation accuracy comparison using CNN (epoch = 50, dropout = 0.25) and SCAE (epoch = 50,100,1000, dropout = NA, 0.1 and 0.25)

Table 7 Comparative analysis of proposed work with recently published work using deep neural networks experimented on IIIT-D CLF database

References	Approach	SA (in %)	MDR (in %)	FDR (in %)
[18]	Fractal Dimensions Weighted extreme learning machine (deep neural network)	90.07	10.07	6.38
[28]	Patch-oriented classification using CNN	94.44	10.5	4.7
	Proposed Hybrid approach	98.45	1	1

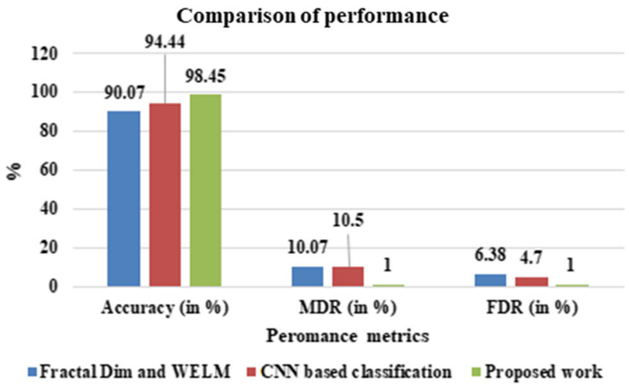


Fig. 19 Graphical representation of comparative analysis of proposed work with recently published work using deep neural networks experimented on IIIT-D CLF database

IIIT-D CLF data with images SNR of more than 2.5. The following points enlightens the improvements in comparison to existing state-of-the-art techniques:

(1) Reduced processing area of the image for the deep learning-based outcome, in the form of extracted contour regions.
(2) Detection of multiple instances due to contour extraction in the first stage of the hybrid proposed system.
(3) Better performance as a hybrid learning-based system with a stable patch size of 28*x*28, effective feature engineering using SCAE and reduced overfitting with dropout.

Table 7 shows the comparison of the performance of the proposed work and other popular and published segmentation and detection techniques using deep neural networks and experimentation performed on the IIIT-D database. Figure 19 shows a graphical representation of the performance metrics of Table 7. The comparison shows the improved SA of the proposed work due to the use of the hybrid approach.

4.4 Recommendations and discussions

The proposed method and empirical evaluation suggest that the system is over efficient and effective. SCAE suggests the following findings and recommendations:

(1) Images below SNR 2.5 are discarded to avoid a)extracting imposter contours and b)learning noise more than the actual signal.
(2) Simple and effective SCAE feature learning mechanism.
(3) Hybrid early detection-cum-classification using stack method for better training.
(4) Dropout as regularization technique to avoid overfitting.

Future investigations considering the results obtained, the following points are suggested:

(1) Different databases such as NIST SD, WVU can be used in experiments for the combination of training and testing. A large amount of data with different noise levels can train the model better and make it a generic solution.
(2) Different masks can be used to include images below SNR 2.5. Apart from colour information, other features such as gradient information, ridge information can be used to differentiate fingerprint from the background.
(3) Different model architecture can be experimented with using different counts of layers. Different optimizers such as SGD can produce stable results along with different activation functions.
(4) Dropout vs batch normalization as regularization techniques can experiment.
(5) Forced overfitting validation to check if overfitting is occurring in a large volume of such noisy data and to what extent. As here, 1000 count of the epoch was identified as the extended limit, and hence, overfitting control measures are applied.
(6) Segmentation on a generalized quality of image irrespective of noise level and noise type, colourmap range for small and large histogram range of the images, a different voting mechanism for early detection of CoIs since saliency detection works better with non-overlapping salient regions. The same application can work better with additional feature sets and feature reduction techniques. Thereby, the segmentation can be improved along with handling technique and performance trade-off.

5 Conclusion

Fingerprint segmentation is required to separate the relevant information from irrelevant information of the image to improve the accuracy of the proceeding steps of fingerprint matching. The major goal of segmentation is to extract the relevant ridge areas of the image accurately; the major challenge of the latent fingerprint is the noisy background overlapping with ridges. This paper has presented and investigated the dual approach of latent fingerprint segmentation. We applied an early distinction of potentially relevant areas of the latent fingerprint image using masks based on the colour and saliency feature of the image. These relevant areas or CoIs can be multiple in a single with single, multiple, partial fingerprints or regions similar to fingerprints called as falsely detected regions, or none if there are no fingerprints in the image. Hence, the irrelevant images are not processed and relevant images are processed only in compartmentalised regions. Where the process guarantees noise reduction, better segmentation is proposed by the application of deep learning to the collection of these CoIs. For that purpose, the classification of equal-sized patches of these CoIs is fed to SCAE. The patches of these CoIs are fed to an SCAE for classification into relevant fingermark or imposter background noise with salient importance in the image. The use of early distinction along with the patch-based technique substantially reduced the misclassification rate and false classification rate. The stack of CAE, an unsupervised method for feature extraction, is used to pre-train CNN. Pre-trained CNN-based classification outperforms the published results of CNN-based classification. Our model was tested on the IIIT-D database, and it outperformed recently published methods in terms of segmentation accuracy, detection rates and execution times. In future work, we can train the model with a database with more images and train on bad to ugly images and fine-tune the model with different CNN layers to achieve a trade-off between accuracy and MDR and FDR with increasing noisy images in experimentation.

Acknowledgements The authors are thankful to IIITD for providing the latent fingerprint database. We would also like to acknowledge UPES, Dehradun, India, for providing the computing resources for experimentation.

Data availability There is IIITD Latent fingerprint dataset with 150 classes and 1046 images. The database can be obtained go to the url:http://www.iab-rubric.org/resources.html.

Declarations

Conflict of Interest The authors find no conflict of interest.

References

1. Agarwal D, Bansal A (2021) A utility of pores as level 3 features in latent fingerprint identification. Multimed Tools Appl 80(15):23605–23624
2. Ali H, Sharif M, Yasmin M, Rehmani MH (2020) Colour-based template selection for detection of gastric abnormalities in video endoscopy. Biomed Signal Process Control 56(101668):101668
3. Amin J, Sharif M, Gul N, Raza M, Anjum MA, Nisar MW, Bukhari SAC (2019) Brain tumor detection by using stacked autoencoders in deep learning. J Med Syst 44(2):32
4. Azman AR, Mahat NA, Wahab RA, Ahmad WA, Puspanadan JK, Huri MAM, Kamaluddin MR, Ismail D (2021) Box-behnken design optimisation of a green novel nanobio-based reagent for rapid visualisation of latent fingerprints on wet. Non-porous substrates. Biotechnol Lett 43(4):881–898
5. Baur C, Denner S, Wiestler B, Navab N, Albarqouni S (2021) Autoencoders for unsupervised anomaly segmentation in brain mr images: a comparative study. Med Image Anal 69(101952):101952
6. Borji A (2015) What is a salient object? A dataset and a baseline model for salient object detection. IEEE Trans Image Process 24(2):742–756
7. Cao K, Jain AK (2015) Latent orientation field estimation via convolutional neural network. In: 2015 International conference on biometrics (ICB). IEEE
8. Chandraprabha K, Akila S (2019) Texture feature extraction for batik images using glcm and glrlm with neural network classification. Int J Sci Res Comput Sci Eng Inf Technol 06–15
9. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
10. Chhabra M, Shukla MK, Ravulakollu KK (2020) Boosting the classification performance of latent fingerprint segmentation using cascade of classifiers. Intell Decis Technol 14(3):359–371
11. Chhabra M, Shukla MK, Ravulakollu KK (2021) Bagging- and boosting-based latent fingerprint image classification and segmentation. Advances in intelligent systems and computing. Springer, Singapore, pp 189–201
12. Chhabra M, Shukla MK, Ravulakollu KK (2021) State-of-the-art: a systematic literature review of image segmentation in latent fingerprint foren. Recent Pat Comput Sci 13(6):1115–1125
13. Cornia M, Baraldi L, Serra G, Cucchiara R (2018) Paying more attention to saliency: image captioning with saliency and context attention. ACM Trans Multimed Comput Commun Appl 14(2):1–21
14. Daluz HM (2021) Courtroom testimony for fingerprint examiners. CRC Press, Florida
15. Deufel B, Mueller C, Duffy G, Kevenaar T (2013) BioPACE: biometric passwords for next generation authentication protocols for machine-readable travel documents. Datenschutz Datensicherheit - DuD 37(6):363–366
16. Diyasa GSM, Fauzi A, Idhom M, Setiawan A (2021) Multi-face recognition for the detection of prisoners in jail using a modified cascade classifier and CNN. J Phys Conf Ser 1844(1):012005
17. Ezeobiejesi J, Bhanu B (2016) Latent fingerprint image segmentation using fractal dimension features and weighted extreme learning machine ensemble. In: 2016 IEEE conference on computer vision and pattern recognition workshops (CVPRW). IEEE
18. Ezeobiejesi J, Bhanu B (2017) Latent fingerprint image segmentation using deep neural network. Deep learning for biometrics. Springer International Publishing, Cham, pp 83–107
19. Fan R, Li X, Lee S, Li T, Zhang HL (2020) Smart image enhancement using CLAHE based on an F-shift transformation during decompression. Electronics 9(9):1374
20. Haralick R, Shanmugam K, Dinstein I (1973) Textural features for image classification. IEEE Trans Syst Man Cybernet SMC 3(6):610–621
21. Henry ER (1900) Classification and uses of fingerprints. Routledge & Sons, London
22. Hsiao C-T, Lin C-Y, Wang P-S, Wu Y-T (2022) Application of convolutional neural network for fingerprint-based prediction of gender, finger position, and height. Entropy (Basel) 24:475. https://doi.org/10.3390/e24040475
23. Huang F, Zhang J, Zhou C, Wang Y, Huang J, Zhu L (2020) A deep learning algorithm using a fully connected sparse autoencoder neural network for landslide susceptibility prediction. Landslides 17(1):217–229
24. Igoe DP, Parisi AV, Downs NJ, Amar A, Turner J (2018) Comparative signal to noise ratio as a determinant to select smartphone image sensor colour channels for analysis in the UVB. Sensors and actuators. A, Physical 272:125–133
25. Jalali S, Boostani R, Mohammadi M (2021) Efficient fingerprint features for gender recognition. Multidimens Syst Signal Process. https://doi.org/10.1007/s11045-021-00789-6
26. Johnson BT, Riemen JAJM (2019) Digital capture of fingerprints in a disaster victim identification setting: a review and case study. Forensic Sci Res 4(4):293–302
27. Kalka ND, Beachler M, Hicklin RA (2020) LQMetric: a latent fingerprint quality metric for predicting AFIS performance and assessing the value of latent fingerprints. JFI 70(4):443–463
28. Khan AI, Wani MA (2019) Patch-based segmentation of latent fingerprint images using convolutional neural network. Appl Artif Intell AAI 33(1):87–100
29. Kharghanian R, Peiravi A, Moradi F, Iosifidis A (2021) Pain detection using batch normalized discriminant restricted boltzmann machine layers. J Vis Commun Image Represent 76(103062):103062
30. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
31. Kumar AS, Nair JJ (2019) Pair wise training for stacked convolutional autoencoders using small scale images. J Intell Fuzzy Syst 36(3):1987–1995
32. Li G, Yu Y (2016) Visual saliency detection based on multiscale deep CNN features. IEEE Trans Image Process Pub IEEE Signal Process Soc 25(11):5012–5024
33. Liu Y, Zhang Y, Coleman S, Bhanu B, Liu S (2020) A new patch selection method based on parsing and saliency detection for person reidentification. Neurocomputing 374:86–99
34. Liu Yisi, Wang X, Wang L, Liu D (2019) A modified leaky ReLU scheme (MLRS) for topology optimization with multiple materials. Appl Math Comput 352:188–204
35. Maltoni D, Maio D, Jain AK, Prabhakar S (2009) Handbook of fingerprint recognition. Springer-Verlag, Berlin
36. Marmanis D, Schindler K, Wegner JD, Galliani S, Datcu M, Stilla U (2018) Classification with an edge: improving semantic image segmentation with boundary detection. ISPRS J Photogramm Remote Sens Off Pub Int Soc Photogramm Remote Sens (ISPRS) 135:158–172
37. Masci J, Meier U, Cireşan D, Schmidhuber J (2011) Stacked convolutional auto-encoders for hierarchical feature extraction. Lecture notes in computer science. Springer, Berlin Heidelberg, pp 52–59
38. Mateescu VA, Bajic IV (2016) Visual attention retargeting. IEEE Multimed 23(1):82–91

39. Mehtre BM, Murthy NN, Kapoor S, Chatterjee B (1987) Segmentation of fingerprint images using the directional image. Pattern Recogn 20(4):429–435
40. Mújica-Vargas D, Kinani JMV, de Rubio J, J. (2020) Color-based image segmentation by means of a robust intuitionistic fuzzy C-means algorithm. Int J Fuzzy Syst 22(3):901–916
41. Murshed MGS, Kline R, Bahmani K, Hussain F, Schuckers S (2021) Deep slap fingerprint segmentation for juveniles and adults. In arXiv [cs.CV]. http://arxiv.org/abs/2110.04067
42. Nguyen D-L, Cao K, Jain AK (2018) Automatic latent fingerprint segmentation. In: 2018 IEEE 9th international conference on biometrics theory, applications and systems (BTAS). IEEE
43. Prasad V, Prasad L, Lukose S, Agarwal P (2021) Latent fingerprint development by using silver nanoparticles and silver nitrate-A comparative study. J Forensic Sci 66(3):1065–1074
44. Prost J, Cihak-Bayr U, Neacşu IA, Grundtner R, Pirker F, Vorlaufer G (2021) Semi-supervised classification of the state of operation in self-lubricating journal bearings using a random forest classifier. Lubricants (Basel, Switzerland) 9(5):50
45. Saguy M, Almog J, Cohn D, Champod C (2021) Proactive forensic science in biometrics: novel materials for fingerprint spoofing. J Forensic Sci. https://doi.org/10.1111/1556-4029.14908
46. Sankaran A, Vatsa M, Singh R (2014) Latent fingerprint matching: a survey. IEEE Access: Pract Innov Open Solut 2:982–1004
47. Sankaran A, Jain A, Vashisth T, Vatsa M, Singh R (2017) Adaptive latent fingerprint segmentation using feature selection and random decision forest classification. Int J Inf Fusion 34:1–15
48. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw Off J Int Neural Netw Soc 61:85–117
49. Shenoy ES, Rosenthal ES, Shao Y-P, Biswal S, Ghanta M, Ryan EE, Suslak D et al (2018) Real-time, automated detection of ventilator-associated events: avoiding missed detections, misclassifications, and false detections due to human error. Infect Control Hosp Epidemiol Off J Soc Hosp Epidemiol Am 39(07):826–833
50. Shi J, Yan Q, Xu L, Jia J (2016) Hierarchical image saliency detection on extended CSSD. IEEE Trans Pattern Anal Mach Intell 38(4):717–729
51. Singh SP, Ayub S, Saini JP (2021) Analysis and comparison of normal and altered fingerprint using artificial neural networks. Int J Knowl Based Intell Eng Syst 25(2):243–249
52. Sonali S, Sahu AK, Singh S.P. Ghrera, Elhoseny M (2019) An approach for De-noising and contrast enhancement of retinal fundus image using CLAHE. Opt Laser Technol 110:87–98
53. Stojanović B, Marques O, Nešković A (2019) Latent fingerprint datasets. Segmentation and separation of overlapped latent fingerprints. Springer International Publishing, Cham, pp 9–20
54. Stojanović B, Marques O, Nešković A (2019) Machine learning based segmentation of overlapped latent fingerprints. Segmentation and separation of overlapped latent fingerprints. Springer International Publishing, Cham, pp 29–34
55. Sudharshan PJ, Petitjean C, Spanhol F, Oliveira LE, Heutte L, Honeine P (2019) Multiple instance learning for histopathological breast cancer image classification. Expert Syst Appl 117:103–111
56. Tembusai ZR, Mawengkang H, Zarlis M (2021) K-nearest neighbor with K-fold cross validation and analytic hierarchy process on data classification. Int J Adv Data Inf Syst. https://doi.org/10.25008/ijadis.v2i1.1204
57. Venosa AD, King DW, Sorial GA (2002) The baffled flask test for dispersant effectiveness: a round robin evaluation of reproducibility and repeatability. Spill Sci Technol Bull 7(5–6):299–308
58. Wan GC, Li MM, Xu H, Kang WH, Rui JW, Tong MS (2020) XFinger-net: pixel-wise segmentation method for partially defective fingerprint based on attention gates and U-net. Sensors (Basel, Switzerland) 20(16):4473
59. Wang S-H, Muhammad K, Hong J, Sangaiah AK, Zhang Y-D (2020) Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. Neural Comput Appl 32(3):665–680
60. Wang H-J, Hou W-Y, Kang J, Zhai X-Y, Chen H-L, Hao Y-W, Wan G-Y (2021) The facile preparation of solid-state fluorescent carbon dots with a high fluorescence quantum yield and their application in rapid latent fingerprint detection. Dalton Trans (Cambridge, England: 2003) 50(35):12188–12196
61. Xu X, Mu N, Chen L, Zhang X (2016) Hierarchical salient object detection model using contrast-based saliency and color spatial distribution. Multimed Tools Appl 75(5):2667–2679
62. Yan Y, Nie F, Li W, Gao C, Yang Y, Xu D (2016) Image classification by cross-media active learning with privileged information. IEEE Trans Multimed 18(12):2494–2502
63. Yang D, Karimi HR, Sun K (2021) Residual wide-kernel deep convolutional auto-encoder for intelligent rotating machinery fault diagnosis with limited samples. Neural Netw Off J Int Neural Netw Soc 141:133–144
64. Yuan X, Feng Z, Norton M, Li X (2019) Generalized batch normalization: towards accelerating deep neural networks. Proceedings of the ...AAAI conference on artificial intelligence. AAAI Conf Artif Intell 33:1682–1689
65. Zdilla MJ, Hatfield SA, McLean KA, Cyrus LM, Laslo JM, Lambert HW (2016) Circularity, solidity, axes of a best fit ellipse, aspect ratio, and roundness of the foramen ovale: a morphometric analysis with neurosurgical considerations: a morphometric analysis with neurosurgical considerations. J Craniofac Surg 27(1):222–228
66. Zhang Y, Gao C, Li Z, Lv Y, Zhu K (2021) A method of fingermark anti-counterfeiting for forensic document identification. Pattern Recogn Lett 152:86–92
67. Zhou D-X (2020) Theory of deep convolutional neural networks: downsampling. Neural Netw Off J Int Neural Netw Soc 124:319–327
68. Zhou H, Yu G (2021) Research on fast pedestrian detection algorithm based on autoencoding neural network and adaboost. Complexity 2021:1–17
69. Zhu F, Kong X, Fu H, Tian Q (2018) A novel two-stream saliency image fusion CNN architecture for person re-identification. Multimedia Syst 24(5):569–582

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.