International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybrid Approach for Biometric Recognition: Integrating Custom Vector Quantization and CNN-Based Feature Extraction

Geetanjali Sawant[1], Vinayak Bharadi[2], Kaushal Prasad[3], Pravin Jangid[4]

Submitted: 24/04/2023 **Revised:** 27/06/2023 **Accepted:** 06/07/2023

Abstract: A biometric recognition is performed with feature extraction, matching, and classification. Before the emergence of deep learning, biometric recognition has completely relied on manual feature extraction. Convolutional neural networks have automated feature extraction. To fetch features manually, domain knowledge and programming expertise are required. A dataset quality affects accuracy of a shallow classifier whereas the performance of a deep learning model succeeds in providing high accuracy only if the training dataset is balanced, qualitative, and large enough to distinguish features from various classes. Constructing a classifier from a large training dataset is time-consuming and causes overfitting. On the other hand, a small dataset-based model suffers from underfitting. To overcome the said issues, this paper proposed a hybrid approach of a concatenation of manually extracted domain-independent features such as Kekre's Median Codebook and Kekre's Fast Codebook and automatically extracted features through CNNs by processing samples from physiological and behavioral biometric traits independently and feeding these to neural networks to achieve best possible accuracy of classification so that the possibility of underfitting and overfitting is avoided. This method is evaluated by applying it to LFW, UPOL, IITD, IITD V1, and UserSignatureDatabase datasets of face, iris, fingerprint, palmprint, and signature respectively, and resulting models achieved improved (certain models achieved equivalent accuracy) with reduced memory and learning time.

Keywords: *Multimodal, Unimodal Biometric System, Kekre's Median Codebook, Kekre's Fast Codebook, Feature Integration*

1. Introduction

Biometric authentication system performs feature extraction, matching and classification based on matching score [1][2]. The vast research has been carried out for classification through shallow classifiers such as SVM, Baye's classification, Decision tree, Random Forest etc. These techniques have a very efficient and effective performance for extracting conventional discriminative features [3] provided that dataset is qualitative. The new era of artificial intelligence has shifted the research interest from shallow classifiers to Deep Neural Network (DNN) based classifiers. Deep neural network involves convolutional layers, pooling layers and fully connected/ dense layers. Convolutional layers aim to generate feature map from the input by applying filters of specified size, pooling layers affect dimensions of feature map and dense layer labels the provided input. DNN requires large volume of data for training. This dataset required to be preprocessed to obtain balanced and cleaned data for training. Model training upon huge amount of dataset becomes computationally expensive and may cause overfitting whereas small dataset causes underfitting due to insufficient training. One way to avoid large computations is to apply transfer learning where the parameters from pretrained networks are shared by the dataset in hand. It saves computational time and memory involved to train model, however prediction ability of such model depends on the relevance between dataset used by original network and the dataset, which share theses parameters. When dataset is small, insufficient-features-based training leads to high misclassification. A Siamese network successfully handles small dataset, by pairing inputs from same class as well as from different classes to form positive and negative pair respectively. DNN combined with Siamese network may aid in reducing misclassification rate in cost of increased time and memory requirement, whereas if number of classes are too many and sufficient positive and negative pairs are not learnt then the model may fail to give the highest possible accuracy[4]. Monica Bianchini presented theoretical results, to support claim that deep learning techniques are effective, but these results are still few and incomplete [5]. Many factors are there such as quality, domain, size of dataset, dimension of dataset-sample, balanced or imbalanced dataset with respect to samples per class, temporal dependent or independent nature of dataset, semantic correlation among samples from dataset; affect the selection of technique to build classifier. These factors also contribute in the selection of biometric trait to build authentication system. Kitsuchart Pasupa, et al. has statistically confirmed that the shallow models achieved

[1]*Department of Information Technology, Research Scholar, Finolex Academy of Management and Technology, India*
[2]*Department of Information Technology, Professor, Finolex Academy of Management and Technology, Ratnagiri, India*
[3]*Department of Mechanical Engineering Professor, Finolex Academy of Management and Technology, Ratnagiri, India*
[4]*Department of Information Technology, Research Scholar, Finolex Academy of Management and Technology, India*

better performance than the deep model that did not use a regularization technique. However, a deep model augmented with a regularization technique-CNN with dropout technique-was competitive to the shallow models [6]. Ali Bou Nassif et al. presented comparative study of semantic analysis of Arabic reviews. He found that random forest is best shallow classifier but transformer (ANN) outperformed when it uses araBERT [7]. Guk Bae Kim experimented with SVM classifier and Deep Neural Network (DNN) to classify the regional pattern of Fiffuse Lung Disease and found DNN as the winner. He claimed that with every additional convolutional layer misclassification rate drops significantly to provide boosted accuracy [8]. As quality of dataset matters while choosing appropriate technique between shallow and deep learning to construct classifier. Xu-Cheng Yin et al. proposed way to inspect training image qualities and depending upon clarity of image, choose shallow classifiers or deep learning classifiers to proceed further with recognition of scene characters and text. Recognition from blurred and small images was done through deep learning techniques [9]. Context of data also contributes in deciding features to be extracted through shallow classifiers. Some of hand-crafted feature descriptors such as SURF, and HOG-3D are used for action recognition as well as some other shape and movement-based feature descriptors [10]-[17] used for behavioral characteristics detection, but these methods have several limitations. Hand-crafted methods need descriptors, unique-designed feature identifiers, and vocabulary build approaches for representation and extracting features. This feature engineering mechanism is difficult and needs experience and expertise from respective domain. This encouraged them to use hybrid model consisting of neural network layers to fetch features and perform classification through shallow classifier. Le Yang et al. had performed depression analysis through hybrid framework, which independently process images using CNNs and textual data using SVM and had applied decision based fusion to obtain final classification [18]. A hybrid model with support vector machines (SVM) and long short-term memory (LSTM) to represent temporal relationship among samples was built and evaluated by Jacek Haneczok, Jakub Piskorski. The task focuses on linking event templates automatically extracted from online news by an existing event extraction system, which contain only short text snippets, and potentially erroneous and incomplete information. Results of the performance of explored shallow learning methods such as decision tree-based random forest and gradient boosted tree ensembles (XGBoost) along with kernel-based (SVM) were presented comparing shallow learners with deep learning approach based on long short-term memory (LSTM) recurrent neural network [19].

Selection of samples for training do affect prediction capability of model. Techniques such as Adaboosting, ensembling improve classifier's prediction power. Boonyawee Grodniyomchai had proposed hybrid model by adopting the AdaBoost algorithm to adjust the weights of weak classifiers to build a strong classifier from odor dataset [20]. Quang Tri Chiem et al. explained their two-Stage 3D Object Recognition from Orthogonal Projections. He has performed 2-stage classification, 3D Object Recognition. In the first stage, a pre-trained Convolutional Neural Network (CNN) classifier (AlexNet) is fine-tuned to generate vectors of soft labels representing class probabilities. These vectors (or the first-stage CNN features) are passed to a shallow neural network trained to generate the final class labels. While the first stage classifier is trained directly on the physical data, the second stage classifier is trained on the probability vectors (or/and features) produced by the first stage classifier, and therefore, classification outcomes of the first stage gets corrected by the second stage assessment [21].

These hybrid approaches focused on improving accuracy but have not taken into account the huge memory and learning time incurred in training the deep learning model. Takayuki Hoshinoet et al. used style transfer mapping (STM) as a data-space-based transfer learning method and fine-tuning (FT) as a parameter-space-based transfer learning method to apply Artificial neural network. The combined use of parameter-space-based transfer learning and deep classifiers has effectively reduced the data measurement time of surface electromyogram (sEMG)-based human-computer interface (HCI) applications [22].

This paper has proposed a hybrid approach of classification through concatenation of manually extracted features with the features fetched from CNNs. Kekre's median codebook (KMC) and Kekre's Fast Codebook (KFC) are Vector Quantization(VQ) based feature vectors. VQ is a proven method used in lossy compression. KMCG and KFCG are the methods used to generate codebooks in less computations compared to Linde Buzo Gray (LBG) algorithm. This codebook is used in variety of research fields such as speech recognition and face detection, pattern recognition, speech data compression, image segmentation, Content Based Image Retrieval (CBIR), Face recognition [23]-[25]. As these custom features capable to distinguish samples from input classes, a hybrid approach with which these features are combined with the features extracted automatically through CNNs is prevailed upon both deep learning classification and shallow classification as standalone technique.

2. Method

A multi-algorithmic, multi-instance or multimodal system can be developed by performing fusion on features or decisions as shown in Figure 1. We have proposed a hybrid approach for biometric recognition. For evaluation of the said approach, three ways are used to perform feature level fusion i.e. to concatenate hand crafted Kekre's Median Codebook(KMC) and Fast Codebook(KFC) features with automatically extracted features from Convolutional Neural Networks:

1. Unimodal multi-algorithmic Feature Level Fusion.
2. Unimodal multi-network Feature Level Fusion
3. Multimodal Feature Level Fusion

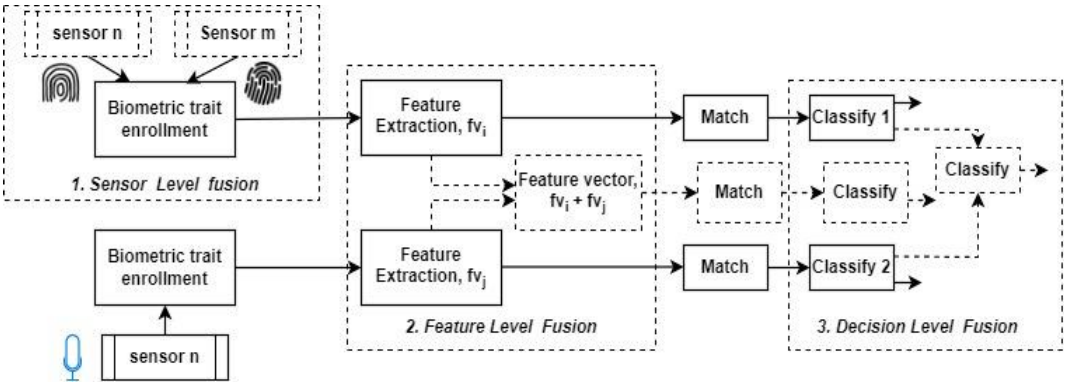


Fig 1. Sensor/Feature/Decision Level fusion in Biometric Authentication system

2.1 Algorithms:

Kekre's Median Codebook and Kekre's Fast Codebook are prepared by implementing the following algorithms:

2.1.1 Kekre's Median Codebook Generation (KMCG) Algorithm:

1. 2-dimensional array representing an image, is partitioned into 2x2 sized non overlapping blocks.
2. These are arranged in row to obtain total 12 values from 4 pixels (Each pixel with red, green and blue channel).
3. Median of first column is used to partition entire dataset into two sets $\{X_{11}, X_{12}, X_{13},….\}$ and $\{X_{21}, X_{22}, X_{23},….\}$ respectively.
4. Each of two sets is further partitioned into two sets by considering median of second column as splitting value. The set with median M_i is partitioned into two clusters with centroid M_{i1} and M_{i2} by assigning X_{ij} to cluster 1 if $X_{ij} < M_{i1}$ else it will be assigned to cluster 2, where j=1,2,…

Clusters are represented with i=1,2

5. Step 4 is repeated till the number of medians equals the size of the codebook.
6. Partitioning is repeated till the desired number of sets are created, maximum 2N, where N is number of columns. We have performed partitioning for 7 times to get codebook of size of 128 x 12.

The codebook is stored as the feature vector for the image[26]-[28].

2.1.2 Kekre's Fast Codebook Generation (KFCG) Algorithm:

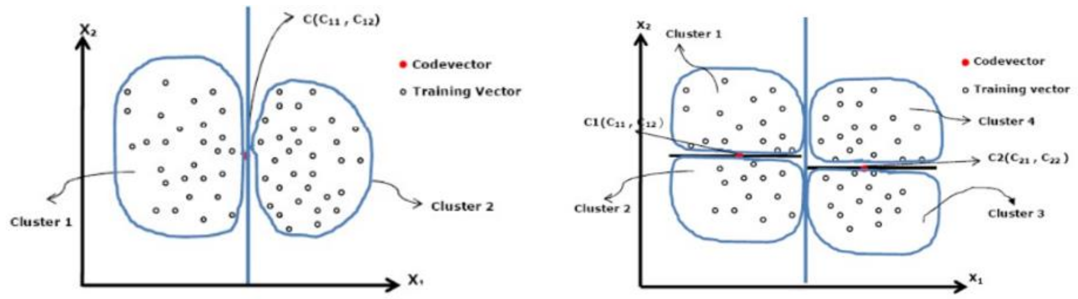
KFCG aims to generate codebook with minimum computations.

1. First two steps from KMCG are followed to get the code vector for the fed input. Code vector is represented as $\{X_1, X_2, …….X_{mn}\}$.

2. A mean of this code vector C is used to partition input vectors from code vector into two clusters with centroids C_1 and C_2. These two clusters are represented as $\{X_{11}, X_{12}, X_{13},….\}$ and $\{X_{21}, X_{22}, X_{23},….\}$ respectively.

The set with centroid C_i is partitioned into two clusters with centroid C_{i1} and C_{i2} by assigning X_{ij} to cluster 1 if $X_{ij} < C_{i1}$ else it will be assigned to cluster 2, where j=1,2,…

3. Step 3 is repeated till the number of centroids equals the size of the codebook as shown in Figure 2.



a) b)

Fig 2. Recursive clustering to obtain KMC & KFC

This codebook is used as the feature vector of the input sample[29][30].

2.2 Datasets:

The features are drawn from the following datasets:

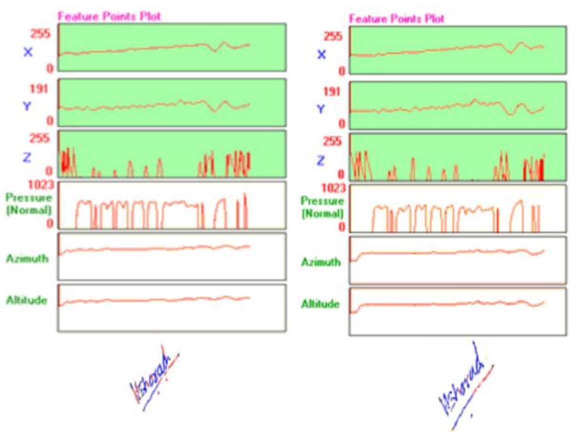
2.2.1 University of Palackého and Olomouc in the Czech Republic (UPOL) : This iris dataset consists of total 384 images of 64 subjects with 3 samples of left and right iris each. The images are of size 576 x 768.

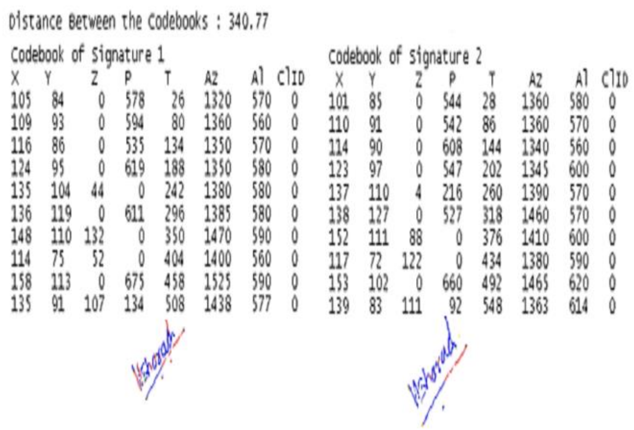
2.2.2 SD 302d and Fig Data: In September 2017, the Intelligence Advanced Research Projects Activity. (IARPA) held a data collection (SD302d) as part of Fingerprint Challenge. SD 302d fingerprint dataset contains plain fingerprint Finger dataset of Dr.V.A. Bharadi. Total 1010 images from 101 subjects used for processing.

2.2.3 IITD palmprint dataset: It consists of 5 to 6 samples of each of left and right hand of 230 users. Total 1380 images of each of left and right palm have been compiled from students and staff of IIT Delhi in duration of July 2006-2007. size 800 x 600. Samples were obtained with touchless image capturing setup.

2.2.4 Labeled Faces in Wild (LFW): This face dataset contains one to various samples of 1680 subjects, so its not balanced dataset. Total 13233 images of 250 x 250 size are available. For the said approaches of feature fusion, 2020 images from 101 classes are used. Due to unavailability of sufficient number of samples, data augmentation was performed increase size of dataset.

2.2.5 User Signature Database: This dataset is prepared by Dr. V.A. Bharadi, consisting of 10 samples of each of 101 subjects. KMC and KFC features were determined from the online characteristics of signature such as X, Y, Z coordinates, pen pressure, azimuth and altitude as shown in Figure 3.





Distance Between the Codebooks : 340.77

Codebook of Signature 1

X	Y	Z	P	T	AZ	Al	ClID
105	84	0	578	26	1320	570	0
109	93	0	594	80	1360	560	0
116	86	0	535	134	1350	570	0
124	95	0	619	188	1350	580	0
135	104	44	0	242	1380	580	0
136	119	0	611	296	1385	580	0
148	110	132	0	350	1470	590	0
114	75	52	0	404	1400	560	0
158	113	0	675	458	1525	590	0
135	91	107	134	508	1438	577	0

Codebook of Signature 2

X	Y	Z	P	T	AZ	Al	ClID
101	85	0	544	28	1360	580	0
110	91	0	542	86	1360	570	0
114	90	0	608	144	1340	560	0
123	97	0	547	202	1345	600	0
137	110	4	216	260	1390	570	0
138	127	0	527	318	1460	570	0
152	111	88	0	376	1410	600	0
117	72	122	0	434	1380	590	0
153	102	0	660	492	1465	620	0
139	83	111	92	548	1363	614	0

(a) (b)

Fig 3. Features extracted from online signature to build KMC and KFC feature vector, represent

a) plot b)Feature values

2.3 Proposed models

The multialgorithmic, multi-network and multimodal models are built by performing feature fusion, which are described as follows.

2.3.1 To perform Unimodal multi-algorithmic Feature Level Fusion:

i. Convolutional neural network(CNN) layers were applied on image data to extract features automatically and were concatenated with manually extracted Kekare's Median and fast codebook(KMCG/KFCG) feature vector generated from the same datasets. Models corresponding to KMCG and KFCG are referred as models $Model_{11}$ and $Model_{12}$. This concatenated feature vector was used to perform matching and to label the input.

ii. From keras, featureconcat method is used to perform direct feature fusion.

iii. Iris, Fingerprint, Palmprint, Face datasets were processed to obtain KFC and KMC and combined with CNN-based features to construct $Model_{11}$ and $Model_{12}$ respectively.

2.3.2 To perform Unimodal multinetwork Feature Level Fusion:

i. We applied CNN layers to get feature vector from physiological datasets and KMCG/KFCG feature vector obtained from behavioural datasets is fed to Long-short term memory (LSTM)network. These feature vectors were fused and used to classify sample.

ii. These are unimodal models since same dataset is fed to a hybrid approach involving CNNs and LSTM. models are referred to as $Model_{21}$ and $Model_{22}$.

2.3.3 To perform Multimodal Feature Level Fusion:

i. Multimodal biometric authentication systems have been proven to be more reliable, robust, and accurate than unimodal authentication system/s. Hence, we have used a hybrid approach to build a multimodal biometric system. We applied CNN layers to obtain feature vectors from physiological datasets (i.e. Iris, Fingerprint, Palmprint, and Face) and KMCG/KFCG feature vectors obtained from a behavioral dataset (UserSignatureDatabase), which was fed to Long-short term memory (LSTM)network. These feature vectors were fused and used to classify samples as shown in Figure 4.

ii. These are multimodal models developed with two different datasets and are referred to as $Model_{31}$ and $Model_{32}$ which were built from KFC and KMC respectively.

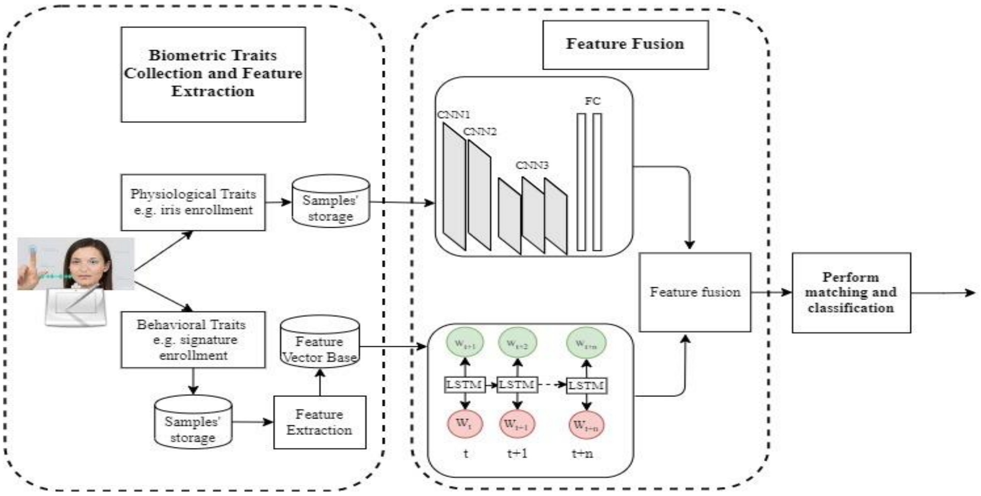


Fig 4. Multimodal classifier with feature level fusion

Concatenated features have been processed through fully connected layers. To construct a classifier, we need to specify following parameters:

i. **Loss function**: To deal with classification of multiple classes categorial cross entropy is used for defining VGG16, VGG19 and Inception network architectures. Categorical cross entropy formula is:

$$L(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log (1 - \hat{y}) \quad (1)$$

Where, y is actual class, ŷ is a predicted class.

ii. **Optimizer**: Optimizers are algorithms work for minimizing the loss and update the weights in backpropagation. Ádam' optimizer is selected for learning model and it updates weight using the formula:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \cdot \widehat{m_t} \quad (2)$$

Where,

$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta w_t}\right] v_t = \beta_2 v_{t-1} + (1 - \beta_2)\left[\frac{\delta L}{\delta w_t}\right]^2$

Where, β1 & β2 = decay rates of average of gradients.

'Adam' is an Adaptive Moment Estimation that combines both RMSprop and momentum-based GD optimizers. We found that to build a model for a small-scaled dataset, RMSprop works well.

iii. **Activation function:** The activation function decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. The purpose of the activation function is to introduce non-linearity into the output of a neuron. We used the 'softmax' activation function.

iv. **Metric**: The performance of architectures is evaluated from accuracy obtained from a constant number of epochs.

3. Experimental Results

The accuracy obtained with the implementation of said approaches varies from dataset to dataset. As they vary in the number of samples per class, dimension, and quality of the sample, the sensors used to acquire samples, etc. But irrespective of these factors, a concatenation of KMC features with abstracted features discovered from CNN has resulted in the highest accuracy among all models for each of the data sets. The models were constructed by considering the nature of input data and feature vector i.e., image data or sequential data. Table 1 represents the results which are obtained from the iris dataset based on different classifiers. To evaluate on the same scale, all models ran for 80 epochs. CNN with three layers has provided 91.86% accuracy, whereas with the hybrid approach only two CNN layers were used which has substantially reduced learning time and improved accuracy as custom features KMC and KFC were added to it. Similarly, results from other datasets are reported in table 2,3 and 4.

Table 1. Experimental results from iris dataset

Approach	**KFCG_L**	**KMCG_L**	**CNN_L**	**Model11 Model12**		**Model21 Model22**		**Model31 Model32**	
Measures				**CNN+ KFCG_L**	**CNN+ KMCG_L**	**CNN + LSTM_ KFCG_L**	**CNN+LSTM_ KMCG_L**	**CNN1 +LSTM2_ KFCG_L**	**CNN1 + LSTM2_ KMCG_L**
Accuracy	88.38	94.02	93.29	91.45	**97.03**	91.86	95.08	92.06	94.16
No.of layers	-	-	3CNN, 2FC	2CNN, 3FC	2CNN, 3FC	2CNN, LSTM	2CNN, LSTM	2CNN, LSTM	2CNN, LSTM
No.of trainable parameters	-	-	12,189,576	24,746,096	24,746,096	24,287,628	24,287,628	80,991,025	80,991,025
TimeTo train(min) / 80 epochs	-	-	115	70	70	83	83	95	95
Comparison	The maximum achieved accuracy is equivalent to the accuracy reported by Sowmya et. al. and Naseem which is 97% from iris datasets CASIA (94 subjects) and MBGC (150 subjects) respectively [31]-[35].								

Table 2. Experimental results from fingerprint dataset

Approach	**KFCG**	**KMCG**	**CNN**	**Model11 Model12**		**Model21 Model22**		**Model31 Model32**	
Measures				**CNN+ KFCG**	**CNN+ KMCG**	**CNN + LSTM_ KFCG**	**CNN+LSTM_ KMCG**	**CNN1+LSTM2_ KFCG**	**CNN1+ LSTM2_ KMCG**
Accuracy	91.50	96.1	91.86	93.98	95.99	93.22	95.35	93.18	94.0
No.of layers	-	-	3CNN, 2FC	2CNN, 3FC	2CNN, 3FC	2CNN, LSTM	2CNN, LSTM	2CNN, LSTM	2CNN, LSTM
No.of trainable parameters	-	-	579,576	1,170,101	1,170,101	1,667,213	1,667,213	579,576	579,576
TimeTo train(min)	-	-	64	49	49	57	57	61	61

/80 epochs									
Comparison	The maximum accuracy 95.99 is higher than the accuracy obtained by Aguilar [94.28%] and Gowthami Using Zone Based Linear Binary Patterns[95%] from FVC2002 and FVC 2004 datasets respectively [34-35] as well as from Jain et al.[74%], Medina-Perez et al using deformable minutie clustering[68.6%], Cao and Jain used an automated latent fingerprint recognition algorithm[78.3] which is reported by Danilo Valdes-Ramirez et al[36]-[39].								

Table 3: Experimental results from the palmprint dataset

Approach	**KFCG_L**	**KMCG_L**	**CNN_L**	**Model11 Model12**		**Model21 Model22**		**Model31 Model32**	
Measures				**CNN+ KFCG_L**	**CNN+ KMCG_L**	**CNN + LSTM_ KFCG_L**	**CNN +LSTM_ KMCG_L**	**CNN1 +LSTM2_ KFCG_L**	**CNN1 + LSTM2_ KMCG_L**
Accuracy	91.01	93.89	90.27	91.16	**96.29**	93.52	94.84	92.87	93.56
No.of layers	-	-	3CNN, 2FC	2CNN, 3FC	2CNN, 3FC	2CNN, LSTM	2CNN, LSTM	2CNN, LSTM	2CNN, LSTM
No.of trainable parameters	-	-	13,257,608	26,904,021	26,904,021	41,682,085	41,682,085	13,257,60 8	13,257,608
TimeTo train(min) /80 epochs	-	-	186	88	88	154	154	170	170
Comparison	The maximum accuracy of 96.29% is higher than the accuracy Double cohesion learning-based Multiview discriminant palmprint recognition method applied to the same IITD dataset [40][41].								

Table 4: Experimental results from the face dataset

Approach	**KFCG**	**KMCG**	**CNN**	**Model11**	**Model12**	**Model21**	**Model22**	**Model31**	**Model32**
Measures				**CNN+ KFCG**	**CNN+ KMCG**	**CNN + LSTM_ KFCG**	**CNN +LSTM_ KMCG**	**CNN1 +LSTM2_ KFCG_L**	**CNN1 + LSTM2_ KMCG_L**
Accuracy	91.8	94.02	91.99	95.23	**96.86**	91.86	93.62	93.12	95.03
No.of layers	-	-	3CNN, 2FC	2CNN, 3FC	2CNN, 3FC	2CNN, LSTM, FC	2CNN, LSTM, FC	2CNN, LSTM	2CNN, LSTM
No.of trainable parameters	-	-	6,788,941	13,893,125	13,893,125	10,783,846	10,783,846	61,924,694	61,924,694
TimeTo train(min) /80 epochs	-	-	537	465	465	490	490	503	503
Comparison	The accuracy is higher than the techniques Fisher Vector Faces [93.03], Simile classifiers [84.72], LBP PLDA [87.33], DFD[84.02], CMD+SLBP[92.58], LBP multishot[85.17] and FR+FCN[96.45] reported in [42]								

Direct concatenation i.e., $Model_{12}$ resulted in the highest 95.99% and the least 49 minutes of learning time for all the shown in Table 1 and Figure 5. Generating KFC is faster than generating KMC; however, KMC has proven to be more effective than KFC. $Model_{12}$ resulted in the highest 95.99% as shown in Figure 6.

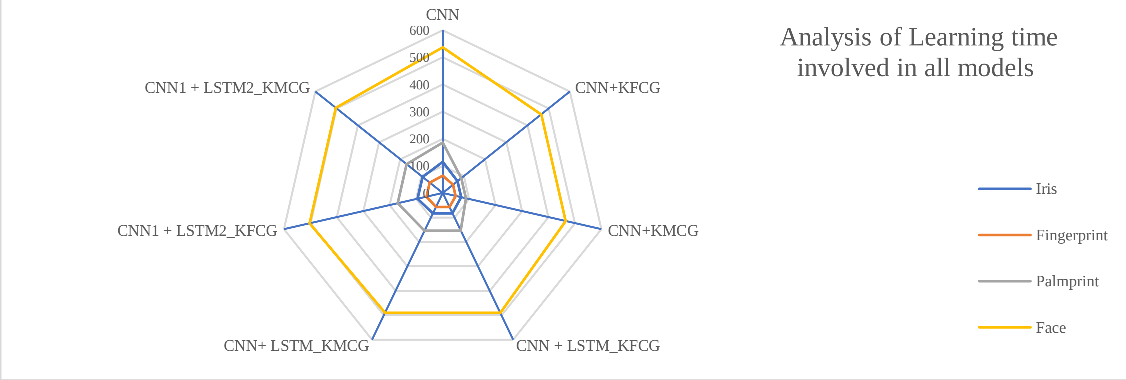


Fig 5. Learning time of all the proposed models

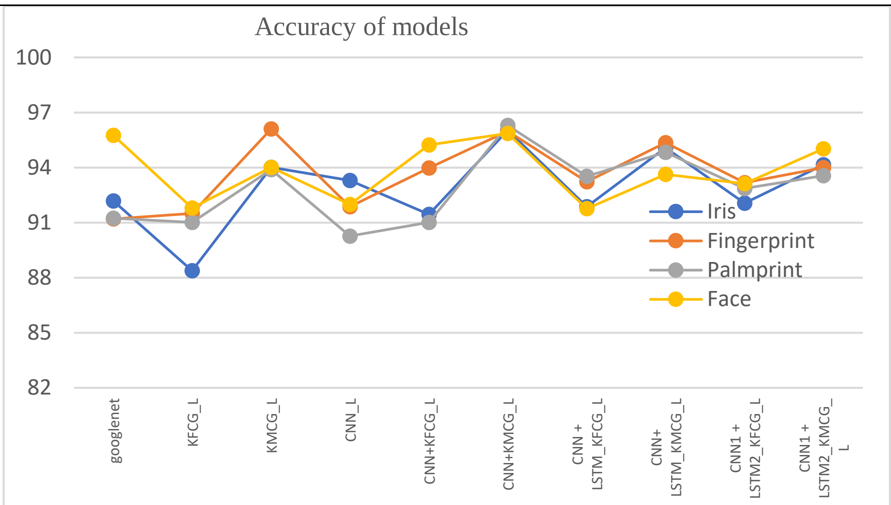


Fig 6. Learning time of all the proposed models

4. Conclusion

Prior to emergence of deep learning techniques', vast research has been carried out and many shallow classifiers have been developed to classify dataset. Shallow and deep classifiers, both have certain benefits, limitations and disadvantages associated with them. A hybrid approach may offer an optimal way when dataset is not fully satisfying requirements to apply any of shallow or deep classification as a standalone technique. If existing handcrafted features combined with CNN's discovered features, then the extensive use of CNN layers can be avoided. Hence, the chances of overfitting also get reduced. By keeping only minimal layers in our architecture, memory requirement and time to train on samples were reduced. This was confirmed by constructing unimodal and multimodal classifiers from biometric traits' datasets. The dataset with 2D samples and 1D KFC/KMC feature vectors encouraged us to use three approaches in order to perform a fusion of hand-crafted Kekre's Median (/fast) codebook feature vector with features identified from CNN layers. The direct concatenation of said features provided the maximum improvement compared to other approaches used for feature-fusion to build unimodal systems, however multimodal classifier has proven to be more robust than unimodal classifier when the system runs for a longer duration.

References

[1] Krishna Dharavath, F. A. Talukdar, R. H. Laskar, "Study on Biometric Authentication Systems, Challenges, and Future Trends: A Review" DOI: 10.1109/ICCIC.2013.6724278

[2] Anil K. Jain, Arun Ross, Salil Prabhakar, "An Introduction to Biometric Recognition", January 2004. DOI:10.1109/TCSVT.2003.818349

[3] Zhang Rui, Zheng Yan, "A Survey On Biometric Authentication: Towards Secure And Privacy-Preserving Identification" IEEE 2017 DOI: 10.1109/ACCESS.2018.2889996

[4] Hoo-Chang Shin, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning", 2016 IEEE.

https://doi.org/10.1109/TMI.2016.2528162

[5] Monica Bianchini, Franco Scarselli "On the complexity of shallow and deep neural network classifiers", ESANN 2014 proceedings. DOI: 10.1109/TNNLS.2013.2293637

[6] H. B. Kekre, Tanuja K. Sarode, Bhakti C. Raul, Jan. 2009. "Color Image Segmentation using Kekre's Fast Codebook Generation Algorithm Based on Energy Ordering Concept", ACM. https://doi.org/10.1145/1523103.1523175

[7] H B Kekre, T K Sarode, V A Bharadi, A A Agrawal, R J Arora, M C Nair, 26 February 2010. "Iris Recognition Using Discrete Cosine Transform and Vector Quantization". https://doi.org/10.1145/1741906.1741913

[8] Kitsuchart Pasupa, Wisuwat Sunhem, "A Comparison between Shallow and Deep Architecture Classifiers on Small Dataset", IEEE 2016. DOI: 10.1109/ICITEED.2016.7863293

[9] Ali Bou Nassif, Abdollah Masoud Darya, Ashraf Elnagar, "Empirical evaluation of shallow and deep learning classifiers for Arabic sentiment analysis", ACM transactions 2021. https://doi.org/10.1145/3466171

[10] Guk Bae Kim , Kyu-Hwan Jung , Yeha Lee , Hyun-Jun Kim , Namkug Kim , Sanghoon Jun , Joon Beom Seo , David A Lynch,"Comparison of Shallow and Deep Learning Methods on Classifying the Regional Pattern of Diffuse Lung Disease", J Digit Imaging 2018 DOI:10.1007/s10278-017-0028-9

[11] Xu-Cheng Yin, Chun Yang, Wei-Yi Pei, "Shallow Classification or Deep Learning: An Experimental Study", IEEE 2014. DOI: 10.1109/ICPR.2014.333

[12] Ali Beikmohammadi, Karim Faez, "Mixture of Deep-based Representation and Shallow Classifiers to Recognize Human Activities", IEEE 2019. DOI: 10.1109/ICSPIS48872.2019.9066014

[13] G. Willems, T. Tuytelaars, and L. VanGool, "An efficient dense and scale-invariant spatio-temporal interest point detector," 2008, pp. 650-663: Springer. https://doi.org/10.1007/978-3-540-88688-4_48

[14] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in BMVC 2008-19th British Machine Vision Conference, 2008 DOI:10.5244/C.22.99

[15] M. Jain, H. Jegou, and P. Bouthemy, "Better exploiting motion for better action recognition", IEEE 2013. DOI: 10.1109/CVPR.2013.330

[16] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," in European Conference, 2012, https://doi.org/10.1007/978-3-642-33718-5_23.

[17] H. Wang and C. Schmid, "Action recognition with improved trajectories," IEEE. DOI: 10.1109/ICCV.2013.441

[18] A. B. Sargano, P. Angelov, and Z. Habib, "Human action recognition from multiple views based on view-invariant feature descriptor using support vector machines," 2016. https://doi.org/10.3390/app6100309

[19] C. Yuan, X. Li, W. Hu, H. Ling, and S. J. Maybank, "Modeling geometric-temporal context with directional pyramid co-occurrence for action recognition," IEEE 2014.

[20] Le Yang, Dongmei Jiang and Hichem Sahli, "Integrating Deep and Shallow Models for Multi-Modal Depression Analysis — Hybrid Architectures", IEEE 2018. DOI: 10.1109/TAFFC.2018.2870398

[21] Jacek Haneczok, Jakub Piskorski, "Shallow and deep learning for event relatedness classification", Elsevier 2020. https://doi.org/10.1016/j.ipm.2020.102371

[22] Boonyawee Grodniyomchai, Khattiya Chalapat, Kulsawasd Jitkajornwanich, Saichon Jaiyen, "A Hybrid of Shallow and Deep Learning for Odor Classification Based on Adaptive Boosting", IEEE 2019. DOI: 10.1109/ICCSCE47578.2019.9068552

[23] Quang Tri Chiem; Margaret Lech; Richardt H. Wilkinson, "A Hybrid Two-Stage 3D Object Recognition from Orthogonal Projections", IEEE 2019. DOI: 10.1109/ICSPCS47537.2019.9008740

[24] Takayuki Hoshino, Suguru Kanoga, Masashi Tsuba ki, Atsushi Aoyama, "Comparing subject-to-subject transfer learning methods in surface electromyogram-based motion recognition with shallow and deep classifiers", Elsevier 2022. https://doi.org/10.1016/j.neucom.2021.12.081

[25] Ajay Shrestha, Ausif Mahmood, "Review of Deep Learning Algorithms and Architectures", IEEE April 2019. DOI: 10.1109/ACCESS.2019.2912200

[26] Fatin Sadiq Alkinani, Abdul Monem S. Rahma February 2019. "A Comparative Study of KMCG Segmentation Based on YCbCr, RGB, and HSV Color Spaces". DOI:10.29304/jqcm.2019.11.1.468

[27] Dr. Sudeep Thepade, Dimple Parekh, Jinali Shah, Bhumin Shah, Paras Vora, July 2012.

"Classification of Fingerprint using KMCG Algorithm", IJSTR. DOI:10.5120/ijca2018917285

[28] Dr. H B Kekre, T K Sarode, V A Bharadi, Tejas Bajaj, February 26–27, 2010. "A Comparative Study of DCT and Kekre's Median Code Book Generation Algorithm for Face Recognition", ACM. https://doi.org/10.1145/1741906.1741961

[29] Dr. H. B. Kekre. Tanuja K. Sarode. Sudeep D. Thepade. Pallavi N. Halarnkar, 2011. "Kekre's fast codebook generation in VQ with various color spaces for colorization of grayscale images", Springer India. https://link.springer.com/chapter/10.1007/978-81-8489-989-4_18

[30] Tanuja K. Sarode, Nabanita Mandal, Sept. 2013. "K-Means Codebook Optimization using KFCG Clustering Technique", International Journal of Computer Applications. DOI:10.5120/13496-1228

[31] Sowmya, et al., "Iris Recognition System for Biometric Identification", International Journal of Emerging Trends & Technology in Computer Science, (2013). DOI: 10.1109/ICCITECHN.2007.4579354

[32] Nassem, "Iris recognition using class-specific dictionaries", Computers and Electrical Engineering, (2016). https://doi.org/10.1016/j.compeleceng.2015.12.017

[33] Vahid Nazmdeh, Shaghayegh Mortazavi, Daniel Tajeddin et. el. "Iris Recognition; From Classic to Modern Approaches", IEEE 2019. DOI: 10.1109/CCWC.2019.8666516

[34] Aguilar, "Fingerprint Recognition", IEEE 2007. DOI: 10.1007/978-0-387-73003-5_47

[35] Gowthami, "Fingerprint Recognition Using Zone Based Linear Binary Patterns", IEEE 2015. https://doi.org/10.1016/j.procs.2015.08.072

[36] A. K. Jain and J. Feng, "Latent fingerprint matching", IEEE Jan. 2011. DOI:10.1109/TPAMI.2010.59

[37] M. A. Medina-Pérez et al., "Latent fingerprint identification using deformable minutiae clustering", Neurocomputing, vol. 175, pp. 851-865, Jan. 2016. https://doi.org/10.1016/j.neucom.2015.05.130

[38] K. Cao and A. K. Jain, "Automated latent fingerprint recognition", IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, pp. 788-800, Apr. 2019.

[39] Danilo Valdes-Ramirez et al., "A Review of Fingerprint Feature Representations and Their Applications for Latent Fingerprint Identification: Trends and Evaluation", IEEE 2019. DOI: 10.1109/ACCESS.2019.2909497

[40] S. Zhao and B. Zhang, ''Joint constrained least-square regression with deep convolutional feature for palmprint recognition,'' IEEE Jan. 2022. DOI: 10.1109/TSMC.2020.3003021

[41] Dele W. S. Alausa, Emmanuel Adetiba, "Contactless Palmprint Recognition System: A Survey", IEEE Dec. 2022. DOI: 10.1109/ACCESS.2022.3193382

[42] Lixiang Li1,2,3 Xiaohui Mu1,3 Siying Li 1,3 Haipeng Peng, "A Review of Face Recognition Technology", IEEE 2020. DOI: 10.1109/ACCESS.2020.301102

[43] Yenneti, L. L. ., Singam, A. ., & Gottapu, S. R. . (2023). Conflicting Parameter Pair Optimization for Linear Aperiodic Antenna Array using Chebyshev Taper based Genetic Algorithm. International Journal on Recent and Innovation Trends in Computing and Communication, 11(1), 161–166. https://doi.org/10.17762/ijritcc.v11i1.6086

[44] Dhabliya, D. (2021). Feature Selection Intrusion Detection System for The Attack Classification with Data Summarization. Machine Learning Applications in Engineering Education and Management, 1(1), 20–25. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/8