JOIV : Int. J. Inform. Visualization, 7(1) - March 2023 178-185

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

End-To-End Evaluation of Deep Learning Architectures for Offline Handwriting Writer Identification: A Comparative Study

Wirmanto Suteddy [a,*], Devi Aprianti Rimadhani Agustini [a], Anugrah Adiwilaga [a], Dastin Aryo Atmanto [a]

[a] *Department of Computer Engineering, Universitas Pendidikan Indonesia, Bandung, 40625, Indonesia*

Corresponding author: [*]wirmanto.suteddy@upi.edu

***Abstract*— Identifying writers using their handwriting is particularly challenging for a machine, given that a person's writing can serve as their distinguishing characteristic. The process of identification using handcrafted features has shown promising results, but the intra-class variability between authors still needs further development. Almost all computer vision-related tasks use Deep learning (DL) nowadays, and as a result, researchers are developing many DL architectures with their respective methods. In addition, feature extraction, usually accomplished using handcrafted algorithms, can now be automatically conducted using convolutional neural networks. With the various developments of the DL method, it is necessary to evaluate the suitable DL for the problem we are aiming at, namely the classification of writer identification. This comparative study evaluated several DL architectures such as VGG16, ResNet50, MobileNet, Xception, and EfficientNet end-to-end to examine their advantages to offline handwriting for writer identification problems with IAM and CVL databases. Each architecture compared its respective process to the training and validation metrics accuracy, demonstrating that ResNet50 DL had the highest train accuracy of 98.86%. However, Xception DL performed slightly better due to the convergence gap for validation accuracy compared to all the other architectures, which were 21.79% and 15.12% for IAM and CVL. Also, the smallest gap of convergence between training and validation accuracy for the IAM and CVL datasets were 19.13% and 16.49%, respectively. The results of these findings serve as the basis for DL architecture selection and open up overfitting problems for future work.**

***Keywords*— Offline writer identification; deep learning; convolutional neural network; comparative study; end-to-end.**

Manuscript received 11 Jan. 2022; revised 20 Apr. 2022; accepted 8 Jun. 2022. Date of publication 31 Mar. 2023.

I. INTRODUCTION

Identification of writers using their handwritten is still challenging for a machine because of the unique individual hand movement. This uniqueness has also become part of human biometrics as one person, and another have different styles [1]. The process of identifying handwriting is categorized into two: the text-based approach to online and offline handwritten documents [2]. The difference in patterns generated from different authors' writings can be in the form of the width and thickness of the ink imprint, angle, slope, height, the direction of movement, and legibility of the writing, thereby making the difference between one person writing to another. With the rapid advancement of technology, especially in computer vision, machines can identify handwritten digits and characters, but identifying the labeled writer from the handwritten text is still a topic of research discussion. Various research has been conducted in this field for more than two decades. The subject matter has received attention because it can be used to solve applied problems in various fields of study. Some of these subjects and fields, e.g., validating the authenticity of a person's handwriting in legal court cases, analyzing a handwritten threat letter by a criminal in forensics, or the authenticity of wills in law. In paleography, for analysis of historical handwriting, in the banking sector for verification of signatures and handwriting, and in other fields that require scientific proof [3]. A person's writing pattern can vary even on the same word, produced by intra-class variances that affect the handwritten, e.g., writing speed, time, and mood, among others [4]. In general, three stages are involved in this identification after acquiring the data, i.e., pre-processing, feature extraction, and classification. Unlike identification which helps to identify a writer, author retrieval focuses on finding similar handwriting from documents.

In the early stages, after the image is cleaned by pre-processing, a few steps need to be conducted for public datasets for better performance before they can be used, e.g., removing special characters and ensuring good segmentation results. In our experiments, the presence of special characters,

such as periods, commas, quotes, colons, and others, produces high similarity between writers, which reduces performance accuracy slightly. For segmentation, if the data is not segmented correctly, it will greatly affect the performance of the built system. On the contrary, Kumar and Sharma [5] tried using opposite approaches, such as the segmentation-free methods, where the identification of the authors was conducted using the region probability map technique and voting mechanism to identify the writer.

Conventional handcrafted machine learning algorithms for handwriting identification, such as texture-based methods [6], [7] and specific descriptors [8], show promising results. Still, with the emergence of Deep Learning, researchers are trying another road with its advantages and convenience. However, more in-depth neural networks are more challenging to train. Consequently, many researchers present their respective methods for a specific problem, e.g., residual learning ResNet for image recognition, new scaling method EfficientNet for CNN itself, depthwise separable convolution MobileNet for embedded and mobile applications, and many others. Moreover, it raises the question of how to obtain the best DL architectures for writer identification problems. As a result, various sophisticated methods are used by researchers based on DL architectures in writer identification problems for different reasons. However, to our knowledge, only a little explain the statistical results of DL architectures which are the basis for selecting DL architectures for writer identification problems.

As far as we comprehend, Fiel and Sablatnig [9] use the Convolutional Neural Network approach (CaffeNet) on writer identification and retrieval problems for the first time. The model contains five layers and three FC. Furthermore, [10] compared handcrafted features with convenience for author identification. The results indicated that their CNN architecture, which had three convolutions and FC layers, was not as good as the handcrafted method. Ni et al. [11], in the experiment regarding noise reduction in identifying authors using DL features, also stated that the results generated did not produce a good performance. On the contrary, where classification learning generally uses supervised learning, Christlein et al. [12] used ResNet DL architecture with unsupervised learning and SIFT keypoint location of databases ICDAR17 on historical documents and ICFHR16 on historical Latin script documents demonstrating promising results. This research is similar to Bria et al. [13], who also experimented using different DL architectures, namely VGG19, ResNet50, InceptionV3, InceptionResNetV2, and NASNetLarge. However, instead of end-to-end, they are used as transfer learning on medieval books for paleograph writer identification, and InceptionResNetV2 demonstrates better performance among others. Following this, Hosoe et al. [14] used the LeNet5 architecture to extract features of personal writing styles using the ETL-1-character and the NIST special databases. Keglevic et al. [15] employed DenseNet CNN Architecture, which had a dense block with five layers, and triplet architecture having multiple CNN branches, to extract information from the ICDAR 2013 dataset. [16] utilized CNN with four convolutional layers for text-independent author identification on JEITA-HP offline handwritten Japanese characters, firemaker dataset, and IAM database as an end-to-end network. Durou et al. [17] conducted a comparison study for handwriter identification using modified AlexNet architecture with SVM and KNN classification on IAM and ICFHR 2012 Arabic handwriting dataset, which resulted in the DL method showing better results than the conventional method. Similarly, Helal et al. [18] experimented CNN DL with an addition dissimilarity approach with an SVM classifier using CVL and BFL databases.

Rehman et al. [19] used transfer learning with AlexNet pre-trained model architecture in their approach. They examined different layers in the model to determine which of the layers influencing the rate of author identification is best for feature extraction and then fed the result to the Support Vector Machine (SVM) classifier on Arabic QUWI dataset. Semma et al. [20] exploit key point detectors such as Harris Corner and FAST to extract the handwritten text region points and feed them to the ResNet CNN model on writer identification. Meanwhile, information context from handwritten usually has its length or language. Sulaiman et al. [21] proposed an opposite approach as a general solver, independent of any samples from a handwritten document using modified AlexNet with a hybrid method of Deep Learning and handcrafted. Furthermore, although CNN has potential development, it has a drawback on word images, according to He and Schomaker [22], and they proposed FragNet as a solution based on a text block or word images.

A summary of different DLs used in writer identification, as described above, showed in Table I.

TABLE I
SUMMARY OF DIFFERENT DL USED IN WRITER IDENTIFICATION

Author	Dataset	DL	Accuracy(%)
Fiel and Sablatnig (2015)	ICDAR 2011, ICDAR 2013, CVL	CaffeNet	99.5 98.6 98.3
Helal et al. (2017)	CVL	ConvNets	Inferior performance
Christlein et al. (2017)	ICDAR 2017	ResNet	88.9
Bria et al. (2018)	Medieval books	VGG19 ResNet50 InceptionV3 InceptionResNetV2 NASNetLarge	88.25 88.13 94 94.25 93.27
Hosoe et al. (2018)	ETL-1, NIST-19 2nd	LeNet-5	97 88
Keglevic et al. (2018)	ICDAR 2013	DenseNet	98.9
Nguyen et al. (2019)	JEITA-HP, Firemaker+ IAM	CNN	99.97 91.81
Durou et al. (2019)	IAM, ICFHR 2012	AlexNet	93 99.5
Rehman et al. (2019)	QUWI arabic+en	AlexNet	92.2 92.78
He and Schomaker (2020)	IAM, CVL, Firemaker, CERUG-EN	FragNet	85.1 90.2 69 77.5
Semma et al. (2021)	IAM, QUWI, IFN/ENIT	ResNet	99.5 99.8 99.8

With many researchers using certain DLs and showing different results on different datasets, it needs a scientific explanation of its DLs to understand their advantages and disadvantages to this particular problem. Therefore, our proposed method in our hypotheses is a quantitative comparative study comparing several DL architectures end-to-end regarding their training and validation accuracy to find which DL architecture is suitable for handwriting writer identification problems for better chances of being developed further for use in real-life scenarios.

II. Materials and Method

For materials and methods consisting of IAM and CVL datasets also for the selected DL architectures, the DLs namely VGG16, ResNet50, MobileNet, Xception, and EfficientNetB0. All datasets will be pre-processed, fed end-to-end to architecture models, and evaluated. Since this is comparative research, the DL architectures' layers were not modified for equality reasons. However, each database has its number of classes, so the last layer was removed, described as follows.

A. IAM Dataset

The IAM dataset [23] contains offline handwriting of 1539 visual image data and 657 class of English-speaking writers. The data was segmented into sentences, lines, and word images. Following this, word images were used as the experiment data, some of which are shown in Fig. 1. (a)-(c). The dataset was cleaned, and all the unidentified texts or incorrectly written words were removed. Also, special characters such as periods, commas, colons, and others were removed because they can affect the feature vector generated by the system. In addition, it was found that not all authors had the same average word data, but 657 authors were selected as the research's sample data.

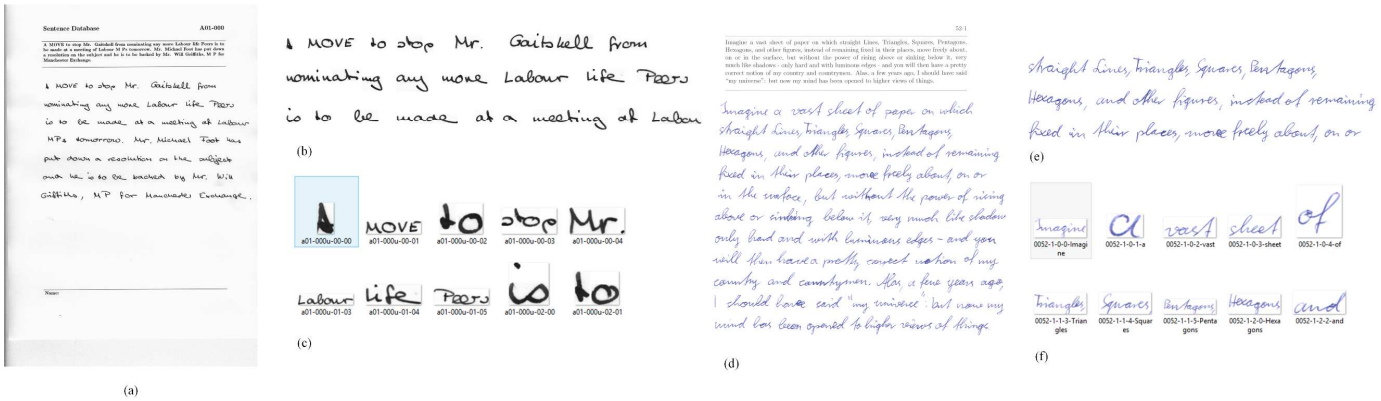


Fig. 1 IAM dataset (a) pages, (b) sentences, (c) words; CVL dataset (d) pages, (e) sentences, (f) words

B. CVL Dataset

The offline handwritten CVL database [24] has 1,600 pages and 310 authors, consisting of 283 and 27 authors who wrote only 5 and 7 texts, respectively. It also consisted of 101,069 words written in German and English, some of which are shown in Fig. 1. (d) – (f). The literature reference mentioned that the total data words generated were only 99.904 because several authors omitted some words.

C. Pre-processing

The IAM dataset and CVL image data were pre-processed before being fed into the CNN for feature extraction. The first step of the pre-processing was carried out by converting all the images to grayscale instead of color for computational convenience, algorithm simplification, and threshold requirements. The next step involves removing noise from the dataset using dilation and erosion operations, after which a gaussian filter was applied to remove gaussian noise, followed by Otsu's threshold, as shown in Fig. 2.

For the IAM dataset, as explained earlier, the data has special characters and unidentified text. The total number of special characters found is 14,963, including periods, commas, question marks, and unidentified text. All the special characters were removed for readable text features, and padding was added for proper feature extraction. After that, the images were resized to 224x224 pixels, and the data was ready for extraction.

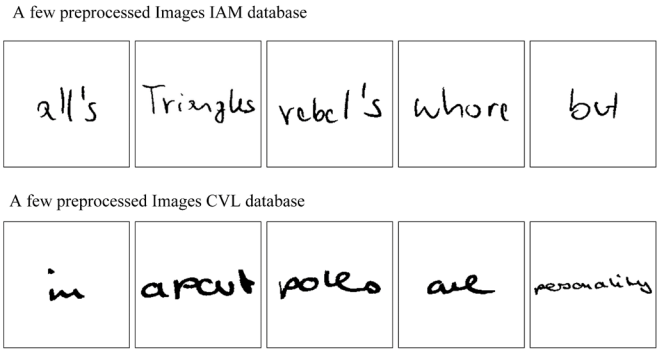


Fig. 2 A few IAM and CVL text images after pre-processed

D. Deep-Learning

In selecting the DL that we will evaluate, for the representativeness of each existing architecture, we choose one of each architecture at random that is not its variance. As aforementioned, the last layer was excluded for five DL architectures, significantly reducing the number of parameters in each DL architecture. Due to space limitations in the layout of the figure locations in this paper, we combine two proposed architectures into one figure for simplification, where we examine the architectures one by one, not with two arrows, as depicted in Fig. 3, or Fig. 4. The following is brief description of the five proposed architectures.

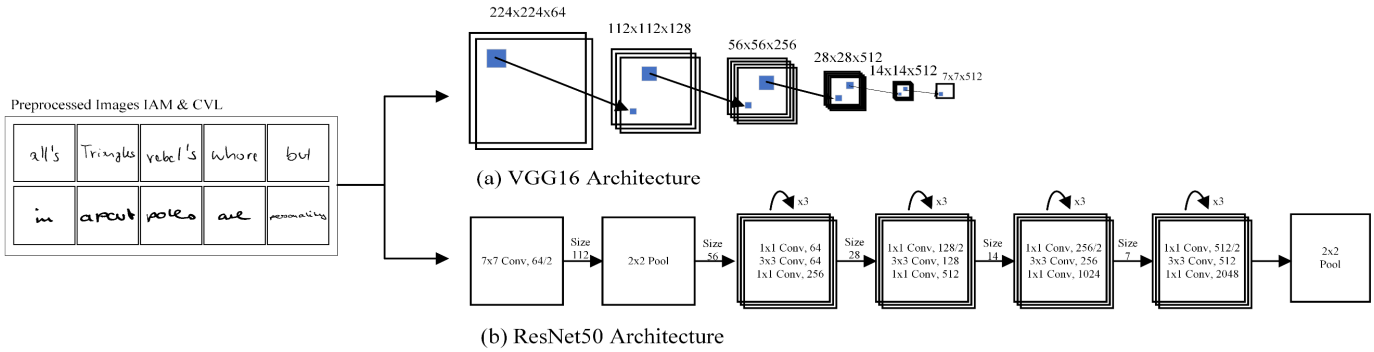


Fig. 3 (a) VGG16 and (b) ResNet50 Deep Learning Architecture

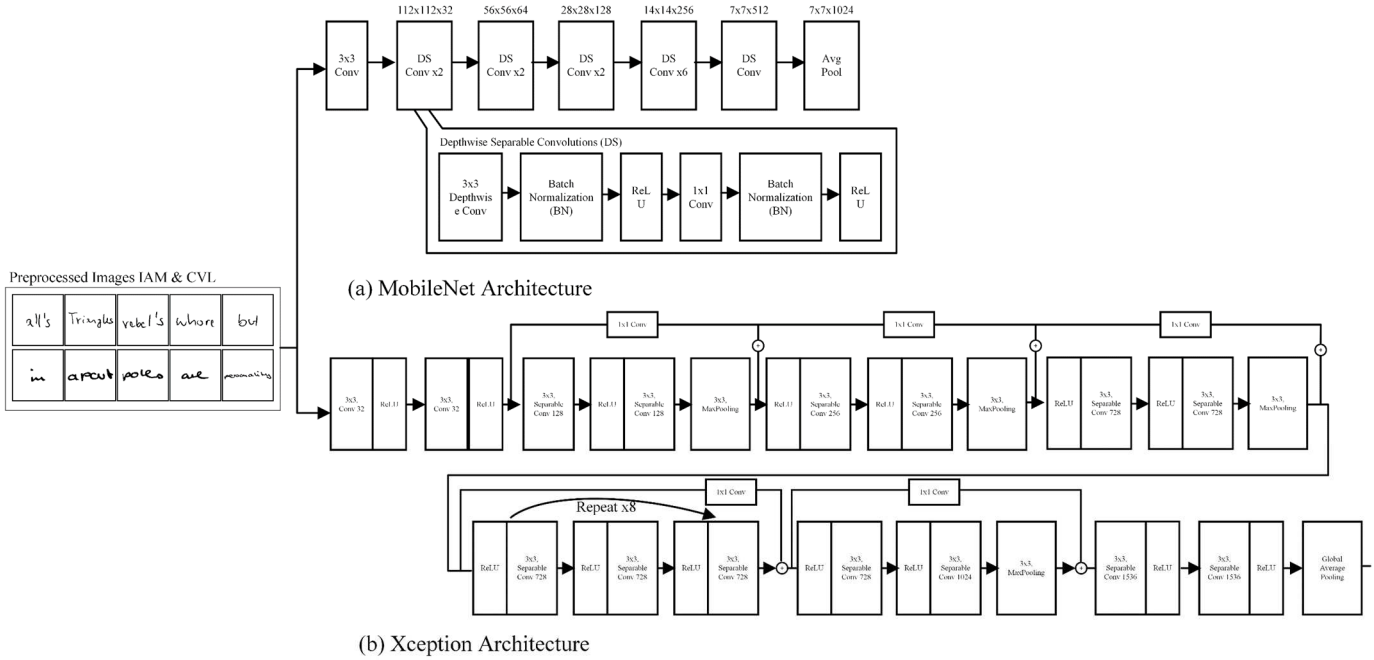


Fig. 4 (a) MobileNet and (b) Xception Deep Learning Architecture

1) VGG16 Architecture: VGG16 [25] is made up of stacks of Convolutional Neural Networks (CNN) having different depths, architecture, and three Fully-Connected (FC) layers. The default VGG16 architecture is 16 layers deep, as seen in Fig. 3. Also, this architecture has more than 14 million parameters.

2) ResNet50 Architecture: Residual Network (ResNet) [26] is a Convolutional Neural Network with skip connections as its residual blocks, making it possible to train very deep neural networks. This architecture won the ILSVRC 2015 classification task and has more than 23 million trainable parameters, as seen in Fig. 3.

3) MobileNets Architecture: MobileNet [27] is a deep CNN aimed at embedded mobile applications. This model has convolutions that can be separated into filtering and combining each in depth to create a lightweight DL, with the tradeoff of lowering accuracy to reduce size and latency. The architecture has more than three million parameters, as shown in Fig. 4.

4) Xception Architecture: Xception [28] architecture is a CNN based entirely on depth-wise separable convolution layers and has 36 layers deep. This model is adopted from the inception model architecture with convolutions that can be separated. Furthermore, this architecture configuration allows for a deep neural network because eight repetitions are in the middle before the final exit flow. This architecture has more than 20 million parameters, as shown in Fig. 4.

5) EfficientNetB0 Architecture: EfficientNet [29] is a CNN that uses the compound scaling coefficient technique to balance important dimensions of network parameters in CNN. The graphic display of this architecture is exceptionally long due to its scaling method in the IAM and CVL databases, and it has more than four million parameters. The simplified version of the architecture is depicted in Fig. 5.

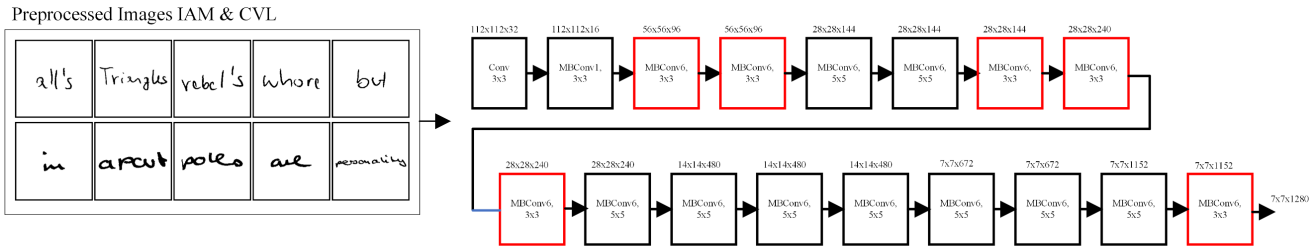


Fig. 5 EfficientNetB0 Deep Learning Architecture (simplified)

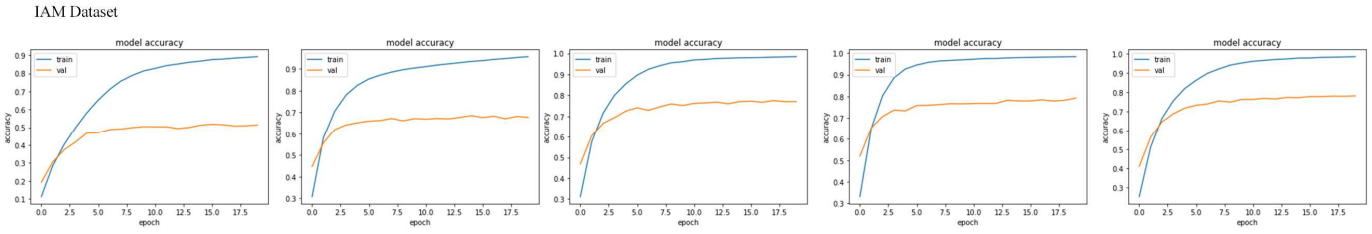
E. Experiments

For comparison, all the DL architectures were trained using the same configuration with a personal computer having the following specifications CPU i7-10700F @ 2.90GHz, 16GB of RAM, and NVIDIA GeForce GTX 1660 6GB GPU. After going through the pre-processing stage and calling the architecture model, global average pooling was added and flattened. The total number of classes was replaced with the number of authors for each database, including 657 authors obtained from IAM and 310 from CVL datasets. In addition, the softmax activation function was used, epochs were set to 20, categorical cross entropy parameter as a function of loss, and we set *include top* parameter to false as we do not want the last layer for every architecture. Furthermore, for both IAM and CVL datasets, first, we remove five images from each author for test data. Next, we divide them into 80% train data and the rest as validation. Furthermore, the classification process was carried out using the Keras library [30] written in python. Feature extraction was conducted automatically by each DL model in order to compare their quality efficiently.

For performance evaluation, only training and validation metrics were used to test the accuracy of the different architectures. These metrics were used because the focus was not on the writers' retrieval but only on identification; thus, test data was not used. Furthermore, this study's primary priority is to find each DL's merits. The obtained results will form the basis for selecting an appropriate DL architecture for research on future projects like writer retrieval with more metric evaluation, e.g., Soft top N, Hard Top N, precision at N, average precision, confusion matrix, or the plot of training and validation loss.

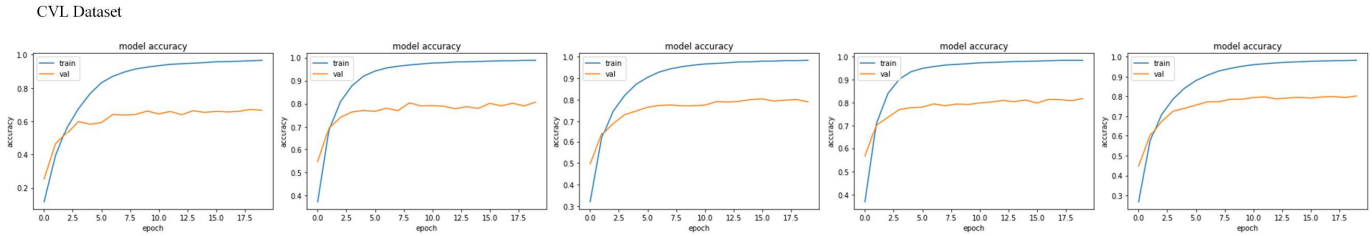
III. Results and Discussion

Feature extraction plays a significant step in the learning process as each DLs extract local features to make feature maps for both image datasets. After each DLs architectures process, the result is depicted as train and validation plot accuracy in Fig. 6 and Fig. 7. The results show convergence gap indicates overfitting. Overfitting and underfitting are not topics discussed in this paper, so we exclude them for future reference as we are also aware that the augmentation technique, dropout method, and fine-tuning layers could overcome those problems in future works.



(a) VGG16 (b) ResNet50 (c) MobileNet (d) Xception (e) EfficientNetB0

Fig. 6 Convergence gap of training and validation plot accuracy for IAM dataset



(a) VGG16 (b) ResNet50 (c) MobileNet (d) Xception (e) EfficientNetB0

Fig. 7 Convergence gap of training and validation plot accuracy for CVL dataset

A. Comparison Result

From Fig. 6, it can be seen that all the observed DL architectures achieved almost the same training and validation accuracy result for IAM datasets, with each architecture having indications of overfitting. The training and validation results obtained from evaluating the architectures quite differ, and the difference between their gaps indicated it. There was a 19.45% increase from the lowest VGG16 to the highest Xception, demonstrating that the Xception method with depthwise separable convolution can produce a slightly better gap with little data than the other DL architectures.

Similarly, the training and validation results on the CVL dataset, as depicted in Fig. 7., show that the difference between the respective gaps of the DL mentioned above increased by 13.3% from the lowest VGG16 to the highest Xception, while the resulting difference between accuracy and validation accuracy is 16.49% which shows that the Xception result has the slightest difference between the deviations. The results in Table II show all performance outcomes. The comparison of the observed DL architectures demonstrates an increase in validation accuracy, shown in Fig. 8 for IAM and Fig. 9 for the CVL dataset.

TABLE II
COMPARISON RESULT PERFORMANCE

Deep Learning Architectures	Database							
	IAM				CVL			
	Training Acc (%)	Validation Acc (%)	TVAG (%)[a]	TVAG Rank	Training Acc (%)	Validation Acc (%)	TVAG (%)[a]	TVAG Rank
VGG16	96,03	57,45	38,58	5	96,48	66,69	29,79	5
ResNet50	**98,86**	75,33	23,53	4	**98,86**	80,71	18,15	3
MobileNet	98,32	76,47	21,85	3	98,27	78,96	19,31	4
Xception	98,37	**79,24**	**19,13**	1	98,30	**81,81**	**16,49**	1
EfficientNetB0	98,32	78,05	20,27	2	98,07	80,23	17,84	2

[a]Train and Validation Accuracy Gap

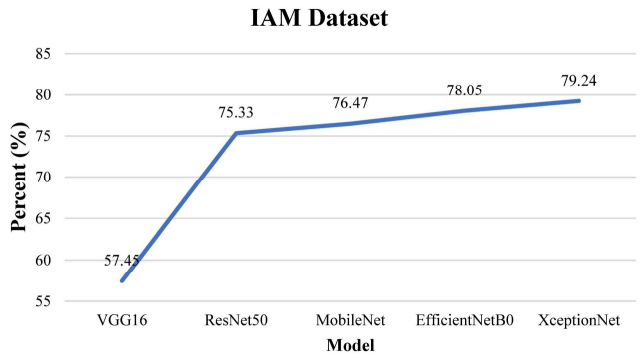


Fig. 8 Statistical chart of validation accuracy on IAM dataset

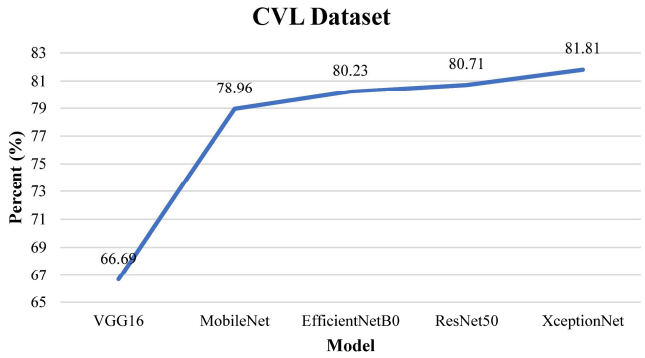


Fig. 9 Statistical chart of validation accuracy on CVL dataset

The increase in validation accuracy indicates that the convergence gap between train and validation accuracy slightly improves from the respective DL. The difference in validation accuracy between the highest and lowest DL architectures is 21.79% and 15.12% for IAM and CVL, respectively. This demonstrates that the validation accuracy in Xception architecture increased in a better approach, and variation occurs only in each dataset's second, third, and fourth positions. The following comparison of training and validation accuracy is shown in Fig. 10. And Fig. 11. for the IAM data set and CVL dataset, respectively.

Fig. 10 and Fig. 11 also demonstrate that Xception has more advantages because its difference between train and validation accuracies is smaller than other DL. This indicates that the DL architecture outperformed others in terms of convergence gap accuracy, giving the model a better chance of being developed further for use in real-life scenarios.

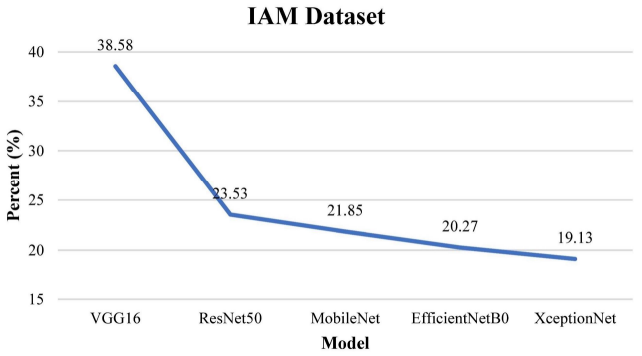


Fig. 10 Statistical chart of convergence gap between train and validation accuracy for IAM dataset

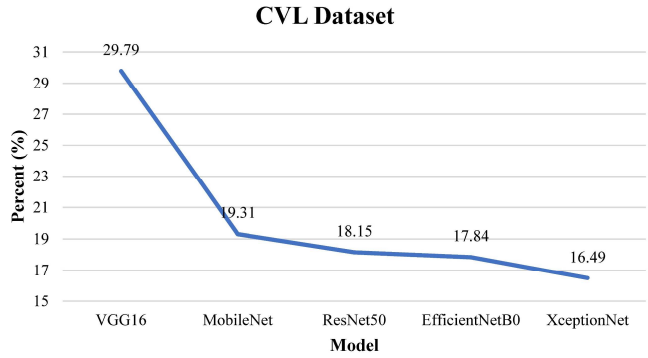


Fig. 11 Statistical chart of convergence gap between train and validation accuracy for CVL dataset

B. Performance comparison with previous research

Most researchers from previous studies mainly focusing developing additional methods or specific approaches, e.g. [7-19] with arbitrary DL selected to get the best accuracy in writer identification with TopN, SoftN, or HardN evaluation metrics performance. On the contrary, our work mainly focuses on getting a scientific explanation by comparative study without any additional method or specific approach to the handwriting writer identification problem. We prove that the best suitable DL in the convergence gap area between training and validation accuracy, which is ExceptionNet

without the additional methods mentioned above, has almost reached state-of-the-art accuracy produced by previous studies in IAM dataset [16], [17], [20], [22] with the highest accuracy 99.5% using ResNet with a difference of 1.2% with our accuracy of 98.3% and CVL dataset [9], [10], [22] which is identical 98.3%. This accuracy gap of 1.2% in the IAM dataset is because we have yet to work out of scope focus on this paper, the overfitting problems. Nevertheless, for comparison regarding methods and objects, we briefly summarize our result as a preliminary comparative study from Table II in Table III for IAM and Table IV for CVL, respectively.

TABLE III
BRIEF METHODS FOR IAM DATASETS IN WRITER IDENTIFICATION

Authors	Methods	Datasets	Accuracy(%)
[16]	CNN	IAM	91.81
[17]	AlexNet	IAM	93
[22]	FragNet	IAM	85.1
[20]	ResNet	IAM	99.5
Proposed DL*	**Xception**	**IAM**	**98.37**

*Preliminary Comparative Study

TABLE IV
BRIEF METHODS FOR CVL DATASETS IN WRITER IDENTIFICATION

Authors	Methods	Datasets	Accuracy(%)
[9]	CaffeNet	CVL	98.3
[10]	ConvNets	CVL	Inferior
[22]	FragNet	CVL	90.2
Proposed DL*	**Xception**	**CVL**	**98.3**

*Preliminary Comparative Study

IV. CONCLUSION

This quantitative comparative study was conducted using five DL architectures end-to-end as a comparison to aid the classification problem of the offline handwriting writer's identification. Each architecture performance was evaluated and then compared to others. The results demonstrate that ResNet50 architecture had the highest train accuracy of 98.86%. However, Xception DL has performed a more suitable convergence gap validation accuracy of 21.79% and 15.12% for IAM and CVL, followed by the convergence between training and validation accuracy of 19.13% and 16.49% for both datasets. It was discovered that this comparison study's results open room for more research on the overfitting problem. Our study provides ground truth of suitable DL selection of offline handwriting writer's identification problems. Furthermore, only a few previous comparative research observed the difference between training and validating loss functions as an indicator for selecting a suitable model for real scenarios. Analyzing the overfitting problems encountered in this research using augmentation technique, dropout, and fine-tuning layers with suitable DL should be considered in future works.

ACKNOWLEDGMENT

This work is supported by the Annual Work Plan and Budget Assignment Fund for Research and Community Service Institute, Universitas Pendidikan Indonesia, Fiscal Year 2022, with the Rector's decree number: 0965/UN40/PT.01.02/2022.

REFERENCES

[1] A. L. Hagström, R. Stanikzai, J. Bigun, and F. Alonso-Fernandez, "Writer Recognition Using Offline Handwritten Single Block Characters." arXiv, Mar. 07, 2022. Accessed: Sep. 02, 2022.

[2] M. Sonkusare and N. Sahu, "A Survey on Handwritten Character Recognition (HCR) Techniques for English Alphabets," *AVC*, vol. 3, no. 1, pp. 1–12, Mar. 2016.

[3] C. Halder, Sk. Md. Obaidullah, and K. Roy, "Offline Writer Identification and Verification—A State-of-the-Art," in *Information Systems Design and Intelligent Applications*, vol. 435, S. C. Satapathy, J. K. Mandal, S. K. Udgata, and V. Bhateja, Eds. New Delhi: Springer India, 2016, pp. 153–163.

[4] C. Adak, B. B. Chaudhuri, and M. Blumenstein, "An Empirical Study on Writer Identification and Verification From Intra-Variable Individual Handwriting," *IEEE Access*, vol. 7, pp. 24738–24758, 2019.

[5] P. Kumar and A. Sharma, "Segmentation-free writer identification based on convolutional neural network," *Computers & Electrical Engineering*, vol. 85, p. 106707, Jul. 2020.

[6] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Texture-based descriptors for writer identification and verification," Expert Syst. Appl., vol. 40, no. 6, pp. 2069–2080, May 2013.

[7] P. Singh, P. P. Roy, and B. Raman, "Writer identification using texture features: A comparative study," Comput. Electr. Eng., vol. 71, pp. 1–12, Oct. 2018.

[8] F. A. Khan, M. A. Tahir, F. Khelifi, A. Bouridane, and R. Almotaeryi, "Robust offline text independent writer identification using bagged discrete cosine transform features," Expert Syst. Appl., vol. 71, pp. 404–415, Apr. 2017.

[9] S. Fiel and R. Sablatnig, "Writer Identification and Retrieval Using a Convolutional Neural Network," in *Computer Analysis of Images and Patterns*, vol. 9257, G. Azzopardi and N. Petkov, Eds. Cham: Springer International Publishing, 2015, pp. 26–37.

[10] L. G. Helal, Y. Maldonado, G. da Costa, D. B. Goncalves, and G. Z. Felipe, "Offline writer identification using handcrafted features versus ConvNets," in *2017 36th International Conference of the Chilean Computer Science Society (SCCC)*, Arica, Oct. 2017, pp. 1–8.

[11] K. Ni, P. Callier, B. Hatch, J. Mastarone, and J. Cline, "On noise reduction for handwritten writer identification," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2017, pp. 1984–1988.

[12] V. Christlein, M. Gropp, S. Fiel, and A. Maier, "Unsupervised Feature Learning for Writer Identification and Writer Retrieval," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Nov. 2017, pp. 991–997.

[13] A. Bria *et al.*, "Deep Transfer Learning for writer identification in medieval books," in *2018 Metrology for Archaeology and Cultural Heritage (MetroArchaeo)*, Cassino FR, Italy, Oct. 2018, pp. 455–460.

[14] M. Hosoe, T. Yamada, K. Kato, and K. Yamamoto, "Offline Text-Independent Writer Identification Based on Writer-Independent Model using Conditional AutoEncoder," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, Aug. 2018, pp. 441–446.

[15] M. Keglevic, S. Fiel, and R. Sablatnig, "Learning Features for Writer Retrieval and Identification using Triplet CNNs," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, Aug. 2018, pp. 211–216.

[16] H. T. Nguyen, C. T. Nguyen, T. Ino, B. Indurkhya, and M. Nakagawa, "Text-independent writer identification using convolutional neural network," *Pattern Recognition Letters*, vol. 121, pp. 104–112, Apr. 2019.

[17] A. Durou, S. Al-Maadeed, I. Aref, A. Bouridane, and M. Elbendak, "A Comparative Study of Machine Learning Approaches for Handwriter Identification," in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, London, United Kingdom, Jan. 2019, pp. 206–212.

[18] L. G. Helal, D. Bertolini, Y. M. G. Costa, G. D. C. Cavalcanti, A. S. Britto, and L. E. S. Oliveira, "Representation Learning and Dissimilarity for Writer Identification," in *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Osijek, Croatia, Jun. 2019, pp. 63–68.

[19] A. Rehman, S. Naz, M. I. Razzak, and I. A. Hameed, "Automatic Visual Features for Writer Identification: A Deep Learning Approach," *IEEE Access*, vol. 7, pp. 17149–17157, 2019.

[20] A. Semma, Y. Hannad, I. Siddiqi, C. Djeddi, and M. El Youssfi El Kettani, "Writer Identification using Deep Learning with FAST

Keypoints and Harris corner detector," *Expert Systems with Applications*, vol. 184, p. 115473, Dec. 2021.
[21] A. Sulaiman, K. Omar, M. F. Nasrudin, and A. Arram, "Length Independent Writer Identification Based on the Fusion of Deep and Handcrafted Descriptors," *IEEE Access*, vol. 7, pp. 91772–91784, 2019.
[22] S. He and L. Schomaker, "FragNet: Writer Identification Using Deep Fragment Networks," *IEEE Trans.Inform.Forensic Secur.*, vol. 15, pp. 3013–3022, 2020.
[23] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, Nov. 2002.
[24] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "CVL-DataBase: An Offline Database for Writer Retrieval, Writer Identification and Word Spotting," in *2013 12th International Conference on Document Analysis and Recognition*, Washington, DC, USA, Aug. 2013, pp. 560–564.
[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," p. 14, 2015.
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[27] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, 2017.
[28] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, Jul. 2017, pp. 1800–1807.
[29] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019.
[30] F. Chollet and others, "Keras." GitHub, 2015. [Online]. Available: https://github.com/fchollet/keras