



# An Efficient Novel Approach for Detection of Handwritten Numericals Using Machine Learning Paradigms

Pavithra Avvari<sup>(✉)</sup>, Bhavani Ratakonda, K. Sandeep, Y. Jeevan Nagendra Kumar, and T. N. P. Madhuri

Department of IT, Gokaraju Rangaraju Institute of Engineering and Technology, JNTU(H), Hyderabad, India

pavithra.griet@gmail.com, bhavani1592@grietcollege.com

**Abstract.** The goal of recognizing handwritten numbers on paper is to extract the characteristics of the entered handwritten number. The human visual system is one of the wonders of the world. People can easily find and recognize the numbers as they are used for recognition and identification. Most people easily recognize numbers without effort. Humans are amazing at deciphering what our eyes show. All work is done subconsciously, and no additional training is required to recognize numbers. Generally, we do not need to assess the difficulty of a problem that our visual system solves. However, recognizing numbers for mechanical devices such as computers is not that easy. If we try to develop a computer program to detect numbers, the difficulty of visually recognizing patterns is significant. When it comes to coding, what seems simple when we see it will be quite complicated. Simple intuitions, for example, are for a system that recognizes strokes, circles, curves, straight lines, curves, and more. Example of recognizing a number like 9, you should recognize the loop at the top and a straight vertical bar at the bottom right. This is not easy to express algorithmically. That is why we came up with a prototype for number recognition. Although identification, classification and recognition are simple tasks with the help of machine learning, we are trying to develop a system that recognizes numbers as accurately as possible. The main goal of focusing on handwriting recognition is to learn about neural networks. Images are divided into pixels and recognized in pixel order.

**Keywords:** K nearest neighbours · Stochastic gradient descent · Support vector machines · Neural networks · Deciphering · Metrics · Dataset hyperplanes · Regression

## 1 Introduction

Our project aims at an automatic recognition of digits written by hand. The tendency of writing the digits in different forms is much higher as each person writes them differently. In such cases also, the system developed must be able to recognize the digits correctly and give out the accurate results. There exist many classification and pattern recognition algorithms in machine learning such as KNN, SVM, Neural Networks (CNN, ANN, and RNN) [1, 2]. We try to study how each algorithm works while training and testing the dataset. The project focuses on developing an easy methodology of recognizing the images for a plethora of things. The goal of this system is to provide information about the research project “An Efficient Novel approach for detection of handwritten numerical using Machine Learning paradigms” [12]. It includes what we did for the completion of project, requirements, description of modelling of the project, the technologies we have used for the project, implementation of the project and future enhancements [1, 2].

This project explains in detail about how a machine can identify human written digits. For this process, we are considering MNIST (“Modified National Institute of Standards and Technology”) dataset which consist of 60000 images of 28\*28 sizes in grey scale[15]. However, considering images as a dataset will not make our work easy. These images need to be pre-processed [2, 4]. As well as many other step need to be performed before entering into training the dataset. We are considering few classification and recognition algorithms like KNN, SVM and CNN [3]. We come up with a conclusion that no algorithm can beat up Neural Network in digit recognizer. We try to achieve the highest accuracy and prediction rates by applying best training methodologies [3, 4].

## 2 Related Works

The application identification of handwritten numerals using machine learning provides the following uses to the user: this application provides a chance for the user to allow their machine to recognize the digits from 0 to 9 written in any format with ease [2, 3]. It is useful for various kinds of future works like usage of this module in number plate recognition, number detection projects, etc. This also helps in performing multiple further operations like calculating attendance percentage, focusing on absentees, sending text messages to parents, etc. [14]. This document furnishes different diagrams for the “An Efficient Novel approach for detection of handwritten numerical using Machine Learning paradigms” using uml that clearly explains the model of building this particular application [3, 4]. The literary survey provides insights into the project in terms of the user experience and functionality of the project [9].

## **2.1 Feasibility Study**

In the software development process, the feasibility study is critical. It provides the developer an opportunity to assess the operational flexibility of the product that is being developed[13]. The operational flexibility, technical support, the output of the project, and many others are treated as various criteria and parameters to analyse the feasibility of the project. Operational Feasibility, Technical Feasibility, as well as Economic Feasibility are the three basic types of feasibility studies.

### **2.1.1 Operational Feasibility**

The application developed must be in such a way that it is feasible for the users to use with ease. How much the user is willing to use the application determines the operational feasibility of the system. Before checking the operational feasibility, the user must be adequately trained with regard to the application [11]. The user must not feel any difficulty in using the system. However, the percentage of the willingness of the user to use the application entirely depends on the way that the system is built with a user-friendly interface and the method that is taken to popularize the user about the system handling and usage. Our project employs a simple yet intuitive interface for the user to navigate through the application and operate it feasibly.

### **2.1.2 Technical Feasibility**

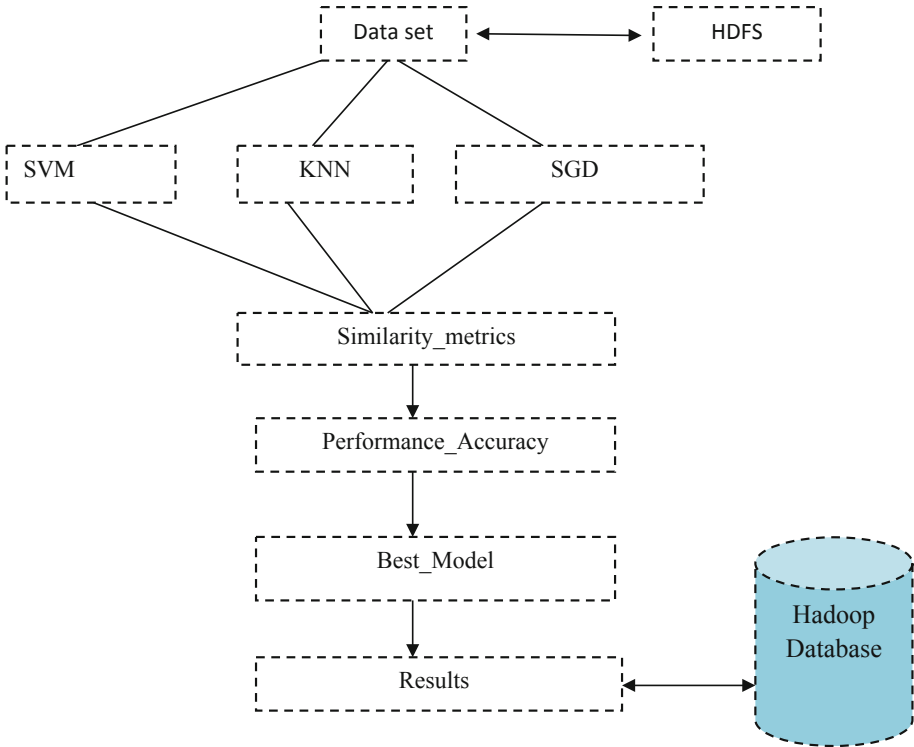
This study is done to find the technical feasibility. This study determines to find out the technical requirements needed to develop the system.. Our project is developed with utmost care such that it is technically feasible [12]. Changes can be easily made and adopted according to the user's requirements. The trending programming language "Python" and its inbuilt libraries made the application development simple and easy.

### **2.1.3 Economic Feasibility**

This study is done to find the economic feasibility of the developing system. The expenditures must be justified. The existing systems have the economy as one main backdrop because they require other specially developed devices [3, 15]. Thus, our project is designed well with a limited budget, and the hidden reason for this would be freely available technologies. The one and only customized product that has been used as an external device is a webcam which is always an economic-friendly device.

### 3 System Architecture

See Fig. 1.



**Fig. 1.** Model framework

## 4 Methodology

### 4.1 Data

Handwritten digits were provided for a total of 70,000 photos from the MNIST (Modified National Institute of Standards and Technology) data set, with 60,000 examples in the training set and 10,000 examples in the test set, both with labeled images of 10 digits (0 to 9). This is a small part of the NIST broad set that was normalized to fit a 20 \* 20 pixel frame without changing the aspect ratio [17]. The handwritten digits are images in the form of 28 \* 28 grayscale intensities of images that represent an image, with the first column of each image being a label (0 to 9). The same has been decided in the case of the test set, which consists of 10,000 photos with labels ranging from 0 to 9 (Figs. 2, 3 and 4).

Name	Type	Size	Value
X_test	float64	(899, 64)	Min: 0.0 Max: 16.0
X_train	float64	(898, 64)	Min: 0.0 Max: 16.0
data	float64	(1797, 64)	Min: 0.0 Max: 16.0
digits	utils.Bunch	7	Bunch object of sklearn.utils module
n_samples	int	1	1797
y_test	int32	(899,)	Min: 0 Max: 9
y_train	int32	(898,)	Min: 0 Max: 9

Fig. 2. Data

Node	Transferring Address	last contact	configured capacity (GB)	used (GB)	non DFS used (GB)	Remaining (GB)	Used (%)	Remaining (%)	blocks	Block pool used(GB)
DN-1	172.16.9.31:50010	3	9112.91	14.80	71.01	8157.09	0.53	83.99	78.00	5.71
DN-2	172.16.9.32:50010	2	1154.19	14.68	77.66	1421.85	1.03	92.77	76.00	5.98
DN-3	172.16.9.33:50010	1	1450.26	14.84	87.56	1417.88	1.08	89.63	79.00	5.36
DN-4	172.16.9.34:50010	4	1450.26	14.84	87.56	1417.88	1.08	89.63	79.00	5.36
DN-5	172.16.9.35:50010	3	1450.26	14.84	87.56	1417.88	1.08	89.63	79.00	5.36

Fig. 3. Table of memory allocation in hadoop ecosystem

Data set(DS)	Size of the_fill	Replication®	Block_size(BS)	Modification_Time(MT)	Permissions (P)	Ownr (O)
				12/11/2020 16:24	rwxr-xr-x	IT
1	15.86 MB	6	128 MB	12/11/2020 16:54	rw-r--r--	IT
2	15.86 MB	6	128 MB	12/11/2020 17:24	rw-r--r--	IT
3	146.53 MB	6	128 MB	12/11/2020 17:44	rw-r--r--	IT
4	146.53 MB	6	128 MB	12/11/2020 18:54	rw-r--r--	IT
5	146.53 MB	6	128 MB	12/11/2020 19:24	rw-r--r--	IT

Fig. 4. Table of memory segregation in hadoop environment for file distribution

### 4.2 Support Vector Machines (SVM)

The ability of a machine with a reference vector to generate the highest level of precision is greater. SVM can be used for classification, which involves drawing a line between two categories or classes to distinguish [6]. Hyperplanes are the lines that connect different classes [8]. However, this distinction is not so clear [11]. In this situation, the dimension of the hyperplane must be changed from 1D to the N-th dimension, which is called the nucleus. Linear nuclei, polynomial nuclei, and functional nuclei with a radial base are the three types of nuclei [8, 9]. In multidimensional space, the created hyperplane divides different classes [17]. SVM iteratively develops an ideal hyperplane that minimizes the error [7, 13]. However, because it takes a long time to train and performs worse with overlapping classes, this classifier is not suitable for large data sets. It is also sensitive to the type of core used [8, 16].

SVM falls into the category of controlled learning and with a bonus for classification and regression problems. In general, SVM draws an optimal hyperplane, which is classified into different categories [18]. In two-dimensional space, we first draw the data points of the independent variable corresponding to the dependent variables. Then begin the classification process by looking at the hyperplane or any linear or nonlinear plane that distinguishes the two classes at their best (Fig. 5).

Algorithm:	
1.	Identify the correct hyperplane which segregates the two classes better.
2.	Look for the maximum distance between nearest data point (of either any class) or hyperplane, the distance is measured as margin. So, look for hyperplane with maximum margin both sides equally. Hyperplane with higher margin is more robust, whereas low margin has changed for misclassification.
3.	SVM selects the classifier accurately to maximized margin.
4.	SVM is robust to the classifier and have a feature to ignore outliers and try to look for a hyperplane with maximum margin

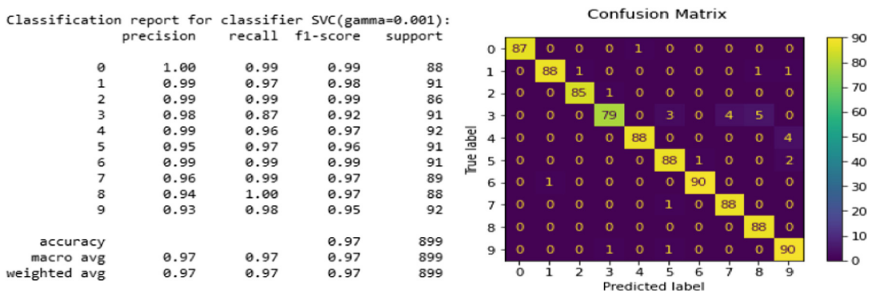


Fig. 5. Support Vector Machines (SVM)

### 4.3 K Nearest Neighbours (KNN)

This is the most basic classifier for categorizing images. This classifier is effectively explained with a simple expression in plain language: “Tell me your neighbors and I will tell you who you are.” This approach is based solely on the distance between two illuminated vectors [7]. Finds the most common data among the closest K samples to distinguish the new data. Euclidean distance can be used as a distance metric [8].

This algorithm gave an accuracy of 92.8%. However, this algorithm has many significant drawbacks in terms of various aspects such as the choice of features [8], dimensionality reduction, etc. KNN is the nonparametric method or classifier used for both classification and regression problems [17]. This is the delayed or late learning classification algorithm, in which all the calculations are done until the last stage of the classification, and these are instance-based learning algorithms where the convergence is done locally. As it is the simplest and easiest to implement, there is no explicit training phase before and the algorithm does not perform training data aggregation (Fig. 6).

<b>Algorithm:</b>	
1.	Compute the distance metric between the test data point and all labeled data points.
2.	Order the labeled data points in increasing order of distance metric.
3.	Select the top K labeled data points and look at class labels.
4.	Look for the class labels that majority of these K labeled data points have and assign it to test data points.

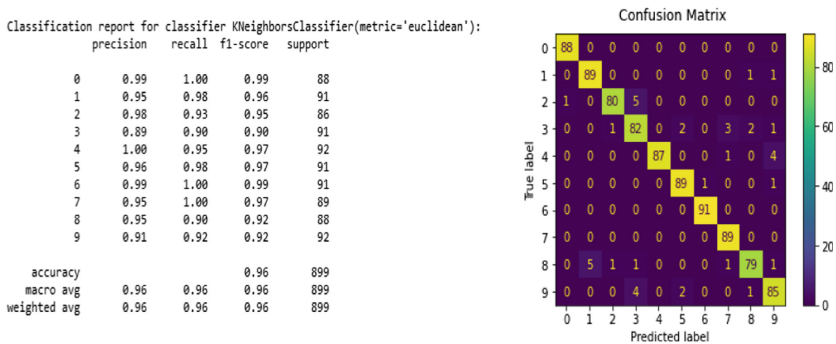


Fig. 6. K Nearest Neighbours (KNN)

### 4.4 Stochastic Gradient Descent

Stochastic gradient descent is a very popular and common algorithm that is used in various machine learning algorithms; the most important thing is that it forms the basis of neural networks. Gradient descent is an iterative algorithm that starts from an arbitrary point on a function and moves down its slope in steps until it reaches the lowest point of that function [18] (Figs. 7 and 8).

<b>Algorithm:</b>	
1.	<i>Find the slope of the objective function with respect to each parameter/feature. In other words, compute the gradient of the function.</i>
2.	<i>Pick a random initial value for the parameters. (To clarify, in the parabola example, differentiate "y" with respect to "x". If we had more features like x1, x2 etc., we take the partial derivative of "y" with respect to each of the features.)</i>
3.	<i>Update the gradient function by plugging in the parameter values.</i>
4.	<i>Calculate the step sizes for each feature as: <math>step\ size = gradient * learning\ rate</math>.</i>
5.	<i>Calculate the new parameters as: <math>new\ params = old\ params - step\ size</math></i>
6.	<i>Repeat steps 3 to 5 until gradient is almost 0.</i>

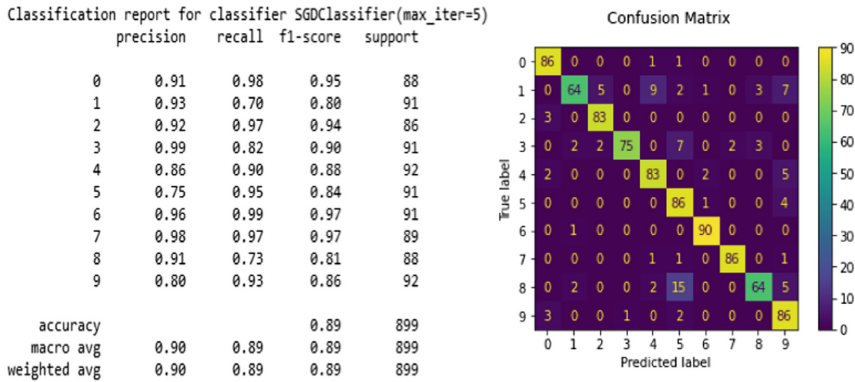


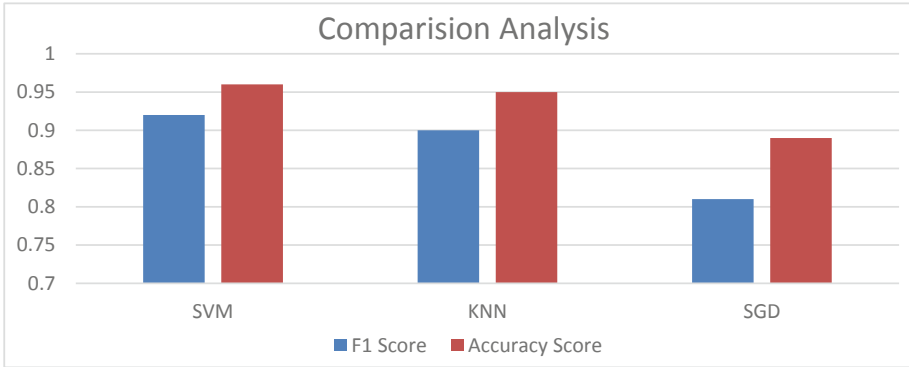
Fig. 7. Stochastic Gradient Descent (SGD).

**Table: Comparison Analysis**

<i>Name of the Classifiers</i>	<i>F1 Score</i>	<i>Accuracy Score</i>
<i>SVM</i>	0.92	0.96
<i>KNN</i>	0.90	0.95
<i>SGD</i>	0.81	0.89

In this section, the processed images are sent as input to various algorithms. The precision and F1 evaluation values are 0.96, 0.95, 0.89 and 0.92, 0.90, 0.81, respectively. Choosing the correct data set, pre-preparing the data with the right techniques, planning the model, and many other tasks combine to create better performance. The CNN model in our model efficiently organizes all the test photos with the names of the individual classes. This did not happen in any case. Each age group has its own data set [18]. Accuracy seems to increase with each generation. The pass set was tested after the





**Fig. 8.** Model accuracy

model was developed on the preparatory data set and showed a precision of 0.96. Then when the model was obtained with test data as input, the model showed an accuracy of 0.97%.

## 5 Conclusion

In this section we saw how the different algorithms work in handwriting recognition. There are many classification and recognition algorithms. K-Neighbor and Support Vector Machine are some of the most famous and widely used. All these algorithms consider various factors during training and testing. Also considered is the MNIST handwritten digit dataset used, consisting of approximately 60,000 images with sizes  $28 * 28$  grayscale images of handwritten digits 0 to 9. All images should be pre-processed using appropriate techniques. In the study, the main characteristics are extracted from each image for further processing for training. The processed images are sent as input to various algorithms. The precision and F1 evaluation values are 0.96, 0.95, 0.89 and 0.92, 0.90, 0.81, respectively. Choosing the correct data set, pre-preparing the data with the right techniques, planning the model, and many other tasks combine to create better performance. The CNN model in our model efficiently organizes all the test photos with the names of the individual classes. This did not happen in any case. Each age group has its own data set. Accuracy seems to increase with each generation. The pass set was tested after the model was developed on the preparatory data set and showed a precision of 0.96. Then when the model was obtained with test data as input, the model showed an accuracy of 0.97%. The excellent performance of the model is a direct result of the inclusion and structure of the authentic image of the model.

## References

1. Abu Ghosh, M.M., Maghari, A.Y.: A comparative study on handwriting digit recognition using neural networks. In: 2017 International Conference on Promising Electronic Technologies (ICPET), 2017, pp. 77–81 (2017). <https://doi.org/10.1109/ICPET.2017.20>

2. Babu, U.R., Venkateswarlu, Y., Chintha, A.K.: Handwritten Digit Recognition Using K-Nearest Neighbour Classifier. *World Congress on Computing and Communication Technologies* **2014**, 60–65 (2014). <https://doi.org/10.1109/WCCCCT.2014.7>
3. G. Vijendar Reddy, Sukanya Ledalla, Avvari Pavithra: A quick recognition of duplicates utilizing progressive methods ‘International Journal of Engineering and Advanced Technology (IJEAT)’ at Volume-8 Issue-4, April 2019
4. Tuba, E., Bacanin, N.: An algorithm for handwritten digit recognition using projection histograms and SVM classifier. In: 2015 23rd Telecommunications Forum Telfor (TELFOR), 2015, pp. 464–467. <https://doi.org/10.1109/TELFOR.2015.7377507>
5. Gil, A.M., Costa Filho, C.F.F., Costa, M.G.F.: Handwritten digit recognition using SVM binary classifiers and unbalanced decision trees. In: Campilho, A., Kamel, M. (eds.) *ICIAR 2014. LNCS*, vol. 8814, pp. 246–255. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-11758-4\\_27](https://doi.org/10.1007/978-3-319-11758-4_27)
6. Ahamed, Hafiz & Alam, Ishraq & Islam, Md. (2019): SVM Based Real Time Hand-Written Digit Recognition System
7. Al-Wzawzy, Haider. (2016): Handwritten Digit Recognition Using Convolutional Neural Networks. *International Journal of Innovative Research in Computer and Communication Engineering*
8. Ratakonda, B., Avvari, P., Rajarao, B., Sreevani, V., Ganapathi Raju, N.V.: Smart Parking for Smart Cities using IoT in 3rd International conference on Design and manufacturing aspects for sustainable energy(ICMED-2021) (2021). <https://doi.org/10.1051/e3sconf/202130901128>
9. Jeevan Nagendra Kumar, Y., Spandana, V., Vaishnavi, V.S., Neha, K., Devi, V.G.R.R.: Supervised machine learning approach for crop prediction in agriculture sector. In: *IEEE - 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 736–741. ISBN: 978-1-7281-5370-4
10. Prasanna Lakshmi, K., et al.: Video genre classification using convolutional recurrent neural networks. *Int. J. Adv. Comput. Sci. Appl.* **11**, 170–176 (2020). ISSN: 2156-5570 (Online) ISSN: 2158-107X (Print)
11. Shailaja, V., Lohitha, R., Musunuru, S., Deepthi Reddy, K., Padma Priya, J.: Predictive analytics of performance of india in the olympics using machine learning algorithms. *Int. J. Emerging Trends Eng. Res.* **8**(5), May 2020. ISSN 23473983
12. Zaguia, A., Raju, V., Jeevan Nagendra Kumar, Y., Rawat, U.: Secure Vertical Handover to NEMO using Hybrid Cryptosystem. *Hindawi*, Article ID 6751423, Hindawi
13. Jeevan Nagendra Kumar, Y., Rajini Kanth, T.V.: GIS-MAP based spatial analysis of rainfall data of Andhra Pradesh and telangana states using R. *Int. J. Electr. Comput. Eng. (IJECE)* **7**(1), February 2017, Scopus Indexed Journal, ISSN: 2088-8708
14. Ratakonda, B., Therala, A., Hanumanthu, C.K.: Driving license detection using QR code. In: *International Conference on Design and Manufacturing Aspects for Sustainable Energy*, 19 August 2020, *ICMED* (2020). <https://doi.org/10.1051/e3sconf/202018401010>
15. Prasanna Lakshmi, K., et al.: Efficient mining of data streams using associative classification approach. *Int. J. Softw. Eng. Knowl. Eng.* **25**(3), 605–631 (2015). ISSN (Online): 1793-6403. ISSN (Print) :0218-1940
16. Ledalla, S., Mahalakshmi, T.S.: Sentiment analysis using legion kernel convolutional neural network with LSTM. *Int. J. Innov. Technol. Exploring Eng.* **8**, 226–229 (2019)
17. Subbarayudu, Y., Sureshabu, A.: Distributed multimodal aspective on topic model using sentiment analysis for recognition of public health surveillance. *Expert Clouds and Applications*, 16 July 2021, doi.[https://doi.org/10.1007/978-981-16-2126-0\\_38](https://doi.org/10.1007/978-981-16-2126-0_38) Springer, Singapore. Print ISBN978-981-16-2125-3 Online ISBN978-981-16-2126-0.

18. Rajiv, K., Rajasekhar, N., Prasanna Lakshmi, K., Srinivasa Rao, D., Sabitha Reddy, P.: Accuracy evaluation of plant leaf disease detection and classification using GLCM and multi-class SVM classifier. In: Sharma, H., Saraswat, M., Kumar, S., Bansal, J.C. (eds.) CIS 2020. LNDECT, vol. 61, pp. 41–54. Springer, Singapore (2021). [https://doi.org/10.1007/978-981-33-4582-9\\_4](https://doi.org/10.1007/978-981-33-4582-9_4)