ConDense: Multiple Additional Dense Layers with Fine-Grained Fully-Connected Layer Optimisation for Fingerprint Recognition

Dane Lang and Dustin van der Haar(✉)

University of Johannesburg, Auckland Park, Gauteng, South Africa
201042463@student.uj.ac.za, dvanderhaar@uj.ac.za

Abstract. Fingerprint recognition is now a common, well known and generally accepted form of biometric authentication. The popularity of fingerprint recognition also makes it the focus of many studies which aim to constantly improve the technology in terms of factors such as accuracy and speed. This study sets out to create fingerprint recognition architectures which improve upon pre-trained architectures - named ConDense - that provide stronger if not comparable accuracy in comparison to related works on the authentication/identification task. Each of these ConDense architectures are tested against databases 1A, 2A, 3A provided by FVC 2006. The ConDense architectures presented in this study performed well across the varying image qualities in the given databases, with the lowest EERs achieved by this study's architectures being 1.385% (DB1A), 0.041% (DB2A) and 0.871% (DB3A). In comparison to related works, the architectures presented in this study performed the best in terms of EER against DB1A, and DB3A. The lowest EER for DB2A reported by a related work was 0.00%.

Keywords: Fingerprint recognition · Convolutional Neural Networks · Transfer learning

1 Problem Background

Fingerprint recognition is now a common, well known and generally accepted form of biometric authentication, an example of which can be seen in modern mobile devices which allow for the authentication of their users through scanning a fingerprint [11]. The popularity of fingerprint recognition also makes it the focus of many studies which aim to constantly improve the technology in terms of factors such as accuracy and speed. This study set out to accomplish two objectives. 1) Create a fingerprint authentication approach, which performs on a comparable if not higher level than related implementations when testing against the study's selected dataset, FVC 2006. 2) Perform a comparison of related fingerprint recognition approaches in order to find the most suitable candidate for use in a fingerprint recognition-based solution. This was enabled

© Springer Nature Switzerland AG 2022
M. El Yacoubi et al. (Eds.): ICPRAI 2022, LNCS 13364, pp. 15–27, 2022.
https://doi.org/10.1007/978-3-031-09282-4_2

by the use of the FVC 2006 [2] competition's fingerprint database which was used to provide a means of evaluating this study's approaches as well as serve to provide other solutions to compare this study's approaches against. Each of the architectures created in this study build additional dense layers on top of pre-trained architectures to improve performance over the base architectures. Few other works evaluate the effect of additional dense layers on CNNs, leading to this approach being named ConDense. The contributions of this study are:

– Results which show that the CNN architectures produced in this study, when tested against FVC 2006's databases DB1A and DB3A, score the lowest EERs when compared against a range of related works from FVC2006's commencement up to modern works.
– A study which shows that optimising the fully connected layers of a CNN can improve the performance of that CNN in terms of EER, ranging from the highest found increase of 60% to the highest reduction of 48%.
– An ablation study which shows that slower learning rates, higher dense layer sizes, and lower dropout percentages result in lower CNN EERs.

In Sect. 2 this paper will first provide an overview of related works. These range from modern implementations to those from the initial FVC 2006 competition. Section 3 provides a high level description of the approach that was used to set up the architectures used in this study. This is followed by the results in Sects. 4, 5 and 6 which detail the architectures created for this study as well as a comparison with the found related work. This paper closes with Sect. 7 which provides a summary of its results.

2 Related Work

A number or related studies on improving fingerprint matching have been performed that uses the FVC 2006's fingerprint databases. This section provides a discussion on these studies. Each which use subsets of the databases provided by FVC 2006 to test their architectures. The results posted by these studies will then be used to compare against the performance of the three architectures presented by this paper.

Priesnitz, Huesmann, Rathgeb, Buchmann and Busch [11] presented a COVID-19 inspired approach to touchless fingerprint recognition. Their approach relies on a mobile device to capture an image of a subject's fingers which run through a series of pre-processing steps which include Otsu's thresholding and a number of processes which isolate each finger's fingertip. One of the databases they tested their solution against was FVC 2006's DB2A where they scored an equal error rate (EER) of 0.15% [11].

He, Liu and Xiang [6] developed an architecture that consisted of two high-level stages: Image alignment through a spatial transformer and image matching via a deep residual network (ResNet). The spacial transformer the authors developed was named AlignNet and was used to allow for rotation invariance in their architecture. The images generated by AlignNet used their ResNet implementation which classified them as either genuine or impostor matches. The authors tested their database against DB1A of FVC 2006 and reported an EER of 3.587% [6].

Sanchez-Fernandez, Romero, Peralta, Medina-Perez, Saeys, Herrera and Tabik [13] created a latent fingerprint identification system - ALFI - which utilised parallelism to quickly process large fingerprint databases. Their study used minutiae-based feature extraction and matching approaches for fingerprint identification. The relevant FVC 2006 databases against which the authors tested their approach are DB2A and DB3A where they scored EERs of 0.48% and 3.70% respectively [13].

Kaggwa, Ngubiri and Tushabe [7] performed a study which compared the results of three separate approaches for feature extraction and matching:

- Minutiae based
- Gabor Filter based
- Combined Feature Level and Score Level Gabor Filter based

The authors' combined feature level and score level gabor filter based approach performed the best in their study, outperforming each of the other approaches in all experiments. The authors begin by using Gabor feature extraction to obtain a set of Gabor features which are then structured into column vectors. These column vectors are then used in "random feature level fusion" [7] before matching. These extracted features are then fed into the matching process, which consists of two steps. Their approach first determines the Euclidean distance between the extracted features to obtain matching scores, followed by score level fusion using the Max Rule [7]. The authors performed their experiments in six permutation sets against DB2A of FVC 2006. The best EER they obtained was in set two, which was 0.00%.

Wahby Shalaby and Omair Ahmad [15] proposed a scheme named "Multilevel Structural Technique for Fingerprint Recognition" (MSFR). In MSFR, fingerprint images are first broken down into a set smaller images that make up the original whole image. The features extracted from these smaller images and the features of the complete image are then used to build a template. The resulting template is then used in the scheme's matching process, which takes into account the features of both the small images and the complete image when matching against another template. The authors tested MSFR against DB1 of FVC 2006 and reported an EER of 5.21%.

Khazaei and Mohades [8] built an approach which makes use of Voronoi diagrams to perform fingerprint matching. Their approach involves first pre-processing images by applying a Gabor filter and skeletonisation to input fingerprint images. Feature extraction is performed on the pre-processed images and Voronoi diagrams are built from the extracted features. The Voronoi diagrams are used to determine central cells which are then provided to a matching algorithm for comparison. The authors tested their approach against databases 1A, 2A and 3A of FVC 2006 and obtained EERs of 2.8% (DB1A), 3.65% (DB2A) and 1.15% (DB3A).

Lastly, the organisers of FVC 2006 posted the results of their competition which included the top three algorithms by EER in both their open and light categories. In the open category, the best performing algorithms for each database scored 5.564% (DB1), 0.021% (DB2) and 1.534% (DB3). The best performing algorithms for the light category scored 5.356% (DB1), 0.148% (DB2) and 1.634% (DB3).

3 Methods

This study aims to use deep learning to provide an improved approach to fingerprint classification in terms of accuracy. Instead of attempting to build a new architecture from the ground up, a solution was created which makes use of the advances made by already existing CNN architectures which provide solutions to similar problems. The three architectures that were selected are Xception [3], Inception V3 [14] and ResNet 152 V2 [5] for their small memory footprint and high base accuracy. Each of these three architectures will form the base of our CNN architectures, after which additional fully connected layers are added to explore their effect on improving the accuracy of the pre-trained architectures. The name selected for the approach of introducing the higher complexity fully connected layers is ConDense, which will be provided as a suffix for each pre-trained architecture's name wherever these modified architectures are used. Sections 4, 5 and 6 provide an overview of our experiments and our results.

4 Experiment Data

Biometric authentication is an ever-evolving field where over time the need for improvements in areas such as accuracy, efficiency and cost effectiveness are a driving force behind advances. The drive for such improvements lead to the creation of datasets and competitions that aid in developing and testing potential advances in biometric authentication. In the case of fingerprint biometrics, an example of these is the Fingerprint Verification Competition whose first edition was run in the year 2000 [9]. This section provides an overview of FVC 2006 the latest iteration outside of FVC-onGoing, whose fingerprint database are used in training and testing the Convolutional Neural Network architectures presented in this paper. FVC 2006 was selected due to its good variety of fingerprint images captured for databases one through three and its continued relevance in evaluating the performance of modern fingerprint recognition solutions [1,6,12,13].

Table 1. The differences between fingerprint capturing/generation technologies and image dimensions in the databases provided by FVC 2006 [2]

Database	Technology	Image dimensions (pixels)
DB1	Electric Field Sensor	96×96
DB2	Optical Sensor	400×560
DB3	Thermal Sweeping Sensor	400×500
DB4	Synthetic Fingerprint Generator	288×384

The most recent FVC outside of the current ongoing online initiation occurred in 2006 and made available four fingerprint databases. The fingerprints for these databases were each captured using different technologies which in turn produced images whose resolution differed per database [2,4]. Databases one to four are suffixed with the letters A and B e.g. "DB1A" and "DB1B". The databases suffixed with A contained 12 samples from each of 120 subjects and those suffixed with B contained 12 samples from each of 10 subjects. A summary of the differences of each database is given in Table 1 and examples of the samples captured for each database can be seen in Fig. 1.

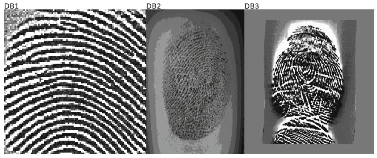

Fig. 1. The differences between fingerprint images in the databases provided by FVC 2006 [2]

To aid in evaluating the performance of the Xception-ConDense, Inception-ConDense and ResNet-ConDense architectures that were produced for this study, the three non-synthetic fingerprint databases were selected for use in training and evaluation. The use of the FVC 2006 databases allows for the derivation of performance metrics and their subsequent comparison against related studies which have also utilised the same databases. In addition this also allows for a comparison of performance for each of the architectures presented in this paper when running against different databases to show how the different image capturing methods and dimensions potentially affect performance. Lastly, the varying sensors and fidelities allow for a good evaluation of fingerprint recognition algorithms.

Using FVC 2006 gives access to the four fingerprint databases which were provided to the competition's contestants. For the purposes of the greater study under which this one falls - which would only make use of non-synthetic fingerprints - only databases DB1, DB2 and DB3 are used. Each of the fingerprint images provided in the databases is in greyscale.

DB1 provides the smallest of the fingerprint images provided for FVC 2006. The images were captured using a AuthenTec Electric Field Sensors and are 96×96 pixels in size [2]. DB1A provides fingerprints for 140 subjects with 12 samples per subject. DB1B provides 12 samples for 10 subjects. The size difference between images provided in FVC 2006 can be seen in Fig. 1. Architectures presented in FVC 2006 tended to perform the poorest in terms of EER when tested against this database, potentially due to the relatively small size and low detail of the images. The lowest EER on DB1 was reported as 5.564% (open category).

Fingerprint images from DB2 are the largest in terms of image size of those provided by the competition. The images from DB2 are 400×560 pixels and were captured by a BiometriKa Optical Sensor [2]. As with DB1, DB2A provides images for 140 subjects with 12 samples each and DB2B provides 10 subjects with 12 samples per subject. FVC 2006 architectures performed the best when using DB2 with the lowest EER reported as 0.021% (open category).

Lastly, in terms of size the images from DB3 fit between those of DB1 and DB2. Images from DB3 are 400×500 pixels and were captured by a Atmel Thermal Sweeping Sensor [2]. Once again, DB3A provided 12 samples for each subject, and DB3B provided 12 samples for 10 subjects. Performance of architectures against DB3 was significantly better compared to those running against DB1, with the best of these (open category) reporting an EER of 1.534%.

5 Experiment Setup

When training the architectures which use Xception-ConDense, Inception-ConDense and ResNet-ConDense, there are two stages: The first is pre-processing where fingerprint images are modified in accordance with requirements from each pre-trained architecture. The second is where the pre-processed images are fed into each of the architectures for training.

Pre-processing, as mentioned above, is the first step in the two-part process. Here a number of modifications to the FVC 2006 fingerprint images are required before they can be passed to each CNN architecture for training/testing. When training, the first 10 of the 12 samples for each subject in any given database are used, giving a 83%:17% split. The remaining two for each subject are then used for testing.

By their nature, the greyscale images provided by the FVC 2006 databases are single-channel images. The pre-trained architectures that are used in this study were trained using ImageNet weights that require three-channel images. To compensate for this, the grey channel of each FVC 2006 image is duplicated twice to create a three-channel image.

The new three-channel image is then fed into a architecture-specific pre-processing function, where Xception-ConDense, Inception-ConDense and ResNet-ConDense each provide their own means of pre-processing images. These pre-processed images are then resized to the dimensions of the images that were used to originally train the architectures. The sizes used are 299×299 (Xception-ConDense), 299×299 (Inception-ConDense) and 224×224 (ResNet-ConDense).

After pre-processing is complete, these pre-processed images are provided to each of the architectures in turn for use in training. First, the ground truth labels for each image is established for use in the training process. These are then used to guide each architecture in terms of making correct classifications.

The structure of each of this study's architecture, can be seen in Fig. 2. The initial layer is the input layer, whose shape is set to the dimensions of the images based on each pre-trained architecture's specific sizes. These input layers then lead into their respective pre-trained architectures. After the pre-trained

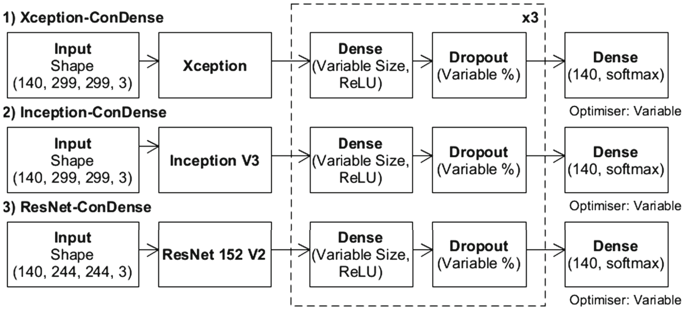

Fig. 2. Overview of classifier architectures for Xception-ConDense (1), Inception-ConDense (2) and ResNet-ConDense (3) architectures.

architecture layers, the structure for each architecture is largely the same. Three sets of a combination of a fully connected (Dense) layer and a dropout layer are then added. The dense layers have an output size which varies by the database to which it is applied with Rectified Linear Unit (ReLU) being the activation function in each case. The dropout layers are each set to a percentage which varies by the database to which it is applied. The final layer in each of the architectures is a dense layer set to the number of subjects that are used to train the model, which is 140. The activation function for the final layer is softmax.

6 Experiment Results

6.1 Ablation Study

The base architectures are initialised using a number of hyperparameters that govern training and certain aspects of our architecture layers. The hyperparameters that were focused on in this study were learning rate, optimiser, dense layer size and dropout percentage used in the architecture with the options for each shown below:

- Learning Rate - 0.01, 0.001 and 0.0001.
- Optimiser - Adam and SGD.
- Dense layer size - 256, 512, 1024 and 2048.
- Dropout - 0.1, 0.2, 0.3, 0.4 and 0.5.

Keras Tuner [10] was used to find the optimal hyperparameter per architecture/database combination. The hyperparameters that were identified for each combination are shown in Table 2. Table 2 shows that in most cases, lower learning rates, higher denser layer sizes, and lower dropout are needed to achieve lower EERs.

Table 2. Hyperparameters used for each combination of ConDense architecture and FVC 2006 database.

ConDense architecture	Inception			Exception			ResNet		
FVC 2006 Database	1A	2A	3A	1A	2A	3A	1A	2A	3A
Learning Rate	0.0001	0.0001	0.0001	0.0001	0.0001	0.01	0.0001	0.0001	0.01
Optimiser	Adam	Adam	Adam	Adam	Adam	SGD	Adam	Adam	SGD
Dense Layer Size	2048	2048	2048	1024	2048	1024	2048	1024	2048
Dropout	0.1	0.1	0.5	0.2	0.1	0.1	0.1	0.2	0.2

6.2 Architecture Performance

The first set of results were obtained by testing the architectures against DB1A of FVC 2006. Of the three architectures, the one that resulted in the lowest EER was the Inception-ConDense architecture with an EER of 1.522%. This was followed by ResNet-ConDense and Xception-ConDense with EERs of 1.778% and 1.829% respectively.

The Inception-ConDense architecture's higher EER is evident from the Receiver Operating Characteristic (ROC) curve generated for the three architectures, shown in Fig. 3, where the Inception-ConDense architecture's true positive rate (TPR) reaches the highest peak at 95% before seeing a large increase in false positive rate (FPR) starting at 2.5%. The Precision-Recall curve given in Fig. 4 shows that the precision of each of the architectures noticeably begin to decrease as recall reaches 0.4. The Xception-ConDense architecture has the highest area under the curve (AUC) with 0.824.

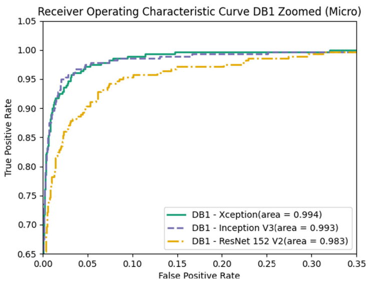

Fig. 3. Micro-average ROC curves for all classifiers (DB1.)

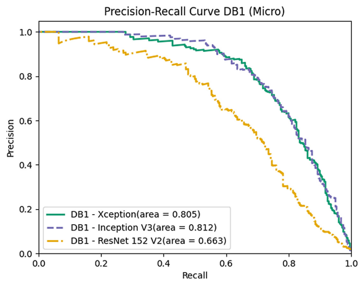

Fig. 4. Micro-average precision-recall curves for all classifiers (DB1.)

For the second set of results, each of the architectures were tested against DB2A of FVC 2006. In this category each of the architectures performed considerably better than in their test against DB1A. The top architecture was Xception-ConDense which scored an EER of 0.041%. The Inception-ConDense and ResNet-ConDense architectures scored 0.103% and 0.144% respectively.

In this category's ROC curve shown in Fig. 5, the Xception-ConDense architecture's FPR begins noticeably increasing at a TPR of 97.5%. The large change in FPR begins rising from 0.5%. The Precision-Recall curve given in Fig. 6, shows that each of the architectures begin losing precision at approximately 75% recall, but the Xception-ConDense architecture maintains the highest precision before a significant drop until a recall of 80% is achieved. At this point the precision is still above 95% precision.

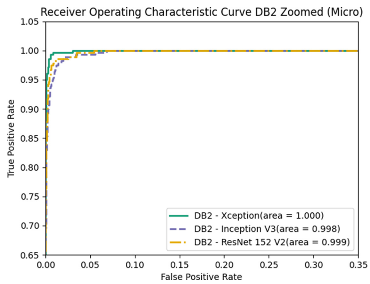

Fig. 5. Micro-average ROC curves for all classifiers (DB2.)

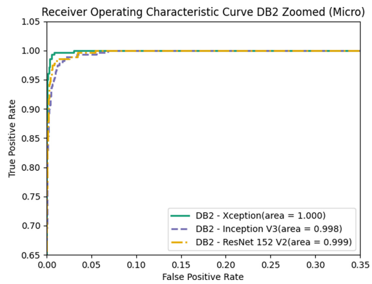

Fig. 6. Micro-average precision-recall curves for all classifiers (DB2.)

The last set of results were obtained were from testing the architectures against DB3A of FVC 2006. The top performing architecture for this set of tests was once again Xception-ConDense. Here the architecture achieved an EER of 0.953%. Inception-ConDense and ResNet-ConDense performed considerably worse with EERs of 1.711% and 2.592% respectively. In broader terms all the architectures performed worse in the tests against DB3 compared to those of DB2.

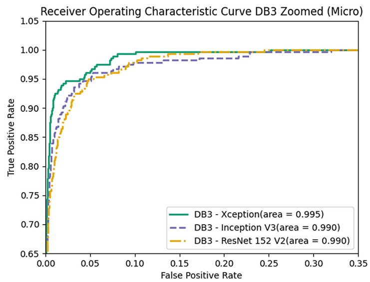

Fig. 7. Micro-average ROC curves for all classifiers (DB3.)

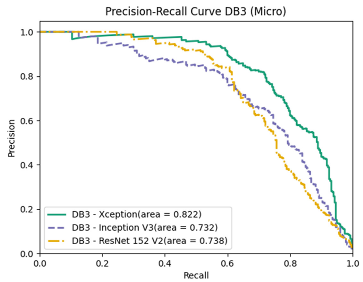

Fig. 8. Micro-average precision-recall curves for all classifiers (DB3.)

Considering the ROC curve shown in Fig. 7, all of the architectures begin showing a large increase in FPR at just below 95% TPR. The best performing of these, the Xception-ConDense architecture, reaches approximately a 95%

TPR with a FPR of 4%. The Precision-Recall curve shown in Fig. 8 shows that the Inception-ConDense and ResNet-ConDense architectures start seeing a decline at approximately 27% recall where the same drop is seen for Xception-ConDense at 44% recall. The Xception-ConDense architecture reaches 60% Recall at approximately 95% precision before seeing a steep decline in recall.

6.3 Comparison to Related Work

This section provides a comparison of the performance of the architectures against the performance reported by the authors of the studies under Sect. 2. The differences in performance will be discussed with reference to the EERs obtained by each of the different architectures. Each of the studies listed under Sect. 2 conducted their tests against subsets of the databases provided by FVC 2006. As a result only a small number of them have used all the databases that were selected from FVC 2006 for use in this study. In addition, where the EERs for ConDense architectures are provided the EERs for their base architectures are also provided along with the percentage change in EER that resulted from the addition of multiple dense layers.

Table 3. Comparison of the performance of related work and this paper's ConDense architectures when trained and tested against DB1.

Method (DB1)	EER %	Non-ConDense EER %	Change %
Xception-ConDense	**1.385**	2.079	34
Inception-ConDense	1.390	1.4825	6
ResNet-ConDense	2.697	1.8191	−48
Khazaei and Mohades [8]	2.8		
He, Liu and Xiang [6]	3.587		
Wahby Shalaby and Omair Ahmad [15]	5.21		
FVC 2006 Light [2]	5.356		
FVC 2006 Open [2]	5.564		

When testing against DB1, the architectures scored the lowest three EERs when compared against the related work. The lowest of the three was Xception-ConDense with 1.385%, with the highest being ResNet-ConDense with 2.697%. The next highest was the work of Khazaei and Mohades [8] with 2.8%. The results show a significant improvement in EER compared to the best original FVC 2006 open and light category results which were both above 5% [2]. The comparison of the DB1 results to all relevant related works can be seeing in Table 3.

The next set of results are those for DB2. Here the work of Kaggwa, Ngubiri and Tushabe scored the lowest EER of 0.00% [7]. This was followed by FVC 2006's best open category result which was 0.021% [2]. These are directly followed by Xception-ConDense with an EER of 0.041%. The full comparison of results for DB2 can be seen in Table 4.

Table 4. Comparison of the performance of related work and this paper's ConDense architectures when trained and tested against DB2.

Method (DB2)	EER %	Non-ConDense EER %	Change %
Kaggwa, Ngubiri and Tushabe [7]	**0.00**		
FVC 2006 Open [2]	0.021		
Xception-ConDense	0.041	0.092	56
FVC 2006 Light [2]	0.148		
Priesnitz et al. [11]	0.15		
Inception-ConDense	0.167	0.121	−38
Sanchez-Fernandez et al. [13]	0.48		
ResNet-ConDense	0.193	0.493	61
Khazaei and Mohades [8]	3.65		

The final set of results are those for DB3. Here the Xception-ConDense architecture achieved the lowest EER of 0.871%. The second and third lowest EERs were reported by Khazaei and Mohades and ResNet-ConDense with EERs of 1.15% [8] and 1.449% respectively. The full comparison of these results can be seen in Table 5.

The results of the study show that across all three databases, the introduction of additional dense layers improved the performance of their base architectures in 2/3 of the cases. The most significant improvement is seen for ResNet-ConDense in Table 4 with an improvement of 61% with the highest reduction in Table 3 for ResNet-ConDense with −48%.

The authors of this study endeavoured to find any relevant studies for performance comparisons. The results of this study are reported to the best of the authors' knowledge. When looking at past and present results for DB1 and DB3, it can be seen that these categories are the most challenging of the three databases that FVC 2006 provided. The architectures attained the lowest EERs in these categories, showing significant improvements when compared against related work. The Xception-ConDense architecture performed consistently well, where it placed in the top three architectures in all of the tests against the FVC 2006 databases. This study set out to build CNN architectures which provide improved accuracy when compared to existing approaches. The performance of the architectures' created in this study when testing against DB1 and DB3 shows that study was successful in achieving that goal.

Table 5. Comparison of the performance of related work and this paper's ConDense architectures when trained and tested against DB3.

Method (DB3)	EER %	Non-ConDense EER %	Change %
Xception-ConDense	**0.871**	1.202	28
Khazaei and Mohades [8]	1.15		
ResNet-ConDense	1.449	2.980	51
FVC 2006 Open [2]	1.534		
FVC 2006 Light [2]	1.634		
Inception-ConDense	1.745	1.647	−6
Sanchez-Fernandez et al. [13]	3.70		

7 Conclusion

This study set out to accomplish two objectives. 1) Create a fingerprint authentication approach which performs on a comparable if not higher level than related implementations when testing against the study's selected dataset, FVC 2006. 2) Perform a comparison of related fingerprint recognition approaches in order to find the most suitable candidate for use in a fingerprint recognition-based solution. The results when testing against the FVC 2006 database in Sect. 6 show that for DB1A, the Xception-ConDense architecture had the lowest EER of 1.385%. For DB2A, the solution from Kaggwa, Ngubiri and Tushabe [7] performed the best with an EER of 0.00%. Lastly, for DB3A the Xception-ConDense pipeline scored the lowest EER of 0.871%. Our implementations for DB1A and DB3A satisfy our first objective where, to the best of our knowledge, we could not find approaches that score lower EERs against those databases. We satisfy our second objective with our comparison for DB2A, where we identified Kaggwa, Ngubiri and Tushabe's approach as the lowest scoring approach [7]. Lastly, this study has shown that - while further study is required to determine the cause - the addition of more complex fully-connected layers to the base pre-trained architectures resulted in solutions which produce the lowest EERs in two of the three categories against which they were evaluated.

References

1. Agarwal, D., Garima, Bansal, A.: A utility of ridge contour points in minutiae-based fingerprint matching. In: Chaki, N., Pejas, J., Devarakonda, N., Rao Kovvur, R.M. (eds.) ICCIDE 2020, pp. 273–286. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-8767-2_24
2. Cappelli, R., Ferrara, M., Franco, A., Maltoni, D.: Fingerprint verification competition 2006. Biometric Technol. Today **15**(7), 7–9 (2007)
3. Chollet, F.: Xception: deep learning with depthwise separable convolutions. CoRR abs/1610.02357 (2016). http://arxiv.org/abs/1610.02357

4. Fierrez, J., Ortega-Garcia, J., Torre-Toledano, D., Gonzalez-Rodriguez, J.: BioSec baseline corpus: a multimodal biometric database. Pattern Recogn. **40**(4), 1389–1392 (2007)
5. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. CoRR abs/1603.05027 (2016). http://arxiv.org/abs/1603.05027
6. He, Z., Liu, E., Xiang, Z.: Partial fingerprint verification via spatial transformer networks. In: 2020 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–10 (2020)
7. Kaggwa, F., Ngubiri, J., Tushabe, F.: Combined feature level and score level fusion Gabor filter-based multiple enrollment fingerprint recognition. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 159–165 (2016)
8. Khazaei, H., Mohades, A.: Fingerprint matching algorithm based on Voronoi diagram. In: 2008 International Conference on Computational Sciences and Its Applications, pp. 433–440 (2008)
9. Maio, D., Maltoni, D., Cappelli, R., Wayman, J., Jain, A.: FVC 2000: fingerprint verification competition. IEEE Trans. Pattern Anal. Mach. Intell. **24**(3), 402–412 (2002)
10. O'Malley, T., et al.: Kerastuner (2019). https://github.com/keras-team/keras-tuner
11. Priesnitz, J., Huesmann, R., Rathgeb, C., Buchmann, N., Busch, C.: Mobile touchless fingerprint recognition: implementation, performance and usability aspects. Sensors **22**, 792 (2021)
12. Sabir, M., Khan, T., Arshad, M., Munawar, S.: Reducing computational complexity in fingerprint matching. Turk. J. Electr. Eng. Comput. Sci. **28**, 2538–2551 (2020)
13. Sanchez-Fernandez, A.J., et al.: Asynchronous processing for latent fingerprint identification on heterogeneous CPU-GPU systems. IEEE Access **8**, 124236–124253 (2020)
14. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. CoRR abs/1512.00567 (2015). http://arxiv.org/abs/1512.00567
15. Wahby Shalaby, M., Omair Ahmad, M.: A multilevel structural technique for fingerprint representation and matching. Sig. Process. **93**(1), 56–69 (2013)