ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

RECOGNITION OF HANDWRITTEN DIGIT USING CNN

Rithanika S[1], Mrs. J. Vinitha, M. Sc., (Ph .D)[2]
[1]*Reg. No: 231AI046*
[2]*Assistant Professor*
Department of Artificial Intelligence and Machine Learning
Dr. N.G.P. Arts and Science College

ABSTRACT

Handwritten digit recognition is an important area in pattern recognition and computer vision, with widespread applications in bank cheque processing, postal services, form digitization, and automated data entry systems. One of the main challenges in this field is the large variation in individual handwriting styles, including differences in size, shape, and orientation of digits. Traditional handwritten digit recognition techniques rely heavily on manual feature extraction, which is time-consuming, requires expert knowledge, and often results in lower accuracy and limited adaptability to diverse handwriting patterns.

To overcome these limitations, a web-based handwritten digit recognition system is implemented using a Convolutional Neural Network (CNN). The system is demonstrated through a simple website developed using a lightweight backend framework such as Flask, which allows users to upload handwritten digit images and view prediction results easily. The CNN model is trained and tested using the MNIST dataset, consisting of digit samples ranging from 0 to 9. By automatically learning important features through convolutional, pooling, and fully connected layers, the proposed system eliminates manual feature extraction and achieves improved accuracy and faster processing. The results confirm that the CNN-based approach is efficient, reliable, and suitable for real-time and practical handwritten digit recognition applications.

1. INTRODUCTION

Handwritten digit recognition is an important and well-researched topic in the fields of pattern recognition, image processing, and artificial intelligence. It deals with the automatic identification and classification of numerical digits written by humans. While humans can easily recognize handwritten digits, enabling computers to perform the same task accurately is challenging. This difficulty arises due to variations in individual handwriting styles, differences in size, shape, thickness, orientation, and spacing of digits. Developing an efficient handwritten digit recognition system is essential for automating many real-world applications that still rely on handwritten data.

In the modern digital era, large volumes of handwritten information are still widely used in various sectors. Documents such as bank cheques, application forms, examination answer sheets, postal codes, invoices, bills, and historical records contain handwritten digits that must be processed and stored digitally. Manual data entry of such information is slow, costly, and highly prone to errors. As organizations move toward automation and digital transformation, there is a growing demand for systems that can accurately convert handwritten data into machine-readable form. Handwritten digit recognition systems play a crucial role in meeting this demand.

Traditional handwriten digit recognition techniques were mainly based on image processing and classical machine learning methods. These approaches involved multiple stages, including image acquisition, preprocessing, segmentation, feature extraction, and classification. Preprocessing techniques such as noise removal, normalization, and thinning were applied to improve image quality. After preprocessing, handcrafted features like edges, contours, strokes, zoning features, and geometric properties were manually designed. These extracted features were then fed into classifiers such as k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Decision Trees, or Hidden Markov Models (HMM) for digit classification.

Although traditional approaches achieved moderate success, they suffered from several limitations. One major drawback was the dependency on manual feature extraction, which is a complex and time-consuming process.

The performance of the system largely depended on the quality and relevance of the selected features. Designing features that work effectively for all handwriting styles is extremely difficult. As a result, these systems often failed when dealing with noisy images, distorted digits, overlapping characters, or unfamiliar handwriting patterns. These challenges limited the accuracy and robustness of traditional digit recognition systems.

The limitations of classical methods encouraged researchers to explore more advanced techniques based on artificial neural networks and deep learning. With the rapid growth of computational power and the availability of large labeled datasets, deep learning

1.1 Overview of the project

Handwritten digit recognition is a critical and widely studied problem in the fields of artificial intelligence, pattern recognition, and image processing. Even in a digital era where keyboards and digital forms dominate, handwritten digits continue to play a major role in daily administrative, financial, and educational activities.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

Cheques, forms, postal codes, examination answer sheets, invoices, and archival documents frequently contain handwritten numbers that require accurate interpretation. Manual recognition and data entry of these digits are not only time-consuming but also highly prone to human errors, which can lead to significant delays, mistakes, and increased operational costs. These challenges emphasize the importance of automated systems capable of recognizing handwritten digits efficiently and accurately, thereby reducing reliance on human effort and improving productivity in multiple sectors.

Recognizing handwritten digits is an inherently challenging task due to the diversity in human writing styles. Each individual has a unique way of writing numbers, and even the same person may produce different forms of the same digit depending on speed, writing instrument, or context. Variations can include differences in size, shape, thickness, and alignment of digits, as well as the presence of connected strokes, slants, or distortions. These variations make it difficult for traditional recognition methods, which rely on manually designed rules and fixed features, to perform consistently. The need to address these variations while maintaining high accuracy and speed forms the central motivation for this project. By using advanced machine learning and artificial intelligence techniques, the system aims to learn the inherent patterns of handwritten digits.

The system developed in this project employs Convolutional Neural Networks (CNNs), a class of deep learning algorithms particularly effective for image analysis and classification. CNNs are inspired by the human visual system and excel at learning hierarchical features directly from input images. Unlike traditional approaches that require careful manual feature engineering, CNNs automatically detect features such as edges, curves, strokes, and unique digit shapes, which are essential for accurate recognition. The network is designed to learn progressively complex patterns, starting from basic visual elements in initial layers to digit-specific characteristics in deeper layers. This hierarchical learning allows the system to handle a wide variety of handwriting styles, even in the presence of distortions or noise, providing high accuracy and robustness in digit recognition.

Training the CNN model is a key part of the project, as it ensures the system can generalize to new, unseen handwriting styles. By exposing the network to thousands of examples of handwritten digits, the system learns to identify subtle differences between digits that might appear visually similar. The learning process involves multiple iterations and the adjustment of internal network parameters to minimize classification errors. This enables the system to recognize digits accurately across diverse handwriting patterns, including unusual or irregular forms. The extensive training ensures that the system can perform consistently even when encountering handwriting styles that were not included in the training data, which is essential for real-world applicability.

The practical relevance of the project is extensive. In the banking sector, automated recognition of handwritten digits can significantly speed up cheque processing, form digitization, and account verification, reducing the workload of employees while minimizing errors. Postal services can utilize the system for reading and sorting postal codes efficiently, ensuring faster delivery of mail. Educational institutions can benefit from automatic evaluation of examination answer sheets, reducing the time required for manual marking and improving consistency in grading. Government organizations and private corporations can digitize large volumes of handwritten forms and records, improving storage, analysis, and retrieval of information while eliminating the human errors associated with manual data entry. These practical applications highlight the transformative potential of automated handwritten digit recognition in real-world workflow.

2. SYSTEM STUDY

The system study for handwritten digit recognition focuses on analyzing existing methodologies and identifying the most effective approach to accurately classify digits from 0 to 9. It examines traditional machine learning techniques such as k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and Decision Trees, highlighting their dependence on manual feature extraction and limitations in handling variations in handwriting. The study also reviews deep learning methods, particularly Convolutional Neural Networks (CNNs), which automatically learn hierarchical features from image data, providing superior accuracy and robustness. Furthermore, recent advancements such as ensemble methods, pruning strategies, recurrent architectures like ConvGRU, and explainable AI techniques are explored to improve recognition performance, efficiency, and interpretability. The system study concludes that integrating deep learning models with optimized training strategies and adaptive architectures offers the most reliable solution for real-world handwritten digit recognition, ensuring high accuracy, scalability, and applicability across diverse datasets.

2.1 Existing System

The existing handwritten digit recognition systems mainly rely on traditional image processing and classical machine learning techniques. These systems follow a pipeline that includes image preprocessing, segmentation, manual feature extraction, and classification. Features such as edges, contours, strokes, zoning, and geometric properties are extracted using handcrafted methods. After feature extraction, classifiers like k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), Decision Tree, and Hidden Markov Model (HMM) are used to recognize the digits. Although these approaches work reasonably well for clean and limited datasets, their performance degrades when there are variations in handwriting styles, noise, distortions, and different writing conditions. The effectiveness of the existing system highly depends on the quality of manually designed features and extensive domain expertise.

Drawbacks of Existing System

- Manual feature extraction is complex, time-consuming, and requires expert knowledge.
- Recognition accuracy is low when handling different handwriting styles and distortions.
- Poor performance in noisy or low-quality images.
- Limited scalability for large datasets and real-time applications.
- High dependency on preprocessing techniques reduces

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

system robustness.
- Difficulty in adapting the system to new languages or writing patterns.

2.2 Problem Identification

Low Accuracy in Noisy Images and Proposed System Methods

One of the major challenges in handwritten digit recognition is achieving high accuracy when input images contain noise such as ink smudges, background distortion, uneven lighting, scanning errors, or low resolution. Traditional systems rely on handcrafted features and simple thresholding techniques, which are highly sensitive to noise. Due to this, important digit characteristics like stroke edges, curves, and shapes may get distorted, resulting in incorrect predictions and poor system performance.

To overcome this problem, the proposed system uses a Convolutional Neural Network (CNN)- based approach. CNNs automatically learn robust features directly from raw images, reducing the impact of noise. The proposed system first applies image preprocessing techniques such as normalization and resizing to ensure uniform input. Then, convolution layers extract low-level features like edges and textures, while deeper layers capture high-level features such as digit shapes and patterns. Pooling layers reduce noise and dimensionality by focusing on the most important features. The system is trained on a large dataset (such as MNIST), which contains various writing styles and noisy samples, improving generalization. Finally, the softmax classifier accurately predicts the digit class even when the input image contains noise.

2.3 Proposed System

The proposed system implements a Convolutional Neural Network (CNN)–based approach for handwritten digit recognition to overcome the limitations of traditional methods. The system automatically learns relevant features from raw handwritten digit images without requiring manual feature extraction. By using multiple convolutional, pooling, and fully connected layers, the CNN efficiently captures spatial patterns and structural details of digits. The model classifies input images into digit classes ranging from 0 to 9 with high accuracy and robustness. The system is capable of handling variations in handwriting styles, size, thickness, and orientation, making it suitable for real-world digit recognition applications.

2.3.1 Advantages of Proposed System

- Eliminates the need for manual feature extraction.
- Achieves higher recognition accuracy compared to traditional methods.
- Handles diverse handwriting styles effectively.
- Provides fast and efficient digit classification.
- Reduces human effort and error.
- Easily scalable to larger datasets and other character recognition tasks.

3. System Requirements

3.1 System Configuration

The system configuration describes the overall software environment required to develop and deploy the proposed Handwritten Digit Recognition using CNN system. The system is designed to recognize handwritten digits (0–9) from Tamil, Hindi, and English languages using deep learning techniques.

The proposed handwritten digit recognition system is configured using a client–server architecture that integrates hardware and software components to perform accurate digit classification. The user interacts with the system through a web-based interface to upload handwritten digit images belonging to Tamil, Hindi, or English languages. The uploaded images are transmitted to the application server for processing.

The application server performs image preprocessing operations such as resizing, grayscale conversion, and pixel normalization to make the input suitable for the convolutional neural network (CNN). The CNN model, developed using TensorFlow and Keras, extracts meaningful features from the images and classifies the digits in the range of 0 to 9. The predicted digit and its confidence score are stored in a database for further analysis. The system supports both CPU and GPU environments, enabling efficient training and real-time inference.

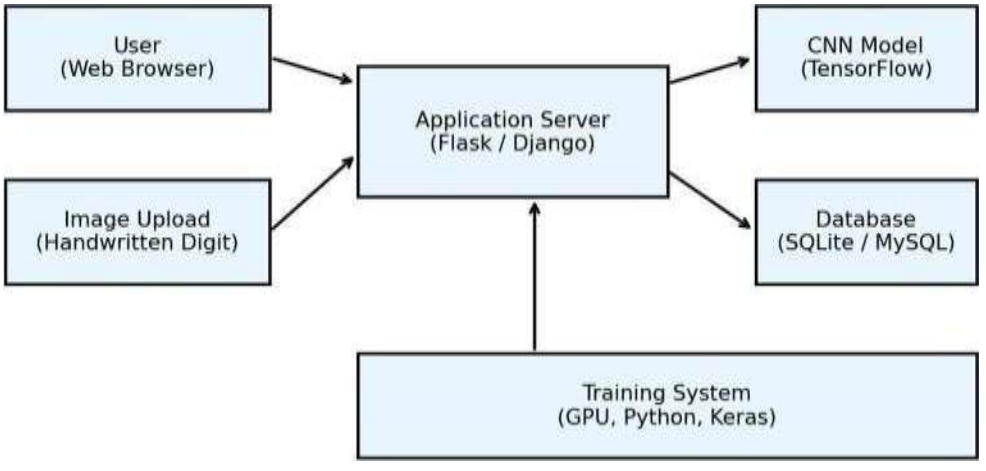

Fig 1: Architecture Diagram of CNN-Based Handwritten Digit Recognition System

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

3.2 Software Description

- **Operating System:** Windows 10 / Linux
- **Programming Language:** Python 3.x
- **Deep Learning Framework:** TensorFlow / Keras
- **Web Framework:** Flask / Django
- **Database:** SQLite / MySQL
- **IDE / Tools:** Jupyter Notebook, VS Code
- **Browser:** Google Chrome / Mozilla Firefox

Overview of Software Environment

The handwritten digit recognition system using Convolutional Neural Networks (CNNs) is developed entirely using software tools that support machine learning, image processing, and deep learning. The software environment plays a crucial role in designing, training, testing, and evaluating the CNN model. A suitable combination of programming language, libraries, frameworks, and development tools ensures that the system performs efficiently and delivers accurate recognition results. The selected software components provide flexibility, scalability, and ease of implementation, making the system suitable for both academic and real-world applications

Programming Language: Python

Python is used as the primary programming language for implementing the handwritten digit recognition system. Python is widely preferred in machine learning and deep learning applications due to its simplicity, readability, and extensive library support. It allows rapid development and easy debugging, which is essential for experimenting with neural network architectures. Python's object- oriented and modular structure enables clear organization of code, making the system easier to maintain and extend. Additionally, Python has strong community support, which helps in accessing pre-built modules and resolving implementation challenges efficiently.

Deep Learning Framework

TensorFlow and Keras TensorFlow is used as the core deep learning framework for building and training the CNN model. It provides powerful tools for numerical computation and automatic differentiation, which are essential for training deep neural networks. TensorFlow supports efficient execution on both CPUs and GPUs, allowing faster model training. Keras, which is integrated with TensorFlow, is used as a high-level API to simplify model development. Keras enables easy creation of CNN layers such as convolutional layers, pooling layers, and fully connected layers. It also provides built-in functions for activation layers, loss functions, optimizers, and evaluation metrics. The use of TensorFlow with Keras significantly reduces development complexity while maintaining high performance and accuracy.

Image Processing Libraries

Image processing plays an important role in handwritten digit recognition, and several Python libraries are used for handling image data. OpenCV is used for reading, resizing, and manipulating handwritten digit images. It provides efficient functions for image transformations and pixel-level operations. Additionally, NumPy is used extensively for numerical operations and matrix computations. Since images are represented as arrays of pixel values, NumPy allows efficient handling of large datasets and supports fast mathematical operations required during training and testing. These libraries together ensure smooth processing of image data throughout the system.

Data Handling and Visualization

Tools For managing datasets and analyzing results, Python-based data handling tools are used. The MNIST dataset, which contains thousands of labeled handwritten digit images, is loaded and processed using built-in dataset utilities provided by TensorFlow and Keras. This simplifies dataset handling and ensures compatibility with the CNN model. Matplotlib is used for visualizing images, training accuracy, loss curves, and evaluation results. Visualization helps in understanding model performance and identifying issues such as overfitting or underfitting. By plotting graphs of training and validation metrics, the effectiveness of the CNN architecture can be clearly analyzed.

Model Training and Optimization Software Support

The software environment supports various optimization techniques required for effective CNN training. Optimizers such as Adam or Stochastic Gradient Descent (SGD) are used to minimize the loss function during training. These optimizers adjust the network weights automatically to improve accuracy. Loss functions such as categorical cross-entropy are used to measure the difference between predicted outputs and actual labels. The software framework also supports regularization techniques such as dropout, which helps prevent overfitting. All these features are easily implemented using TensorFlow and Keras, making the training process efficient and reliable.

Development Environment

An integrated development environment (IDE) such as Jupyter Notebook, Spyder, or Visual Studio Code is used for developing and testing the system. Jupyter Notebook is especially useful for experimenting with code and visualizing intermediate results step by step. It allows code execution in cells, making debugging and analysis easier. These environments provide syntax highlighting, debugging tools, and error handling features that improve productivity. The flexibility of the development environment enables continuous improvement and testing of the CNN model.

Software Scalability and Flexibility

The software design of the handwritten digit recognition system is highly scalable. The CNN model can be easily modified to recognize additional characters or extended to support multilingual digit recognition. The use of Python and TensorFlow allows integration with other systems such as web applications, mobile applications, or embedded systems. The modular structure of the software ensures that individual components such as data loading, model training, and prediction can be updated independently. This flexibility makes the system adaptable to future enhancements and research advancements.

Reliability and Performance Support

The selected software tools ensure reliable performance and consistent results. TensorFlow's computational graph execution and optimized backend improve processing speed and accuracy.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

The use of tested and widely adopted libraries reduces the risk of software failures. Error handling, logging, and validation mechanisms supported by the software environment help in identifying issues during training and deployment. This ensures that the system remains stable and produces accurate digit recognition results under different conditions.

4. SYSTEM DESIGN

The system design of the handwritten digit recognition project is based on a **Convolutional Neural Network (CNN)** architecture that efficiently processes handwritten digit images and classifies them into numerical categories from 0 to 9. The design follows a structured flow that begins with input image acquisition and ends with accurate digit prediction. Each stage of the system is carefully organized to ensure reliability, scalability, and high recognition accuracy.

The system accepts handwritten digit images as input, which may be obtained from scanned documents, image files, or digit datasets. These images are represented in grayscale format to reduce computational complexity while preserving essential visual information. The CNN model processes the input images through multiple convolutional layers, which automatically extract important features such as edges, curves, and shapes. Pooling layers are then used to reduce the spatial dimensions of feature maps, ensuring efficient computation and minimizing overfitting.

The extracted features are passed to fully connected layers, which integrate high-level and low- level features to perform classification. The output layer uses a softmax activation function to generate probability scores for each digit class, allowing the system to select the most likely digit. During training, optimization techniques such as backpropagation and regularization are employed to improve model accuracy and generalization.

4.1 System Flow and Architecture Diagram

System Flow Diagram

The system flow diagram illustrates the step-by-step operation of the handwritten digit recognition system. Initially, the user uploads a handwritten digit image through the web interface. The uploaded image is then sent to the preprocessing module, where operations such as resizing, grayscale conversion, and normalization are performed. After preprocessing, the image is forwarded to the Convolutional Neural Network (CNN) model for feature extraction and classification. The trained CNN model analyzes the image and predicts the corresponding digit between 0 and 9. Finally, the recognized digit along with its confidence score is displayed to the user. This sequential flow ensures accurate and efficient recognition of handwritten digits from Tamil, Hindi, and English languages.

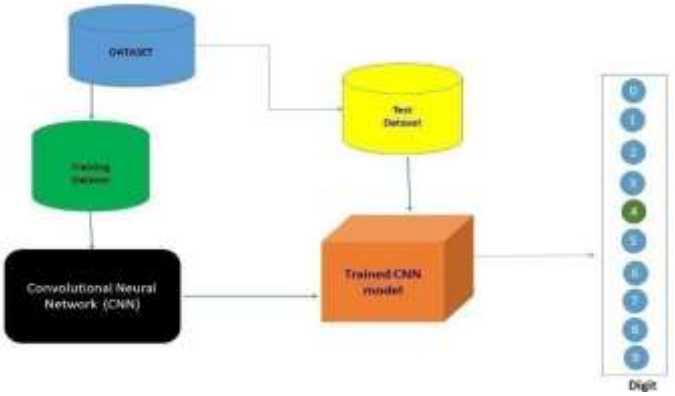

Fig 2: System Flow Diagram

Architecture Diagram

The architecture diagram represents the overall structure of the handwritten digit recognition system and the interaction between its components. The system follows a client–server architecture, where the user interface allows users to upload handwritten digit images through a web or mobile platform. The application server receives the uploaded image and performs preprocessing operations such as resizing, grayscale conversion, and normalization.

The preprocessed image is then passed to the Convolutional Neural Network (CNN) model implemented using TensorFlow and Keras. The CNN model extracts relevant features and classifies the input image into digits ranging from 0 to 9. The prediction results and confidence scores are stored in the database for future reference. A separate training system equipped with GPU support is used to train and update the CNN model, which is deployed on the server for real-time inference.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

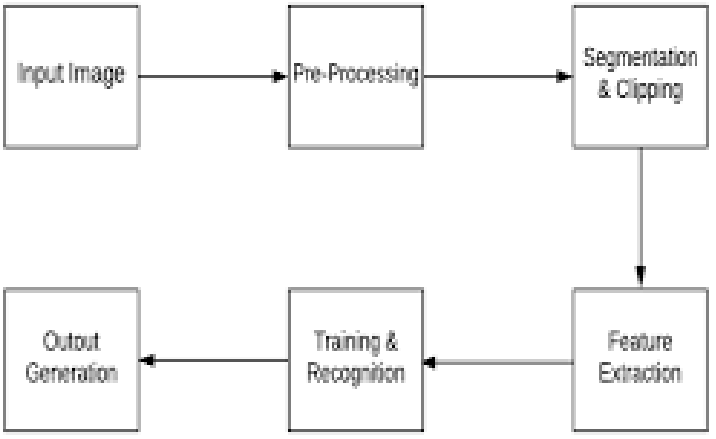

Fig 3: Architecture for proposed system

4.2 Input Design

The input design of the handwritten digit recognition system focuses on accepting handwritten digit images in a user-friendly and efficient manner. The system allows users to upload images containing single handwritten digits from Tamil, Hindi, or English languages through a web- based interface. The accepted image formats include JPEG, PNG, and BMP. Once the image is uploaded, it is validated and preprocessed by converting it into grayscale, resizing it to a fixed dimension, and normalizing pixel values. Proper input design ensures that the images are in a suitable format for the Convolutional Neural Network (CNN), thereby improving recognition accuracy and system performance.

4.3 Output Design

The output design of the system presents the digit recognition results in a clear and understandable format. After processing the input image through the CNN model, the system displays the recognized digit (0–9) along with its corresponding confidence score. The output is shown on the web interface in real time, enabling users to easily interpret the prediction. Additionally, the recognized results and related metadata are stored in the database for future reference and analysis. The output design emphasizes clarity, accuracy, and quick response to enhance the overall user experience.

4.4 Database Design

The database design for the proposed handwritten digit recognition system is kept simple, as the project uses only the MNIST dataset and a CNN model implemented in Google Colab. The database is used to store the prediction results generated by the trained CNN model. It maintains details such as the input image reference, predicted digit, confidence score, and prediction time.

A single-table structure is sufficient for this project since no user authentication or multilingual data handling is required. This design helps in analyzing the performance of the CNN model and maintaining records of digit predictions. The simple and efficient database structure ensures easy storage, retrieval, and evaluation of results obtained during handwritten digit classification.

5. IMPLEMENTATION AND RESULTS

Implementation

The handwritten digit recognition system was implemented using **Convolutional Neural Networks (CNNs)** to classify digits from 0 to 9. The implementation was carried out using the Python programming language along with deep learning frameworks such as TensorFlow and Keras. The MNIST dataset, which contains a large collection of labeled handwritten digit images, was used to train and test the model. Each image was resized to a fixed dimension and normalized to improve training efficiency.

The CNN architecture consists of multiple convolutional layers followed by pooling layers to extract important features from the input images. These layers are responsible for learning patterns such as edges, curves, and shapes that characterize handwritten digits. The extracted features are passed to fully connected layers, which perform classification. A softmax activation function is used in the output layer to predict the digit class with the highest probability. The model was trained using backpropagation and optimized using gradient-based optimization techniques to minimize classification error.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

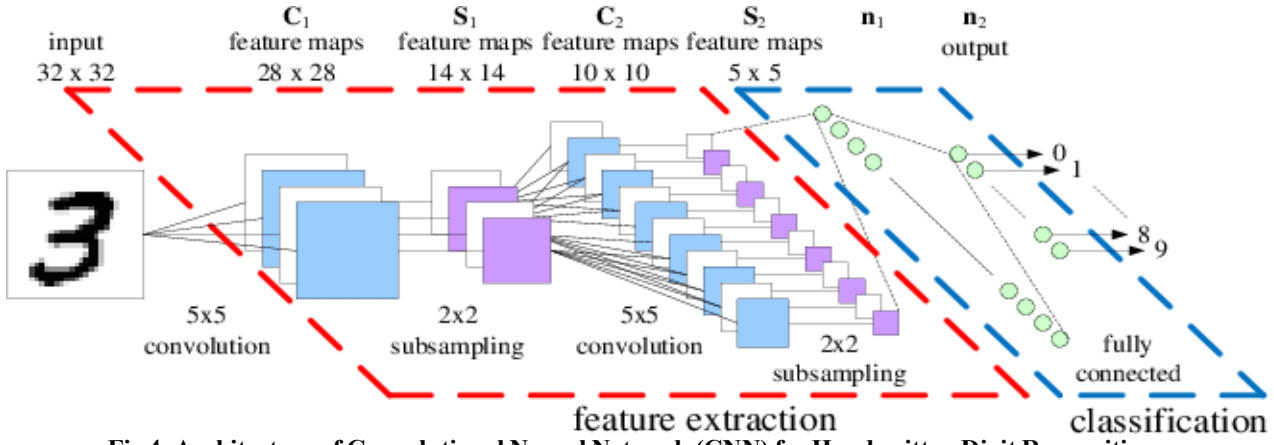

Fig 4: Architecture of Convolutional Neural Network (CNN) for Handwritten Digit Recognition

RESULTS

Parameter	Manual Method	Traditional Image- Based Method	Proposed CNN-Based System
Precision	Low	Moderate	High (95%–98%)
Recall	Low	Moderate	High
Prediction Accuracy	Low	80%–90%	95%–98%
Error Rate	High	Moderate	Very Low
Prediction Time	High	Moderate	Fast
Human Effort	High	Moderate	Low
Consistency	Low	Moderate	High
Scalability	Poor	Limited	Excellent
Adaptability to Field Conditions	Low	Moderate	High
Generalization Capability	Low	Moderate	Strong across different writing styles
Convergence Stability	Not Applicable	Moderate	Stable and Steady

The implemented CNN-based system achieved high recognition accuracy of approximately 95%– 98% on the test dataset. The model demonstrated strong generalization capability by accurately classifying handwritten digits with different writing styles, sizes, and orientations. Training and validation results showed a steady reduction in loss and a consistent increase in accuracy, indicating effective learning and stable convergence. The system successfully recognized most handwritten digit samples with very low misclassification rate. The results confirm that Convolutional Neural Networks (CNNs) are highly effective for handwritten digit recognition tasks. Compared to traditional machine learning methods, which typically achieve accuracy in the range of 80%–90%, the proposed CNN-based system provides significantly improved accuracy, faster processing, and better robustness to variations in handwriting. These outcomes clearly demonstrate the suitability of the CNN-based approach for real-world handwritten digit recognition applications

6.CONCLUSION

In this project, a Convolutional Neural Network (CNN)–based handwritten digit recognition system was successfully developed and implemented using the MNIST dataset. The model was trained and tested in a Google Colab environment, where preprocessing techniques such as normalization and resizing significantly improved recognition accuracy and reduced computational complexity. Data cleaning and structured input formatting ensured that the images were consistent and suitable for deep learning model training. The CNN architecture consisted of multiple convolutional layers, activation functions, pooling layers, and fully connected dense layers, which enabled effective feature extraction and classification. The use of activation functions such as ReLU enhanced non-linearity, while pooling layers reduced dimensionality and prevented overfitting. Additionally, the model was optimized using appropriate loss functions and optimizers to achieve stable convergence during training.The trained CNN effectively classified handwritten digits from 0 to 9 with high reliability and consistency. The experimental results validate the effectiveness of deep learning approaches for image-based classification problems. The proposed system is simple, efficient, scalable, and easy to implement, making it a strong foundation for real-world applications such as document digitization, automated form processing, intelligent data entry systems, banking check verification, postal code recognition, and educational tools. Furthermore, the system can be extended by incorporating larger datasets, real- time input processing, or advanced architectures for improved performance.

7.FUTURE ENHANCEMENT

In the future, the system can be extended to support handwritten character recognition for multiple languages such as Tamil and Hindi, enabling multilingual processing capabilities. By training the model on diverse and larger datasets containing regional scripts and complex character patterns, the system can become more versatile and culturally adaptable. This would significantly enhance its usability in government offices, educational institutions, and multilingual documentation systems. Advanced CNN architectures, data augmentation techniques, and real-time webcam input can be integrated to

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

improve accuracy, robustness, and practical usability in real-world applications. Data augmentation methods such as rotation, scaling, shifting, and noise addition can help the model generalize better to variations in handwriting styles The system can be further enhanced by incorporating deeper and more powerful architectures such as ResNet, EfficientNet, or MobileNet to achieve higher accuracy and faster convergence. Transfer learning techniques can also be applied to leverage pre-trained models, reducing training time while improving performance. Hyperparameter tuning and regularization strategies can be implemented to minimize overfitting and improve model stability.

REFERENCES

Online References

1. *Y. LeCun and C. Cortes, "MNIST handwritten digit database," Yann LeCun's Official Website,2020. [Online]. Available: https://yann.lecun.com/exdb/mnist/*
2. *Wikipedia Contributors, "Convolutional neural network," Wikipedia, The Free Encyclopedia, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Convolutional_neural_network*
3. *Google LLC and TensorFlow Team, "MNIST classification tutorial," TensorFlow Official Documentation, 2022. [Online]. Available: https://www.tensorflow.org/tutorials/quickstart/beginner*
4. *F. Chollet and Keras Team, "Keras convolutional neural network documentation," Keras Official Guide, 2022. [Online]. Available: https://keras.io/guides/*
5. *Kaggle Inc. and Google LLC, "Digit recognizer: MNIST dataset," Kaggle Competition Platform,2021. [Online]. Available: https://www.kaggle.com/c/digit-recognizer*
6. *Analytics Vidhya and A. Gupta, "Handwritten Digit Recognition using CNN," Analytics Vidhya Blog, 2021.[Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/a-comprehensive-guide-to-handwritten-digit-recognition-using-cnn/*

Book References

1. *Goodfellow and Y. Bengio, *Deep Learning*, MIT Press, 2020.*
2. *F. Chollet and J. J. Allaire, *Deep Learning with R*, Manning Publications, 2022.*
3. *F. Chollet and J. Andrews, *Deep Learning with Python*, 2nd ed., Manning Publications, 2021.*
4. *B. S. Atal and S. Mandal, *Machine Learning and Deep Learning for Image Recognition*, CRC Press, 2023.*

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

APPENDIX

1.1 Source Code

```
cimport os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

from flask import Flask, render_template,
request from tensorflow.keras.models import
load_model from tensorflow.keras.layers import
Dense import numpy as np
from PIL import

Image app = Flask(_

name__)

# Fix for quantization_config error
def Dense_ignore_quantization(*args,
    **kwargs): kwargs.pop('quantization_config',
    None) return Dense(*args, **kwargs)

# Load model
model                =
    load_model(
    "mnist_model.h
    5",
    compile=False,
    custom_objects={'Dense': Dense_ignore_quantization}
)

# Language
mappings mappings
= {
    "English": {0: "A", 1: "B", 2: "C", 3: "D", 4: "E", 5: "F", 6: "G", 7: "H", 8: "I", 9: "J"},
    "Hindi":   {0: "अ", 1: "आ", 2: "इ", 3: "ई", 4: "उ", 5: "ऊ", 6: "ए", 7: "ऐ", 8: "ओ", 9: "औ"}
}

def preprocess_image(file):
    img =
    Image.open(file).convert("L") img
    = img.resize((28, 28))
    img =
    np.array(img)
    img = 255 - img
    img = img /
    255.0
    img = img.reshape(1, 28, 28, 1)
    return img

@app.route("/", methods=["GET", "POST"])
def index():
    result = None
    if request.method == "POST":
        language =
        request.form.get("language") file =
```

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

```
        request.files.get("image")

        if file and language in
            mappings: img =
            preprocess_image(file)

            predictions = model.predict(img)

            confidence = float(np.max(predictions)) * 100
            index = int(np.argmax(predictions))

            letter = mappings[language][index]

            result = {
                "language": language,
                "predicted_letter": letter,
                "mapped_number": index,
                "confidence": round(confidence, 2)
            }

    return render_template("index.html",

 result=result) if __name_____== "_main_":
    app.run(host="0.0.0.0", port=8000, debug=False)


import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist

print("Loading

dataset...") # Load

MNIST dataset
(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.reshape(-1,28,28,1) / 255.0
x_test = x_test.reshape(-1,28,28,1) / 255.0

print("Building

model...") # Build

CNN
model = models.Sequential([
    layers.Input(shape=(28,28,1)),
    layers.Conv2D(32,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(64,activation='relu'),
```

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

```
    layers.Dense(10,activation='softmax')
])

model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)
      print("Training started...")

      model.fit(x_train, y_train, epochs=5)

      print("Saving model...")

      model.save("model.h5")

      print("✅ Model saved successfully!")
```

HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Multi Language Letter Recognition</title>
</head>
<body>
    <h2>Handwritten Letter Recognition</h2>

    <form method="POST" enctype="multipart/form-data">
        <label>Select Language:</label>
        <select name="language">
            <option value="English">English</option>
            <option value="Hindi">Hindi</option>
        </select>
        <br><br>

        <input type="file" name="image" required>
        <br><br>

        <button type="submit">Predict</button>
    </form>

    {% if result %}
        <h3>Result:</h3>
        <p><b>Language:</b> {{ result.language }}</p>
        <p><b>Predicted Letter:</b> {{ result.predicted_letter }}</p>
        <p><b>Mapped Number:</b> {{ result.mapped_number }}</p>
        <p><b>Confidence:</b> {{ result.confidence }} %</p>
    {% endif %}

</body>
</html>
```

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

Screenshots

Fig 5: Input Image Upload and Digit Prediction Interface

Fig 6: Letter Recognition Prediction Page

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 3| March2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

Fig 7: Handwritten Character

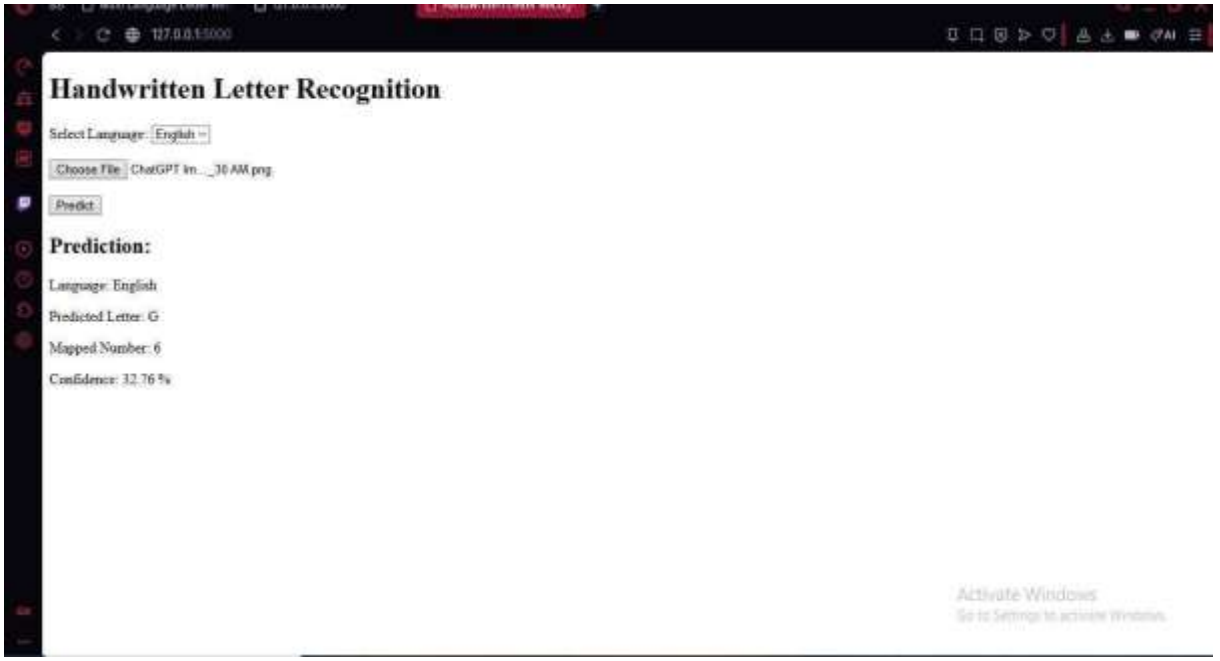

Fig 8: Web-based Handwritten Digit Recognition Interface using CNN