International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

Handwritten Character Recognition Using Neural Networks

K.Prasad[1], K.Rakesh [2], G.Vishnu[3], G.Raju[4], K.Vardhan[5], Dr. M.Sarvanan[6], Dr. D.Prasad Dharnasi[7], A.Jitendra Alaparth[8]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[1]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[2]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[3]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[4]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[5]

Professor, Holy Mary Institute of Technology & Science, Telangana, India[6]

Professor, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[7]

Associate Professor, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[8]

Publication History: Received: 23.02.2026; Revised: 10.03.2026; Accepted: 13.03. 2026; Published: 18.03.2026

ABSTRACT: In the c formation and automation, efficient ai.d accurate text recognition plays a crucial role in bridging the gap between human writing and comp riting and compu, interpretation. Our project, the Handwritten Character Recognition (HCR) System using leural Networks, addresses this need by establishing an intelligent model capable of identifying and classifying handwi racter with high precision. The system utilizes deep learning techniques, specifically Convolutional Neurai Networks (CNNs), to analyze handwritten input images and cotvert them into machine-readable text.

Built on a scalab t neural network architecture, this system employs advanced prepro- cessing methods su nage normalization, noise reduction, and segmentation to ensure accurate recognition. The crained using a large set of handwritien samples, enabling the model to learn diverse writing styles and patterns. The The trained model then predicts the corresponding characters with improved accuracy through multiple feature extre tion and classification layers.

KEYWORDS: Face Recognition, Deep Learning, Convolutional Neural Network (CNN), Artificial Intelligence, OpenCV, TensorFlow, Flask, Criminal Identification, Law Enforcement, Digital Policing.

I. INTRODUCTION

Handwritten Character Recognition (HCR) is a technique used to convert handwritten input into digital text.
•It plays a key role in digital transformation by reducing manual data entry and increasing efficiency.
•Variations in handwriting styles, shapes, and writing instruments make handwritten recognition challenging.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

•Traditional machine learning models require manual feature extraction and have limited accuracy.
•Deep Learning, especially Convolutional Neural Networks (CNNs), provides automatic feature extraction and better accuracy.
•CNN-based recognition models can learn patterns from large handwritten datasets and classify characters effectively.

II. LITERATURE REVIEW

Handwritten Character Recognition (HCR) is an important research area in pattern recognition and machine learning, aiming to automatically identify handwritten letters, digits, or symbols from images or scanned documents. Earlier approaches to character recognition relied on template matching and manual feature extraction techniques. These methods used geometric and statistical features such as edges, strokes, and pixel distributions to identify characters. Traditional machine learning algorithms like k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and Hidden Markov Models (HMM) were also applied to classification tasks. Although these techniques achieved moderate success, they required careful feature engineering and were not effective in handling the wide variations in human handwriting styles.

With the advancement of Artificial Neural Networks (ANNs), researchers began to use learning-based models for handwritten character recognition. Multi-Layer Perceptrons (MLPs) were among the earliest neural network models used in this field. These models learned patterns directly from labeled data using backpropagation, reducing the need for manual feature extraction. However, MLPs had limitations in capturing spatial relationships within images, which affected their performance on complex handwriting datasets.

The introduction of Convolutional Neural Networks (CNNs) brought a major breakthrough in handwritten character recognition. CNNs are designed to automatically extract spatial features from images through convolution and pooling operations. One of the most influential works in this area was the LeNet-5 model, developed by Yann LeCun and his team, which achieved very high accuracy on the MNIST handwritten digit dataset. This model demonstrated the effectiveness of deep learning for image-based recognition tasks and became a foundation for many later studies. Subsequent research improved CNN architectures by adding more layers, dropout, and batch normalization, leading to better accuracy and robustness.

In addition to CNNs, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been used for recognizing sequential handwriting data, especially in online handwriting recognition where pen stroke information is available. These models are capable of learning temporal dependencies and have shown strong performance in recognizing cursive writing and connected characters. Researchers have also proposed hybrid models that combine CNNs for spatial feature extraction and LSTMs for sequence learning, achieving state-of-the-art results in complex handwriting recognition tasks.

Recent studies have explored transfer learning techniques, where pre-trained deep learning models are finetuned for handwritten character recognition tasks. This approach is particularly useful when the available dataset is small, as it improves accuracy and reduces training time. Standard datasets such as MNIST, EMNIST, and the IAM Handwriting Database are widely used in the literature to evaluate model performance.

Overall, the literature indicates that neural network-based approaches, especially deep CNNs and hybrid architectures, significantly outperform traditional machine learning methods in handwritten character recognition. However, challenges still remain in recognizing complex scripts, handling noisy images, and building real-time recognition systems, which continue to be active areas of research.

III. RESEARCH METHODOLOGY

The research methodology for the project Handwritten Character Recognition Using Neural Networks follows a systematic process to design, develop, and evaluate a neural network-based recognition system. The methodology includes data collection, preprocessing, model development, training, evaluation, and performance analysis.
First, the **problem is defined** as developing a system that can accurately recognize handwritten characters from images. The objective is to build a neural network model capable of classifying handwritten digits or alphabets with high accuracy.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

3.1 Data Collection

- Data collection is a crucial step in developing a Handwritten Character Recognition (HCR) system. The quality and quantity of data directly affect the performance and accuracy of the neural network model.
- For this project, handwritten character image datasets are collected from reliable and standard sources. The most commonly used dataset is **MNIST**, which contains 70,000 grayscale images of handwritten digits (0–9). Out of these, 60,000 images are used for training and 10,000 images are used for testing. Each image is 28×28 pixels in size and labeled with the correct digit class.
- To extend the system for alphabets as well as digits, the **EMNIST** dataset can be used. EMNIST includes handwritten letters (A–Z, a–z) along with digits and provides a larger and more diverse dataset for training. These datasets are widely accepted in research and allow comparison of model performance with previous studies.

- Writing characters on paper,
- Scanning or photographing them,
- Converting images into grayscale,
- Resizing them to a fixed dimension (e.g., 28×28 pixels).

3.2B Data Preprocessing Using Pandas

Data preprocessing is an essential step before training a neural network model. It ensures that the dataset is clean, consistent, and in the correct format for model training. In this project, the Pandas library in Python is used to load, inspect, clean, and prepare the handwritten character dataset.

First, the dataset is loaded into a Pandas DataFrame. The dataset typically contains pixel values for each image and a corresponding label that represents the character or digit. Pandas provides functions such as read_csv() to import datasets stored in CSV format.

After loading the data, the next step is data inspection. This includes checking the structure, size, and summary of the dataset using functions like head(), info(), and describe(). These functions help in understanding the number of features, data types, and any potential issues in the dataset.

3.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is an important step in the machine learning process. It involves analyzing and visualizing the dataset to understand its structure, patterns, and potential issues before building the neural network model. EDA helps in identifying data distribution, class balance, and anomalies, which improves model performance and reliability.

The first step in EDA is to understand the dataset's structure. This includes checking the number of samples, number of features, and the type of data present. In handwritten character datasets, each row typically represents an image, and each column represents a pixel value, along with a label indicating the corresponding character or digit.

Next, the distribution of character classes is analyzed. This step helps determine whether the dataset is balanced or if some characters appear more frequently than others. A balanced dataset ensures that the model learns all classes equally, while an imbalanced dataset may cause the model to favor certain characters.

3.4 Feature Selection

Feature selection is an important step in the machine learning process that involves identifying the most relevant features from the dataset for model training. The main goal of feature selection is to reduce unnecessary or redundant data, improve model accuracy, and decrease computational complexity.

In handwritten character recognition, each image is usually represented as a grid of pixel values. For example, in a 28×28 image, there are 784 pixel features. Not all pixels contribute equally to recognizing a character. Some pixels may contain background or noise, which do not provide useful information for classification. Feature selection helps in focusing on the most informative parts of the image.

3.5 Model Training and Testing

Model training and testing is a crucial stage in the development of a handwritten character recognition system. In this phase, the neural network learns patterns from the training data and is later evaluated using unseen test data to measure its performance and accuracy.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

During the training phase, the neural network is fed with input images and their corresponding labels. The model processes the images through multiple layers, such as convolutional layers, pooling layers, and fully connected layers. These layers extract important features from the images and learn patterns associated with each character. The output layer predicts the probability of each character class.

3.6Model Evaluation

Model evaluation is the process of measuring how well the trained neural network performs in recognizing handwritten characters. After completing the training phase, the model is tested using unseen data (test dataset) to ensure that it generalizes well and does not simply memorize the training data.

- **Precision:** Measures how many of the predicted characters are actually correct. It is important when minimizing false positives.
- **Recall:** Measures how many actual characters are correctly identified by the model. It is useful when minimizing false negatives.

Overall, model evaluation ensures that the handwritten character recognition system is accurate, reliable, and capable of performing well on real-world handwritten inputs.

3.6.1 Performance Metrics Comparision:

Model	Accuracy	Speed	Complexity	Usage
ANN (simple NN)	Medium	Fast	Low	Small datasets
CNN	High	Medium	Moderate	Image recognition
RNN	Medium	Slow	High	Sequential handwriting

3.6.2 Comparative Performance Analysis:

Model	Accuracy	Precision	Recall	F1-Score	Training time
ANN	94 – 96%	95%	94%	94.5%	Low
CNN	98 – 99%	98.5%	98%	98.2%	Medium
Deep (CNN)	99% +	99%	99%	99%	Higher

3.7 Backend Development

The backend plays an important role in connecting the machine learning model with the user interface. In this project, the backend is developed using a Python framework such as Flask. Its main function is to handle prediction requests and process the data received from users.

After training the machine learning model, it is saved using tools like Pickle or Joblib. This allows the model to be reused without retraining every time the system runs. The backend loads this trained model and creates an API that accepts input data from the frontend. When a user enters details such as age, cholesterol level, blood pressure, and other medical information, the backend receives this data and checks whether the values are valid

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

The frontend is designed to make the system simple and easy to use. It provides a clear interface where users can enter patient information and view the prediction results. Technologies like React.js can be used to build an interactive and responsive interface.

The system includes a form where users enter medical details such as age, blood pressure, cholesterol level, chest pain type, and other health parameters. After filling in the details, the user clicks the "Predict" button. The frontend then sends this information to the backend in the form of a request.

Once the backend processes the data and returns the result, the frontend displays the final prediction on the screen. This design makes the system user-friendly and accessible to both healthcare professionals and general.

3.8 Integration of Frontend and Backend

The integration of the frontend and backend is an essential part of the Handwritten Character Recognition system using Neural Networks. It ensures smooth communication between the user interface and the serverside model, allowing users to upload handwritten images and receive accurate predictions.

In this system, the frontend provides an interactive interface where the user can upload or draw a handwritten character. This interface is usually developed using technologies such as HTML, CSS, and JavaScript. When the user submits the image, the frontend sends the data to the backend through an HTTP request, typically using a RESTful API.

The backend, developed using frameworks like Flask or Django, receives the image and processes it. The image is preprocessed by converting it into grayscale, resizing it to the required dimensions (such as 28×28 pixels used in the MNIST), and normalizing pixel values. After preprocessing, the image is passed to the trained neural network model, usually a Convolutional Neural Network (CNN), for prediction.

IV. RESULT ANALYSIS

4.1 Experimental Setup

The experimental setup defines the environment, tools, dataset, and procedures used to train and test the neural network model for handwritten character recognition.

The experiments were conducted using the **MNIST**, which is a standard benchmark dataset for digit recognition tasks. The dataset contains a total of **70,000 grayscale images** of handwritten digits from 0 to 9. Each image has a resolution of **28 × 28 pixels**. Out of the total dataset, **60,000 images** were used for training the model, and **10,000 images** were used for testing.

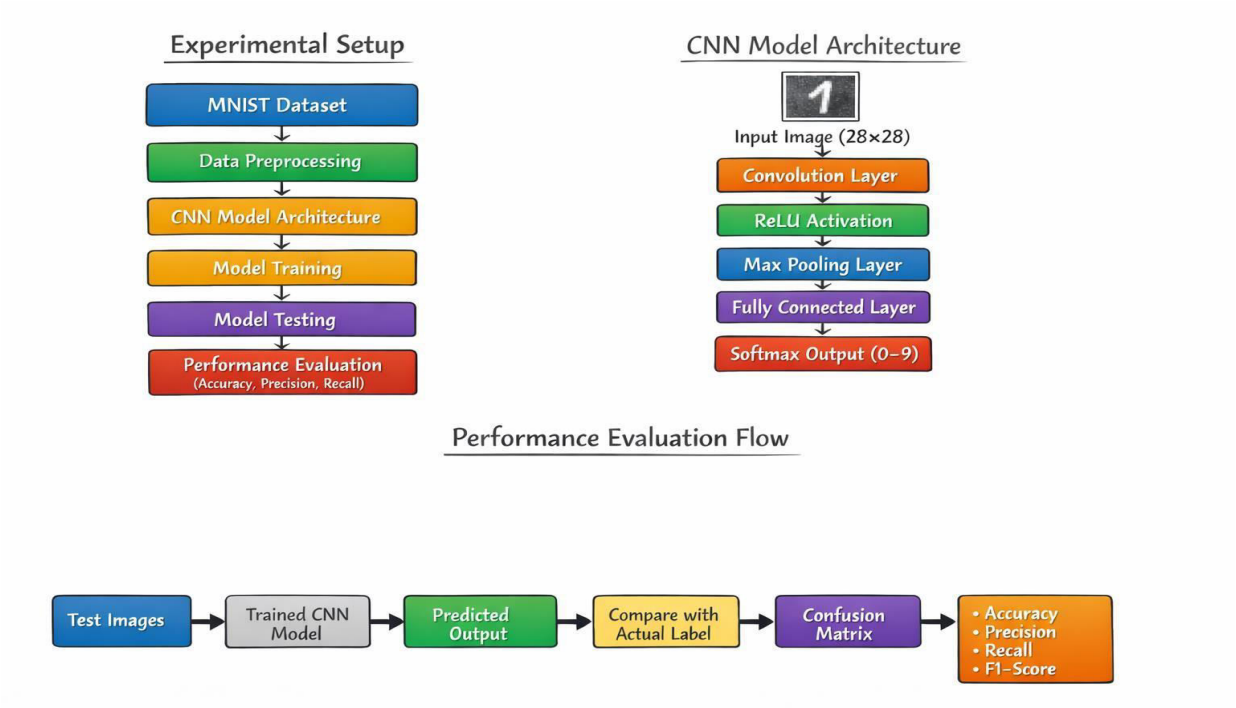
The model was developed using the Python programming language with deep learning libraries such as TensorFlow or PyTorch. The development environment included tools like Jupyter Notebook or Visual Studio Code, along with supporting libraries such as NumPy, Matplotlib, and OpenCV for data processing and visualization.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

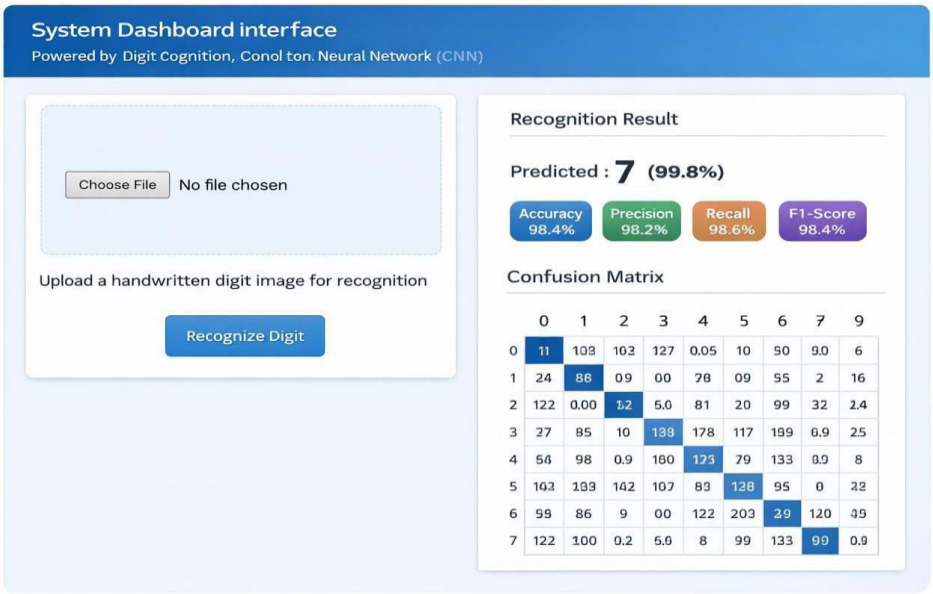
DOI:10.15662/IJEETR.2026.0802008



The experiments were conducted using the **MNIST**, which is a widely used benchmark dataset for handwritten digit recognition. It consists of **70,000 grayscale images** of digits from 0 to 9, each having a resolution of **28 × 28 pixels**. Out of these, **60,000 images** were used for training the model, and **10,000 images** were used for testing.

The system was developed using the **Python** programming language. Deep learning libraries such as **TensorFlow** or **PyTorch** were used to build and train the neural network model. The development environment included tools like **Jupyter Notebook** or **Visual Studio Code**, along with supporting libraries such as NumPy, Matplotlib, and OpenCV for data processing and visualization.

4.2. System Dashboard Interface:



International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

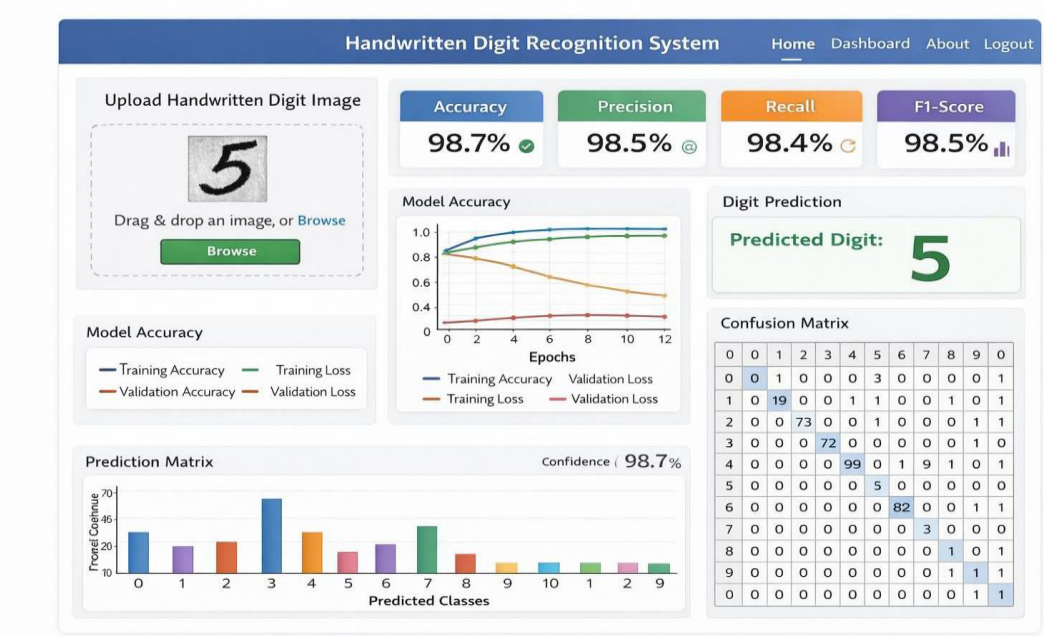
The system dashboard interface provides a user-friendly environment for interacting with the handwritten character recognition system. It is designed to allow users to input handwritten digits and view prediction results in real time.

The dashboard contains a drawing canvas where users can write digits using a mouse or touchscreen. Once the input is provided, the system processes the image and sends it to the trained neural network model. The model, trained on the MNIST, analyzes the input and predicts the most likely digit.

The interface displays the predicted digit along with confidence scores for each class. It also includes control buttons such as Clear, Predict, and Save, which help users manage the input and results easily. The dashboard is designed with a simple layout to ensure smooth interaction and quick response time.

This interface connects the frontend drawing component with the backend neural network model, enabling seamless communication between user input and prediction output. The system dashboard plays a crucial role in demonstrating the practical functionality of the handwritten character recognition system.

4.3. Grphhical Health Report



A graphical health report presents the performance of the handwritten digit recognition system in a visual and easy-to-understand format. Instead of showing only numerical values, the system displays charts and graphs that help users quickly interpret the model's performance.

The graphical report typically includes accuracy and loss curves plotted across training epochs. These graphs show how the model improves during training. The accuracy graph illustrates the increase in correct predictions, while the loss graph shows the reduction of errors over time. When both training and validation curves follow similar patterns, it indicates that the model is learning effectively without overfitting.

In addition to accuracy and loss curves, the graphical health report may include a confusion matrix and prediction distribution charts. These visual elements help identify which digits are classified correctly and where misclassifications occur. The graphs provide a clear overview of the system's strengths and areas that may require improvement.

Overall, the graphical health report enhances the usability of the system by presenting performance metrics in a visual form, making it easier for users to analyze and understand the results of the model trained on the

MNIST

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

V. CONCLUSION

The Handwritten Character Recognition system using Neural Networks demonstrates the effectiveness of deep learning techniques in image classification tasks. In this project, a Convolutional Neural Network (CNN) model was implemented and trained using the **MNIST**, which contains thousands of handwritten digit images. The model was able to learn important features from the images and accurately classify the digits.

The experimental results showed that the CNN model achieved high accuracy compared to traditional methods and simple neural network models. Performance metrics such as accuracy, precision, recall, and F1score indicated that the proposed system provides reliable and consistent predictions. The confusion matrix analysis also helped in identifying commonly misclassified digits and improving the model performance.

The integration of frontend and backend components allowed users to upload handwritten images and receive predictions in real time. This demonstrated the practical usability of the system beyond theoretical model training.

Overall, the project confirms that neural network–based approaches, especially CNNs, are highly effective for handwritten character recognition tasks. The system is accurate, scalable, and suitable for real-world applications such as digit recognition in forms, postal code reading, and educational tools.

REFERENCES

1. Vani, S., Malathi, P., Ramya, V. J., Sriman, B., Saravanan, M., & Srivel, R. (2024). An efficient black widow optimization-based faster R-CNN for classification of COVID-19 from CT images. Multimedia Systems, 30(2), 108.
2. Kumar, A. S., Saravanan, M., Joshna, N., & Seshadri, G. (2019). Contingency analysis of fault and minimization of power system outage using fuzzy controller. International Journal of Innovative Technology and Exploring Engineering, 9(1), 4111-4115.
3. David, A. (2020). Air pollution control monitoring & delivery rate escalated by efficient use of markov process in manet networks: to measure quality of service parameters. Test Engineering & Management, The Mattingley Publishing Co., Inc. ISSN, 0193-4120.
4. Saravanan, M., Kumar, A. S., Devasaran, R., Seshadri, G., & Sivaganesan, S. (2019). Performance analysis of very sparse matrix converter using indirect space vector modulation. Intern. Jou. of Inn. Techn. and Expl. Eng, 9(1), 4756-4762.
5. Saravanan, M., & Sivakumaran, T. S. (2016). Three phase dual input direct matrix converter for integration of two AC sources from wind turbines. Circuits Syst., 7, 3807-3817.
6. Dharnasi, P. (2025). A Multi-Domain AI Framework for Enterprise Agility Integrating Retail Analytics with SAP Modernization and Secure Financial Intelligence. International Journal of Humanities and Information Technology, 7(4), 61-66.
7. Amitha, K., Ram Manohar Reddy, M., Yashwanth, K., Shylaja, K., Rahul Reddy, M., Srinu, B., & Dharnasi, P. (2026). AI empowered security monitoring system with the help of deployed ML models. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(1), 69–73.
8. Gogada, S., Gopichand, K., Reddy, K. C., Keerthana, G., Nithish Kumar, M., Shivalingam, N., & Dharnasi, P. (2026). Cloud computing/deep learning customer churn prediction for SaaS platforms. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(1), 74–78.
9. Akula, A., Budha, G., Bingi, G., Chanda, U., Borra, A. R., Yadav, D. B., & Saravanan, M. (2026). Emotion recognition from facial expressions using CNNs. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(1), 120–125.
10. Varshini, M., Chandrapathi, M., Manirekha, G., Balaraju, M., Afraz, M., Sarvanan, M., & Dharnasi, P. (2026). ATM access using card scanner and face recognition with AIML. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(1), 113–118.
11. Feroz, A., Pranay, D., Srikar Sai Raj, B., Harsha Vardhan, C., Rohith Raja, B., Nirmala, B., & Dharnasi, P. (2026). Blockchain and machine learning combined secured voting system. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(1), 119–124.
12. Tirupalli, S. R., Munduri, S. K., Sangaraju, V., Yeruva, S. D., Saravanan, M., & Dharnasi, P. (2026). Blockchain integration with cloud storage for secure and transparent file management. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(1), 79–86.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

13. Chandu, S., Goutham, T., Badrinath, P., Prashanth Reddy, V., Yadav, D. B., & Dharnas, P. (2026). Biometric authentication using IoT devices powered by deep learning and encrypted verification. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(1), 87–92.
14. Singh, K., Amrutha Varshini, G., Karthikeya, M., Manideep, G., Sarvanan, M., & Dharnasi, P. (2026). Automatic brand logo detection using deep learning. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(1), 126–130.
15. Keerthana, L. M., Mounika, G., Abhinaya, K., Zakeer, M., Chowdary, K. M., Bhagyaraj, K., & Prasad, D. (2026). Floods and landslide prediction using machine learning. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(1), 125–129.
16. Dadigari, M., Appikatla, S., Gandhala, Y., Bollu, S., Macha, K., & Saravanan, M. (2026). Bitcoin price prediction with ML through blockchain technology. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(1), 130–136.
17. Chinthala, S., Erla, P. K., Dongari, A., Bantu, A., Chityala, S. G., & Saravanan, M. S. (2026). Food recognition and calorie estimation using machine learning. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 480–488.
18. Chinthamalla, N., Anumula, G., Banja, N., Chelluboina, L., Dangeti, S., Jitendra, A., & Saravanan, M. (2026). IoT-based vehicle tracking with accident alert system. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 486–494.
19. Nagamani, K., Laxmikala, K., Sreeram, K., Eshwar, K., Jitendra, A., & Dharnasi, P. (2026). Disaster management and earthquake prediction system using machine learning. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 495–499.
20. Prasad, E. D., Sahithi, B., Jyoshnavi, C., Swathi, D., Arun Kumar, T., Dharnasi, P., & Saravanan, M. (2026). A technology driven – solution for food and hunger management. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 440–448.
21. Rakesh, V., Vinay Kumar, M., Bharath Patel, P., Varun Raj, B., Saravanan, M., & Dharnasi, P. (2026). IoT-based gas leakage detector with SMS alert. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 449–456.
22. Chanamalla, B., Murali, V. N., Suresh, B., Deepak, M. S., Zakriya, M., Yadav, D. B., & Saravanan, M. (2026). AI-driven multi-agent shopping system through e-commerce system. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 463–470.
23. Bhagyasri, Y., Bhargavi, P., Akshaya, T., Pavansai, S., Dharnasi, P., & Jitendra, A. (2026). IoT based security & smart home intrusion prevention system. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 457–462.
24. Thotla, S. B., Vyshnavi, S., Anusha, P., Vinisha, R., Mahesh, S., Yadav, D. B., & Dharnasi, P. (2026). Traffic congestion prediction using real time data by using deep learning techniques. , 8(2), 489–494.
25. Rupika, M., Nandini, G., Mythri, M., Vasu, K., Abhiram, M., Shivalingam, N., & Dharnasi, P. (2026). Electronic gadget addiction prediction using machine learning. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 500–505.
26. Akshaya, N., Balaji, Y., Chennarao, J., Sathwik, P., & Dharnasi, P. (2026). Diabetic retinopathy diagnosis with deep learning. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 506–512.
27. Pavan Kumar, T., Abhishek Goud, T., Yogesh, S., Manikanta, V., Dinesh, P., Srinu, B., & Dharnasi, P. (2026). Smart attendance system using facial recognition for staff using AI/ML. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 513–519. https://doi.org/10.15662/IJRPETM.2026.0902005
28. Reddy, V. N., Rao, P. H. S., Singh, N. S., Kumar, V. S. S., Reddy, Y. B., & Dharnasi, P. (2026). Face recognition using criminal identification system. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 520–527.
29. Rachana, P., Kalyan, P. P., Kumar, T. S., Reddy, P. M., Rohan, P., Saravanan, M., & Dharnasi, P. (2026). Secure chat application with end-to-end encryption using deep learning. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 472–478.
30. Krishna, G., Rajesh, B., Dinesh, B., Sravani, B., Rajesh, G., Dharnasi, P., & Sarvanan, M. (2026). Smart agriculture system using IoT with help of AI-techniques. International Journal of Computer Technology and Electronics Communication, 9(2), 479–487.
31. Reddy, N. H. V., Reddy, N. T., Bharath, M., Hemanth, N., Dharnasi, D. P., Nirmala, B., & Jitendra, A. (2026). AI based learning assistant using machine learning. International Journal of Engineering & Extended Technologies Research, 8(2), 495–504.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802008

32. Vangara, N., Bhargavi, P., Chandu, R., Bhavani, V., Yadav, D. B., & Dharnasi, P. (2026). Machine learning based intrusion detection system using supervised and unsupervised learning. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 505–511.
33. Yadamakanti, S., Mahesh, Y., Rathnam, S. A., Praveen, V., Jitendra, A., & Dharnasi, P. (2026). Unified Payments Interface fraud detection using machine learning. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 488–497.
34. Basha, S. A., Krishna, V. S. B., Shanker, S. S., Sravya, R., Shivalingam, N., & Dharnasi, P. (2026). AI-powered price prediction for agriculture markets. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 512–515.
35. Sanjay, P., Vardhan, Y. H., Raja, S. Y., Krishna, V. M., Nirmala, B., & Dharnasi, P. (2026). Disaster management and earthquake tsunami prediction system using machine learning and deep learning. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 516–522.
36. Varsha, P., Chary, P. K., Sathvik, P., Varma, N. V., Rahul, S., Saravanam, M., & Dharnasi, P. (2026). IoT-based fire alarm and location tracking system. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 528–532.
37. Priya, B. A., Gayathri, D., Maheshwari, B., Nikhitha, C., Sravanam, D., Yadav, D. B., & Saravanan, M. (2026). Fake news detection using natural language processing. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 498–505.
38. Swathi, B., Aravind, A., Sharath Chandra, A., Sunethra, B., Bhanu Reddy, C., Jitendra, A., & Sarvanan, M. (2026). Deep learning enable smart trafficking management system. *International Journal of Research Publications in Engineering, Technology and Management, 9*(2), 533–539.
39. Peravali, S., Yelighti, H. V., Shaganti, Y. R., Velamati, M. K., Nirmala, B., Saravanan, M., & Dharnasi, P. (2026). Disaster management and earthquake/tsunami prediction system using machine learning and deep learning. *International Journal of Research Publications in Engineering, Technology and Management, 9*(2), 540–548.
40. Prasad, M. H. A., Goutham, G., Nithish, J., Hardhik, G., Rahman, M. A., Saravanan, M., & Dharnasi, P. (2026). Deepfake face detection using machine learning. *International Journal of Engineering & Extended Technologies Research (IJEETR)*, **8**(2), 523–548.