World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:18, No:5, 2024

An Architecture Based on Capsule Networks for the Identification of Handwritten Signature Forgery

Luisa Mesquita Oliveira Ribeiro, Alexei Manso Correa Machado

Open Science Index, Computer and Information Engineering Vol:18, No:5, 2024 publications.waset.org/10013659.pdf

Abstract—Handwritten signature is a unique form for recognizing an individual, used to discern documents, carry out investigations in the criminal, legal, banking areas and other applications. Signature verification is based on large amounts of biometric data, as they are simple and easy to acquire, among other characteristics. Given this scenario, signature forgery is a worldwide recurring problem and fast and precise techniques are needed to prevent crimes of this nature from occurring. This article carried out a study on the efficiency of the Capsule Network in analyzing and recognizing signatures. The chosen architecture achieved an accuracy of 98.11% and 80.15% for the CEDAR and GPDS databases, respectively.

Keywords—Biometrics, deep learning, handwriting, signature forgery.

I. Introduction

A Way of unique identification of different individuals around the world is by their handwritten signature. This becomes a sign of the person and expresses different characteristics about the subscriber. Different types of signatures currently exist, such as handwritten, electronic, digital and scanned. Among those mentioned, the safest signatures are digital and electronic, since the integrity of the data is guaranteed, because they are legally valid and authenticity verification is carried out by various means, therefore, reducing the likelihood of forgery crime. However, handwritten signatures remain the most common way of document authentication.

The handwritten signature has a high possibility of forgery. If a graphotechnical examination takes place, different characteristics and factors will be evaluated, such as writing speed, pressure placed at the time of signing, among others. However, the expertise is carried out by a specialist and has the potential to become a lengthy and subjective process. This process does not have a standard that dictates which basic steps must be followed, that is, each graphotechnical expert follows what he/she considers to be correct.

The process of acquiring and identifying signatures can be grouped into 2 methods [1]. The first is *off-line*, where the signature is on a document that needs to be scanned by a device. Although easy, the dynamic characteristics of signatures, such as the pressure applied to the pen when writing, the color of the pen used, and speed are lost. In the *on-line* method, the signature is obtained using special electronic devices, such as tablets and cell phones. In this type of signature verification, fraud is more difficult to occur and dynamic properties are preserved. The characteristics achieved in the off-line method are not as accurate as those obtained in on-line, and the off-line images may contain noise, making the process more complex.

The main purpose of signature verification is to differentiate authentic signatures from tampered ones. There are 2 approaches for checking signatures off-line: *writer-dependent* (WD) and *writer-independent* (WI). In the writer-dependent, a model is trained for each person. If a new signature is added for an individual, the model needs to be retrained for that person. In the writer-independent approach, a model is trained for all individuals. That is, if a new signature is added, it is not necessary to train a new model.

Deep learning models such as the Convolution Networks (CNNs) are actively used in the area of image recognition and, consequently, in the area of signature recognition. The main objective of this article is to evaluate the effectiveness of the *Capsule Network* (CapsNet) [2] on detecting signature forgery so that this task may be automated, avoiding slow, subjective and imprecise analysis. The experiments were applied to the CEDAR [3] and the GPDS Synthetic On-line & Off-line Signature databases [4].

This article is organized as follows: Section II presents the works related to the theme of signature analysis; Section III explains the main concepts and the architecture of Capsule Network; Section IV describes the databases used, pre-processing steps, and the evaluation metrics, as well as the methodology; Section V presents and discusses the obtained results and, finally, Section VI presents the concluding remarks.

II. Related Work

Considering the context of Convolutional Neural Networks, [5] evaluated whether the use of CNN resources provides good results for verifying independent-writer signatures. A Support Vector Machine (SVM) was used as a classifier and the Dichotomy Transformation (DT) as the transformer of a multiclass problem into binary. The GPDS-960 databases (union of the GPDS-160 and GPDS-300 databases) [4] and a Brazilian database (PUC-PR) [1] were used. Statistical functions such as the maximum, minimum, median and mean were used, aiming to determine satisfactory results for the partial decision fusion rule. The article's proposal achieved better values with the maximum function and the PUC-PR dataset when compared with other works.

CapsNet was used by [6] and its performance compared to other CNN models in the task of signature analysis. Only the reduced CEDAR database was used in the experiments

L. Ribeiro and A. Machado are with the Department of Computer Science, Pontifical Catholic University of Minas Gerais, Belo Horizonte, MG, 30350 Brazil (e-mail: alexeimcmachado@gmail.com).

and the images were reduced to resolutions of 64x64 and 32x32. For identification, only genuine signatures were used in the 2 resolutions mentioned. For verification, original and falsified images in 64x64 resolution were used. The results obtained in the verification achieved an average accuracy of 90%. The authors concluded that CapsNet had a potential to the verification and identification tasks, indicating that it was reliable for classifying signatures and could deal satisfactorily with the division of genuine and forged signatures.

From another perspective, [7] proposed the *Local Difference Feature* (LDF), a new texture feature with the aim of verifying handwritten signatures off-line, where the difference is calculated between the central pixel and 8 neighbors taken within a specific radius. SVM is used in the verification task where only genuine signatures are used for trained, while testing is done on genuine and forged ones. The new feature was compared with other common classification features, such as Gradient-Oriented Histogram and Local Binary Patterns (LBP) with improvements in accuracy. Following the same idea, [8] suggests multiscale LDF to improve the verification of handwritten signatures. The proposed descriptor is similar to LBP in that it highlights the difference between the pixel and its neighbors. SVM was used and verification performed using a writer-dependent approach. The authors highlighted that the values achieved for the CEDAR and GPDS-300 databases were satisfactory and when compared with other articles, high accuracy could be obtained using a reduced database.

Using histograms, [9] suggests a feature-based multiscale fusion to perform signature description. This combines texture and shape information, seeking to improve the characterization of handwritten signatures. LDF is introduced as a new descriptor combined with *Histogram of Templates*. To verify signatures, the writer-dependent approach was used with the SVM classifier. The results obtained were significant for the GPDS-300 and MYCT-75 bases. Therefore, the multiscale calculation allowed an improvement in accuracy compared to other available methods.

Considering the writer-dependent and writer-independent approaches, [10] used CNN to train the WI approach, where the network was used to extract the characteristics of the off-line signatures in the WD approach. SVM was employed to classify genuine and forged signatures. The tests were carried out on 3 databases, where the GPDS-40000 achieved 92.03% of accuracy. The results obtained using different databases proved the efficiency of the suggested method for verifying off-line signatures. [11] proposed a new method that uses *Deep Neural Network* (DNN) with pairwise loss for off-line verification of WI signatures. The CEDAR database was used and 92.76% of accuracy was achieved, surpassing other available techniques.

A literature review on signature recognition using Machine Learning was carried out in [1], where the main objective was to determine the best algorithm to perform signature recognition based on their type. The baseline model used for off-line identification was the CNN. For on-line signature recognition, Recurrent Neural Networks (RNN) was used most of the time, in conjunction with other architectures.

On the other hand, [12] proposed a modification to the Siamese Network. The first suggestion would be to use a 2-phase CNN in order to simultaneously verify images of handwritten signatures. The second would be to apply the Focal Loss function to deal with the imbalance between genuine and counterfeit samples. The proposed method achieved better accuracy for the chosen databases and the Focal Loss function effectively solved the imbalance of genuine and falsified data.

In view of new proposed networks, [13] presented a model called Adversarial Variation Network (AVN) in order to verify handwritten signatures and generate new data by extracting features efficiently. The accuracy results obtained were 96.16% and 90.32%, for the CEDAR and GPDS-4000 databases. A total of 4 databases from different languages were used.

III. Capsule Network

CNNs were first published by [14]. The model is composed of convolutional and fully-connected layers, where the convolutional ones are responsible for extracting the basic characteristics that portray the content of the image, in addition to applying filters in order to facilitate the recognition of patterns that will be carried out in the next step. The second sequence of layers is responsible for classifying the attributes extracted from the first. CNNs require a minimum level of pre-processing when compared to other classification algorithms and have become extremely popular in image recognition and object detection problems.

CNNs have limitations, such as the significant loss of spatial relationships, due to the pooling process. Therefore, CNNs are not as efficient in investigating spatial relationships between features. From this perspective, [2] proposed the *Capsule Network*, a network focused on image recognition, aiming to solve some of the problems that CNNs present.

CapsNet is invariant to geometric transformations, such as rotation and translation, but it is sensitive to the appearance of specific features and their orientations, among other advantages. It protects the hierarchical links existing in the image, with the aim of determining the existence of a characteristic, as well as the spatial relationship with the other properties of the image. This is accomplished with vectors that contain information such as position, size, orientation, deformation, object texture of the entities.

CapsNet's main difference is the capsule concept. Capsules are sets of neurons structured in hierarchical layers that represent the instantiation properties of an object or part of an object. This group of neurons receives input elements and generates an output in the form of an activity vector. The capsule activity portrays the various properties that are present in the image, such as position, size and texture. A *squashing* function is applied to the activity vector, aiming to restrict its length to a value between 0 and 1.

The CapsNet architecture, shown in Fig. 1, has a total of 3 layers. The first is a convolutional layer, *Conv1*, which has 256 filters with 9x9 convolutions, with stride equal to 1 and Rectified Linear Unit (ReLU) activation function. The second layer is of primary capsules, *PrimaryCaps*, consisting of 32 convolutional capsules in which each capsule includes

8 convolutional units with a 9x9 kernel and stride 2. The last is a fully connected layer, *DigitCaps*, composed of a 16-dimensional capsule for each digit class, so that each capsule of these dimensions is connected to all capsules in the last layer. Finally, the output of the last layer is the activity vector.

Dynamic routing and routing by agreement is performed between the PrimaryCaps and DigitCaps capsules, allowing capsules at higher levels to serve active capsules at lower levels and ignore the others. This process allows the model to recognize multiple objects in the image, even if they overlap.

IV. Materials and Methods

A. Data sets

The CEDAR database [3] used in this article consists of off-line signatures where 1,320 are genuine and 1,320 are forged, totaling 2,640 off-line signatures. The samples are in gray scale, all are in English and in PNG format. They are divided into a folder of original images and a folder of forged images. Fig. 2 shows examples of genuine and forged signatures taken from the database.

The GPDS Synthetic On-line & Off-line Signature data set (GPDS) proposed by [4], is a database of off-line and on-line signatures which contains 10,000 signatures. It has 1,124 folders, where there are 24 genuine samples and 30 fake samples in each, totaling 26,976 original signatures and 33,720 forged ones. The static samples were obtained with different pen models, where all images are in PNG format and at 600 dpi resolution. Dynamic samples are in the MAT format, which contains the x and y coordinates of the signatures.

The images in each of the databases have different dimensions and a varied number of forged and original signatures. Table I presents a summary of the information for each of the databases.

TABLE I
Dataset Information

Dataset	**CEDAR**	**GPDS**
Language	English	Spanish
Number of samples	55	10,000
Number of folders	2	1,124
Original signatures per person	24	24
Forged signatures per person	24	30
Format	PNG	PNG
Resolution	NA	600 dpi

B. Pre-processing

In order to balance the number of images in the 2 databases, different procedures were applied. For CEDAR, the signatures of each person were divided into folders with 24 falsified images and 24 original images. For GPDS, 6 forged images were excluded from each of the folders, as there was a discrepancy in the number of originals and forged images. A partition in the databases was carried out, so that images of the same person would not used for the training, testing and validation stages.

Training, validation and test folders were created for the CEDAR and GPDS bases. The folders with the images were shuffled and divided with the following percentages, 70% for training, 20% for testing and 10% for validation.

A k-fold cross validation strategy with $k = 10$ was used, which consists of dividing the database into sets and using each of these sets for training and the other part for testing. K-fold divides the database randomly into k subsets and at each iteration a set consisting of $k - 1$ subsets are used for training and the rest are used for testing.

The files were saved in specific directories related to the training, validation and test images. For the part where the k-fold method was applied, the test section contained only 1 directory with the necessary images. For validation and training, directories were created for each of the splits carried out and each contained the corresponding images.

Initially, the image paths and their labels were stored. Original signatures were assigned label 0 and the remaining forged images the label 1. Subsequently, all images were read, converted to RGB mode, normalized in a range from 0 to 255, and a anti-aliasing filter was applied. The anti-aliasing filter was used in order to reduce the pixelated effect in signatures. This means that a low-pass filter was applied, aiming to make them look more smooth. Images have been resized to 64x64 pixels.

C. Evaluation Metrics

The metrics of False Acceptance Rate (FAR), False Rejection Rate (FRR), Avarage Error Rate (AER), Precision, Recall and Accuracy were used to evaluate the model quality. They were based on the number of true positives (TP) (original signatures considered valid), true negatives (TN) (forged signatures considered fake), false positive (FP) (forged signatures considered valid) and false negatives (FN) (original signatures considered fake).

The False Acceptance Rate (FAR) measures the number of forged signatures accepted as original. The value of this metric remaining low is interesting, as it aims to ensure that only genuine signatures have access:

$$FAR = \frac{FP}{FP + TN}. \tag{1}$$

The False Rejection Rate (FRR) is the number of genuine signatures rejected by the system:

$$FRR = \frac{FN}{FN + TP}. \tag{2}$$

The Average Error Rate (AER) is the average of the results obtained in FAR and FRR. The lower the value of this metric, the higher the signature recognition accuracy:

$$AER = \frac{FAR + FRR}{2}. \tag{3}$$

Accuracy is a percentage that represents the fraction of correct instances in relation to the total number of predictions made, that is, the model's ability to avoid both false positives and false negatives. Recall measures the percentage of original predictions correctly classified in relation to the total number of original instances, that is, the model's ability to find positive cases.

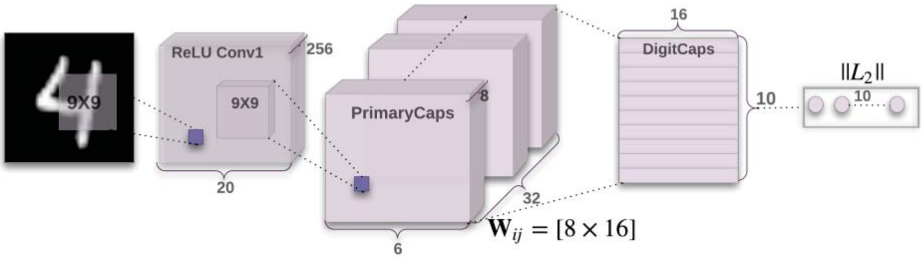
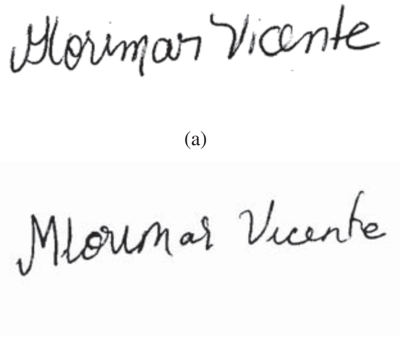


Fig. 1 The Capsule Network Architecture consisting of a convolutional layer, *Conv1*, a layer of primary capsules, *PrimaryCaps*, and a fully connected layer, *DigitCaps* applied to digit recognition [2]



(a)

(b)

Fig. 2 Sample of signatures from CEDAR: (a) True signature (b) Forged signature

D. Methods

The results obtained from CapsNet were compared with those from *MobileNetV2*, in order to verify CapsNet's performance. The reason for choosing *MobileNetV2* over other CNN models is its simplicity allied to high precision, as a compromise solution that is suitable to simpler hardware systems. The Python programming language was chosen together with TensorFlow, Keras and Sklearn. A machine with an 8th generation Intel Core i7 processor, 20 GB RAM and A100 GPU was used for k-fold training.

The reasons why CapsNet was chosen over other more recent networks in the literature are the following: they have hierarchical learning through the concept of capsules, they have greater interpretability due to the fact that capsules are responsible for representing specific entities. They are invariant to transformations, that is, the network can recognize an object, regardless of its orientation or position within the image. After the necessary investigations, not many reliable articles were found about CapsNet being considered a potential network for the area of signature recognition.

The network used in this article used the hyperparameters presented in Table II. The loss function used for CapsNet was *Margin Loss*, as proposed in the original paper. For MobileNetV2, the loss function was the *Binary Cross-Entropy*.

TABLE II
List of Hyper-Parameters

Parameters	Vaue
Optimizer	Adam
Learning rate	0.0001
Batch size	128

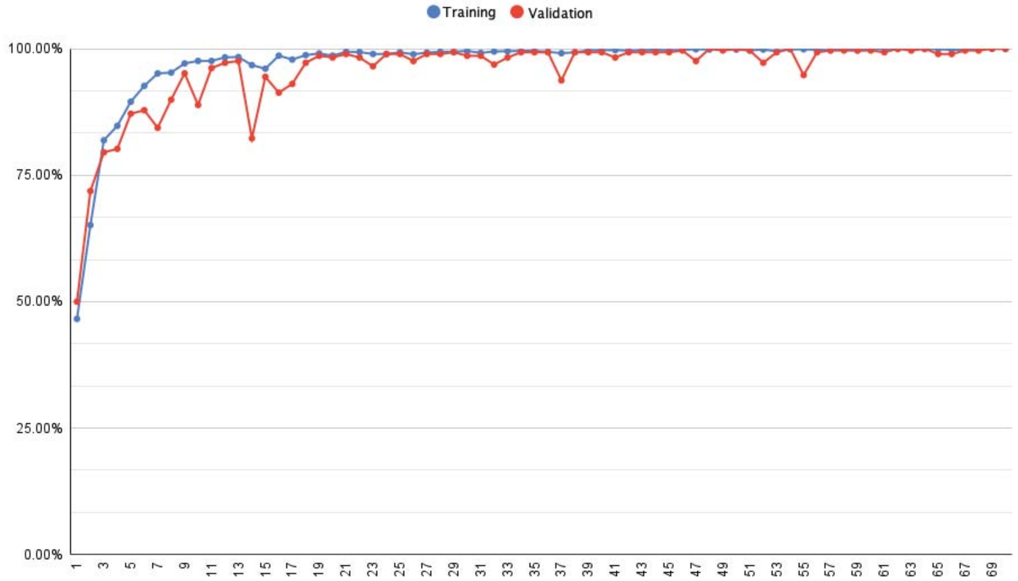
V. Results

Certain points need to be clarified before starting a discussion of the results obtained with the chosen neural networks. Firstly, it was not possible to compare the results achieved from the CEDAR and GPDS bases, as they are completely different sets, but mainly due to the fact that the GPDS has synthetic data. Second point, the article [15] in which the CapsNet network was presented uses the writer-dependent approach, where the network is trained for each person. However, the present work employed the writer-independent approach, in which one model is trained for all individuals.

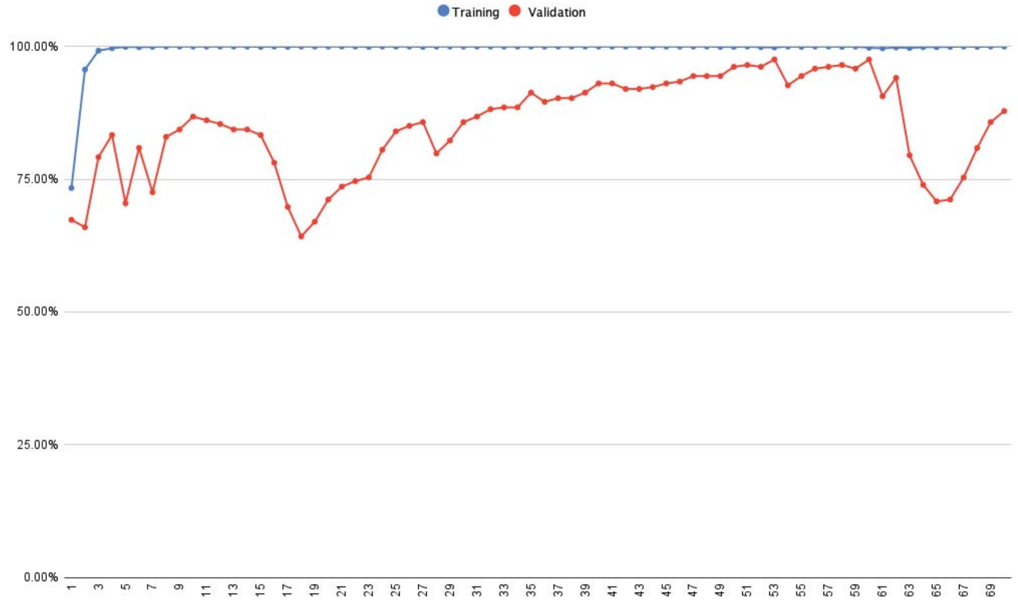
Fig. 3 presents two plots of the accuracy results obtained for training and validation for the CEDAR database using CapsNet and MobileNetV2. The blue curve represents the training accuracy and the red curve represents the validation accuracy throughout the training.

For the CapsNet, it can be seen that at certain epoch values, the validation curve decreases, however, in the next epoch it returns to be close to the training curve. After epoch 55, it is clear that the training and validation curves are close. The behavior that the CapsNet presents is that the network managed to learn the signatures from the CEDAR database. As for the MobileNetV2 plot, it is observed that for the validation curve there are several variations throughout the epochs. The training curve presents a more stable behavior, and from epoch 3 onwards, the values remain stable without a sudden change in the curve. Therefore, it is possible to notice that MobileNetV2 did not reach stable learning, as CapsNet did.

In Table III it can be seen that the CapsNet test results outperformed those of MobileNetV2. CapsNet achieved an accuracy of 98.11%, while MobileNetV2 reached only 81.63%. The most relevant point is presented in the False Acceptance Rate, False Rejection Rate and Average Error Rate metrics. The FAR and FRR values for CapsNet were 3.41% and 0.38%, respectively. In contrast, MobileNetV2 achieved a FAR of 94.85% and FRR of 60.09%. High FAR and FRR values tell that MobileNetV2 is considering forged signatures as genuine and is rejecting originals. The AER values for CapsNet and MobileNetV2 were 1.89% and 77.47%, respectively. The low AER value for CapsNet



(a)



(b)

Fig. 3 Accuracy results obtained for CEDAR during training (blue) and validation (red) for 70 epochs with: (a) CapsNet, and (b) MobileNetV2

indicates that the network has high signature recognition accuracy. MobileNetV2 has low accuracy in this task.

The results achieved for CEDAR, using the CapsNet network, were extremely satisfactory, being demonstrated by the accuracy metric, which obtained a difference of 22.9 percentage points. This demonstrates that CapsNet learned effectively in training and the accuracy when identifying signatures is high. On the other hand, MobileNetV2 did not obtain such good results, despite an accuracy of 81.63%. The values achieved with the FAR, FRR and AER metrics indicated that the model is not performing well enough to recognize signatures, and consequently, it did not reached the necessary learning level.

Fig. 4 shows a plot of the accuracy values for the training and validation with GPDS for the MobileNetV2 network. The blue and red curve are the training and validation accuracy, respectively. Firstly, it is noted that training and validation do not intersect at any time throughout the execution. At some epochs validation values show a behavior of slight increasing followed by decreasing. For training, the values begin to rise

TABLE III
RESULTS OBTAINED FOR CEDAR

Network	FAR [1]	FRR [2]	AER [3]	Precision	Recall	Accuracy
CapsNet	3,41%	0,38%	1,89%	96,69%	99,62%	98,11%
MobileNetV2	94,85%	60,09%	77,47%	73,79%	98,11%	81,63%

[1] False Acceptance Rate, [2] False Rejection Rate, [3] Avarage Error Rate

TABLE IV
INITIAL RESULTS OBTAINED FOR THE GPDS DATABASE

Neural Network	FAR [1]	FRR [2]	AER [3]	Accuracy	Recall	Accuracy
CapsNet	21.54%	18.15%	19.85%	76.17%	81.85%	80.15%
MobileNetV2	63.20%	14.18%	38.69%	62.92%	85.82%	64.03%

[1] False Acceptance Rate, [2] False Rejection Rate, [3] Avarage Error Rate

TABLE V
t STATISTICS AND P-VALUES OBTAINED FOR EEACH METRIC

Metric	Accuracy	Recall	Precision	FAR [1]	FRR [2]	AER [3]
t	94.5	19.09	33.45	-13.81	-19.09	-93.98
p	4.2e-15	6.8e-9	4.7e-11	1.2e-7	6.8e-9	4.4e-5

[1] False Acceptance Rate, [2] False Rejection Rate, [3] Avarage Error Rate

TABLE VI
MEAN AND STANDARD DEVIATION VALUES OBTAINED BY CAPSNET FOR EACH METRIC

Metric	Accuracy	Recall	Precision	FAR [1]	FRR [2]	AER [3]
Average	82.6%	81.5%	83.4%	16.3%	18.5%	17.4%
Standard Deviation	0.5%	3.3%	1.9%	3%	3.3%	0.5%

[1] False Acceptance Rate, [2] False Rejection Rate, [3] Avarage Error Rate

progressively and after epoch 20, they approach 100% but do not decrease at any time. The plots shows that the network was unable to efficiently learn the recognition of original and forged signatures.

Fig. 5 presents a plot of the training and validation results obtained for the GPDS database using CapsNet, where the blue curve is the accuracy for training and the red curve is the validation. Initially, for epochs 1 and 2, the validation values achieved are higher than the training values. However, from epoch 3 onwards, training begins to overlap validation. For the training curve, an increase in accuracy values is observed over the 30 epochs. However, it is notable that the values of the validation curve, after epoch 7, begin to decline. In this way, the plot illustrate the problem of overfitting, which basically occurs when the model has excellent results in training while performing poorly in validation and testing.

Table IV shows the CapsNet and *MobileNetV2* test results, where it can be seen that the performance of neither of the 2 networks was satisfactory. However the results of CapsNet were superior to those of MobileNetV2. A metric in which it is possible to observe a difference between the 2 networks is accuracy, where MobileNetV2 obtained 63.04%, while CapsNet obtained 80.15%, a difference of 17.71 percentage points. The only metrics that stood out from CapsNet were Recall where it reached 85.82% and False Rejection Rate where it reached 14.18%, it is important to highlight that a reduced value of the latter is desirable. However, as previously stated, neither of the results from the 2 networks were surprisingly good. This indicates that the networks are not achieving good accuracy when identifying signatures, where many of them are being considered forged, although they are original.

An evaluation of the statistical significance of the results from the k folds was carried out using the paired t test with a type-I error rate of 0.05. The same null hypothesis was considered for all metrics, i.e. that using CapsNet would not impact the results (equal averages).

Table V shows the t statistics and p-values for each metric. For the accuracy, precision and recall, the null hypothesis is rejected, as the p-values are less than 0.05. Therefore, the mean difference is high enough to claim that it is statistically significant. The smaller the p-value, the more it supports the alternative hypothesis. For the FAR, FRR and AER metrics, the p-values are less than 0.05, indicating that the null hypothesis is rejected.

Tables VI and VII present the mean and standard deviation values for the experiments with the CapsNet and MobileNetV2 networks, respectively. Considering the alternative hypothesis that the average values for accuracy, recall and precision are higher for CapsNet, it is noted that it stood out compared to the results of MobileNetV2, achieving accuracy of 82.6%, recall was 81.5% and precision was 83.4%. As for the average values of FAR, FRR and AER, where the alternative hypothesis is that the values obtained with CapsNet are lower than those of MobileNet, it is observed that CapsNet, again, obtained lower values for these metrics. For the FAR metric, CapsNet achieved an average of 16.3%, while MobileNetV2 achieved an average of 71.8%. Therefore, this indicates that CapsNet is significantly better, that is, the learning of genuine and forged signatures occurred more effectively than MobileNetV2. The standard deviation values for CapsNet were low, as evidenced with the False Acceptace Rate metric in

TABLE VII
MEAN AND STANDARD DEVIATION VALUES OBTAINED BY MOBILENETV2 FOR EACH METRIC

Metric	Accuracy	Recall	Precision	FAR [1]	FRR [2]	AER [3]
Average	22.9%	17.6%	18.5%	71.8%	82.4%	77.1%
Standard Deviation	1.9%	9.8%	5.8%	12.1%	9.8%	2%

[1] False Acceptance Rate, [2] False Rejection Rate, [3] Average Error Rate

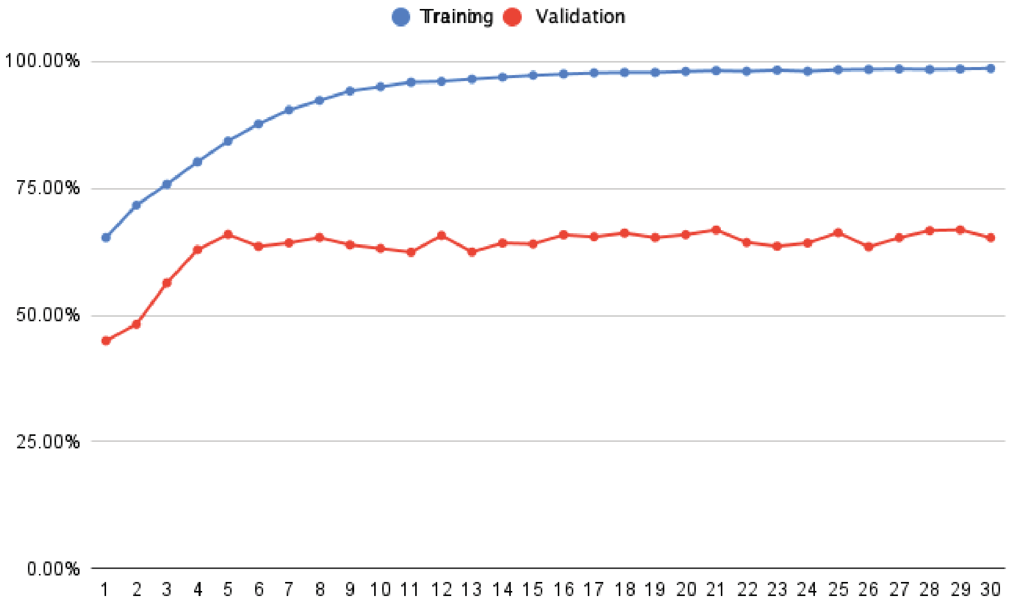


Fig. 4 Accuracy results obtained for GDPS during training (blue) and validation (red) for 30 epochs using MobileNetV2

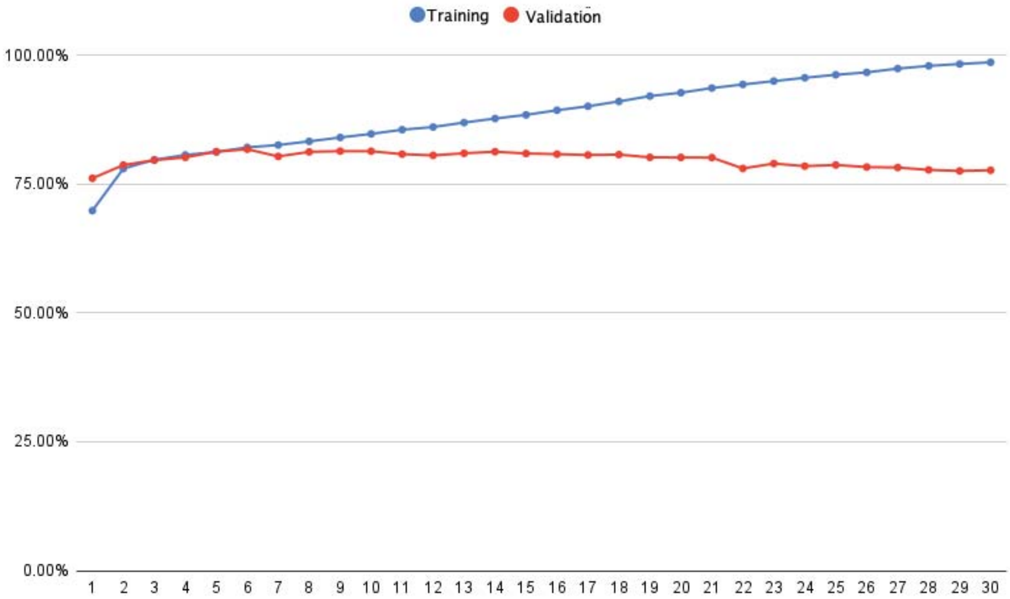


Fig. 5 Accuracy results obtained for GDPS during training (blue) and validation (red) for 30 epochs using CapsNet.

which CapsNet achieved 3% while MobileNetV2 achieved 12.1%. It is important to highlight that the consistency of the model is directly related to a reduced value of standard deviation.

After analyzing the values obtained in Tables V, VI and VII it is possible to state that the CapsNet is significantly superior for GPDS, as evidenced by the p-value equal to 4.2e-15 for accuracy. When analyzing the MobileNetV2 scenario, CapsNet proved to be more reliable for identifying signatures.

Finally, CapsNet was also superior to MobileNetV2 for CEDAR, as it obtained a difference of 16.48 percentage points. Furthermore, CapsNet demonstrated to be statically significant in the task of signature recognition. However, neither of the 2 networks achieved results higher than 90% in accuracy for GPDS, making it necessary to apply Overfitting control techniques.

VI. Conclusion

The main objective of the article was to evaluate the performance of the Capsule Network compared to another CNN of the same size for the task of identifying genuine and forged signatures. The chosen network was the MobileNetV2 because of its simplicity and high performance on many other image recognition applications. Two public databases were used to investigate CapsNet's performance. The CEDAR and GDPS data sets have different natures and have undergone the same normalization and pre-processing techniques. K-Fold cross-validation was applied, aiming to verify the model's level of reliability.

The results obtained for CapsNet were superior to those for MobileNetV2, especially for CEDAR, achieving accuracy of 98.11% for CapsNet and 81.63% for MobileNetV2. The same pattern was observed for the GDPS dataset so that it could be possible to verify that the performance of Capsule Network was superior compared to MobileNetV2.

Future work include comparing Capsule Networks with other CNNs that require more computational effort. The application of procedures to solve Overfitting, such as Droupout, data augmentation to increase the number of images in the base, changes in the loss function, among other possibilities may also improve the already high performance obtained with Capsule Networks.

Acknowledgment

The authors thank the Pontifícia Universidade Católica de Minas Gerais – PUC-Minas and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES – (Grant PROAP 88887.842889/2023-00 – PUC/MG, Grant PDPG 88887.708960/2022-00 – PUC/MG - INFORMÁTICA and Finance Code 001).

References

[1] E. A. Soelistio, R. E. Hananto Kusumo, Z. V. Martan, and E. Irwansyah, "A review of signature recognition using machine learning," in *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, vol. 1, 2021, pp. 219–223.

[2] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *ArXiv*, vol. abs/1710.09829, p. 3859–3869, 2017.

[3] S. Srihari, S.-H. Cha, H. Arora, and S. Lee, "Individuality of handwriting," *Journal of forensic sciences*, vol. 47, pp. 856–72, 08 2002.

[4] M. A. Ferrer, M. Diaz, C. Carmona-Duarte, and A. Morales, "A behavioral handwriting model for static and dynamic signature synthesis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1041–1053, 2017.

[5] V. L. F. Souza, A. L. I. Oliveira, and R. Sabourin, "A writer-independent approach for offline signature verification using deep convolutional neural networks features," in *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, 2018, pp. 212–217.

[6] D. Gumusbas and T. Yildirim, "Offline signature identification and verification using capsule network," in *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 2019.

[7] N. Arab, H. Nemmour, and Y. Chibani, "New local difference feature for off-line handwritten signature verification," in *2019 International Conference on Advanced Electrical Engineering (ICAEE)*, 2019, pp. 1–5.

[8] ——, "Improved multi-scale local difference features for off-line handwritten signature verification," in *2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)*, 2020, pp. 266–270.

[9] ——, "Multiscale fusion of histogram-based features for robust off-line handwritten signature verification," in *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, 2020, pp. 1–5.

[10] S. V. Bonde, P. Narwade, and R. Sawant, "Offline signature verification using convolutional neural network," in *2020 6th International Conference on Signal Processing and Communication (ICSC)*, 2020, pp. 119–127.

[11] Z. Mohammad, I. Jahan, M. M. Kabir, M. A. Ali, and M. Mridha, "An offline writer-independent signature verification system using autoembedder," in *2021 24th International Conference on Computer and Information Technology (ICCIT)*, 2021, pp. 1–6.

[12] W. Xiao and D. Wu, "An improved siamese network model for handwritten signature verification," in *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2021, pp. 1–6.

[13] H. Li, P. Wei, and P. Hu, "Avn: An adversarial variation network model for handwritten signature verification," *IEEE Transactions on Multimedia*, pp. 594–608, 2022.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *CoRR*, vol. abs/1710.09829, 2017. [Online]. Available: http://arxiv.org/abs/1710.09829