Grenze International Journal of Engineering and Technology, January Issue

Recognizing using Deep Learning and Machine Learning to Handwrite Text

Mutloori Prashanthi[1] and Mrutyunjaya. S. Yalawar[2]
[1]M. Tech Scholar, Department of Computer Sciences and Engineering, CMR Engineering College, Hyderabad, Telangana, India
Email: prashanthibaby@gmail.com
[2]Associate Professor, Department of Computer Sciences and Engineering, CMR Engineering College, Hyderabad, Telangana, India

***Abstract*— Traditional handwriting recognition methods have relied on manually crafted features and extensive prior knowledge, making Optical Character Recognition (OCR) systems challenging to train. However, recent advancements in deep learning have significantly improved handwriting recognition performance. With the exponential growth of handwritten data and enhanced processing capabilities, ongoing improvements in accuracy are crucial. The intricate structural details of handwritten characters can be captured by Convolutional Neural Networks (CNNs) with remarkable success, enabling automatic feature extraction essential for recognition tasks. Our study explores various CNN design configurations, including layer depth, kernel size, padding, receptive field, and stride size, and dilation, to optimize handwritten digit recognition. Additionally, we evaluate different Stochastic Gradient Descent (SGD) optimization algorithms to enhance performance. While ensemble models have traditionally achieved superior recognition accuracy, our goal aims to achieve similar outcomes with a CNN architecture that is pure, reducing computational complexity and testing costs. Through extensive experimentation and fine-tuning of learning parameters, we achieved a new benchmark accuracy of 99.87% on the MNIST handwritten digit dataset.**

***Index Terms*— Handwriting recognition, Optical Character Recognition (OCR), DL techniques, CNN.**

I. INTRODUCTION

Handwriting recognition contributes significantly to digitization by converting handwritten text into machine-readable formats. This technology is widely used in applications such as scanning of the Cheque Truncation System (CTS), sorting mail letters, identifying license plates on cars, and preserving historical documents, where accuracy, efficiency, and reliability are essential [16]. Deep neural architectures, Over conventional shallow models, Convolutional Neural Networks (CNNs) in particular are favored because of their capacity to automatically extract features from data without manual intervention. CNNs, a specialized form of deep neural networks, are extensively utilized in the fields of facial recognition, signal processing, natural language processing, object recognition, and picture classification. Their hierarchical feature learning mechanism allows them to process complex data efficiently, making them highly effective for various recognition tasks [1][2].
One type of Multi-Layer Perceptron (MLP) network is called a Convolutional Neural Network (CNN), were introduced in 1980, drawing inspiration from human visual perception.

Grenze ID: 01.GIJET.12.1.169

Similar to how humans recognize objects through repeated exposure, CNNs efficiently analyze visual data.[3][4] Prominent CNN architectures such as GoogLeNet, AlexNet, VGG, and ResNet integrate feature extraction and classification with minimal preprocessing, enabling them to automatically learn rich and interconnected features from images. Unlike MLPs, which struggle with high-definition pictures because of the curse of dimensionality, CNNs leverage topological information, making them robust to variations like rotation and translation [5]. They have been widely applied to using the MNIST dataset to recognize handwritten digits, attaining accuracy levels of up to 99%. While ensemble models enhance accuracy, they introduce higher computational costs and testing complexity. The proposed [6] work focuses on achieving comparable accuracy using a pure CNN architecture, optimizing learning parameters, and fine-tuning CNN design choices for MNIST digit recognition. This approach aims to outperform ensemble models in both recognition precision as well as computational effectiveness. Handwriting recognition plays an important part in digitizing historical documents and assisting individuals with disabilities in interacting with technology [7][8]. This project aims to enhance Using machine learning and deep learning to recognize handwritten writing techniques, addressing the challenges of varied handwriting styles, distortions, noise, and the lack of standardized fonts—issues that traditional OCR systems struggle with [17]. By leveraging advanced ML and DL models, the project seeks to improve accuracy and consistency while ensuring adaptability across different languages, scripts, and writing styles. Training on diverse datasets, combined with data augmentation and transfer learning, enhances how well the model generalizes across various handwriting patterns and real-world environments [6]. A key focus is user-friendliness and accessibility, incorporating features such as real-time transcription, batch processing, and integration with document management systems. These enhancements make the system more practical for a wide range of users and industries. Ultimately, this project aspires to advance handwritten text recognition by prioritizing accuracy, adaptability, and usability. By delivering a reliable and efficient recognition system, it aims to streamline digitization, improve productivity, and enhance accessibility across multiple sectors [7].

II. LITERATURE SURVEY

A. Recognition of Handwritten Digits Using Deep Learning and Machine Learning Techniques[1]

This project focuses on developing a system that combines deep learning and machine learning methods to recognize handwritten digits. It entails categorization, feature extraction, and picture preparation to enhance recognition accuracy.[9] The system integrates traditional ML algorithms used conjunction with deep learning models like Convolutional Neural Networks (CNNs) and Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN) to enhance performance. Designed for flexibility, scalability, and real-time inference, the project aims to maximize precision and robustness in digit recognition. The authors contribute by implementing, experimenting, and evaluating multiple algorithms to determine their effectiveness in accurately identifying handwritten digits.[10]

This project focuses on developing a tensorflow is used in a deep learning system to recognize handwritten text. Under the direction of Kanithi Santosh Kumar, Jayaram Kinthali, Kowshik Kotha, and Yugandhar Manchala , the system is designed to accurately transcribe handwritten text from images.[11][12] By utilizing TensorFlow, a widely adopted deep learning framework, the project ensures efficient development and deployment of the recognition model. The methodology includes Classification, feature extraction, and picture preprocessing with deep neural networks to enhance accuracy. Through extensive training and experimentation, the system effectively recognizes various handwriting styles and languages. The project significantly contributes to handwritten text recognition technology, offering a scalable and adaptable solution for applications such as document digitization, data entry automation, and historical document preservation[13].

B. A Computationally Effective Pipeline for Handwritten Text Recognition on Entire Pages Offline[3]

In their 2020 study, Thomas Delteil and Jonathan Chung present a pipeline that is computationally effective for offline handwritten text recognition (HTR) that maintains high accuracy while optimizing resource usage. Their approach consists of multiple key stages: image preprocessing, text line segmentation, handwriting recognition, and language modeling. By optimizing each phase, the system achieves competitive performance with minimal computational overhead. Key features include adaptive binarization, robust text line segmentation using a modified U-Net architecture, and an efficient deep learning model for handwriting recognition[3]. The study also emphasizes leveraging GPU acceleration to enhance processing speed. Findings from experiments using reference datasets show the efficiency and real-world applicability of this method in document digitization, archival, and text transcription. Overall, this project offers a balanced solution that integrates computational efficiency and high recognition accuracy for offline handwritten text recognition on a whole page.[14]

C. Transformer-Based and RNN-Based Model Comparison for Handwritten Text Recognition

The usefulness of Transformer-based and Recurrent Neural Network (RNN)-based models for handwritten text recognition (HTR) is examined by L.R.B. Schomaker and M. Ameryan in their 2022 study. This research provides a comparative analysis of these architectures, evaluating key performance metrics such as accuracy, computational efficiency, and adaptability to diverse handwriting styles and languages. Through empirical evaluations and experimentation, the study highlights the strengths and limitations of each model type in HTR tasks. By examining their suitability for different applications, the research offers valuable insights for practitioners and researchers, guiding the development of more efficient and effective systems for the recognition of handwritten text.[13]

D. Using ConvNet and RNN for Deep Learning in Handwritten Text Recognition

Manisha Gupta's 2021 study, "Deep Learning for Handwritten Text Recognition (ConvNet & RNN)," explores the improvement of handwritten text recognition through the use of deep learning algorithms. Recurrent neural networks (RNN) and convolutional neural networks (ConvNet) are integrated in this project to accurately transcribe handwritten text [5]. ConvNets are utilized for feature extraction, identifying spatial patterns and structures within input images, while RNNs excel in processing sequential data, ensuring contextual understanding. By combining these two architectures, the project aims to overcome challenges associated with diverse handwriting styles and multiple languages. Gupta's research makes a notable contribution to optical character recognition (OCR) by improving the automated digitization of handwritten documents, streamlining data entry, and enhancing accessibility for individuals with disabilities. Through comprehensive experimentation and evaluation, In handwritten text recognition, the study shows how well deep learning models work. The results reveal the way for practical applications in document processing, archival digitization, and intelligent data extraction across various industries.[15]

III. METHODOLOGY

In the current system, data preprocessing primarily focuses on structured data. Although this step is a time-intensive component of the machine learning (ML) pipeline, automation in this domain remains limited. While existing preprocessing techniques are sufficient for structured data, further advancements are necessary to effectively handle unstructured data. By integrating data mining techniques, AutoML pipelines could bridge this gap, enabling models to learn from diverse internet sources. Additionally, in feature engineering, most methods have been designed specifically for supervised learning. However, due to the high specificity of datasets, AutoML frameworks should aim for greater generalization to efficiently process a wider variety of datasets. As a result, there is a growing shift toward incorporating unsupervised learning paradigms, enhancing the adaptability and flexibility of AutoML pipelines.

A. Proposed System

The proposed handwritten text recognition system intends to utilize both deep learning and machine learning's advantages techniques to enhance accuracy, efficiency, and robustness. This system follows a structured approach, beginning with preprocessing to improve input image quality, followed by feature extraction and classification utilizing both conventional machine learning techniques and cutting-edge deep learning architectures, such as Transformer-based models, Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs). In order to reduce mistakes and improve overall performance, the system prioritizes the use of CNNs for handwritten digit recognition. Our approach integrates multiple convolutional and pooling layers, utilizing a 3×3 kernel size for optimal feature extraction. During training, the model is trained using a collection of 60,000 grayscale images, each sized 28×28 pixels. After five training epochs, the model attains an astounding level of precision of 99.16%, outperforming conventional classification algorithms such as Support Vector Machines Random Forest, Bayes Net, and Multilayer Perceptrons are frequently employed in handwritten digit recognition.

Our project's goal is to identify handwritten numbers that are entered by the user and interpret them as a complete number in the decimal system by default. Based on the user's preference, this number will then be converted into binary, octal, or hexadecimal formats. To achieve this, we will develop a graphical user interface (GUI) with a canvas widget, allowing users to draw handwritten digit sequences for recognition and conversion. After each recognition process, the canvas can be cleared to enable new inputs, ensuring a seamless and interactive user experience.

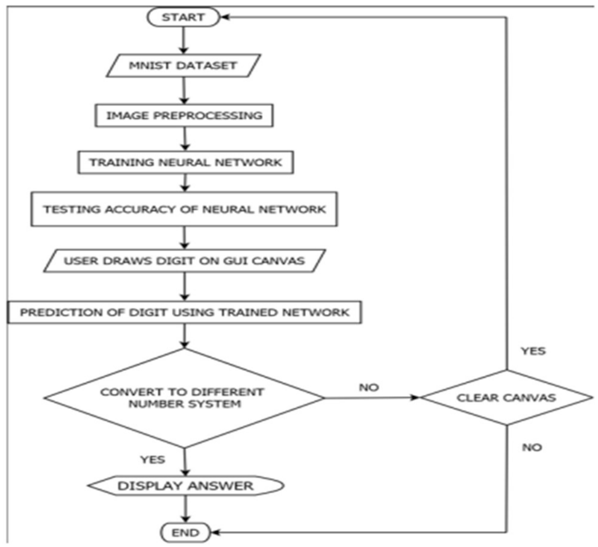

Figure 1 shows the project's activity diagram.[7]

A. Dataset

Training of the proposed model is done using the MNIST[7] dataset. A predetermined ratio is used to separate the 70,000 digital photos into training and testing sets. Before proceeding with model training, the image data undergoes cleaning and preprocessing to enhance its quality and ensure optimal performance.

B. Image Preprocessing

Image preprocessing involves applying various techniques resizing pictures, turning them into grayscale, and augmenting data to enhance the suitability of digital image data for effective use in machine learning models.

C. Neural Network Training

Following data preparation, a CNN model with a 3x3 kernel size will be built, combining several convolutional and pooling layers. With the help of several preloaded Python libraries, including TensorFlow, Pillow, OpenCV, Tkinter, and NumPy, the model will thereafter be trained using training and validation data.

D. Neural Network Accuracy Testing

Following training on the training dataset, the testing dataset is used to assess the model's performance. A specific subset of the MNIST dataset is used as the testing set, which enables the calculation of the model's accuracy and assessing its effectiveness.

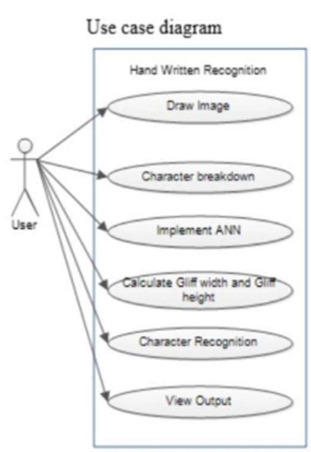

Fig 2: Use Case Diagram

E. User Sketches a Digit on the GUI Canvas

Following evaluation, or training and testing, of the model using the MNIST dataset, the trained model is then utilized through a canvas based on a Graphical User Interface (GUI), in which the user uses the mouse cursor to draw numbers by clicking and dragging the mouse in the appropriate locations.

F. Identify the Clear Canvas or Number

Two options are presented to the user once they have drawn the numbers on the GUI canvas as they see fit.

- Identify Number: The CNN model is used in this option to anticipate the user-drawn string of numbers.
- With the Clear Canvas option, the user can create additional digits for continued continuation after clearing the canvas.

G. Change to a Different Number System

Three options are shown to the user following the digit prediction.

- Convert to Binary: This choice changes the known decimal number to its counterpart in binary.
- Hexadecimal Conversion: This option changes the recognized decimal number to its corresponding hexadecimal value. To convert a recognized decimal number to its octal equivalent, use the Convert to Octal option.
    - Direct Word Classification: Utilizing Convolutional Neural Networks (CNNs), we train models capable of accurately classifying entire words.
    - Character Segmentation: Leveraging Long Short-Term Memory networks (LSTMs) with convolution, we construct bounding boxes for each character within a word. After the segmented characters are sent to a CNN for classification, we may use the outcomes of both segmentation and classification to reconstruct the complete word.

IV. RESULTS

The given bar chart showcases the accuracy performance of various deep learning (DL) and machine learning (ML) techniques classifiers used for handwritten text recognition. The classifiers evaluated in this study include Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), and Multi-Layer Perceptron with Adam Optimizer (MLP-ADAM). At 99.5%, the RF classifier has the highest accuracy making it the best-performing model among the four. MLP-ADAM, a deep learning-based approach, follows closely with 98.4% accuracy, demonstrating the effectiveness of neural networks. The KNN model achieves 97.7% accuracy, while DT lags slightly behind at 97.6%. These results indicate that while traditional ML models like RF and KNN perform well, deep learning approaches (such as MLP-ADAM) also yield competitive results in Recognition of handwritten text. The results demonstrate how ML and DL approaches can be combined to improve text recognition task accuracy.

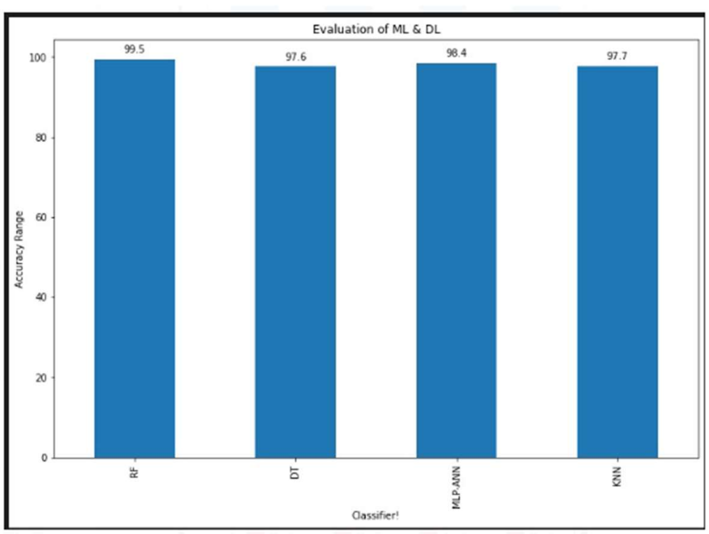

Fig 3

The bar chart compares Accuracy, Sensitivity, and Specificity of different classifiers—Decision trees (DT), K-Nearest Neighbors (KNN), Random Forest (RF), and Multi-Layer Perceptrons (MLP-ANN)—for handwritten text recognition. RF achieves the highest accuracy (99%), followed by MLP-ANN (98%), while DT and KNN both reach 97%. Sensitivity, which evaluates how well the model captures positive examples , is recorded only for MLP-ANN (1), whereas other models show 0. Specificity, indicating the ability to identify negative instances correctly, remains negligible, except for DT with 0.2. The performance of these models is evaluated using

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} \times 100$ Sensitivity (Recall or True Positive Rate) = $\frac{TP}{TP+FN} \times 100$

Specificity (True Negative Rate) = $\frac{TN}{TN+FP} \times 100$

While RF emerges as the most accurate model, MLP-ANN demonstrates better sensitivity, highlighting the importance of balancing these metrics for optimal handwritten text recognition.

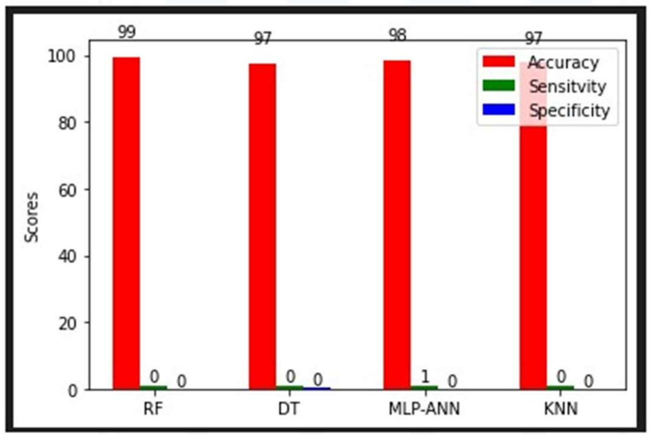

Fig 4

The confusion matrix for the Random Forest (RF) classifier in handwritten text recognition shows excellent performance, with most values concentrated along the diagonal, indicating correct classifications. Minimal misclassifications are observed, with only a few instances scattered in non-diagonal positions, suggesting strong generalization. The model effectively distinguishes different classes, but slight confusion exists between certain categories, as reflected in small off-diagonal values. Overall, the RF model demonstrates high accuracy and reliable performance, making it a robust choice for handwritten text recognition, though minor refinements such as hyperparameter tuning or data augmentation could further enhance its precision.

```
*Confusion Matrix for RF:
[[1193    0    0    0    0    0    1    0    1    0]
 [   0 1349    0    0    0    2    0    0    1    0]
 [   4    0 1149    2    0    0    0    0    2    0]
 [   0    0    2 1249    0    1    1    1    2    2]
 [   0    1    0    0 1135    0    1    0    1    2]
 [   1    1    0    3    0 1065    1    0    3    2]
 [   1    0    0    0    1    0 1165    0    0    0]
 [   0    0    2    0    0    0    0 1262    1    3]
 [   0    1    1    1    1    0    3    0 1166    1]
 [   0    0    3    3    3    0    0    2    1 1201]]
```

Fig 5

An analysis of the Random Forest (RF) model's classification in handwritten text recognition demonstrates exceptional performance, with all classes exhibiting precision, recall, and F1-score values near 1.00. With 99% accuracy indicates that the model effectively recognizes handwritten characters with minimal errors. The macro and weighted averages further confirm the model's consistency across different classes, ensuring balanced performance. With high precision, the RF classifier makes accurate predictions with fewer false positives, while its high recall ensures it captures most true instances. This report highlights RF’s strong ability to generalize well across diverse handwritten inputs, making it a reliable model for text recognition tasks.

*Classification Report for RF:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	1195
1	1.00	1.00	1.00	1352
2	0.99	0.99	0.99	1157
3	0.99	0.99	0.99	1258
4	1.00	1.00	1.00	1140
5	1.00	0.99	0.99	1076
6	0.99	1.00	1.00	1167
7	1.00	1.00	1.00	1268
8	0.99	0.99	0.99	1174
9	0.99	0.99	0.99	1213
accuracy			0.99	12000
macro avg	0.99	0.99	0.99	12000
weighted avg	0.99	0.99	0.99	12000

Fig 6

V. Conclusion

This study aimed to enhance the performance of handwritten digit recognition by evaluating different Convolutional Neural Network (CNN) architectures. Our goal was to eliminate the need for complex preprocessing, costly feature extraction, and intricate ensemble methods commonly used in traditional recognition systems. Through extensive experimentation with the MNIST dataset, we analyzed the impact of various hyperparameters on recognition accuracy, emphasizing the importance of fine-tuning these parameters for optimal performance. Using the Adam optimizer, we achieved an exceptional recognition rate of 99.89%, surpassing all previously reported results for the MNIST database. Our findings highlight how Handwritten digit identification accuracy is directly impacted by the number of convolutional layers. What's new about our approach lies in the comprehensive exploration of CNN architecture parameters, achieving superior accuracy without relying on ensemble CNN models, which typically add computational complexity. While some researchers have utilized ensemble methods to boost accuracy, our pure CNN model delivers comparable results with greater efficiency. Looking forward, we propose exploring hybrid CNN architectures such as In addition to domain-specific recognition systems, there are CNN-RNN and CNN-HMM models. Furthermore, looking at evolutionary techniques to maximize CNN parameters, such as kernel sizes, learning rate, and layer counts—holds great potential for further enhancing recognition performance.

References

[1] Ritik Dixit, Rishika Kushwah, Samay Pashine. (2021).Handwritten Digit Recognition using Machine and Deep Learning Algorithms

[2] Yugandhar Manchala, Jayaram kinthali, Kowshik Kotha, Kanithi Santosh Kumar. (2021). Handwritten Text Recognition using Deep Learning with Tensorflow.

[3] Jonathan Chung, Thomas Delteil.(2020). A Computationally Efficient Pipeline Approach to Full Page Offline Handwritten Text Recognition.

[4] Lalitha Kumari, Sukhdeep Singh, VVS Rathore (2022). Lexicon and Attention based Handwritten Text Recognition System. Rohini G.Khalkar, Adarsh Singh Dikhit, Anirudh Goel,

[5] Manisha Gupta. (2021). Deep Learning for Handwritten Text Recognition (ConvNet & RNN)

[6] L.R.B Schomaker, M. Ameryan. (2022). Comparing Transformer based to RNN based models in a Handwritten Text Recognition task.

[7] M., U. Swetha, Handwritten Character Recognition Using Machine Learning and Deep Learning (August 06, 2023). Available atSSRN: https://ssrn.com/abstract=4878511 or http://dx.doi.org/10.2139/ssrn.4878511

[8] Hanadi Hassen Mohammed, Junaid Malik, Somaya Al-Madeed, Serkan Kiranyaz.. (2022). 2D self - Organized ONN Model For HandWritten Text Recognition.

[9] Evans Ehiorobo, Rukayat Koleoso and Charles Uwadiaa. (2022). Training of Offline Handwritten Text Recognisers using Computer-Generated Text.

[10] Salvador Espana-Boquera, Maria Jose Castro-Bleda, Jorge Gorbe-Moya, and Francisco Zamora-Martinez.(2018). Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models.

[11] Sudhagar, D., Saturi, S., Choudhary, M., Senthilkumaran, P., Howard, E., Yalawar,M. S., & Vidhya, R. G. (2024). Revolutionizing data transmission efficiency in IoT-enabled smart cities: A novel optimization-centric approach. International Research Journal of Multidisciplinary Scope (IRJMS), 5(4), 592-602. https://doi.org/10.47857/irjms.2024.v05i04.01113
[12] Tiwari, R., Yalawar, M.S., Naresh Kumar, V., Kranthi Rathan, P., Sai Kumar, P.,Chandra Sekhar Reddy, L. (2025). A New and Effective Cryptography-Based Data-Hiding Technique for Image Compression. In: Bhateja, V., Dey, M., Senkerik, R.(eds) Innovations in Information and Decision Sciences. FICTA 2024 2024. Smart Innovation, Systems and Technologies, vol 422. Springer, Singapore. https://doi.org/10.1007/978-981-96-0147-9_35
[13] Mattaparthi, S., Kumar, S., Yalawar, M.S. (2023). Fake Currency Detection: A Survey on Different Methodologies Using Machine Learning Techniques. In: Kumar,A., Mozar, S., Haase, J. (eds) Advances in Cognitive Science and Communications.ICCCE 2023. Cognitive Science and Technology. Springer, Singapore. https://doi.org/10.1007/978-981-19-8086-2_45
[14] Kumar, S., Yalawar, M.S. (2020). K-NN SEMANTIC Inquiry on Scrambled Social Information BASE. In: Kumar, A., Paprzycki, M., Gunjan, V. (eds) ICDSMLA 2019.Lecture Notes in Electrical Engineering, vol 601. Springer, Singapore. https://doi.org/10.1007/978-981-15-1420-3_108
[15] Amutha, M.; Lokeshwaran, K.; Rajendiran, M.; Mohapatra, Subhasis; Shameer, A. P.;Vinodha, S, Mrutyunjaya S Yalawar, "Green AI Revolution Machine Learning For Environmental-Friendly Communication Networks" In Journal of Environmental Protection and Ecology (Vol. 25, Issue 2, pp. 537–544).-2024. https://scibulcom.net/en/article/T5RlmrQ1KrCWpgQpDjC8
[16] Arulananth TS, Kuppusamy PG, Ayyasamy RK, Alhashmi SM, Mahalakshmi M et al (2024) Semantic segmentation of urban environments: leveraging U-Net deep learning model for cityscape image analysis. PLoS ONE 19(4):e0300767.
[17] Amutha, M.; Lokeshwaran, K.; Rajendiran, M.; Mohapatra, Subhasis; Shameer, A. P.; Vinodha, S, Mrutyunjaya S Yalawar, "Green Ai Revolution Machine Learning For Environmental-Friendly Communication Networks" In Journal of Environmental Protection and Ecology (Vol. 25, Issue 2, pp. 537–544).-2024.