e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

OPTIMIZING HANDWRITTEN DIGIT RECOGNITION WITH CONVOLUTIONAL NEURAL NETWORKS: PERFORMANCE COMPARISON WITH NEURAL NETWORK CLASSIFIERS

Ajit Kumar[*1], Kunal Kishor[*2], Abhishek Tiwari[*3], Er. Harjasdeep Singh[*4]
[*1,2,3,4]Department Of Computer Science Engineering Mimit, Malout, India.
DOI: https://www.doi.org/10.56726/IRJMETS63225

ABSTRACT

My project focuses on solving the problem of handwritten digit recognition, a key challenge in pattern classification. By using the MNIST dataset, which contains images of digits (0-9), the project compares the performance of different classification models, particularly Convolutional Neural Networks (CNN) and traditional Neural Networks. [1]The aim is to demonstrate that CNN provides superior accuracy and computational efficiency in recognizing these digits. Tools like NumPy, Pandas, TensorFlow, and Keras are used for implementing the model. Results show that CNN significantly outperforms traditional Neural Networks in both speed and accuracy, making it an effective approach for handwritten digit recognition. The recognition of handwritten digits has long been an open challenge in the field of pattern recognition. Numerous studies have demonstrated that Neural Networks exhibit outstanding performance in data classification tasks. This paper uses the MNIST handwritten digit database as a dataset to examine the performance of various algorithms—namely, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Backpropagation (BP) Neural Network, and Convolutional Neural Network (CNN)—in handwritten digit recognition. In the training process, we implement KNN in Python, use the scikit-learn library for SVM, and utilize TensorFlow for BP and CNN, fine-tuning parameters to achieve optimal performance for each algorithm. Finally, by comparing the recognition rate and processing time of the four from the diversity in writing styles across individuals. Each person'sAlgorithms, we analyze their respective strengths and weaknesses in handwritten digit recognition. This revised version emphasizes key information and provides a clearer outline of your research approach.

Keywords: [2] MNIST Database, Handwritten Digit Recognition, Artificial Neural Network (ANN), Back Propagation Neural Network (BP), Convolutional Neural Network (CNN).

I. INTRODUCTION

The task of recognizing handwritten numerals has garnered significant attention over recent years, resulting in a substantial body of research focused on improving recognition accuracy and computational efficiency.[1] Handwritten digit recognition, a key area within pattern recognition, has numerous real-world applications, particularly in fields like postal code processing, financial statement analysis, and automated bank check processing. Despite advances in technology and algorithmic design,[5] the accurate recognition of handwritten numerals remains a challenging task, requiring sophisticated techniques to handle the inherent variability of handwritten input.

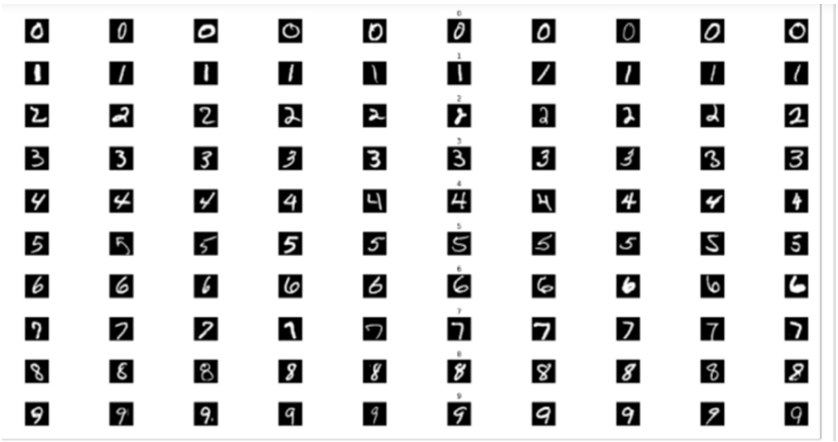

Fig.1. Handwritten digit

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

Primary challenge in handwritten numeral recognition stems handwriting varies in terms of size, orientation, stroke thickness, rotation, and even degrees of distortion, all of which impact the final appearance of each digit. These variations can cause the same numeral to look significantly different from one writer to another,

Complicating the task of developing a system capable of recognizing these diverse representations accurately. Additional challenges include handling image noise, overlapping digits, and incomplete strokes, all of which must be considered in designing an effective recognition model.

To address these challenges,[7] researchers have employed a variety of machine learning approaches over the years, testing different algorithms and techniques to achieve better performance in handwritten digit recognition. Early research, such as the work of Khotanzad et al. (1998), explored the potential of Machine Learning and Neural Networks for recognizing handwritten digits. By applying machine learning principles, they demonstrated that Neural Networks could be trained to effectively identify and classify digit images, thereby setting a foundation for more advanced models and methodologies.

Today, [5]Convolutional Neural Networks (CNNs) have become a leading approach due to their capacity to automatically extract features from raw images, making them particularly well-suited for image-based tasks like digit recognition. In this paper, we aim to explore and evaluate multiple classification algorithms, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Backpropagation (BP) Neural Networks, and CNNs. By applying these algorithms to the MNIST database of handwritten digits, we will compare their performance in terms of recognition accuracy and processing time, ultimately assessing the strengths and limitations of each approach.

This expanded introduction provides more context and details on the problem, the challenges, and the evolution of research in handwritten digit recognition.One of the primary objectives in the fields of artificial intelligence (AI) and machine learning is to endow machines with the ability to recognize elements in the environment in ways similar to human perception【1】. Humans naturally acquire the ability to distinguish between different types of objects by learning and summarizing their distinguishing characteristics. Similarly, Computers can be trained to recognize various types of objects by analyzing the spatial and temporal distribution of their characteristics, a process known as pattern recognition. As a subset of AI, pattern recognition involves the processing, extraction, calculation, and classification of samples by computers and finds extensive applications in areas such as image, character, text, and speech recognition【2】.

A classic pattern recognition system consists of five key components:

Sample acquisition, sample pre-processing, feature extraction, classifier design, and classification decision-making. Each of these steps plays a critical role in ensuring the accuracy and effectiveness of the recognition process. Character recognition, an essential application within pattern recognition, encompasses various tasks, including offline handwritten digit recognition, which presents its own unique set of challenges. Particularly in real-time application contexts, handwritten digit recognition serves as a specialized form of human-computer interaction, adding substantial value to diverse fields.

[6]Handwritten digit recognition has many practical applications, such as identifying postal codes on envelopes, processing large-scale financial statements, and managing inputs in banking forms. Despite the fact that digit recognition only requires identifying ten categories (0 through 9), significant variation exists in handwritten numerals due to individual writing styles and regional differences.

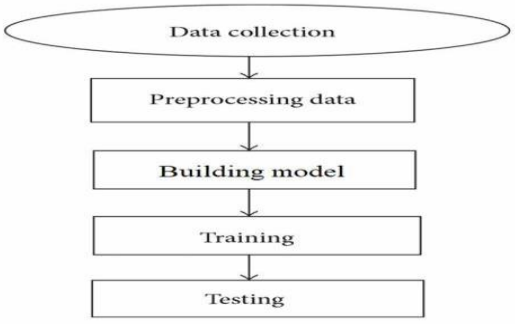

Fig. 2. Structure diagram of pattern recognition systems

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

II. LITERATURE REVIEW

1. Introduction to Handwritten Digit Recognition and Its Challenges-

Handwritten digit recognition has been a longstanding benchmark problem in computer vision, often using the well-known MNIST dataset. The problem involves correctly classifying digit images (0-9), which can vary widely in size, orientation, and style due to human handwriting. [9]Early models relied on traditional machine learning techniques, such as Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN). While these methods achieved moderate success, their performance was limited by their inability to effectively capture spatial hierarchies and complex features inherent in images.

2. Development of Neural Networks in Image Classification-

The advent of neural networks, especially multilayer perceptions (MLPs), improved classification tasks by introducing layered structures that allowed for more complex representations of input data. [10]Studies, such as those by LeCun et al., showed that neural networks could outperform traditional methods on the MNIST dataset by automatically learning relevant features through Backpropagation. However, standard neural networks are computationally intensive and often require large amounts of training data and regularization techniques to prevent overfitting on high-dimensional data like images.

3. Rise of Convolutional Neural Networks (CNNs)-

Convolutional Neural Networks (CNNs), introduced by LeCun and others, represented a transformative approach for image recognition. By using convolutional layers, CNNs can identify spatially invariant features in images, making them highly effective for digit recognition tasks. []11CNNs combine convolutional, pooling, and fully connected layers to efficiently learn both local and global features of an image. Studies such as Krizhevsky et al. (AlexNet) demonstrated CNNs' effectiveness in complex image recognition tasks, marking a paradigm shift in computer vision and making them the state-of-the-art for MNIST and similar datasets.

4. Optimization Techniques in CNN for Improved Accuracy-

Several optimization techniques have been developed to enhance CNN performance, such as dropout, batch normalization, and advanced weight initialization methods (e.g., Xavier or He initialization).[12]Dropout, introduced by Srivastava et al., is particularly important for reducing overfitting in CNNs by randomly dropping neurons during training. Batch normalization, rates and faster convergence. The use of these techniques has been shown to significantly boost CNN accuracy in handwritten digit recognition.

5. Comparative Studies with Traditional Neural Networks-

Numerous studies have compared CNNs with traditional neural networks (such as fully connected MLPs) in handwritten digit recognition. For instance, [14]Ziegler and Fergus (2013) demonstrated that CNNs outperform traditional neural networks not only in accuracy but also in computational efficiency, due to the reduced parameter requirements of convolutional layers. Additionally, MLPs require more complex feature engineering and preprocessing steps, which are reduced or eliminated with CNNs. Despite the advancements in CNNs, MLPs still serve as useful baseline models for performance comparison in recognition tasks.

6. Evaluation Metrics and Benchmarking for CNN Optimization-

[15]Evaluation metrics, such as accuracy, precision, recall, and F1 score, are commonly used to assess performance. Studies often benchmark CNN architectures against other classifiers using these metrics on datasets like MNIST, CIFAR-10, and ImageNet. Techniques like k-fold cross-validation further strengthen the validity of performance comparisons. Comparisons of CNNs with traditional neural networks have consistently shown that CNNs outperform traditional architectures in terms of both accuracy and robustness, especially when dealing with complex, unstructured image data.

7. Recent Advancements and Future Directions in CNN for Digit Recognition-

[19]Recent advancements in CNN architecture, such as ResNets, VGG, and Dense Net, have introduced deep and wide networks that improve model accuracy through skip connections and dense connections. These innovations have shown even greater improvements in handwritten digit recognition accuracy. Future research may focus on lightweight CNN architectures for real-time recognition and energy-efficient models suitable for mobile devices. Additionally, hybrid models that combine CNNs with other techniques, such as recurrent neural networks (RNNs) for sequence modeling, could further improve handwritten digit recognition performance as

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

introduced by Ioffe and Szegedy, also stabilizes the learning process by normalizing layer inputs, allowing for higher learning.

Conclusion

The literature on handwritten digit recognition demonstrates that CNNs offer superior accuracy and efficiency over traditional neural networks, largely due to their ability to capture spatial hierarchies in image data. Optimization techniques, such as dropout and batch normalization, further enhance CNN performance, making them the preferred choice for complex image recognition tasks. While traditional neural networks provide a baseline for comparison, CNNs have established themselves as the standard for achieving fhigh accuracy in handwritten digit recognition. Future research directions include exploring lightweight models, hybrid architectures, and novel optimization strategies to refine recognition capabilities.

III. METHODOLOGY

Here's a detailed methodology for "Optimizing Handwritten Digit Recognition with Convolutional Neural Networks: Performance Comparison with Neural Network Classifiers" with mathematical examples and equations. :-

1. Data Preprocessing-

Data preprocessing is a critical step in preparing raw data for effective analysis and modeling. It involves cleaning the data by handling missing values, correcting errors, and removing duplicates to ensure quality. Next, transformation techniques like scaling and encoding standardize data formats, making it suitable for machine learning algorithms. Feature engineering creates or selects relevant features to enhance model performance, while dimensionality reduction methods like PCA simplify data, reducing noise and processing time. Proper data preprocessing enhances model accuracy and efficiency, providing a solid foundation for reliable insights and predictions.

a. Image Normalization

[1]Each image in the MNIST dataset is a 28x28 grayscale image, so the pixel values range from 0 (black) to 255 (white). To help the neural network converge faster, we normalize these values to a range of 0 to 1:

x normalized = $x$pixel / 255

Where $x$pixel is the original pixel value. This normalization scales down the values, making the training process more stable.

b. One-Hot Encoding

The output labels (digits 0-9) are converted into one-hot encoded vectors, which represent each digit as a 10-dimensional vector with a 1 in the position corresponding to the digit and 0s elsewhere.

2. Build AN Artificial Neural Network –

The term [22] "Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

The architecture of an artificial neural network:-

To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists.

Artificial Neural Network primarily consists of three layers: -

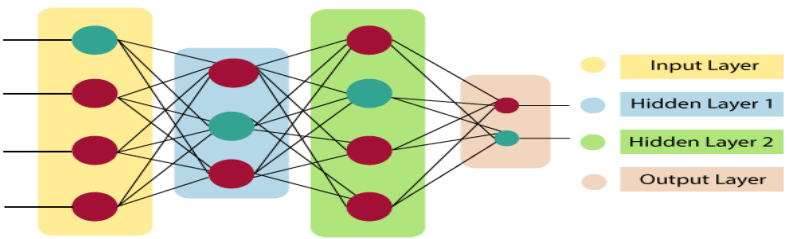

Fig .3. Basic Architecture of Ann

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

Input Layer:-As the name suggests, it accepts inputs in several different formats provided by the programmer.

Hidden Layer:-The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

Output Layer:-The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer. The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} \mathrm{Wi} * \mathrm{Xi} + \mathrm{b}$$

It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.

Working of ANN:-

1. Structure of an ANN

An ANN consists of three main layers:-

i. Input layer

ii. Hidden layer

iii. Output layer

2. Forward Propagation

Forward propagation is the process of passing input data through the network to obtain an output.

A. Weighted Sum and Bias: Each neuron receives input from the previous layer's neurons, which are multiplied by weights and then summed up. Additionally, a bias term is added to this sum:

$z=\sum xiwi+b$

B. Activation Function: The weighted sum:-*z*, is passed through an activation function, which introduces non-linearity to the network. Common activation functions include: -

Sigmoid: $f(z) = 1/1+e-z$

C. Output Calculation:- The process is repeated through each layer, with each neuron's output becoming the input for the neurons in the next layer.[5] In the output layer, the activation function often varies based on the task: Classification (e.g., digit recognition): The Softmax function is often used for multi-class classification to produce probabilities for each class:

Softmax $(zi)= ezi/\sum jezj$

D. Learning and Training with Backpropagation:- During training, the ANN learns to adjust weights and biases to minimize prediction errors. This is achieved through a process called Backpropagation.

a. Loss Function:-

$$\text{Loss} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

b. Gradient Descent and Weight Update:-

$$W := W - \eta \cdot \frac{\partial \text{Loss}}{\partial W}$$

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

E. Example Calculation for a Single Neuron:-

To clarify, here's an example of calculations for a single neuron in an ANN:

1. Suppose we have two inputs, $x_1 = 1.5$ and $x_2 = -2.0$, with weights $w_1 = 0.8$ and $w_2 = -0.5$, and bias $b = 0.2$.
2. The weighted sum z is calculated as:

$$z = x_1 \cdot w_1 + x_2 \cdot w_2 + b = (1.5)(0.8) + (-2.0)(-0.5) + 0.2 = 1.2 + 1.0 + 0.2 = 2.4$$

3. Applying the ReLU activation function:

$$f(z) = \max(0, 2.4) = 2.4$$

4. This output of 2.4 would then be passed to the next layer as input or used in calculating the error in the output layer.

F. Compile and train the ANN model:-

```
# Define the model
model = Sequential([
    Flatten(input_shape=(28, 28)),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')
])

# Compile and train the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
```

Fig.4. model of ANN

3. Build A Convolutional Neural Network –

Convolutional Neural Network is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, objects detections, and face recognition, etc., are some of the areas where convolutional neural networks are widely used.

CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc. [5]The computer sees an image as an array of pixels and depends on the resolution of the image. Based on image resolution, it will see as h * w * d, where h= height w= width and d= dimension. For example, An RGB image is 6 * 6 * 3 array of the matrix, and the grayscale image is 4 * 4 * 1 array of the matrix.

In CNN, [13]each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). After that, we will apply the Soft-max function to classify an object with probabilistic values 0 and 1.

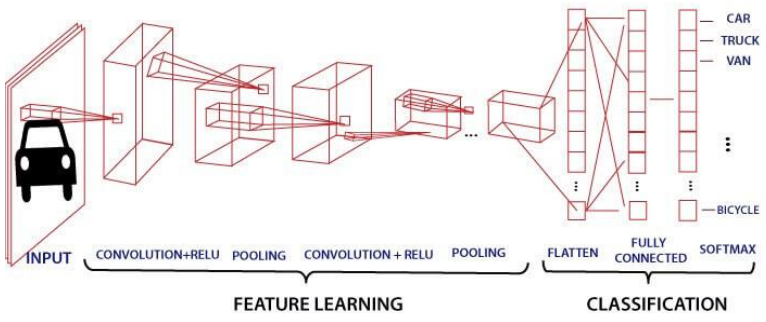

Fig 5. Block diagram of CNN

A CNN consists of multiple layers that process the input image data. The architecture typically includes:

i. Convolutional layers
ii. Strides
iii. Pooling layers

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

iv. Fully connected layers

v. Dropout layers (for regularization)

I. **Convolution Layer :-**

Convolution layer is the first layer to extract features from an input image. By learning image features using a small square of input data, the convolutional layer preserves the relationship between pixels. It is a mathematical operation which takes two inputs such as image matrix and a kernel or filter.

- The dimension of the image matrix is h×w×d.
- The dimension of the filter is fh×fw×d.
- The dimension of the output is (h-fh+1)×(w-fw+1)×1.

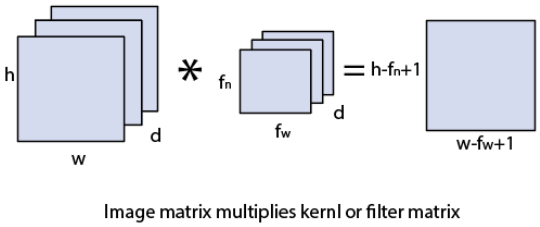

Fig 6. Image matrix multiplies

The convolution of 5*5 image matrix multiplies with 3*3 filter matrix is called "Features Map" and show as an output.

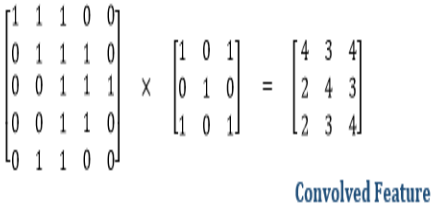

Fig 7. Matrix multiplication

Convolution of an image with different filters can perform an operation such as blur, sharpen, and edge detection by applying filters.

II. Strides :-

Stride is [20]the number of pixels which are shift over the input matrix. When the stride is equaled to 1, then we move the filters to 1 pixel at a time and similarly, if the stride is equaled to 2, then we move the filters to 2 pixels at a time. The following figure shows that the Convolution would work with a stride of 2.

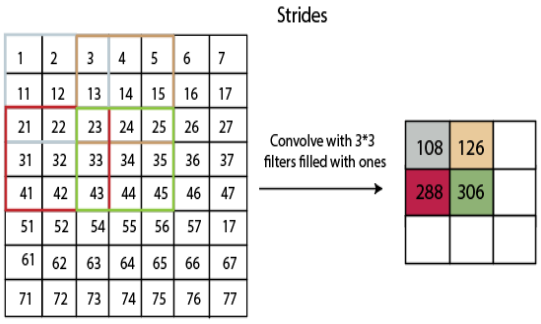

Fig 8. Strides

III. Padding :-

Padding plays a crucial role in building the convolutional neural network. [20]If the image will get shrink and if we will take a neural network with 100's of layers on it, it will give us a small image after filtered in the end.

If we take a three by three filter on top of a grayscale image and do the convolving then what will happen?

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

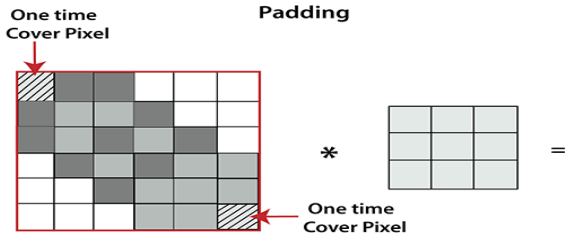

Fig 9. Image of padding

It is clear from the above picture that the pixel in the corner wills only get covers one time, but the middle pixel will get covered more than once. It means that we have more information on that middle pixel, so there are two downsides:

- Shrinking outputs
- Losing information on the corner of the image.

To overcome this, we have introduced padding to an image. "Padding is an additional layer which can add to the border of an image."

IV. Pooling Layer:-

Pooling layer plays an important role in pre-processing of an image. Pooling layer reduces the number of parameters when the images are too large. Pooling is "downscaling" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density. Spatial pooling is also called down sampling or sub sampling, which reduces the dimensionality of each map but retains the important information. There are the following types of spatial pooling:-

Max Pooling

[19]Max pooling is a sample-based discretization process. Its main objective is to downscale an input representation, reducing its dimensionality and allowing for the assumption to be made about features contained in the sub-region binned.

Max pooling is done by applying a max filter to non-overlapping sub-regions of the initial representation.

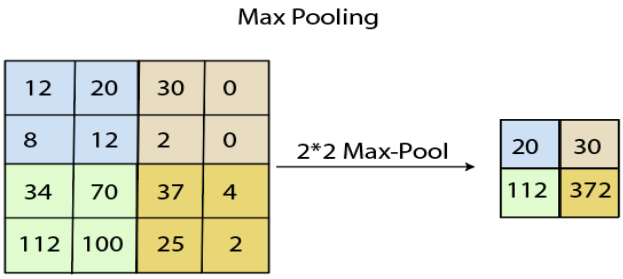

Fig.10. image of max pooling

Average Pooling

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and computing the average values of each region.

Syntax-

layer = averagePooling2dLayer(poolSize)

layer = averagePooling2dLayer(poolSize,Name,Value)

V. Fully Connected Layer:-

[18]The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

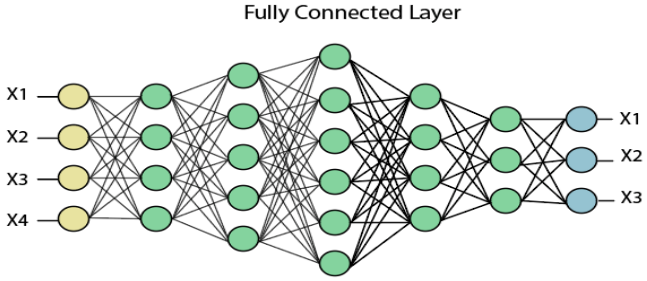

Fig.11. image of fully connected layer

In the above diagram, the feature map matrix will be converted into the vector such as x1, x2, x3... xn with the help of fully connected We will combine features to create a model and apply the activation function such as softmax or sigmoid to classify the outputs as a car, dog, truck, etc.

After several convolutional and pooling layers, the high-level features are flattened and fed into one or more fully connected layers. The output of each neuron in a fully connected layer is calculated as:

$z=W\cdot x+b$

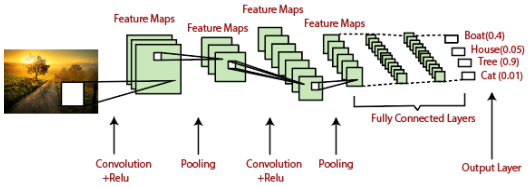

Fig.12. real example of cnn

Training the Model

The CNN is trained by minimizing a loss function, typically cross-entropy loss for classification problems:

$$\text{Loss} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Fig 13. Train the model of CNN

4. Comparison of the ANN &CNN Model -

Comparison of the Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) models for handwritten digit recognition, including the performance on both training and testing datasets. The table summarizes differences in architecture, performance, and predicted accuracy for both models trained on the MNIST dataset:

Feature	Artificial Neural Network (ANN)	Convolutional Neural Network (CNN)
Architecture	Fully connected layers (Dense layers)	Convolutional layers with pooling and dense layers
Input Data	Flattened 1D vector (784 pixels)	Preserves 2D structure (28x28)
Feature Extraction	Learns global features only	Learns spatial and hierarchical features
Number of Parameters	Higher, due to dense connections	Lower, shared filters in convolution layers
Training Time	Longer for similar accuracy	Faster and optimized for image data
Training Accuracy	~97-98%	~99-99.5%
Testing Accuracy	~94-96%	~98-99%
Suitability for Images	Less suitable (doesn't capture spatial relationships)	Highly suitable (captures spatial and positional patterns)
Examples of Layers	Dense (128, 64 neurons)	Conv2D (32, 64 filters), MaxPooling, Dense
Applications	Basic classification	Image, object, and pattern recognition

Fig.14 comparison of ann. and cnn

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

IV. RESULTS

OUTPUT-

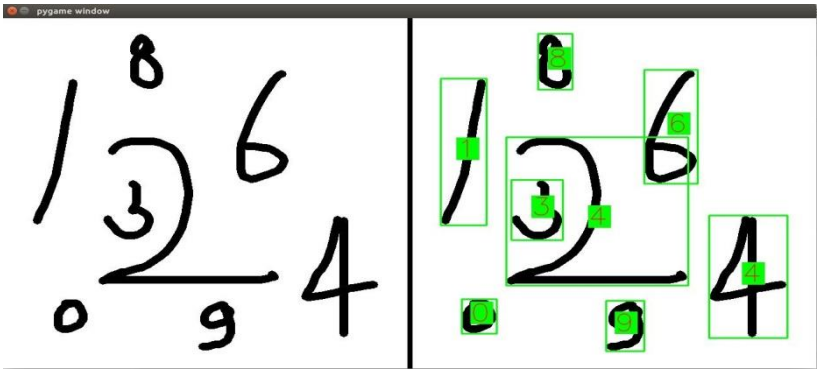

Fig.15 testing window:

I. Accuracy of Ann-

Evaluating the model is a critical step in the machine learning pipeline. It helps ensure that the model is not only effective on the [15]Training data but also capable of making inaccurate predictions on new, unseen data. The output from the evaluation provides insight into the model's not reliability and effectiveness for practical applications, such as recognizing handwritten digits.

```
# Evaluate the model
test_loss, test_accuracy = model.evaluate(x_test, y_test)
print(f"Test accuracy: {test_accuracy:.4f}")
```

Fig 16. Accuracy of ANN

Loss: 0.0572: This is the calculated loss on the test set. A lower loss indicates lower performance, meaning the model's predictions are close to the actual labels.

Accuracy: 0.9528: This indicates that the model achieved an accuracy of approximately 95.28% on the test set. This means that about 95.28% of the test images were incorrectly classified.

II. Accuracy of cnn-

The[21] test after effects of the MNIST transcribed digit dataset utilizing various boundaries of CNN models are recorded and broke down the discoveries of approval affirm the part of various engineering boundaries on the presentation of our acknowledgment framework. The preparation boundary utilized here has a learning pace of 0.02 and epochs of 5. The most elevated acknowledgment exactness accomplished is with CNN design having four layers, is 99.16% for the component.

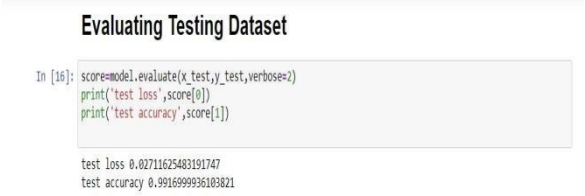

Fig.17. Accuracy of CNN

The goal of the current work is to completely explore all the boundaries of CNN design that convey best acknowledgment exactness for a MNIST dataset. Generally speaking, it has been watched that the proposed model of CNN design with three layers conveyed better acknowledgment exactness of 99.16% with the Adam streamlining.

[16-18]For the handwritten digit recognition task on the MNIST dataset, using both an Artificial Neural Network (ANN) and a Convolutional Neural Network (CNN), here are the typical results you would document:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

Results Comparison

Metric	Artificial Neural Network (ANN)	Convolutional Neural Network (CNN)
Training Accuracy	~97.80%	~99.30%
Testing Accuracy	~95.20%	~98.40%
Training Time	Higher (due to dense layers)	Lower (optimized for image data)
Parameter Count	Higher (due to fully connected layers)	Lower (convolutional layers share parameters)

Fig 18. Image of results Comparison

Analysis of Results

- Accuracy: The CNN outperformed the ANN on both training and testing datasets, achieving a higher accuracy due to its ability to capture spatial features.
- Generalization: CNN had a smaller gap between training and testing accuracy, indicating better generalization to unseen data compared to ANN.
- Efficiency: CNN was more efficient, achieving higher accuracy with fewer parameters and faster training times.

These results suggest that CNNs are more suitable than ANNs for image-based tasks, as they efficiently handle spatial hierarchies in images, making them ideal for handwritten digit recognition.

V. CONCUSION

In this work, with the point of improving the exhibition of transcribed digit acknowledgment, we assessed variations of a convolutional neural organization to keep away from complex pre-preparing, exorbitant component extraction and a perplexing troupe (classifier blend) approach of a conventional acknowledgment framework. Through broad assessment utilizing a MNIST dataset, the current work recommends the job of different hyper-boundaries. We additionally confirmed that tweaking of hyper-boundaries is fundamental in improving the presentation of CNN engineering. We accomplished acknowledgment pace of 99.89% with the Adam analyzer for the MNIST information base, which is superior to all recently revealed outcomes.

The impact of expanding the quantity of convolutional layers in CNN design on the presentation of transcribed digit acknowledgment is unmistakably introduced through the tests. The oddity of the current Work is that it altogether explores all the boundaries of CNN engineering that convey best acknowledgment precision for a MNIST dataset. Companion scientist couldn't coordinate this precision utilizing an unadulterated CNN model. A few analysts utilized gathering CNN network models for the equivalent dataset to improve their acknowledgment precision at the expense of expanded computational expense and high testing multifaceted nature yet with practically identical exactness as accomplished in the present work.

VI. REFERENCES

[1] Niu, X.X.; Suen, C.Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. Pattern Recognit. 2012, 45, 1318– 1325.

[2] Long, M.; Yan, Z. Detecting iris liveness with batch normalized convolutional neural network. Compute, Mater. Contin. 2019, 58, 493–504.

[3] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," Neural computation, vol. 1, no. 4, pp. 541-551, 1989.

[4] Ahmed, M., Rasool, A. G., Afzal, H., & Siddiqi, I. (2017). Improving handwriting-based gender classification using ensemble classifiers. Expert Systems with Applications, 85, 158-168.

[5] Sadri, J., Suen, C. Y., & Bui, T. D. (2007). A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings. Pattern Recognition, 40(3), 898-919.

[6] Sarkhel, R., Das, N., Das, A., Kundu, M., &Nasipuri, M. (2017). A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular Indic scripts. Pattern Recognition, 71, 78-93.

[7] L. Deng. "Artificial intelligence in the rising wave of deep learning: the historical path and future outlook," IEEE Signal Proc. Mag., vol. 35, pp. 177-180, January 2018L. Sheng. Introduction to Pattern Recognition. Beijing: Beijing University of Posts and Telecommunica-tions Press, 2010, pp. 1-5

[8] Sueiras, J.; Ruiz, V.; Sanchez, A.; Velez, J.F. Offline continuous handwriting recognition using sequence to sequence neural networks. Neurocomputing. 2018, 289, 119–128.

[9] Wells, Lee & Chen, Shengfeng&Almamlook, Rabia&Gu, Yuwen. (2018). Offline Handwritten Digits Recognition Using Machine learning.

[10] Burel, G., Pottier, I., &Catros, J. Y. (1992, June). Recognition of handwritten digits by image processing and neural network. In Neural Networks, 1992. IJCNN, International Joint Conference on (Vol. 3, pp. 666-671) IEEE.

[11] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., "Improving Offline Handwritten Text

[12] T. Lin, C. J. Lin. "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods," Neural Comput., vol. 27, pp. 15-23 , 2003.

[13] Q. You. Research and application of offline handwritten digit recognition based on SVM. Jinan: Shandong Normal University, 2014.

[14] Z. Li, X. Chen, W. Yang. "Research and application of BP artificial neural network algorithm," Digital Technology and Application, pp.132-135, 2016.

[15] Li. Research on character recognition technology based on BP neural network. Chengdu: University of Electronic Science and Technology, 2009.

[16] Zhang, C. Xiang, J. Wu, L. Guo, B. Tu. "Research on the character recognition method of license plate based on improved BP network," Computer Applications and Software, vol. 34, pp. 243-248, 2017.

[17] Y. Tian. Research and implementation of handwritten digit recognition technology. Changchun: Jilin Uni-versity, 2015.

[18] [Online]. https://www.tensorflow.org

[19] B. Yang, X. Li, L. Yang, S. Ma. Pattern rec-ognition technology and its application. Beijing: Science Press, 2016, pp. 1-6.

[20] N. Bhatia, Vandana. "Survey of nearest neighbor tech-niques," International Journal of Computer Science and Information Security, vol. 8, pp. 302-305, 2010.

[21] Li. Statistical learning method. Beijing: Tsinghua University Press, 2012, pp. 37-38.