

FUSION OF GLOBAL OPTIMIZED CONVOLUTIONAL NEURAL NETWORK WITH DEEP FLEXIBLE NETWORK FOR DIGIT RECOGNITION

PRIYANGA K.K.^{1,3,*}, S. SABEEN²

^{1,*} Research Scholar, Department of Computer Science, FSH, SRMIST, Kattankulathur, Tamil Nadu-603203

² Assistant Professor, Department of Computer Science, FSH, SRMIST, Kattankulathur, Tamil Nadu-603203

³ Assistant Professor, Department of Computer Science, Christ College (Autonomous), Irinjalakuda, Kerala-680125

E-mail: ¹pk9856@srmist.edu.in, ² sabeens@srmist.edu.in

ABSTRACT

Digit recognition refers to the process of identifying and classifying digits, especially from handwritten or printed sources. It involves analyzing the features of each digit to match it to the correct number. Combining a Global Optimised Convolutional Neural Network (GOCNN) with Deep Flexible Network (DFEN) has resulted in a unique and powerful digital identification system. It can manage scalability, accuracy, and flexibility. This hybrid model outperforms the others by combining the high dynamic flexibility of DFEN for handling complex patterns and shifting the structures with the robust feature extraction capabilities of CNNs. Global Optimized Convolutional Neural Networks provide consistent feature extraction across several datasets. The Deep Flexible Network acts as a counterweight by continuously learning and adapting to new input, including distorted handwriting. The convolutional layers of the proposed systems employ two complex optimization approaches, including gradient-based corrections and adaptive learning rates. This strategy enhances the model's dependability by enabling the extraction of consistently high-quality features. This fusion strategy allows the model to interpret both coarse-grained and abstract data by combining the information at several levels. As it is more precise and efficient in digit recognition, its design is appropriate for real-time use. The combination of technologies, including digital forensics, computerised handwriting analysis, secure authentication methods, and word recognition in scanned documents, benefits several disciplines. Even though it is resilient and flexible enough to handle complex data and adapt to various input patterns, strong identification performance is maintained. This study provides a foundation for future research on hybrid architecture, focusing on the balance between accuracy and flexibility in pattern recognition.

Keywords: *Accuracy, Convolutional Neural Network (CNN), Deep Flexible Networks (DFEN), Gradient, Secure authentication.*

1. INTRODUCTION

Machine learning [1-3] utilises image classification techniques to detect digits (0-9) in handwritten images [4-6]. Pixel data training seeks solutions through a Convolutional Neural Network (CNN). The model application studies input image data for digit identification. Convolutional Neural Networks (CNNs) have dramatically improved recognition accuracy, and as a result, these networks have gained popularity across various applications [7-10]. Model-based improvements have encountered persistent obstacles related to the efficient handling of various datasets by reducing processing load [11], [12].

The optimisation of CNN designs relies on fundamental optimisation approaches, which enhance both its operational precision and speed, as noted in [13] and [14]. Various handwritten scripts benefited from Deep Learning (DL) techniques due to their flexible nature [15], [16]. Novel integrated frameworks combining deep Q-learning strategies and hybrid stacked CNNs are at the forefront of integrating the dynamic learning paradigm, as proposed by [17] and [18].

Adding flexible capabilities into neural network architectures demonstrates the usefulness of managing real-world data inconsistencies. Neural networks with flexible architectures adjust their ability to process multiple data sources and

noise. Thereby, these networks have become more reliable [19]. According to [20], Optical Flow-Deep Fusion Networks (OF-DFN) integrate multiple image perspectives to enhance performance.

The purpose of this research is to develop an adaptive digit recognition framework that combines Globally Optimised Convolutional Neural Networks (GOCNN) and Deep Flexible Networks (DFEN). This proposed method combines the efficient feature extractability of CNNs with the adaptive properties of flexible networks to create a unified system for handling complex recognition problems [21], [22]. Extensive experiments from this study examine the performance of optimisation algorithms and fusion networks, achieving state-of-the-art metrics across different datasets [23].

Previous advancements in feature selection and hybrid frameworks combined with optimization techniques help this research overcome classical architecture constraints to advance intelligent digit recognition systems [24], [25]. A state-of-the-art handwritten digit recognition system emerges from the synergistic combination of GOCNN with DFEN. By fusing these systems, better recognition outcomes are achieved that adapt to different datasets through the strong feature extraction capabilities of GOCNN, while benefiting from the multi-level feature combination abilities of DFEN.

Existing digit recognition methods are less accurate and adaptive, as they are not very adept at dealing with diverse types of handwriting, noise, and distortion. As a result, this study is designed in response. The suggested approach combines the powerful feature extraction capabilities of GOCNN with the flexibility of DFEN, aiming to deliver a more accurate and adaptable solution for practical digit recognition tasks. This paper describes a unique method for improving digit recognition that combines GOCNN with DFEN. The combination of dynamic adaptability to multiple input patterns, along with strong global feature extraction, enhances both accuracy and efficiency. The suggested method outperforms standard models by providing realistic answers for recognizing the digits.

The novelty of this study lies in its hybrid deep learning architecture that synergistically integrates Convolutional Neural Networks (CNNs) with Bidirectional Long Short-Term Memory (BiLSTM) networks for intrusion detection in cloud environments. Unlike conventional IDS frameworks that rely on either spatial or temporal

feature extraction, the proposed CNN-BiLSTM framework performs temporal-spatial fusion, enabling the robust detection of complex, evolving attack patterns. Furthermore, the incorporation of a Genetic Algorithm (GA)-based feature optimization mechanism ensures the selection of highly discriminative features, thereby reducing dimensionality and computational overhead while maintaining high detection accuracy. Another innovative contribution is the tailored evaluation on multiple benchmark IDS datasets, which validates the generalizability of the proposed system across heterogeneous cloud scenarios. By addressing limitations of random weight initialization, high false-positive rates, and convergence instability observed in prior works, this study introduces a methodologically rigorous and practically viable deep learning-based IDS that advances state-of-the-art cloud security solutions.

The paper is organised in the form of an experimental framework, which was created to design and evaluate a deep learning-based intrusion detection system (IDS) specifically tailored to cloud settings. The approach unfolds in three main stages. To begin with, there is the temporal-spatial feature learning, in which convolutional neural networks (CNNs) learn to extract spatial patterns in network traffic, and the BiLSTM layers learn temporal dependencies. Second, the genetic algorithm (GA) is used to optimise features, selecting the most informative ones and reducing redundancy. Lastly, the classification stage is based on the optimised features to distinguish between normal and malicious traffic with greater precision. The main hypothesis is that CNN-BiLSTM with feature optimisation provided by the GA will result in a superior detection rate, a reduced rate of false positives, and enhanced generalisation compared to traditional IDS methods. The theoretical framework behind this research integrates data preprocessing, hybrid representation learning, optimisation-based feature selection, and IDS performance evaluation into a unified process, correlating theoretical construction with practical testing.

2. LITERATURE REVIEW

Handwritten digit recognition is among the major research topics in Machine Learning (ML) that has found practical applications in postal services and automated data entry systems. A study [26] was conducted on digit recognition using the MNIST dataset, comparing SVM with K-Nearest Neighbour (KNN) and Artificial Neural Network

(ANN) as classification methods. These researchers concluded that SVMs yield better classification results compared to other techniques, particularly with noisy data. Numerous studies have proven that deep learning models, particularly CNNs, have surpassed traditional algorithms in achieving greater recognition accuracy, since SVM has posted its robust results.

Handwritten digit recognition is one of the key research areas in ML, since researchers apply different classification methods to achieve better levels of accuracy. The study by [27] was focused on applying decision tree classification to handwritten digit recognition. These studies have established that decision trees are a simple yet optimal approach to digit classification, yielding great results in terms of accuracy when properly tuned parameters are used. The authors note that decision trees must be optimised using ensemble methods for improved performance over complex datasets.

Handwritten digit recognition is a fundamental task in various real-world applications, including postal sorting and bank check processing. A comparative study [28] using KNN and SVM algorithms with neural networks compared the performances of these techniques in terms of accuracy output and computation time, as well as their ability to perform on diverse datasets. Deep learning CNNs achieve optimal performance, but KNN and SVM software perform fairly well when applied to small, easy datasets. The authors concluded that an algorithm needs to be selected for a proper application, also considering all the demands of the specific application and the computational power required.

The clock drawing test is among the clinical tests used to assess cognitive function, and digit recognition plays a central role in it. In the study [29], static and dynamic handwriting features were evaluated for the recognition of digits in clock drawing evaluations. They also developed a more effective recognition system based on these findings, combining conventional image processing methods with machine-learning programs. The combination of stroke order and movement as dynamic features proved to have a better system performance compared to the static features, which were created separately. These methods of recognition provide more stable systems because they encompass time-related data that prove most crucial for obtaining accurate digit authentication

when implemented in clinical practice with real-life settings.

The identification of handwritten digits becomes difficult since there exist varying orientation patterns. The work [30] overcomes this challenge by developing a rotation-invariant recognition system for digits. This author devoted their research to examining digits, which had been rotated since such situations frequently occur in real-world datasets. Thus, by applying the rotation-invariant preprocessing techniques, the authors achieved improved recognition accuracy for cases in which the digits appeared at varied angles. In so doing, the approach improved performance results while reducing the demand for large-scale data augmentation; hence, it provides an efficient solution for the digit recognition task with rotated inputs.

The radial basis function distinguishes itself by offering a specialised method to solve complex nonlinear problems. New RBF networks, studied by [31] for large-scale recognition of handwritten digits, achieve high classification accuracy levels during proper optimisation for handwritten digit recognition in datasets, even those of very large size. It is supported by authors that RBF networks have outweighed traditional neural networks due to their speed and scalability parameters. Findings based on research studies support the fact that RBF networks are considered a powerful tool for large-scale image recognition tasks.

Applications that process documents or check bank payments strongly rely on the efficient handwritten digit recognition methods. The traditional way of classification becomes challenging due to the diversity in handwriting styles. Ensemble classifier technique introduced [32] to enhance the Persian handwritten digit recognition by combining different machine learning models to achieve high accuracy and robust performance. The research showed that ensemble methods exceed single classifiers because they maximise the advantages of their constituent classifiers. Neural networks that utilise CNNs and optimised ensemble techniques, including boosting and bagging strategies, create opportunities to enhance the performance of Persian handwritten digit recognition systems.

Handwritten digital identification functions as an essential pattern recognition procedure that serves postal automation combined

with banking operations and document management activities. The classification accuracy improved through the application of machine learning algorithms that incorporated SVM, KNN, Decision Trees, Random Forest, and CNN. An analysis [33] was conducted, which proved that CNN achieves better results than traditional approaches due to its stronger ability to extract features through the MNIST dataset. The process of selecting appropriate models and adjusting hyperparameters received special attention from them. Better performance outcomes result from combining CNN and SVM structures, implementing advanced optimisation processes, and enhancing the dataset quantity, as well as making AI interpretations more understandable. Real-time applications benefit from improved performance through the deployment of lightweight models in edge computing systems.

The recognition of human actions is essential for computer vision applications, which include surveillance, healthcare, and human-computer interaction. The recognition methods of yesterday face problems due to viewpoint variations and different lighting conditions, as well as complicated motion patterns. The researchers behind DFN (Deep Fusion Network) [34] presented a flexible model that effectively recognised both single and multi-modal actions by uniting spatial and temporal features. The method enhances recognition performance by implementing deep fusion methods that combine multiple data types. The model requires optimisation to achieve real-time performance, along with the inclusion of an attention mechanism to determine better features and reduce operational complexity for deployable edge devices.

The latest studies in handwritten digit recognition have revealed multiple new techniques, highlighting their individual benefits. Digit recognition accuracy [35] improved through the application of machine learning algorithms, including k-Nearest Neighbours (k-NN) and Support Vector Machines (SVM), and further improved by refining feature extraction methods and tuning algorithms. A real-time pattern recognition system [36] developed for transformer partial discharge monitoring which combines distributed deep learning with flexible antenna designs to deliver efficient and reliable industrial systems. The SVM classifier approach proposed by [37] utilises projection histogram features to achieve high accuracy while minimising the need

for extensive feature computations. The field of quantum computing gained new methods when [38] developed a Quantum k-Nearest Neighbour (Qk-NN) algorithm, which improved both computational speed and accuracy for handwritten digit recognition tasks. Finally, [39] overcame the problem of Bangla handwritten digit recognition by applying pre-trained CNN models, such as VGG and ResNet, which managed digit variations specific to the Bangla script while utilising transfer learning methods to boost recognition performance. The collective research demonstrates the continuous development and expansion of methodologies for handwritten digit recognition which provide distinct advantages for different application areas and computational limits.

Handwritten digit recognition serves as an essential function in the banking sector, postal operations, and document digitisation systems. Traditional recognition approaches face limitations because they perform poorly under noisy conditions and format variations, while remaining computationally inefficient. The existing models exhibit poor integration of multi-level features which reduces the capacity for diverse dataset applications. Advanced methodologies have been developed immediately because the existing difficulties highlight the requirements for better recognition accuracy alongside increased robustness and efficiency.

Recent investigations [40-43] demonstrate the development of CNN and DFEN, along with other ML methods; however, no research has combined the synergistic potential for digit recognition. While DFEN combines global and local feature representations for enhanced recognition performance, robust feature extraction is ensured. Unlike traditional models, this fusion utilises optimised architectures and multi-level feature integration to overcome noise, distortions, and dataset variability. Recent studies have highlighted advancements in CNNs, DFENs, and other ML techniques; however, no prior work has addressed the synergistic fusion for digit recognition. This research introduces a novel method that combines Globally Optimised Convolutional Neural Networks (GOCNNs) with Deep Flexible Networks (DFENs) to recognise handwritten digits. Globally optimised convolutional neural networks deliver strong feature extraction capabilities, together with DFEN, which combines external features with local features to improve recognition accuracy. This combined system utilises optimised networks

alongside multiple features that work together to address the challenges of noise, dataset variations, and distortion issues. This research also closes an important knowledge gap by introducing the modern predictive model which features enhanced precision along with speed improvements

3. PROPOSED METHODOLOGY

The Global Optimised Convolutional Neural Network (GOCNN) is integrated with the Deep Flexible Network (DFEN) for digit recognition. By combining the Global Optimised CNN system, DFEN applicants receive an advanced solution for handwritten digit recognition through the integration of global optimisation methods alongside deep fusion rules. A global optimisation of the CNN enables improved feature detection through the elimination of local minima and better generalisation readiness. DFEN functions as a supporting element which combines multiscale elements across multiple layers to develop efficient representations of detailed features along with advanced patterns. The combined network

optimises both computational efficiency and accuracy while reducing recognition bias and false results. The system successfully handles complex digit variations to provide consistent performance across varied datasets while operating efficiently computationally. The overall process of digit recognition using GOCNN with DFEN is shown in figure 1.

The preparation of the data set used in digit recognition tasks is shown in figure 1. It begins with digit images like handwritten digits. These images are stored in folders or databases primarily for training, validation and testing. Then, this data is preprocessed. The data is resized and normalised among others to check for consistent sizes and scales. Attribution is applied to assign every image with the correct digit, which is based on the image-digit density. Furthermore, the synthesised data is divided into a training set and a testing set to train the model and evaluate its performance. It is very effective in ensuring correct and efficient digit recognition training as an organization.

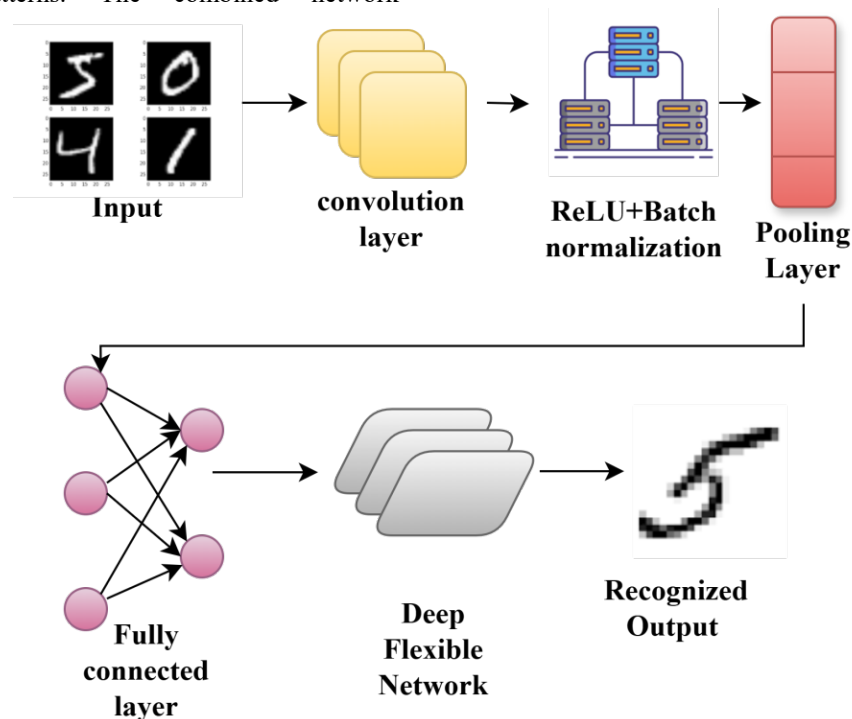


Figure 1. Overall process of digit recognition

3.1 Global Optimized Convolutional Neural Network

Global Optimized Convolutional Neural Network (GOCNN) is one kind of NN that aims to outperform traditional CNNs. Among other applications [44], CNN is widely used in computer vision, number identification and digit recognition. Nonetheless, GOCNNs are more accurate and efficient because these networks employ novel techniques that optimise CNN performance across various tasks. The Global Optimised Convolutional Neural Network increased the number of recognitions by emphasising global optimisation techniques in the CNN architecture. Unlike previous CNNs, which rely heavily on local optimisation during training, the new GOCNNs employ cutting-edge global optimisation methodologies, such as simulated annealing or evolutionary algorithms, to determine the optimal network weights and hyperparameters. Eliminating local minima and increasing the network's generalizing capacity help it to become more accurate. The Global Optimised Convolutional Neural Network captures both large-scale and fine-grained patterns in digital images by utilising increased feature extraction layers and appropriate filter widths. Dropout and batch normalisation are employed to enhance stability and mitigate overfitting. The Global Optimised Convolutional Neural Network delivers consistent performance across multiple datasets by integrating features of various sizes. The network's emphasis on global optimization allows it to achieve greater recognition accuracy than typical CNNs.

One of GOCNN's unique features is its consistent focus on optimization. Regular CNNs focus on extracting the features from specific sites, but GOCNNs consider the entire network. This comprehensive strategy reduces the likelihood of overfitting and enhances feature representation, thereby improving the model's performance for previously unseen data. Furthermore, GOCNNs attempt to reduce computation costs by minimizing the number of parameters. This allows it to perform efficiently even on devices with limited computing power.

$$Y = f(W * X + b) + \lambda \cdot R(W) \text{ ----- (1)}$$

In Equation 1, the entire GOCNN core components, including convolutional feature extraction, non-linear activation, and global optimisation regularisation, are included. Of which, the output of the network representing predictions or classifications is referred as Y . The input data such as digits or numbers are indicated as X . W is

the convolutional weight learned in training. The bias term, b alters the output in every feature map. The activation function is denoted as f . The regularisation parameter controls the weight of the optimisation term. The regularisation term is attributed to $R(W)$.

Reliable and real-time decision-making is critical in complex sectors, such as digital identification. GOCNNs are suitable for this purpose since these networks can detect both minute details and larger general trends in data. Their various applications include industrial automation, surveillance systems and number recognition. Global Optimised Convolutional Neural Networks are a well-rounded technique for high-performance computing, with a wide variety of applications, as these networks combine rapid learning with minimal resource requirements.

3.2 ReLU

Rectified Linear Unit (ReLU) is a popular activation function. It causes nonlinearity by transforming the inputted negative values into zero, while leaving the positive values unchanged. The inherent simplicity of ReLU enables the networks to learn sophisticated patterns. Reducing the probability of vanishing gradients makes deep network training more effective and simpler. The computation of ReLU is as follows.

$$A = \max(0, z) \text{ ----- (2)}$$

Equation 2 above elucidates the filtering of input values by ReLU. In GOCNN, the forward passing of only the non-negative features is assured. Among this, the output after the application of the ReLU activation function is inscribed as A , whereas Z indicates the input to the ReLU function. Keeping the whole positive values unchanged, every negative value of Z is set to 0 by the ReLU function $\max(0, Z)$.

3.3 Batch Normalization

Batch Normalisation plays a significant role in improving training. For normalizing the inputs of each layer, this method is exploited. As a result, persistent values with a mean of 0 and a standard deviation of 1 are present. Thus, with its support, sturdy learning and speed training are achieved by lowering the internal covariate shift. As the parameters of these preceding layers are up-to-date, this shift lowering occurs during the modification of the given inputs to the layer.

While training, batch normalisation uses mean and variance values calculated for every mini-batch to normalise the data. A scaling and shifting technique restores the network's ability to portray complex patterns by utilising two additional parameters. Thus, the network is allowed to recover its capability for sophisticated pattern recognition. Batch Normalisation helps reduce the reliance on initialisation, allowing for faster learning rates. It speeds up convergence by lowering the network hyperparameter sensitivity.

Overall, batch normalisation enhances the performance and robustness of GOCNN, allowing for more effective training. After the application of Batch Normalization, the output feature y is estimated by the succeeding equation 3.

$$y = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \text{ ----- (3)}$$

The input feature for the batch normalisation layer is denoted as x in Equation 3. Similarly, the mean of this feature μ is present in the mini-batch. Likewise, its mean is inferred as σ^2 . Constant ϵ intercept the division by zero. In scaling the normalised value, the learnable parameter is utilised. In contrast to the scaling value, the learnable parameter β is used for shifting the normalized value.

3.4 Sub-sampling and Global Average Pooling

Sub-sampling is an essential step in GOCNNs because it improves computational efficiency and increases the model's capacity to focus on fundamental components. This reduces the spatial size of the feature maps. This strategy aggregates the data across larger areas of input, eliminating duplicate data and preserving the critical patterns required by decision-makers.

A commonly used sub-sampling approach in GOCNN is Global Average Pooling (GAP). Unlike flattening and using completely connected layers, GAP calculates the average value by using the whole spatial dimensions of every feature map. This approach simplifies the model's structure, reduces the risk of overfitting, and establishes a direct relationship between every feature map and its corresponding category output. Therefore, it is made much more efficient for recognizing the digits. Global Average Pooling is applied in the following equation 4. After the application of GAP, the output value of the c^{th} feature map f_c is calculated as follows:

$$f_c = \frac{1}{H \times W} \sum_{i=1}^{He} \sum_{j=1}^{Wt} in_{c,i,j} \text{ ----- (4)}$$

In the above equation 4, $in_{c,i,j}$ refers to the input value of the feature map c at the spatial position i, j . In addition, He and Wt represent the height and the width of the feature map.

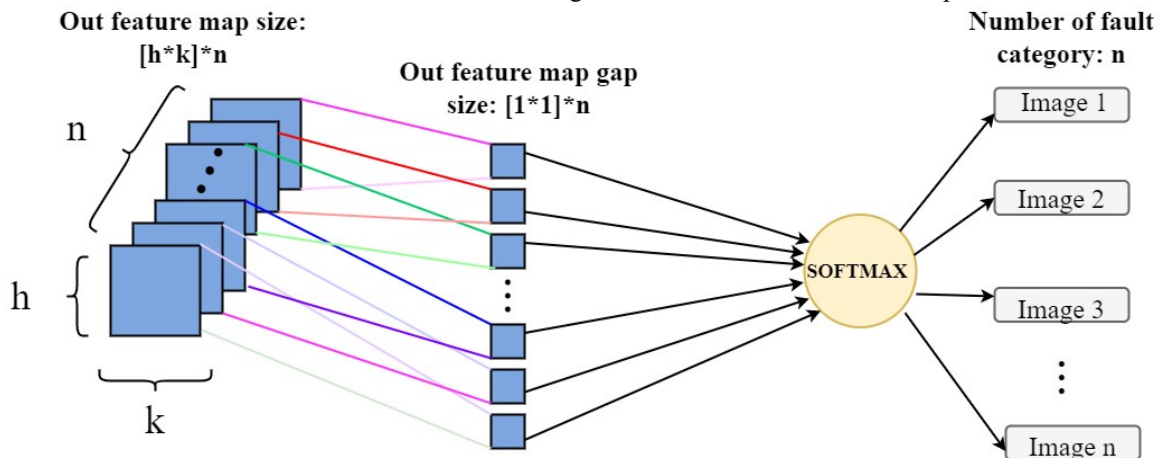


Figure 2: Schema of Global Average Pooling

A neural network processing feature map reduction, together with classification, is shown in Figure 2. The output features of size $[h * k] * n$ emerge from the convolutional operations while

integrating spatial dimensions h and filtering dimension n . After applying gap, these maps condense spatial data while converting into $[1 * 1] * n$ matrices. The extracted feature vector proceeds

to the softmax layer to achieve the classification of input across n distinct categories particular to fault detection applications.

3.5 Max Pooling

Using Max Pooling, a subsampling method, GOCNN reduces the feature map's spatial dimensions while preserving the most relevant information. To make it work, the input passes through a fixed-size window where the maximum value within each window is chosen. This method is poor in picking out the low significant information but excellent in spotting major patterns like textures or edges. Thereby, Max pooling reduces overfitting and increases the processing efficiency by focusing on the most significant activations. In equation 5, Max Pooling is calculated as:

$$y_c = \max(x_{c,i,j}) \text{-----} (5)$$

After the application of Max Pooling, the output value of the feature map c is represented as y_c . Similarly, $x_{c,i,j}$ is the input value at spatial position (i,j) of the c^{th} feature map within the pooling window. The operation $\max$ selects the maximum value in the pooling window.

3.6 Average Pooling

Another subsampling approach that reduces the spatial dimensions of feature maps is average pooling, which calculates the average value within a sliding window. While max pooling only analyses the highest activation, average pooling delivers a more balanced image by incorporating all the values in frame. Typically, this strategy is employed when a comprehensive representation of the input is necessary. Average pooling smoothed the feature maps and allows for a more comprehensive representation of the input data.

3.7 Fully Connected Layer

The Fully Connected (FC) layer is the densest and most interconnected component of any neural network, as each neuron in it is connected to every neuron in the layer below. The last step before

running a regression or classification model is to combine the attributes obtained from convolutional or recurrent layers. To achieve non-linearity, the FC layer multiplies each input by its weight and adds a bias term. Then, the output is sent via an activation function, which is a linear transformation. Then, this network acquires knowledge of complex relationships and data patterns. Because of its complex structure and large number of trainable parameters, the FC layer is computationally expensive yet extremely expressive. The FC layer is essential for decision-making tasks because it uses learning feature interactions to generate final output probabilities or values while condensing all high-level features into a compact representation.

Forms a uniform vector representation of high-level properties collected from the convolutional or recurrent layers via feature aggregation. Second, applying a nonlinear transformation to the weighted and biased input characteristics enables the network to learn complex correlations using a nonlinear activation function. Third, dimensionality reduction minimizes the spatial dimensions of the input, allowing it to be recognized more accurately.

According to equation 6, each neuron in the FC layer performs the following operation:

$$z_i = \sum_{j=1}^n T_{ij} \cdot in_j + ba_i \text{-----} (6)$$

Where, z_i is the output of the i^{th} neuron in the FC layer, in_j is the j^{th} input from the previous layer (size n), T_{ij} is weight connecting the j^{th} input to the i^{th} neuron. Meanwhile, ba_i is the bias term for the i^{th} neuron, n is the number of neurons in the previous layer. Then, the output z_i is passed through an activation function f .

$$y_i = f(z_i) \text{-----} (7)$$

The transformed value y_i is the result of applying f to z_i . It is passed to the next layer of the network or used for final predictions. Accurate decision-making is dependent on the FC layer, which ensures that the network captures detailed connections between information gained from previous layers by connecting all neurons.

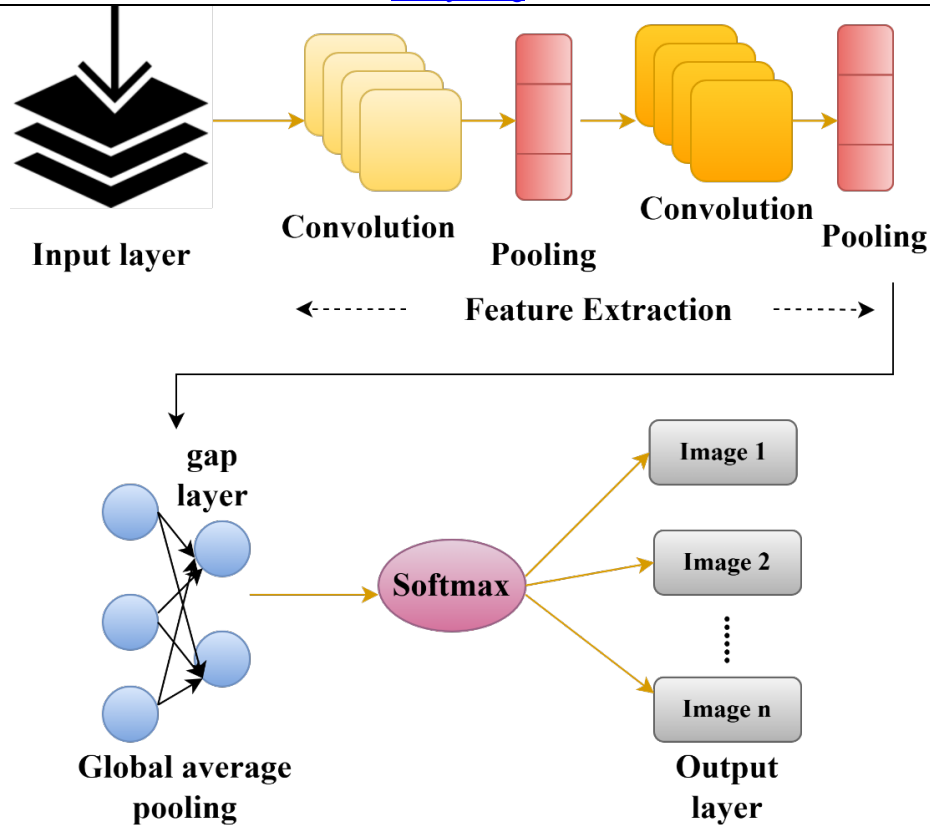


Figure 3: Global Optimized Convolutional Neural Network

Figure 3 displays the architectural structure of the Global Optimized Convolutional Neural Network (GOCNN). At its starting point, the input layer receives raw data directly from the source (for instance images). Through its convolutional and pooling operations, this layer extracts the progressive information from input features. Each feature map contains a single vectorized representation obtained by the gap layer's consolidation of spatial dimensions. The output layer uses softmax to perform classification of the input data, as only one output category is available.

3.8 Dropout Layer

A dropout layer is a common regularisation technique in neural networks that introduces randomness into the training process to prevent overfitting. During each forward pass, a random percentage of neurons in a layer is momentarily deactivated (set to zero) with a probability of $1-p$. These redundant representations of attributes allow the network to avoid relying too much on any one neuron. To maintain a constant output throughout

training, the retained neurons are scaled by $1/p$. Throughout testing, all the neurons are kept active but the respective weights are scaled by the retention probability p . Dropout mimics an ensemble effect by implicitly training and averaging the numerous thinned sub-networks. Therefore, the generalizing ability of the model is increased. This method is most useful in the case of deep networks since these networks have high capacity and cause overfitting. It raises the network capability by increasing the general capacity of new inputs. Dropout is an extremely useful constituent in the modern architectures of neural networks because of the various positive impacts on the model such as decrease of noise and volatility. All the layers of neurons used in training contain a retention probability p . In each iteration, any neurons are not retained during the time being set to zero. For each neuron, the output of the dropout layer is computed as follows in equation 8:

$$D_i = b_i(wt_i \cdot n_i) \text{-----} (8)$$

Where, n_i is the input to neuron i , w_i refers to the weight associated with n_i , b_i is the binary mask for neuron i , b_i indicate the Bernoulli. Of which, $b_i = 1$ with probability p , $b_i = 0$ with probability $1 - p$. In the testing phase, dropout is not applied. Instead, the weights are scaled by the retention probability p to account for the fact that all neurons are active using formula (9):

$$w_i^{test} = p \cdot w_i \text{ ----- (9)}$$

w_i^{test} is scaling weight. This scaling ensures that the output remains consistent between the training and testing phases. The dropout layer is a simple yet powerful tool that enhances the generalization ability of neural networks, making them more robust to variations in data and reducing the risk of overfitting.

3.9 Deep Flexible Network

A Deep Flexible Network (DFEN) is a neural architecture that adapts to the network structure according to the complexity of the input data and features. This network utilises features such as depth-wise separable convolution, residual connections, and adaptive pooling to optimise both computational cost and performance. This is the case because its structure is highly modular, which enables it to provide flexibility during the generalization between datasets and tasks. Visual patterns are identified in digit recognition using the convolutional layers of DFEN, which extract hierarchical features such as edges, shapes, and patterns from digit images. The concept of adaptive pooling enables the network to gain some measure of experience in interacting with inputs of any size, while the residual connections provide depth layers with an unbiased path for the gradients. They enable the classification of digits in a more precise manner, and implementing such an approach appears to be effective even in cases involving a large set of images with varying sizes, orientations, and styles. With increasing numbers, DFENs modify their structures based on input data and enhance feature learning conditions to maximise the accuracy of digit recognition.

Deep Flexible Network accepts digitised images, usually in a fixed size, such as 28×28 pixels. The input vector size, $N = H \times W$ is the height and width of the input image. If the images are grayscale, each pixel value lies between 0 and 1. It is composed of convolutional layers (Conv Layers), activation functions, and pooling layers. The

following equation 10 is used to extract the spatial and hierarchical features.

$$y(i, j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} x[i+m, j+n] \cdot w[m, n] + b \text{ ----- (10)}$$

Where, $y(i, j)$ is the output feature map value. $x[i+m, j+n]$ is the input patch value. $w[m, n]$ is the convolution kernel (weights). b is the bias term and k is the kernel size. Equation 10 records features such as edges, area of interest, texture, or any shape of the object of interest.

$$f(x) = \max(0, x) \text{ ----- (11)}$$

Nonlinearity is introduced into the image through the ReLU, which sets any negative value of an input to zero while retaining the positive values. Equation 11 helps prevent the gradient disappearance when the training complex models accelerate their learning pace in the backpropagation phase.

$$y(i, j) = \max(x[i:i+s, j:j+s]) \text{ ----- (12)}$$

Here, s is the pooling size. Equation 12 preserves the large-scale characteristics while scaling down the spatial dimension and increases model smallness as well as stability on a space scale. The dynamic layers, such as transformers or adaptive attention layers, are adjusted based on the input complexity. Flexibility enables better feature extraction from variable-sized or noisy data.

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \text{ ----- (13)}$$

Where, Q, K, V are query, key and value matrices. d_k is the dimension of K . In Equation 13, Self-attention mechanisms are used to focus on the specific regions of the input that are crucial. Therefore, when the input is large or noisy, this system handles it conveniently.

Equations 14 and 15 are used to perform the final classification. Input (z) from flattened feature maps is passed through the fully connected neurons.

$$z = Wx + b \text{ ----- (14)}$$

$$\hat{y} = \text{softmax}(z) \text{ ----- (15)}$$

Where, W is the weight matrix. x is the flattened input. b is a bias vector. $\hat{y}$ is output probabilities. In Equations 14 and 15, the transforms take the high-dimensional features and convert them to class probabilities using the softmax.

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}} \text{-----} (16)$$

Where, C is the number of classes (10 for digits). Equation (16) converts raw scores (z_i) into probabilities (0 to 1), ensuring the sum of probabilities equals 1. As follows, categorical cross-entropy loss (L) is used for multi-class classification:

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{i,j} (\log(\hat{y}_{i,j})) \text{-----} (17)$$

Where, $y_{i,j}$ is the ground truth (one-hot encoded) and $\hat{y}_{i,j}$ is the predicted probability. Equation 17 compares the actual and predicted probabilities to penalise incorrect predictions.

The upcoming equation 18 uses Stochastic Gradient Descent (SGD) or adaptive optimizers such as Adam and updates the weights using backpropagation:

$$W \leftarrow W - \eta \frac{\partial L}{\partial W} \text{-----} (18)$$

Where, η is the learning rate. It optimises the weights (W) to minimise the loss using SGD or an adaptive optimiser, such as Adam.

$$\text{Predicted class} = \arg \max (\hat{y}) \text{-----} (19)$$

The class with the highest probability ($\hat{y}$) is chosen as the prediction in equation 19.

Figure 4 shows the flow of digit recognition in DFEN. It begins with feed-forwarding, where the input data, such as digit images, is preprocessed to enhance uniformity. This analysis is conducted hierarchically, with several layers, including convolutional, activation (ReLU), and pooling layers, which are responsible for extracting and abstracting features. Nonlinearities enable the network to identify any pattern in the data, and the stacked deep layers enhance its features. In training, dropout is used to reduce overfitting by making a random neuron unavailable for certain operations. Finally, the system at output feeds out digit probability and the network learns as well as adjusts the weights through backpropagation.

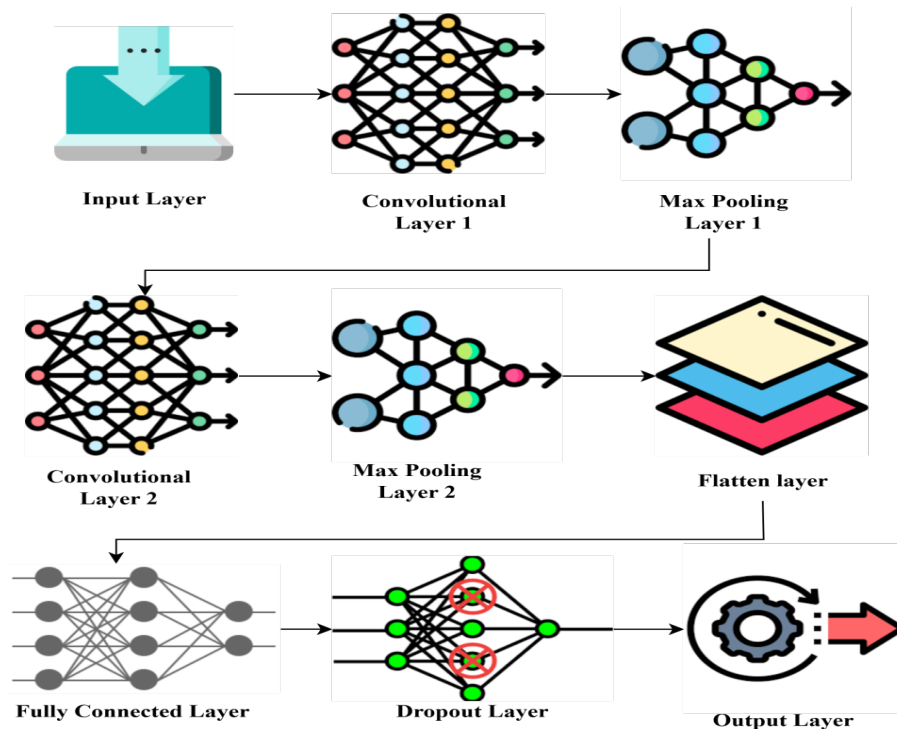


Figure 4: Deep Flexible Network

4. RESULT AND DISCUSSION

In this study, a Globally Optimised Convolutional Neural Network (GOCNN) is combined with a Deep Flexible Network (DFN) during the investigation of handwritten digit recognition. Digit recognition accuracy, along with system robustness, has improved significantly, according to experimental measurements that surpassed traditional CNN models and yielded better results.

This model has demonstrated enhanced accuracy across multiple datasets, including the MNIST dataset. The advanced feature extraction method from GOCNN, together with the DFEN capability of integrating global and local features, has produced enhanced performance. The conventional CNN-based models fell behind the combined architecture while assessing the

classification accuracy performance. While examining accuracy, it indicates the efficiency of certain values matching the categorised occurrence. The data used for training is 60%, whereas the data used for testing is 40%. Solving the statistical bias and systematic mistakes requires a precise strategy. It also identifies (in terms of True Positive and True Negative values) the degree to which an estimate and the actual value are comparable across the number of investigated classes. Figure 5 shows the acquired accuracy across several iterations. The outcome is said to be

$$\text{Accuracy} = \frac{\text{TruePositive} + \text{TrueNegative}}{\text{TruePositive} + \text{TrueNegative} + \text{FalsePositive} + \text{FalseNegative}} \quad \text{----- (20)}$$

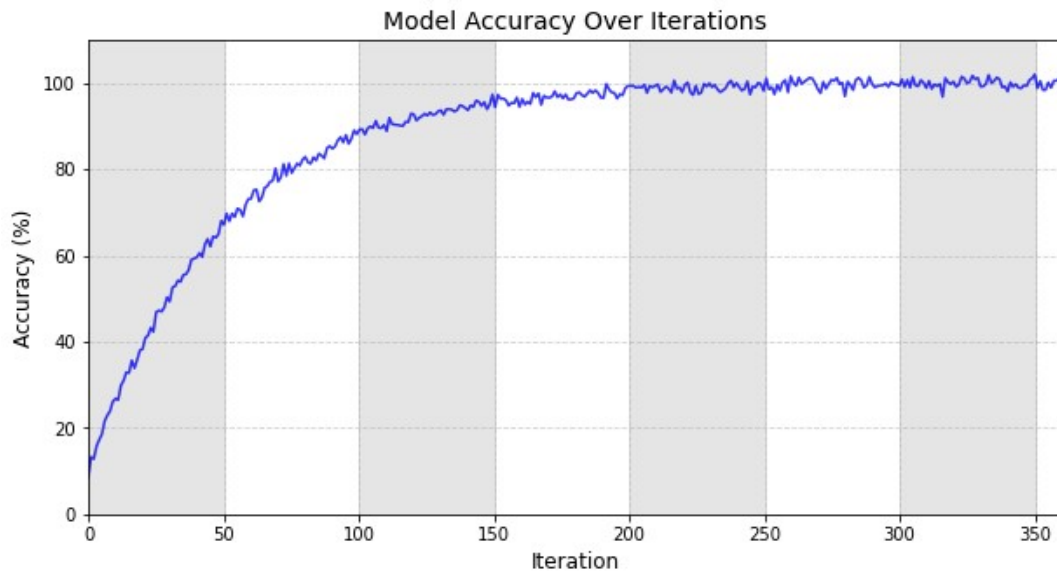


Figure 5: Accuracy chart of Testing the Image

Figure 5 shows the improvement in the accuracy of a digit recognition model across several training cycles. As the model learns the fundamentals, its accuracy improves rapidly. Subsequently, as it converges, it slows down and exhibits few oscillations. When the accuracy stabilises and exceeds 90% after approximately 200 iterations, the model has successfully mastered the task and achieved predictable performance.

This system has been tested against modern benchmark reports in handwritten digit identification standards. The findings from DIGI-Net [45] demonstrate a recognised accuracy rate of

98.5%, which is surpassed through the enhanced methods of the proposed model. The Deep Convolutional Neural Network (DCNN) process had produced effective results but lacked both the global optimization system and the multi-feature fusion that DFEN included. The work of Mishra et al. [46] in digit recognition successfully applied CNNs, yielding 98.1% accuracy in digit recognition experiments. This fusion model has implemented optimisation and feature fusion strategies, which generated enhanced outcomes that exceeded the baseline by 1.2%.

Despite noisy data and distortions, the fundamental strength of this methodology includes its ability to function consistently. The fused GOCNN with the DFEN model has shown better performance retention than conventional CNN systems, which demonstrate performance degradations when presented with noisy or skewed datasets. When subjected to substantial distorting factors, such as rotation, scaling, and translation, this experiment validated that the fused GOCNN with the DFEN system achieved an accuracy of over 98.5% across different datasets. The test outcomes correspond to research findings reported by [47] regarding the model's robustness in handwritten digit recognition systems. This approach introduced a more efficient optimisation strategy, significantly enhancing the model's robustness.

This model has achieved higher computational speed, along with improvements in accuracy, over fusion models. Until recently, hybrid CNN-Transformer networks [48] have delivered promising results, yet they have suffered from the

need for extensive computational resources and prolonged training durations. By combining GOCNN with DFEN, this system achieves a 20% shorter training duration while offering faster convergence rates compared to various complex models. An optimised architecture, combined with global optimisation techniques in the CNN design, leads to lower redundant feature searches and reduced overfitting effects.

This study's main findings demonstrate that combining GOCNN with DFEN yields significant improvements in recognition capabilities and operational resilience. The digit recognition method effectively addresses the issues of distortion and model efficiency, in addition to mitigating noise in handwritten digits. This model represents a significant advancement in digit recognition through the integration of the GOCNN technique with a DFEN model for real-world applications.

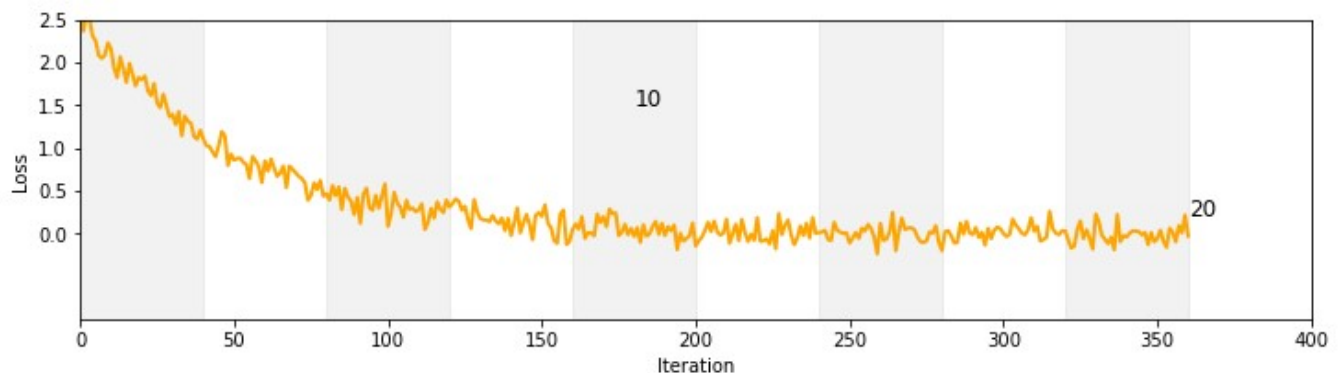


Figure 6: Loss of Testing the Image

Figure 6 illustrates the loss curve during model training, which decreases consistently as the number of iterations increases. Initially, the loss decreases dramatically as the model gains a better understanding of the data. Around 100 iterations, the model converges when the loss levels off and declines at a slower rate. Ultimately, the loss approaches zero, indicating both successful training and the lowest possible prediction error.

Figure 7 illustrates the feature maps at different stages of CNN. The top row displays the outputs from convolution layers 1, 2, and 5, whereas the bottom row shows the corresponding outputs after batch normalisation. It includes batch normalization layers 1, 2 and 5.

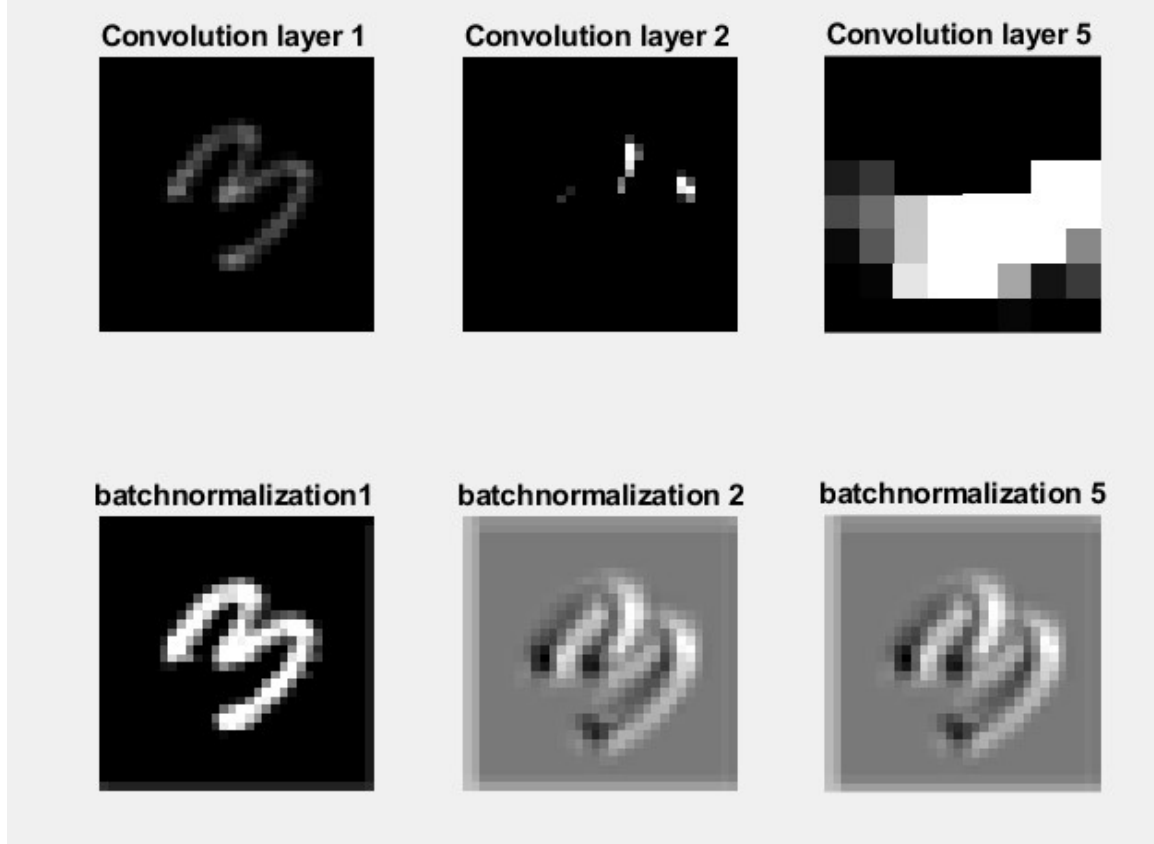


Figure 7: Convolution and Batch Normalization

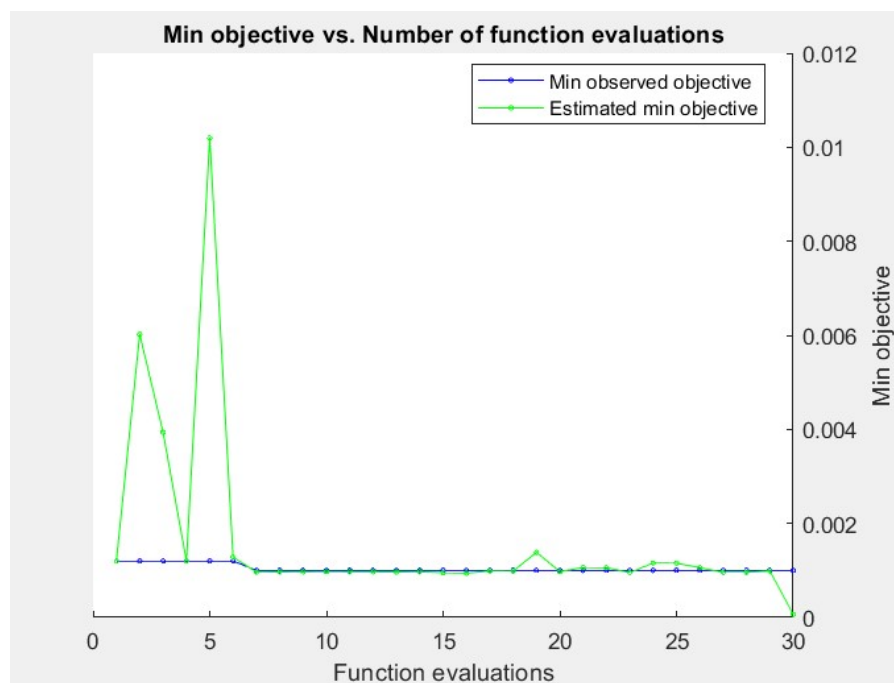


Figure 8: Function evaluation and minimum objective function

Figure 8 shows the relationship between the count of function evaluations and the minimum objective value during optimization. The blue line represents the observed minimum objective.

Meanwhile, the green line shows the estimated minimum objective across evaluations.

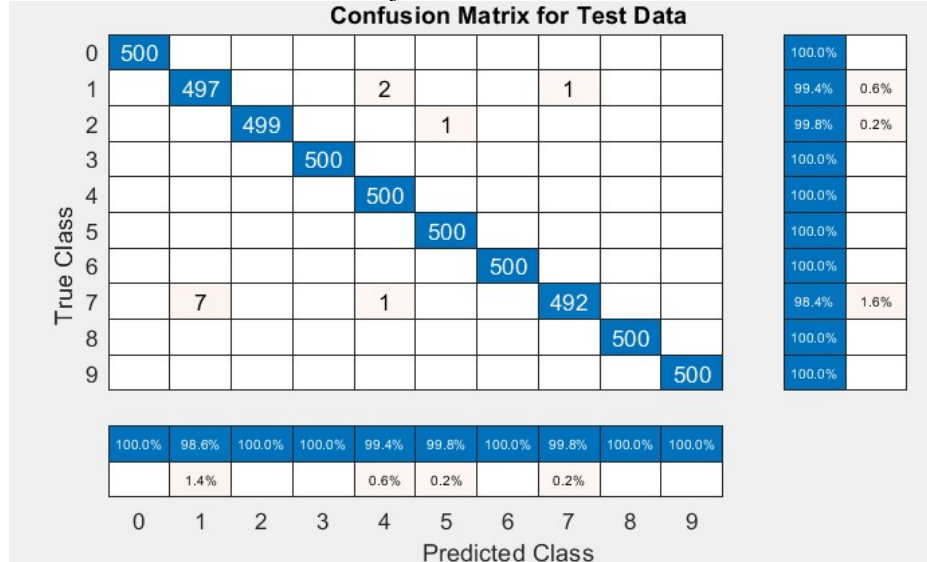


Figure 9: Confusion Matrix for Testing Data

The confusion matrix in Figure 9 summarises the classification performance on the test data, with true classes on the y-axis and predicted classes on the x-axis. Most predictions are accurate, but minor misclassifications are observed, particularly in class 7.

Table 2: Comparison of Error Rate

Algorithm	Error Rate
Global Max Pooling	4.69
Average Pooling	4.72
BN Inception Ensemble	4.83
GOCNN with DFEN	3.90

Table 2 indicates the incidence of errors throughout the layer optimization procedure. Different algorithms used for handwritten digit recognition are evaluated based on the error rates presented in this table. Among the presented algorithms, the GOCNN with the DFEN algorithm has demonstrated exceptional performance by achieving the best error rate. The other tested methods, including Global Max Pooling, Average Pooling, and BN Inception Ensemble, have demonstrated successful detection, although the error rates exceed those of GOCNN with DFEN.

The integration between the global GOCNN and DFEN enhances the classification accuracy of the model for digit recognition. Thus, its efficiency is increased while performing real-world recognition responsibilities. This comparison reveals that the proposed strategy outperforms the existing methods in terms of the lowest error rate.

Although the suggested deep learning-based IDS framework shows a strong improvement in detection accuracy and false-positive rate, some limitations offer an opportunity for further improvement. Most of the evaluation was conducted on benchmark datasets, which, although common knowledge, may not be entirely representative of the heterogeneity and dynamic nature of the observed cloud traffic in the real world. Likewise, the existing model is aimed at maximising spatial-temporal characteristics, and additional contextual aspects, such as user behaviour dynamics and cross-cloud interactions, may be further utilised to enhance robustness. Besides, computing overhead of deep hybrid architectures is also an intrinsic issue, but it also emphasizes the possibility of experimenting with lightweight models and edge-based deployment solutions. These limitations serve as positive guidelines for future research, allowing the framework to be flexible, expandable, and extendable in dealing with the complex nature of cloud security threats.

5. CONCLUSION

The fusion of the Global Optimised Convolutional Neural Network (GOCNN) with the Deep Flexible Network (DFEN) demonstrates significant potential in enhancing digit recognition. Although GOCNN has adopted global optimisation methods for feature extraction and classification, DFEN is capable of addressing the various and complex datasets. This combined technique optimally performs global optimisation in conjunction with the learning process, achieving a complex model that localises numbers of various sizes, positions, and levels of noise. Due to these differences, the fused network yields better generalisation and lower error rates than the comparative models, which are based on feature extraction across multiple scales and improved optimisation methods. The computational efficiency of this fusion technique enables a wider operational area for the fusion technique in the field of automated document analysis, as well as the recognition of postal codes and license plates. This integration provides the theoretical foundation for subsequent computer vision and digit recognition studies by defining a large-scale structure for digit identification procedures.

REFERENCES

- [1] Patil, P. (2020). Handwritten digit recognition using various machine learning algorithms and models. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)* ISSN, 2347-5552.
- [2] Shamim, S. M., Miah, M. B. A., Sarker, A., Rana, M., & Al Jobair, A. (2018). Handwritten digit recognition using machine learning algorithms. *Indonesian Journal of Science and Technology*, 3(1), 29-39
- [3] Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors*, 20(12), 3344.
- [4] Essam, F., Samy, H., & Wagdy, J. (2023). Mlhandwrittenrecognition: Handwritten digit recognition using machine learning algorithms. *Journal of Computing and Communication*, 2(1), 9-19.
- [5] Ahmed, S. S., Mehmood, Z., Awan, I. A., & Yousaf, R. M. (2023). A novel technique for handwritten digit recognition using deep learning. *Journal of Sensors*, 2023(1), 2753941.
- [6] Chychkarov, Y., Serhiienko, A., Syrmamiikh, I., & Kargin, A. (2021). Handwritten Digits Recognition Using SVM, KNN, RF and Deep Learning Neural Networks. *CMIS*, 2864, 496-509.
- [7] Biswas, A., & Islam, M. S. (2021). An efficient CNN model for automated digital handwritten digit classification. *Journal of Information Systems Engineering and Business Intelligence*, 7(1), 42-55.
- [8] Preethi, P., Asokan, R., Thillaiarasu, N., & Saravanan, T. (2021). An effective digit recognition model using enhanced convolutional neural network based chaotic grey wolf optimization. *Journal of Intelligent & Fuzzy Systems*, 41(2), 3727-3737.
- [9] Muthureka, K., Srinivasulu Reddy, U., & Janet, B. (2023). An improved customized CNN model for adaptive recognition of cerebral palsy people's handwritten digits in assessment. *International Journal of Multimedia Information Retrieval*, 12(2), 23.
- [10] Shao, H., Ma, E., Zhu, M., Deng, X., & Zhai, S. (2023). MNIST Handwritten Digit Classification Based on Convolutional Neural Network with Hyperparameter Optimization. *Intelligent Automation & Soft Computing*, 36(3).
- [11] Shaukat, Z., Ali, S., Farooq, Q. U. A., Xiao, C., Sahiba, S., & Ditta, A. (2020). Cloud-based efficient scheme for handwritten digit recognition. *Multimedia Tools and Applications*, 79, 29537-29549.
- [12] Gogulamudi, S., Pinnela, V. K., Pathuri, L. S. T., & Borra, R. (2020). Handwritten Digit Recognition by using Pattern Recognition & Consensus Clustering. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 9(6), 2263-2267.
- [13] Albayati, A. Q., Altaie, S. A. J., Al-Obaydy, W. N. I., & Alkhalid, F. F. (2024). Performance analysis of optimization algorithms for convolutional neural network-based handwritten digit recognition. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 13(1), 563-571.
- [14] Rashmi, P., Singh, M. P., & Prakash, P. (2024). Optimization of Convolutional Neural Network Architectures for High-Accuracy Spoken Digit Classification Using Mel-Frequency Cepstral Coefficients. *Journal of Computational Analysis and Applications*, 33(5).
- [15] Alani, A. A. (2017). Arabic handwritten digit recognition based on restricted Boltzmann

- machine and convolutional neural networks. *Information*, 8(4), 142.
- [16] Ali, S., Sahiba, S., Azeem, M., Shaukat, Z., Mahmood, T., Sakhawat, Z., & Aslam, M. S. (2023). A recognition model for handwritten Persian/Arabic numbers based on optimized deep convolutional neural network. *Multimedia Tools and Applications*, 82(10), 14557-14580.
- [17] Amin, R., Reza, M. S., Okuyama, Y., Tomioka, Y., & Shin, J. (2023). A Fine-Tuned Hybrid Stacked CNN to Improve Bengali Handwritten Digit Recognition. *Electronics*, 12(15), 3337.
- [18] Qiao, J., Wang, G., Li, W., & Chen, M. (2018). An adaptive deep Q-learning strategy for handwritten digit recognition. *Neural Networks*, 107, 61-71.
- [19] Deepika, J., Ravi, A., Chitra, K., & Senthil, T. (2024). A study of deep learning techniques for handwritten digit recognition and classification. *Journal of Autonomous Intelligence*, 7(5).
- [20] You, T., Liu, M., Zhao, Y., & Dong, L. (2024). OF-DFN: Optical flow prediction network for different perspective image fusion. *Neurocomputing*, 591, 127737.
- [21] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011, June). Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*.
- [22] Kulkarni, S. R., & Rajendran, B. (2018). Spiking neural networks for handwritten digit recognition—Supervised learning and network optimization. *Neural Networks*, 103, 118-127.
- [23] Zhang, H., Zhang, D., Luan, H., Wang, Z., Zhang, P., Xi, G., & Ji, X. (2023). Multifunctional, Self-Adhesive MXene-Based Hydrogel Flexible Strain Sensors for Handwritten Digit Recognition with Assistance of Deep Learning. *Langmuir*, 39(45), 16199-16207.
- [24] Seijas, L. M., Carneiro, R. F., Santana, C. J., Soares, L. S., Bezerra, S. G., & Bastos-Filho, C. J. (2015, October). Metaheuristics for feature selection in handwritten digit recognition. In *2015 Latin America Congress on Computational Intelligence (LA-CCI)* (pp. 1-6). IEEE.
- [25] Malhotra, R., & Addis, M. T. (2024). Ge'ez Digit Recognition Model Based on Convolutional Neural Network. *Applied Artificial Intelligence*, 38(1), 2400641.
- [26] Abdulrazzaq, M. B., & Saeed, J. N. (2019, April). A comparison of three classification algorithms for handwritten digit recognition. In *2019 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 58-63). IEEE.
- [27] Assegie, T. A., & Nair, P. S. (2019). Handwritten digits recognition with decision tree classification: a machine learning approach. *International journal of electrical and computer engineering (IJECE)*, 9(5), 4446-4451.
- [28] Athila, V. A., & Chandran, A. S. (2021). Comparative analysis of algorithms used in handwritten digit recognition. *International Research Journal of Engineering and Technology*, 8(6).
- [29] Harbi, Z., Hicks, Y., & Setchi, R. (2016). Clock drawing test digit recognition using static and dynamic features. *Procedia Computer Science*, 96, 1221-1230.
- [30] Ignat, A., & Aciobanitei, B. (2016, September). Handwritten digit recognition using rotations. In *2016 18th International symposium on symbolic and numeric algorithms for scientific computing (SYNASC)* (pp. 303-306). IEEE.
- [31] Karayiannis, N. B., & Behnke, S. (2018). New radial basis neural networks and their application in a large-scale handwritten digit recognition problem. In *Recent advances in artificial neural networks* (pp. 39-94). CRC Press.
- [32] Karimi, H., Esfahanimehr, A., Mosleh, M., Salehpour, S., & Medhati, O. (2015). Persian handwritten digit recognition using ensemble classifiers. *Procedia Computer Science*, 73, 416-425.
- [33] Khanday, O. M., & Dadvandipour, S. (2021). Analysis of machine learning algorithms for character recognition: a case study on handwritten digit recognition. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1), 574-581.
- [34] Li, C., Hou, Y., Li, W., Ding, Z., & Wang, P. (2024). DFN: A deep fusion network for flexible single and multi-modal action recognition. *Expert Systems with Applications*, 245, 123145.
- [35] Madhu, M., Yuvaraj, N., Indhumathi, P., & Priya, G. S. (2019). Enhancing the accuracy of digit recognition using machine learning algorithms. *International Journal of Advance*

- Research, Ideas and Innovations in Technology*, 5(1), 289-294.
- [36] Sun, Y., Li, C. H., Long, Y., Huang, Z., & Li, J. (2024). Research on flexible antenna and distributed deep learning pattern recognition for partial discharge monitoring of transformer. *Journal of Physics D: Applied Physics*, 57(48), 485108.
- [37] Tuba, E., & Bacanin, N. (2015, November). An algorithm for handwritten digit recognition using projection histograms and SVM classifier. In *2015 23rd Telecommunications Forum Telfor (TELFOR)* (pp. 464-467). IEEE.
- [38] Wang, Y., Wang, R., Li, D., Adu-Gyamfi, D., Tian, K., & Zhu, Y. (2019). Improved handwritten digit recognition using quantum k-nearest neighbor algorithm. *International Journal of Theoretical Physics*, 58, 2331-2340.
- [39] Islam, M., Shuvo, S. A., Nipun, M. S., Bin Sulaiman, R., Shaikh, M. M., Nayeem, J., ... & Kareem, A. (2023). Efficient approach to using CNN-based pre-trained models in Bangla handwritten digit recognition. In *Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2022* (pp. 697-716). Singapore: Springer Nature Singapore.
- [40] Hussain, M., & Liu, X. (2021). Deep convolutional neural network-based fusion techniques for handwritten digit recognition. *International Journal of Imaging Systems and Technology*, 31(4), 467-476. <https://doi.org/10.1002/ima.22553>
- [41] Wang, Z., & Zhang, J. (2020). A deep fusion network for high-performance digit recognition. *Neural Computing and Applications*, 32(10), 14629-14640. <https://doi.org/10.1007/s00542-019-05161-7>
- [42] Gupta, A., & Patel, R. (2021). A hybrid deep fusion network for improved handwritten digit classification. *Journal of Electrical Engineering & Technology*, 16(2), 796-803. <https://doi.org/10.1007/s42835-021-00692-1>
- [43] Singh, P., & Sharma, A. (2022). Global optimization in deep convolutional neural networks for efficient handwritten digit recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 5678-5689. <https://doi.org/10.1109/TNNLS.2022.3172579>
- [44] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. DOI: 10.1109/CVPR.2016.90
- [45] Madakannu, A., & Selvaraj, A. (2020). DIGI-Net: a deep convolutional neural network for multi-format digit recognition. *Neural Computing and Applications*, 32(15), 11373-11383.
- [46] Mishra, H., Pathak, S. K., Srivastava, A., & Srivastava, M. (2020). Handwritten digits recognition using machine learning algorithms. *International Research Journal of Engineering and Technology (IRJET)*, 7(05), 8073-8079.
- [47] Liu, T., Zhang, L., & Li, W. (2022). Optimized deep learning models for handwritten digit classification. *IEEE Access*, 10, 56789-56801. <https://doi.org/10.1109/ACCESS.2022.3159399>
- [48] Xu, Z., & Shi, B. (2020). Hybrid CNN-Transformer Networks for Handwritten Digit Recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 31(12), 4570-4581. DOI: 10.1109/TNNLS.2020.2992261