International Journal on Engineering Technology and Infrastructure Development (InJET-InDev)
www.lec.edu.np/journal
Volume 2, Issue no. 2, Page 26-36
Received Date: 10/31/2025
Revised Date: 5/20/2026
Accepted Date: 5/26/2026
ISSN: 3021-999X (Print)
https://doi.org/10.3126/injet-indev.v2i2.95695

Signature Verification using ResNet-based Custom CNN, Advanced Augmentation, and Explainable AIs

Prajwal Chaudhary[1], Yuvraj Aryal[2], Rajad Shakya[3]

[1]*Department of Electronics and Computer, IOE Thapathali Campus, Nepal, prajwalchy25@gmail.com*
[2]*Department of Electronics and Computer, IOE Thapathali Campus, Nepal, yuvrajaryal83@gmail.com*
[3]*Department of Electronics and Computer, IOE Thapathali Campus, Nepal, shakyarajad1@gmail.com*

Abstract

Handwritten signature verification has long served as one of the most trusted forms of identity authentication, widely relied upon in banking, legal, and administrative settings where an individual's presence cannot be guaranteed. However, manual verification by human examiners remains susceptible to error, highlighting the need for accurate and automated alternatives. To address this, a writer-dependent offline signature verification system is developed, built around a custom ResNet-based Convolutional Neural Network (CNN) enhanced with advanced data augmentation and Explainable AI. Since collecting real forged signatures is inherently difficult, synthetic forgeries are generated using augmentation techniques, including ElasticTransform, RandomAffine, GridDistortion, ThinStroke, and ThickStroke, allowing the model to learn meaningful distinctions between genuine and forged samples. Signatures are preprocessed through median blur, grayscaling, denoising, segmentation, padding, resizing, and negation before being fed into the network. Training is guided by the Adam optimizer with exponential learning rate decay, complemented by early stopping, L2 regularization, and dropout to prevent overfitting. Grad-CAM is further incorporated to shed light on the model's decision-making process, offering transparency often absent in deep learning systems. Across ten writers, the system achieved an average accuracy of 83.55%, a False Acceptance Rate (FAR) of 14.92%, and a False Rejection Rate (FRR) of 18.35%.

Keywords: Convolutional Neural Network, Explainable AI, ResNet, Advanced Augmentations

1. Introduction

Handwritten signature verification is perhaps one of the oldest and most universally trusted forms of identity authentication known to modern society. Long before fingerprint scanners or facial recognition systems existed, signatures were already being used to authorize documents, validate agreements, and confirm identity across banking, legal, and administrative settings (Garhawal, S. and Shukla, N., 2013). What makes signature verification particularly appealing is its simplicity; unlike other biometric modalities such as fingerprint scanning, retinal recognition, or voice authentication, it requires no specialized hardware and fits naturally into documentation practices that people and institutions have depended on for centuries. That said, not all signature verification works the same way. The field is broadly divided into two camps: online and offline (Hafemann, L.G., Sabourin, R. and Oliveira, L.S., 2017). Online verification is the more data-rich of the two, capturing dynamic attributes like pen pressure, stroke speed, and trajectory as the signature is being written, using digital input devices. Offline verification takes a quieter, more retrospective approach, working from static images of signatures already captured on physical documents, such as bank cheques, official contracts, or application forms, and subjecting them to image processing and pattern recognition techniques (Garhawal, S. and Shukla, N., 2013). While online systems have the advantage of richer behavioral data, offline verification is far more deeply embedded in real-world workflows, where documents change hands and get reviewed long after the ink has dried.

The trouble, however, is that leaving this verification to human examiners has never been particularly reliable. Even trained professionals have been shown to struggle when asked to consistently tell a skilled forgery apart from a genuine signature, and in a world where the stakes can be extraordinarily high, that inconsistency is a serious problem. It is precisely this gap that has driven growing interest in automated approaches, with deep learning methods and Convolutional Neural Networks (CNNs) in particular emerging as strong candidates for

the task (Hafemann, L.G., Sabourin, R. and Oliveira, L.S., 2017; Khalajzadeh, H., Mansouri, M. and Teshnehlab, M., 2016). Among the strategies explored, writer-dependent (WD) classification, where a dedicated model is trained for each individual signer rather than across all signers at once, has consistently proven more effective at capturing the subtle, personal characteristics that make each person's signature uniquely their own (Garhawal, S. and Shukla, N., 2013; Hafemann, L.G., Sabourin, R. and Oliveira, L.S., 2017). And yet, despite these advances, many existing systems still fall short in ways that matter. Training data is often scarce (Batista, L., Granger, E. and Sabourin, R., 2012); forged samples are hard to come by; and perhaps most frustratingly, the models themselves tend to operate as black boxes, offering predictions without any real explanation of how they arrived at them. In domains like banking or legal authentication, that lack of transparency is not just inconvenient; it is a genuine barrier to trust and adoption.

This work sets out to tackle these challenges head-on. A writer-dependent offline signature verification system is proposed, built around a custom ResNet-based CNN architecture and enriched with Squeeze-and-Excitation (SE) blocks that allow the network to be more selective about which features it pays attention to. To get around the persistent problem of limited forgery data, advanced augmentation techniques, including ElasticTransform, RandomAffine, GridDistortion, ThinStroke, and ThickStroke, are used to synthetically generate realistic forged signatures, giving the model enough exposure to learn meaningful boundaries between genuine and forged samples. The system is evaluated on the UTSig Persian offline signature dataset (Bajestani, A.S., Fouladi, K. and Araabi, B., 2016), a well-established benchmark in the field. And because accuracy alone is not enough, interpretability is built directly into the pipeline: Gradient-weighted Class Activation Mapping (Grad-CAM) is used to produce visual explanations of the model's decisions, highlighting the exact regions of a signature image that tipped the scales one way or the other. The goal, ultimately, is a system that earns trust not just through its numbers, but through its transparency.

2. Literature Review

At the heart of signature verification research lies a deceptively simple question: Is it better to build a system that deeply understands one person's signature or one that can spot forgeries across many different signers? These two schools of thought, writer-dependent (WD) and writer-independent (WI) verification, have shaped the field for decades (Garhawal, S. and Shukla, N., 2013). Writer-dependent systems take a personal approach, training a dedicated classifier for each individual signer so the model can learn the subtle quirks and habits that make that person's signature uniquely theirs. Writer-independent systems, by contrast, take a broader view, training a single model across all writers and using a dissimilarity space to tell genuine signatures apart from forgeries (Kumar, M., 2012). While writer-independent systems are clearly more scalable, writer-dependent approaches have consistently delivered better accuracy in offline settings, where the dynamic richness of real-time signing data simply isn't available (Kumar, M., 2012).

Before deep learning arrived, researchers relied heavily on handcrafted features, carefully designed representations meant to capture what makes each person's signature unique. Kumar (2012) took a structured approach, extracting global and texture-based features from binarized signature images and feeding them into an Artificial Neural Network (ANN). Batista, Granger and Sabourin (2012) cast a wider net, surveying the range of feature extraction strategies available at the time and proposing generative-discriminative ensemble methods as a way to get more mileage out of limited training data; a challenge that would prove remarkably persistent. Guerbai, Chibani and Hadjadji (2015) tried something altogether different, building a writer-independent system around a One-Class SVM trained only on genuine signatures, neatly sidestepping the need for forgery samples. It worked to a point, but the system struggled whenever even genuine samples were scarce. Across these early works, a few recurring tensions became clear: the need for robust features, the chronic shortage of training data, and the difficulty of building classifiers that could hold up on unseen examples. Data augmentation gradually emerged as one of the most practical responses to the data problem (Kingma, D.P. and Ba, J., 2015).

The arrival of deep learning shifted the landscape considerably. Convolutional Neural Networks, with their ability to learn directly from raw image data, turned out to be a natural fit for signature verification, and performance improved noticeably. Khalajzadeh, Mansouri and Teshnehlab (2016) were among the first to show what CNNs could do for Persian signatures, reaching an impressive 95% accuracy on their dataset. Around the same time, Miah et al. (2015) developed an ANN-based model for bank cheque verification, though it fell short

of working on real physical cheques — a gap that limited how far its findings could travel. Cozzens et al. (2017) pushed the CNN approach further, achieving 83% accuracy on the SIGCOMP 2011 dataset by using layered receptive fields to detect inconsistencies between genuine and forged samples, with banking fraud detection squarely in mind. Perhaps the most influential contribution of this era came from Hafemann, Sabourin and Oliveira (2017), who demonstrated convincingly that deep CNNs could learn highly discriminative features for offline verification, leaving handcrafted methods well behind. On the simpler end of the spectrum, Rezaei and Naderi (2019) showed that a Fully Convolutional Network could still reach 76.71% accuracy on a Persian dataset with no preprocessing at all, a result that raised interesting questions about the trade-off between simplicity and performance. More recently, Zois et al. (2019) conducted a thorough investigation of sparse representation techniques, showing that strong results were achievable even in writer-dependent scenarios where only a handful of reference samples were on hand.

Despite all this progress, several stubborn challenges remain. Collecting real forged signatures for training is genuinely difficult. Skilled forgeries are hard to obtain ethically and at any meaningful scale (Khalajzadeh, H., Mansouri, M. and Teshnehlab, M., 2016). Geometric and elastic augmentation techniques have been proposed as a workaround (Kingma, D.P. and Ba, J., 2015), but their use for simulating signature-specific forgeries has been explored only superficially. Perhaps more pressing is the question of interpretability. Deep learning models, for all their predictive power, are notoriously difficult to read, and in a domain where verification decisions can carry serious legal and financial consequences, a system that cannot explain itself is a system that is hard to trust. Selvaraju et al. (2017) offered a meaningful step forward with Gradient-weighted Class Activation Mapping (Grad-CAM), a technique that generates visual explanations of CNN decisions by highlighting the image regions most responsible for a given prediction, yet this capability has seen little uptake in existing signature verification systems. The present work builds on the UTSig Persian offline signature dataset (Hafemann, L.G., Sabourin, R. and Oliveira, L.S., 2017) as its evaluation platform, a well-established benchmark that provides a realistic and challenging testbed for the methods proposed here.

3. Methodology

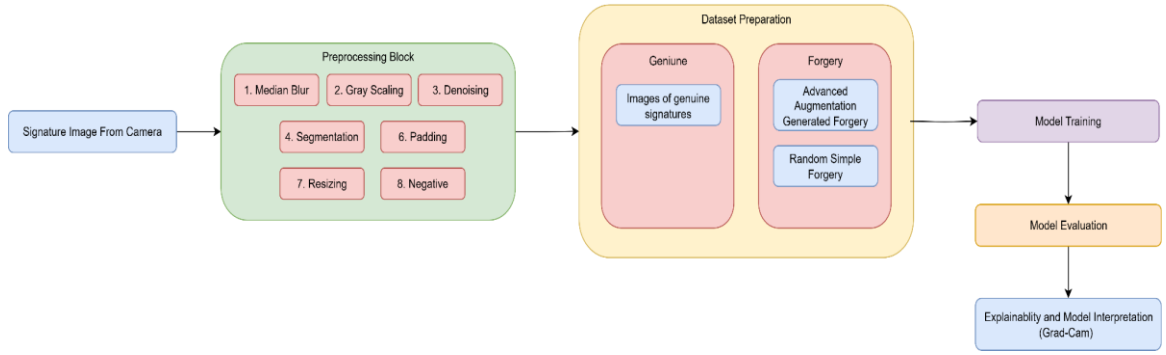


Figure 1. Proposed Methodology

3.1 Dataset Collection

Signature samples were drawn from the UTSig Persian offline signature dataset (Bajestani, A.S., Fouladi, K. and Araabi, B., 2016), a well-known benchmark in the signature verification field. Signatures from 10 different writers were selected for experimentation. For each writer, 27 genuine signatures, 3 opposite-hand forgeries, 66 simple forgeries, and 6 skilled forgeries were included, giving a reasonably diverse set of samples to work with.

Given that the system was built around writer-dependent binary classification, all signatures were organised into two classes: genuine and forged. To supplement the available forged samples, additional forgeries were generated synthetically through augmentation, allowing the model to be exposed to a wider range of variation than the original dataset alone could provide.

3.2 Data Preprocessing

Before training, a series of preprocessing steps was applied to all signature images to improve quality and ensure consistency across the dataset.

Median Blur: Median blur was first applied to reduce grainy noise present in the raw captured images.

Gray Scaling: Each 3-channel RGB image was then converted to a single-channel grayscale image by averaging the pixel values across all three channels.

Denoising: Fast non-local mean denoising was subsequently carried out using the OpenCV library to remove any remaining fine-grained noise.

Image Segmentation: Global thresholding was applied to isolate the signature from the background, with the threshold set to the mean pixel value of each image.

Padding: Padding was added along the shorter sides of each image to produce a square crop, ensuring the signature itself was not distorted in the process.

Resizing: All images were resized to 224×224 pixels to match the input requirements of the CNN model.

Negative: Finally, the negative of each image was computed using the transformation: negative = 255 − image, effectively inverting the pixel intensities.

Dataset Splitting: The full dataset was divided into training, validation, and test sets following a 70:20:10 ratio.

3.3 Data Augmentation

To address the inherent scarcity of both genuine and forged signature samples within the UTSig dataset, advanced augmentation techniques were employed for artificial expansion of the training set. Rather than the application of generic transformations, specific methods were selected to simulate the physical and psychological variations encountered in real-world signature verification.

3.3.1 Spatial Distortions for Forgery Simulation

To model the inconsistencies characteristic of forged signatures, such as hand tremors, muscle fatigue, and poor spatial judgment, two distinct spatial transformations were applied (Simard, Steinkraus and Platt, 2003):

Elastic Transformation: Local, non-linear muscle jitters were simulated through this technique. A random displacement field was generated and subsequently smoothed using a Gaussian filter to ensure local coherence. The displacement for a pixel at coordinates (x, y) was defined as:

$$\Delta x = \alpha . (R * G_\sigma) \quad \text{(Equation 1)}$$

$$\Delta y = \alpha . (R * G_\sigma) \quad \text{(Equation 2)}$$

where R represents a random field $R \in [-1,1]$, G_σ denotes a Gaussian kernel with standard deviation σ, and α is the scaling factor utilized to control deformation intensity.

Grid Distortion: Macro-structural misjudgments, where the correct proportions of the original signature are not maintained by the forger, were modeled using this method (Buslaev et al., 2020). The image was divided into a $n \times n$ mesh; the vertices $V_{i,j}$ were randomly shifted, and the new positions of pixels were calculated via bilinear interpolation:

$$P_{new}(x, y) = \sum_{i=1}^{4} w_i \cdot V_i \quad \text{(Equation 3)}$$

Where V_i represents the four displaced corner vertices and w_i denotes the weights relative to the distance of the pixel from each vertex.

3.3.2 Morphological Variations for Genuine Samples

Natural "intra-class" variations of a genuine writer such as changes in pen type or variations in downward pressure, were accounted for through the utilization of morphological operators (Gonzalez and Woods, 2018).

The stroke thickness was adjusted by these transformations without the fundamental structure of the signature being altered:

ThickStroke (Dilation): Higher pen pressure or a broader nib was simulated by the expansion of the signature boundaries:

$$S_{thick} = S \oplus B \qquad \text{(Equation 4)}$$

ThinStroke (Erosion): Light pressure or a fine-tipped pen was simulated by the contraction of the boundaries:

$$S_{thin} = S \ominus B$$

Through the implementation of these methods, the dataset was expanded to a final count of 81 genuine and 204 forged images. By this robust expansion strategy, the CNN was enabled to learn features invariant to pen type and minor spatial inaccuracies, thereby significantly reducing the likelihood of overfitting.

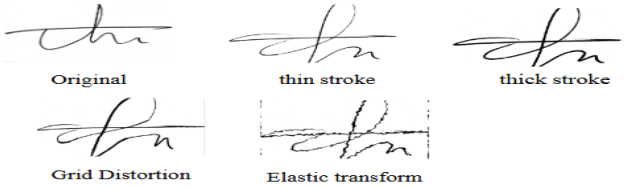


Figure 2. Advance Augmentation Techniques

3.4 Proposed CNN Architecture

A custom CNN architecture was developed, drawing inspiration from residual neural networks and other modern deep learning designs. The architecture was structured around three components: a stem, a backbone, and a head.

The stem handled initial low-level feature extraction through a single convolutional layer, followed by batch normalization and Swish activation. The resulting feature maps were then down sampled using max pooling to reduce spatial dimensions before being passed to the backbone.

The backbone was responsible for learning deeper, more abstract features. Residual blocks were used here to address the vanishing gradient problem; by allowing a portion of the input to bypass processing and pass directly to the output, information retention and gradient flow were both improved. Squeeze-and-Excitation (SE) blocks were also incorporated at this stage, enabling the network to learn which feature channels were most informative and weight them accordingly. This channel-wise attention mechanism improved classification performance without meaningfully increasing computational cost.

Finally, the head took the features produced by the backbone and converted them into a classification decision. Global average pooling was first applied to remove spatial dimensions, after which two fully connected layers were used to learn the most relevant combinations of features. A Softmax activation was applied at the output to convert the raw scores into class probabilities.

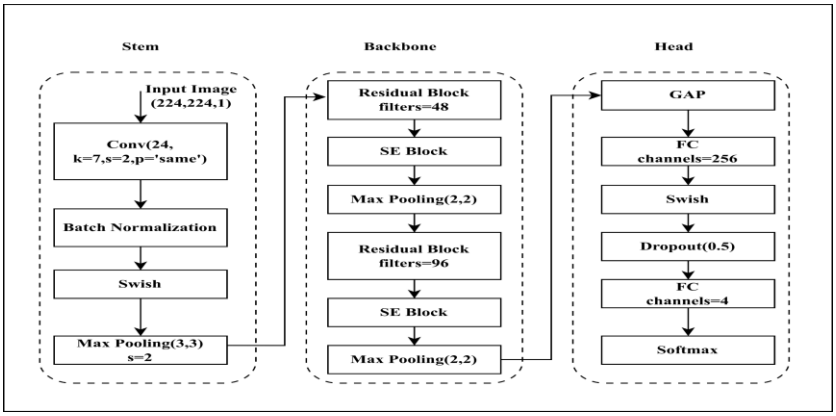


Figure 3. Residual Network-based Custom CNN Architecture

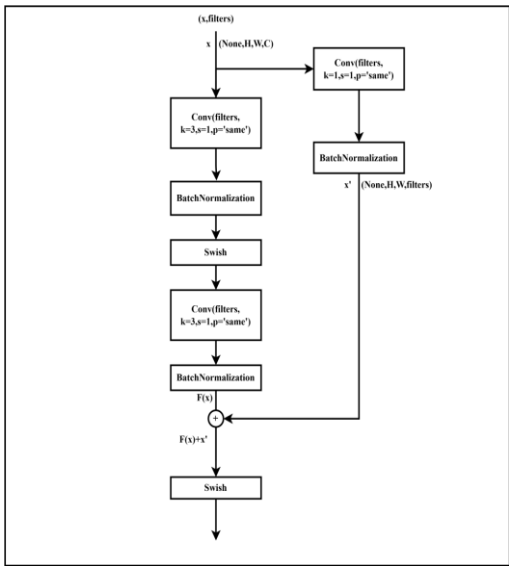


Figure 4. Residual Block

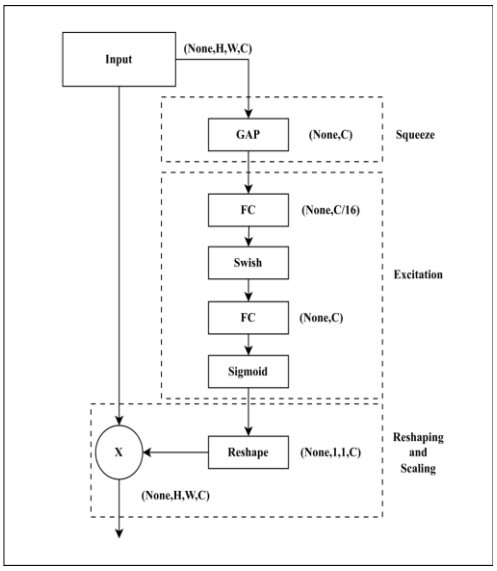


Figure 5. Squeeze and Excitation Block

3.5 Model Training

The model was trained on the preprocessed and augmented dataset described in the preceding sections. The Adam optimizer (Kingma, D.P. and Ba, J., 2015) was selected for its adaptive learning rate properties, with an initial learning rate of 0.001 that was decayed exponentially over training using decay steps of 10,000 and a decay rate of 0.9. This allowed the model to take larger steps early in training and progressively finer ones as it converged.

Several measures were taken to prevent overfitting. Early stopping was applied with a patience of 25 epochs, monitoring validation loss and saving the best-performing model checkpoint encountered during training. L2 regularization was used to discourage excessively large weights, and a dropout rate of 0.5 was applied within the fully connected layers. Label smoothing of 0.1 was also incorporated to improve generalization. Since the system was writer-dependent, a separate model was trained for each of the 10 writers, with each training run allowed up to 200 epochs at a batch size of 16.

3.6 Model Evaluation

Three metrics were used to evaluate the performance of the trained models: FAR, FRR, and overall accuracy.

3.6.1 FAR (False Acceptance Rate)

FAR captured how often forged signatures were mistakenly accepted as genuine; in other words, how frequently the system was fooled. It was computed as the number of forged signatures incorrectly accepted, divided by the total number of forged signatures presented.

$$FAR = \frac{Number\ of\ Forged\ Signatures\ Accepted}{Total\ Number\ of\ Forged\ Signatures} \qquad \text{(Equation 5)}$$

3.6.2 FRR (False Rejection Ratio)

FRR, on the other hand, measured how often genuine signatures were incorrectly turned away. It was calculated as the number of genuine signatures wrongly flagged as forgeries, divided by the total number of genuine signatures tested.

$$FRR = \frac{Number\ of\ Genuine\ Signatures\ Rejected}{Total\ Number\ of\ Genuine\ Signatures} \qquad \text{(Equation 6)}$$

3.6.3 Accuracy

$$Accuracy = \frac{\sum_{i=1}^{N} Genuine_i + \sum_{j=1}^{N} 1 - Forgery_j}{Total\ Number\ of\ Genuine\ Images} \qquad \text{(Equation 7)}$$

3.7 Explainability and Model Interpretation (Grad-CAM)

Once training and evaluation were complete, attention was turned to understanding why the model made the decisions it did. Gradient-weighted Class Activation Mapping, or Grad-CAM (Selvaraju, R.R. et al., 2017), was used for this purpose. Rather than treating the model as a black box, Grad-CAM allowed the regions of each signature image that most influenced the model's prediction to be visualised directly.

The technique worked by computing the gradients of the predicted class score with respect to the activations of the final convolutional layer. These gradients were globally averaged to produce a set of channel-wise importance weights, which were then multiplied by their corresponding feature maps and summed to generate a coarse localisation map. A ReLU activation was applied to this map to retain only the regions that positively contributed to the predicted class, and the resulting heatmap was overlaid on the original signature image. Through this process, the parts of the signature that the model found most diagnostic, whether a particular loop, stroke, or region, could be seen at a glance, offering a level of transparency that is rarely built into signature verification systems.

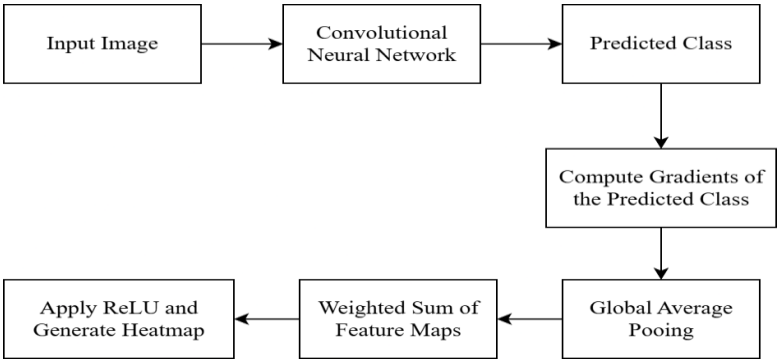


Figure 6. Grad-CAM Working Mechanism

4. Results and Analysis

The training behaviour of one of the writer-dependent classifiers is shown in Figure 7, tracking how both training and validation loss changed over 175 epochs. Early on, both losses fell sharply. The model was quickly and visibly picking up the most prominent patterns in the data. As training progressed, the pace of improvement naturally slowed, and both curves began to settle. By around the 50-epoch mark, the two lines had drawn close together and stayed that way, which is a good sign. When training and validation loss track each other closely over time, it generally means the model has learned to generalise rather than simply memorise, and that held true here. There were no dramatic divergences, no signs of runaway overfitting, just steady and well-behaved convergence through to the end of training.

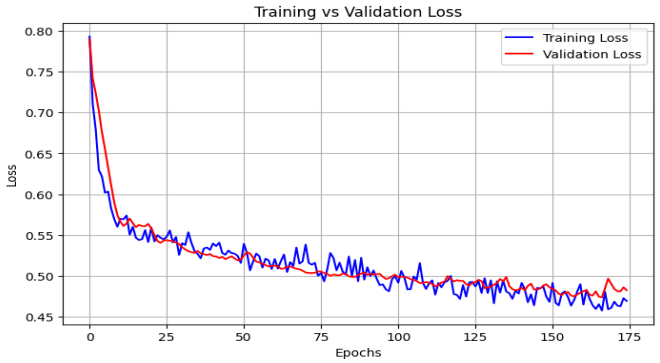
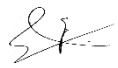
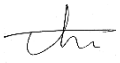
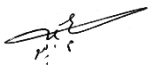
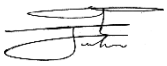
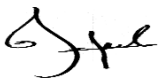


Figure 7. Training Graph

The per-writer results are laid out in Table 1, covering FAR, FRR, and accuracy for each of the 10 writers tested.

Table 1. Accuracy, FAR, FRR table

SN	Signature	FAR	FRR	Accuracy
1		9%	50.13%	63.22%
2		13.64%	11.11%	87.10%
3		9.09%	0%	93.55%
4		9.09%	0%	93.55%
5		50%	11.11%	61.29%
6		18.18%	44.44%	74.19%
7		0%	11.11%	97.77%
8		13.64%	11.11%	87.10%
9		13.64%	22.22%	83.87%
10		13.64%	22.22%	93.87%

Across all ten writers, an average FAR of 14.92%, FRR of 18.35%, and overall accuracy of 83.55% were recorded. Some writers returned particularly impressive numbers — Writer 7 stood out with a FAR of 0% and an accuracy of 97.77%, while Writers 3 and 4 both hit 93.55% accuracy with a FRR of 0%, meaning every genuine signature presented for those two writers was correctly accepted. These results suggest that for signers with consistent and distinctive signing styles, the model performed very reliably.

Not every writer told such a clean story, though. Writer 1 came with an FRR of 50.13%; genuine signatures were being turned away at an unusually high rate, which most likely points to considerable variability in how that person signs. When genuine samples differ too much from one another, even a well-trained model can struggle to decide what counts as authentic. Writer 5 presented the opposite problem, with a FAR of 50%; half of all forged signatures were accepted as genuine. This kind of error tends to occur when a writer's signature is relatively simple or lacks highly distinctive features, making it easier for forgeries to pass undetected. Writer 6 sat uncomfortably in the middle, with elevated error rates on both ends (FAR of 18.18% and FRR of 44.44%), suggesting a signature that was neither consistent enough to learn well nor distinctive enough to defend against imitation. These cases are a reminder that writer-dependent systems are only as strong as the signatures they are trained on. When the underlying samples are variable or simple, the model's job becomes considerably harder. Placed alongside related work, the results hold up well given the constraints of the experimental setup. Cozzens et al. (2017) reported 83% accuracy on the SIGCOMP 2011 dataset using a CNN with layered receptive fields, and the proposed system reached a closely comparable 83.55% average, despite working with a much smaller

dataset and relying entirely on synthetically generated forgeries rather than real ones. Rezaei and Naderi (2019) achieved 76.71% accuracy on a Persian dataset using a Fully Convolutional Network with no preprocessing at all, a figure that the present system comfortably exceeded through careful preprocessing and targeted augmentation. At the upper end of the comparison, Khalajzadeh, Mansouri and Teshnehlab (2016) reached 95% accuracy on Persian signatures, though their work had the advantage of a larger and more balanced dataset. Taken in that context, the results reported here are competitive, particularly given that only 27 genuine samples per writer were available and all forgery data was synthetically produced.

The figure above demonstrates a Grad-CAM heatmap overlaid on a sample signature image from the test set. The colors represent the intensity of the model's attention during the classification process. Red and yellow areas are the regions that the model considers most important in making its decision. Blue or faded regions have less influence on the model's decision. The upper-left loop and lower-left stroke exhibit strong activation, indicating that these regions are critical to the model's classification decision. The background and unrelated areas are faint or blue, which is expected and desirable behavior, as it suggests the model is correctly focusing on the signature rather than on noise.

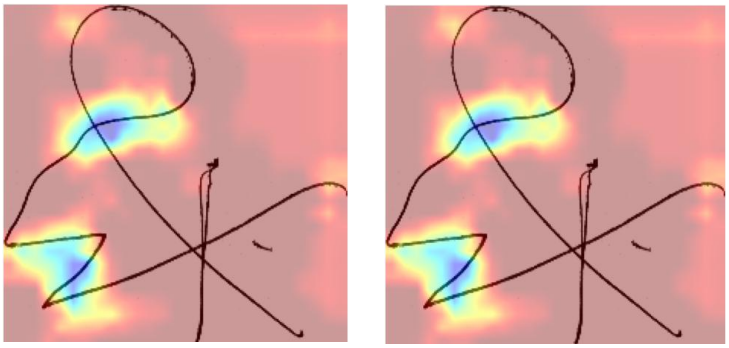


Figure 8. Grad-CAM Visualization

The experimental results demonstrate that the proposed ResNet-based custom CNN can effectively distinguish between genuine and forged signatures despite limited training samples. Residual and squeeze-and-excitation blocks improved feature learning capability. Advanced augmentation techniques significantly improved generalization performance by simulating realistic forged signatures. Similarly, Grad-CAM visualizations confirmed that the CNN focused on meaningful signature stroke regions rather than background noise.

5. Conclusion

Offline signature verification is a problem that sits at the intersection of pattern recognition, security, and trust, and building a system that handles it well, reliably, and transparently is harder than it might first appear. This work took on that challenge by developing a writer-dependent verification system built around a custom ResNet-based CNN, supported by advanced data augmentation and Grad-CAM explainability.

One of the most persistent obstacles in this space is the scarcity of forged signature samples for training. That problem was addressed here by synthetically generating forgeries using ElasticTransform, GridDistortion, ThinStroke, and ThickStroke augmentation techniques, giving the model enough exposure to meaningful variation without relying on difficult-to-obtain real forgeries. The approach paid off across ten writers, an average accuracy of 83.55% was achieved, with a FAR of 14.92% and FRR of 18.35%, and several individual writers saw results well above 90%.

Each of the core objectives set out at the start of this work was met. A writer-dependent binary classification system was successfully built and shown to distinguish genuine signatures from forgeries using deep learning. The data scarcity problem was tackled through targeted augmentation, which meaningfully expanded the training set without distorting the underlying signature characteristics. The ResNet-inspired architecture, enriched with Squeeze-and-Excitation blocks, proved effective at extracting the fine-grained features that separate one person's signature from an imitation of it. And the integration of Grad-CAM brought something

that is too often missing from deep learning systems: a way to see, not just trust, why a decision was made. In a domain where verification outcomes can carry real legal and financial weight, transparency matters.

That said, the results were not uniform across all writers, and that variability is worth taking seriously. Writers with simpler or less consistent signatures posed a genuine challenge, and those cases point clearly toward where the next round of improvements should be focused: better data balancing, larger per-writer sample sets, and training strategies that adapt more gracefully to individual signing habits.

6. Future Enhancement

There is plenty of room to build on what has been done here. On the data side, Generative Adversarial Networks (GANs) could be explored as a more powerful alternative to augmentation for producing realistic forged signatures, one that goes beyond spatial distortions and captures the subtler stylistic patterns that make a convincing forgery. Larger and more diverse datasets, spanning multiple languages and writing traditions, would also help push robustness further than a single Persian dataset can.

On the architecture side, Transformer-based models and Vision Transformers (ViTs) are worth investigating as alternatives to the CNN backbone used here, given the strong results they have shown on image recognition tasks more broadly. A writer-independent system, one model that handles all signers rather than a separate model per writer, would be a significant step forward in terms of scalability and practical deployability.

For explainability, more expressive techniques such as SHAP or LIME could be integrated alongside or in place of Grad-CAM, offering finer-grained insight into what the model is responding to. And finally, real-time and mobile deployment remain the natural end goal for a system like this, bringing signature verification out of the research setting and into the banking counters, legal offices, and administrative workflows where it is actually needed.

References

Garhawal, S. and Shukla, N., 2013. A study on handwritten signature verification approaches. *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, 2(8), pp.2497–2503.

Bajestani, A.S., Fouladi, K. and Araabi, B., 2016. UTSig: A Persian offline signature dataset. *Preprint, submitted to IET Biometrics*.

Hafemann, L.G., Sabourin, R. and Oliveira, L.S., 2017. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70, pp.163–176.

Kumar, M., 2012. Signature verification using neural network. *International Journal on Computer Science and Engineering (IJCSE)*, 4(9).

Batista, L., Granger, E. and Sabourin, R., 2012. Dynamic selection of generative–discriminative ensembles for off-line signature verification. *Pattern Recognition*, 45(4), pp.1326–1340.

Khalajzadeh, H., Mansouri, M. and Teshnehlab, M., 2016. Persian signature verification using convolutional neural networks. *International Journal of Engineering Research and Technology*, 1(2), pp.7–12.

Miah, M.B.A. et al., 2015. Handwritten courtesy amount and signature recognition on bank cheque using neural network. *International Journal of Computer Applications*, 118(5).

Guerbai, Y., Chibani, Y. and Hadjadji, B., 2015. The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recognition*, 48(1), pp.103–113.

Cozzens, B. et al., 2017. Signature verification using a convolutional neural network. *BigData*, pp.2644–2651.

Rezaei, M. and Naderi, N., 2019. Persian signature verification using fully convolutional networks. arXiv preprint arXiv:1909.09720.

Darbon, J. et al., 2008. Fast nonlocal filtering applied to electron cryomicroscopy. *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, pp.1331–1334.

Kingma, D.P. and Ba, J., 2015. Adam: A method for stochastic optimization. *University of Toronto*.

Shorten, C. and Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), pp.1–48.

Selvaraju, R.R. et al., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp.618–626.

Zois, E.N. et al., 2019. A comprehensive study of sparse representation techniques for offline signature verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(1), pp.68–81.

Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Kurichev, M. and Kalinin, A.A., 2020. Albumentations: Fast and flexible image augmentations. Information, 11(2), p.125.

Gonzalez, R.C. and Woods, R.E., 2018. Digital Image Processing. 4th edn. New York: Pearson.

Simard, P.Y., Steinkraus, D. and Platt, J.C., 2003. Best practices for convolutional neural networks applied to visual document analysis. Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR), pp.958–963.