iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.62254

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Handwritten Digit Recognition System using Machine Learning

Tannu Tyagi[1], Harshita Sharma[2], Khushi Jain[3], Chirag Mittal[4]
Computer Science Department, Dronacharya Group of Institution ,Greater Noida

Abstract: Handwritten digit recognition remains a crucial area of research in pattern recognition and machine learning. In this paper, we present a novel approach to enhance handwritten digit recognition systems by incorporating deep learning techniques and an interactive graphical user interface (GUI).
Our system employs convolutional neural networks (CNNs) for feature extraction and classification, allowing for improved accuracy and robustness in digit recognition tasks.
The integrated GUI facilitates real-time interaction, enabling users to input handwritten digits directly for instant recognition. We evaluate the performance of our system on the MNIST dataset and demonstrate its effectiveness in achieving high accuracy and usability.

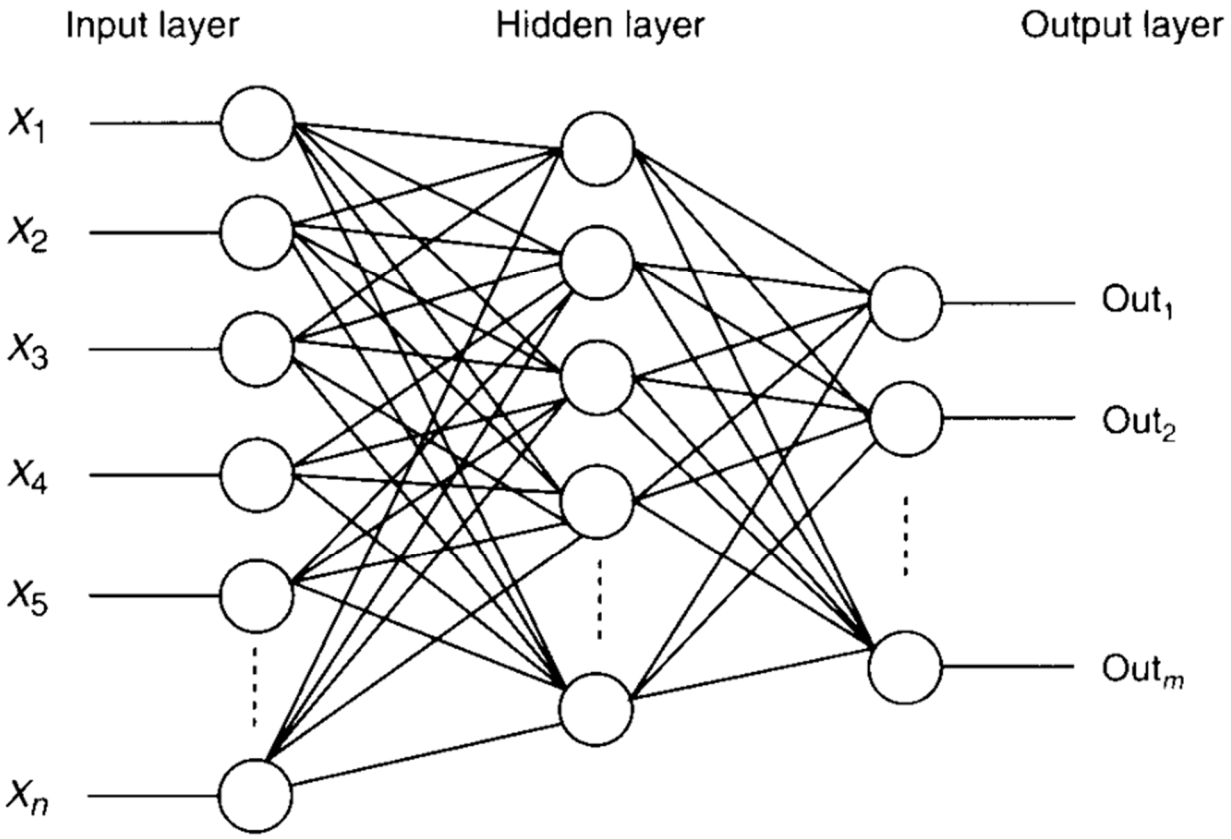
I. INTRODUCTION

Handwritten digit recognition has garnered significant attention due to its applications in various fields such as optical character recognition (OCR), digital document processing, and automated data entry.
Traditional methods often rely on handcrafted features and shallow learning algorithms, which may struggle to capture complex patterns in handwritten digits.
This has been a topic of boundless-research in the field of deep learning. Digit recognition has many applications like number plate recognition, postal mail sorting, bank check processing, etc. [1]
Deep learning, particularly convolutional neural networks (CNNs), has emerged as a powerful approach for feature learning and classification tasks, offering superior performance in image recognition tasks. In this paper, we propose a deep learning-based approach to handwritten digit recognition, complemented by an interactive GUI interface for seamless user interaction. Recently Deep Convolutional Neural Networks (CNNs) becomes one of the most appealing approaches and has been a crucial factor in variety of recent success and challenging machine learning applications such as challenge ImageNet [3, 4, 5,], object detection [6, 7], image segmentation [9,], and face recognition.
Therefore, CNNs is considered our main model for our challenging tasks of image classification. Specifically, it is used for handwriting digits recognition which is one of high academic and business transactions. Handwriting digit recognition application is used in different tasks of our real life time purposes. Precisely, it is used in banks for reading checks, post offices for sorting letter, and many other related tasks

II. METHODOLOGY

The comparison between Support Vector Machines (SVM), Multi-layered Perceptron (MLP), and Convolutional Neural Network (CNN) algorithms revolves around their performance on the MNIST dataset. MNIST comprises 70,000 handwritten digit images, with 60,000 for training and 10,000 for testing. Each image is 28x28 pixels and comes with a label denoting the digit it represents.
The model used in the proposed system is a Convolutional Neural Network (CNN). CNNs are a type of deep learning architecture specifically designed for processing and classifying images.
They have shown remarkable performance in various computer vision tasks, including handwritten digit recognition. In this context, the CNN is employed for both feature extraction and classification of handwritten digits, contributing to the system's high accuracy and robustness.

©IJRASET: All Rights are Reserved



A. *Data Set*

Handwritten character recognition is a well-explored domain, featuring diverse implementation approaches, including various datasets, algorithms, and feature engineering techniques. The MNIST dataset, derived from the NIST database, amalgamates data from Special Database 1 and Special Database 3, representing digits scribed by high school students and census bureau employees, respectively. Comprising a total of 70,000 handwritten digit images, MNIST is divided into a training set of 60,000 images and a test set of 10,000 images. Each image is sized 28x28 pixels, with anti-aliasing applied. Moreover, every image is paired with a corresponding label indicating the depicted digit.

B. *Support Vector Machine*

The Support Vector Machine (SVM) is a supervised learning algorithm. It operates by plotting data points in an n-dimensional space, where each dimension represents a feature. SVM seeks to find a hyperplane that effectively separates the data points belonging to different classes. This hyperplane is chosen to maximize the margin between the classes, ensuring optimal classification. Support vectors, which are the data points closest to the hyperplane, play a crucial role in determining its position. SVM can be categorized into two main types: linear and non-linear. In this study, we employ Linear SVM for handwritten digit recognition.[10]

C. *Multilayered Perception*

A multilayer perceptron (MLP) belongs to the category of feedforward artificial neural networks (ANN). It comprises three primary layers: the input layer, hidden layer(s), and output layer. Each layer contains multiple nodes, also known as neurons, with connections to every node in the subsequent layer.[12] While the basic MLP has three layers, the number of hidden layers can vary, with no constraints on the number of nodes within them. The input and output layer sizes are determined by the number of attributes and classes in the dataset, respectively. Determining the specific number of hidden layers and nodes is often done experimentally due to the model's unpredictable nature. Each hidden layer can utilize different activation functions for data processing. For training, MLP employs a supervised learning method known as backpropagation, adjusting the weights of connections iteratively to align with the training data.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

D. Convolutional Neural Network

CNN, or Convolutional Neural Network, is a powerful deep learning algorithm widely applied in image recognition and classification tasks. Unlike traditional methods, CNN requires minimal preprocessing of images. It processes images in small segments rather than individual pixels, enhancing its ability to detect intricate patterns, such as edges, efficiently. The network comprises an input layer, an output layer, and multiple hidden layers, including Convolutional layers, Pooling layers (such as Max and Average pooling), Fully Connected layers (FC), and normalization layers. [12] CNN utilizes filters, or kernels, which are arrays of weights, to extract features from input images. Each layer in CNN employs different activation functions to introduce non-linearity [13] into the model. Throughout the CNN layers, the height and width of the data decrease while the number of channels increases, resulting in a column matrix used for predicting the output.[14].

E. Visualization

In this study, we utilized the MNIST dataset, which comprises handwritten digit images, to compare various levels of deep and machine learning algorithms, namely SVM, MLP, and CNN. Our comparison focused on execution time, complexity, accuracy rate, number of epochs, and number of hidden layers (specifically for deep learning algorithms). To present the insights gleaned from our detailed analysis, we employed bar graphs and tabular charts using the matplotlib module. These visualizations offer precise representations of the algorithmic advancements in digit recognition, with graphs provided at crucial stages of the programs to enhance the interpretation of results.

III. IMPLEMANTATION

To conduct a comprehensive comparison of the algorithms based on working accuracy, execution time, complexity, and the number of epochs (specifically for deep learning algorithms), we employed three distinct classifiers:

1) Support Vector Machine Classifier
2) Multilayer Perceptron (MLP) Classifier
3) Convolutional Neural Network (CNN) Classifier

Each algorithm's implementation was extensively detailed below to ensure a smooth and accurate flow of analysis, facilitating a thorough comparison between them.

A. Pre – Processing

Pre-processing serves as an initial step in both machine and deep learning, aiming to enhance input data quality by reducing unwanted impurities and redundancy. To simplify the input data, all images in the dataset were reshaped into 2-dimensional arrays of size (28,28,1). Since pixel values range from 0 to 255, normalization was performed by converting the dataset to 'float32' and dividing by 255.0, ensuring input features fall within the range of 0.0 to 1.0. Additionally, one-hot encoding was applied to transform y values into binary arrays, making each number categorical. For example, the output value 4 would be represented as [0,0,0,0,1,0,0,0,0,0]

B. Support Vecctor Machine

In scikit-learn [16], the Support Vector Machine (SVM) module supports both dense (numpy.ndarray) and sparse (any scipy.sparse) sample vectors as input. Within scikit-learn, classes such as SVC, Nu SVC, and LinearSVC are capable of performing multi-class classification on datasets. In this study, we opted for Linear SVC to classify the MNIST datasets, leveraging a Linear kernel implemented using LIBLINEAR [17]. The implementation made extensive use of various scikit-learn libraries including NumPy, Matplotlib, Pandas, Sklearn, and Seaborn. The process began with downloading the MNIST datasets, followed by loading and reading the CSV files using Pandas. Subsequently, we conducted the plotting of sample data and conversion into matrices, followed by normalization and feature scaling. Finally, a linear SVM model was created, along with a confusion matrix utilized to evaluate the accuracy of the model.[9]

C. Multi Layered Perceptron

The implementation of Handwritten Digits Recognition using Multilayer Perceptron (MLP) [18] , also known as a feedforward artificial neural network, was carried out with the assistance of the Keras module. We utilized the Sequential class to create an MLP model and incorporated various hidden layers with distinct activation functions to process input images of size 28x28 pixels. After establishing the sequential model, we added Dense layers with different specifications and Dropout layers, as depicted in the provided block diagram.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

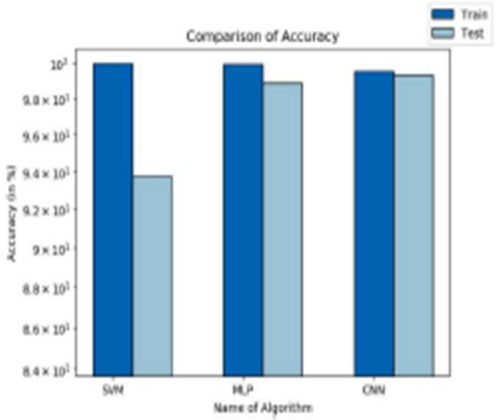
We employed a neural network with 4 hidden layers and an output layer comprising 10 units, corresponding to the total number of labels. Each hidden layer comprised 512 units. The input to the network was a 784-dimensional array derived from the 28x28 image. The Sequential model facilitated the systematic addition of desired layers. We utilized the Dense layer, also referred to as a fully connected layer, given its characteristic of connecting all neurons from one layer to those in the previous layer. Additionally, we incorporated the Rectified Linear Unit (ReLU) activation function to introduce non-linearity, aiding the network in learning non-linear decision boundaries. The final layer adopted a softmax function, suitable for multiclass classification problems.[19]

D. Conventional Neural Network

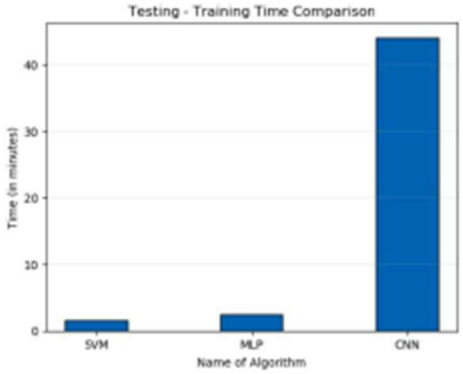
Handwritten digit recognition using Convolutional Neural Network (CNN) was implemented using Keras, an open-source neural network library.[15] We utilized the Sequential class in Keras to construct the model layer-by-layer. The input image dimension was set to 28 (Height) x 28 (Width) x 1 (Number of channels) [20]. The initial layer of the model was a Convolutional layer, which convolves a filter matrix across the input data to extract features. We employed 32 filters with dimensions (3,3) and a stride of 1. The activation function used in this layer was Rectified Linear Unit (ReLU) to introduce non-linearity. Subsequently, another convolutional layer was added with 64 filters of dimensions (3,3) and ReLU activation. Pooling layers were then applied to reduce image dimensionality and computational load, using MAX-pooling to retain only the maximum value from each pool.[22] We set the pool size to (2,2) with a stride of 2. Dropout layers were introduced to prevent overfitting by randomly dropping neurons. [23] A dropout probability of 0.25 (25%) was used. Following this, a Flatten layer was applied to convert the 2-dimensional matrix into a column vector, which was then fed into a fully connected layer consisting of 128 neurons.[23] A dropout probability of 0.5 (50%) was set for this layer. The output of the fully connected layer was passed through a final output layer with 10 neurons representing the classes (numbers 0 to 9). [24] The SoftMax function was employed in the output layer to perform classification, returning a probability distribution over all 10 classes.[25] The class with the highest probability was considered the output.

COMPARISON ANALYSIS OF DIFFERENT MODELS.

MODEL	TRAINING RATE	TESTING RATE	EXECUTION TIM
SVM	99.98%	94.005%	1:35 min
MLP	99.92%	98.85%	2:32 min
CNN	99.53%	99.31%	44:02 min



Bar graph depicting accuracy comparison (SVM (Train: 99.98%, Test: 93.77%), MLP (Train: 99.92%, Test: 98.85%), CNN (Train: 99.53%, Test: 99.31%)).



Bar graph showing execution time comparison of SVM, MLP and CNN (SVM: 1.58 mins, MLP: 2.53 mins, CNN: 44.05 mins).

IV. RESULTS

We evaluate the performance of our system using the MNIST dataset, a widely used benchmark for handwritten digit recognition. The system achieves high accuracy rates, surpassing traditional methods and demonstrating robustness to variations in handwriting styles and noise. Furthermore, user feedback on the integrated GUI interface is positive, highlighting its intuitiveness and ease of use.

COMPARISON ANALYSIS OF DIFFERENT MODELS.

MODEL	TRAINING RATE	TESTING RATE	EXECUTION TIM
SVM	99.98%	94.005%	1:35 min
MLP	99.92%	98.85%	2:32 min
CNN	99.53%	99.31%	44:02 min

V. LIMITATIONS

One of the most difficult parts of a handwritten digit recognition system is that there are a lot of different handwriting styles which is a very personal behavior. Numbers may have different parts of stress, may be written in different angles, may have different lengths of particular segments. Although these challenges are faced by machine learning developers, several steps such as fine tuning the already defined models and creating state-of-the-art classification methods for predicting handwritten digits effectively by reducing computational cost, time and also improving accuracy have been taken up already. Extensive research is also being conducted in this field for sustaining it accordingly. Several issues may arise if this model is implemented on a huge scale. Recognizing handwritten digits, if used with bad intention, could ultimately lead to several issues. People may use such a technology to identify bank pins, atm pins, etc. to perform monetary felonies. On the contrary, even though issues like these may come up, measures could be taken to handle these issues effectively and sustain the use of this technology to automate many processes such as banking, address recognition, shipping systems, postal industry, etc. which will make it beneficial to use as it will ultimately have more pros than cons.

VI. CONCLUSION

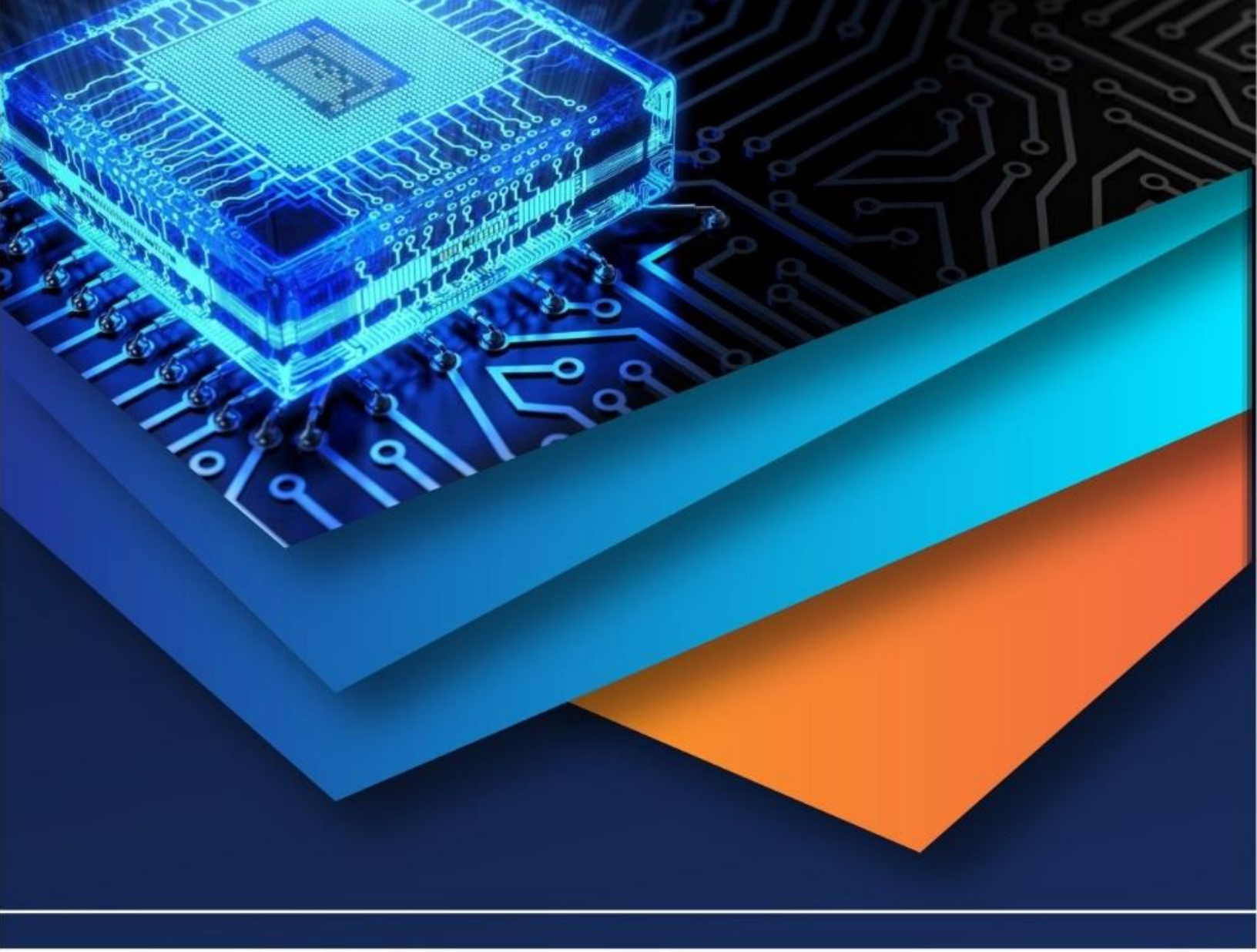
In conclusion, we present a deep learning-based handwritten digit recognition system augmented with an interactive GUI interface. The integration of CNNs and the GUI enables accurate and user-friendly digit recognition, with potential applications in various domains such as digital document processing, educational tools, and interactive interfaces. Future research directions may involve extending the system to recognize the digits in other languages as well and exploring advanced CNN architectures for further performance improvements.

REFERENCES

[1] "Handwriting recognition": https://en.wikipedia.org/wiki/Handwriting_recognition
[2] "What can a digit recognizer be used for?": https://www.quora.com/ What can a digit recognizer be used for?

[3] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, Zhuowen Tu, " Deeply-Supervised Nets " NIPS 2014.
[4] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures", IEEE Transactions Computer, vol. 22, no. 1, 1973.
[5] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato and Yann LeCun "What is the Best Multi-Stage Architecture for Object Recognition" CCV'09, IEEE, 2009.
[6] Kaiming, He and Xiangyu, Zhang and Shaoqing, Ren and Jian Sun " Spatial pyramid pooling in deep convolutional networks for visual recognitionl European", Conference on Computer Vision, arXiv:1406.4729v4 [cs.CV] 23 Apr 2015.
[7] X. Wang, M. Yang, S. Zhu, and Y. Lin. " Regionlets for generic object detection". In ICCV, 2013.
[8] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey. "ImageNet classification with deep convolutional neural networks". In Advances in Neural Information Processing Systems 25 (NIPS'2012). 2012.
[9] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. " Indoor semantic segmentation using depth information". International Conference on Learning Representation, 2013. "Support Vector Machine Algorithm": https: www.javatpointcommachine learning support vector machine algorithm
[10] https://machinelearningmastery.com/neural-networks-crash-course/
[11] "An Introduction to Convolutional Neural Networks research gate:net=publication=285164623 An Introduction to Convolutional Neural Networks.
[12] "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning", Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall: https : ==arxiv:org=pdf=1811:03378:pdf
[13] "Basic Overview of Convolutional Neural Network": https://medium.com/dataseries/basic-overview-of-convolutional-neuralnetwork-
[14] https: //github.com/dixitritik17/Handwritten-Digit-Recognition .
[15] https://scikit-learn.org/stable/modules/svm.html
[16] https: //en.wikipedia.org/wiki/LIBSVM .
[17] https://github:com/Flame-Atlas-HandwrittenDigitRecognitionusing MLP=blob=master=ANN:py
[18] https://learnopencv.com/understanding-feedforward-neural-networks/
[19] "How Do Convolutional Layers Work in Deep Learning Neural Networks": https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-
[20] https://machinelearningmastery.com/rectified-linear-activation-functionfor-deep-learning-neural-networks/
[21] https://machinelearningmastery.com/pooling-layers-for-convolutionalneural-networks/.
[22] https://medium.com/@amarbudhiraja/https-medium-comamarbudhiraja-learning-less-to learn-better-dropout-in-deep-machinelearning-74334da4bfc5.
[23] https://medium.com/dataseries/basic-overview-of-convolutional-neuralnetwork-cnn-4fcc7dbb4f17.
[24] https://medium.com/data-science-bootcamp/understand-the-softmaxfunction-in-minutes-f3a59641e86d.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)