



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 5, May 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

 9940 572 462

 6381 907 438

 ijirset@gmail.com

 www.ijirset.com

Tomato Leaf Disease Detection Using CNN

Priyanka Dodake, Prof Pushpa Tandekar, Prof Lowlesh Yadav

Department of Computer Science and Engineering, Shri Sai College of Engineering and Technology, Chandrapur, India

Department of Computer Science and Engineering, Shri Sai College of Engineering and Technolog, Chandrapur, India

Department of Computer Science and Engineering, Shri Sai College of Engineering and Technology, Chandrapur, India

ABSTRACT: Plant diseases cause low agricultural productivity. Plant diseases are challenging to control and identify by the majority of farmers. In order to reduce future losses, early disease diagnosis is necessary. This study presents a deep learning approach for detecting tomato leaf diseases using Convolutional Neural Networks (CNNs). The proposed method involves preprocessing the tomato leaf images, followed by training the CNN model to classify them into one of ten categories: healthy, yellow leaf curl virus (YLCV), bacterial spot (BS), early blight (EB), leaf mold (LM), septoria leaf spot (SLS) target spot (TS), two spotted spider mite spot(TSSMS), mosaic virus(MV) and late blight (LB). The model was trained using a dataset of 16021 tomato leaf images. The training was conducted for 10 epochs, 20 epochs, and 50 epochs, and the accuracy achieved was 64%, 94%, and 97%, respectively. These results demonstrate the effectiveness of the proposed approach in detecting tomato leaf diseases, and the performance improves with increasing epochs. The automated approach can aid in the early detection and prevention of tomato diseases, which can ultimately help in improving the yield and quality of tomato crops.

I. INTRODUCTION

Tomatoes are one of the most widely cultivated and consumed crops globally, making it a significant part of the agriculture industry. Unfortunately, tomato plants are susceptible to various diseases, which can lead to significant economic losses due to reduced yield and quality. One of the most common diseases that affect tomato plants is grey leaf spot, which damages and kills the leaves, ultimately hindering the plant's ability to produce fruit. The infection caused by the pathogen responsible for grey leaf spot in tomato plants progresses through four phases: contact, invasion, latency, and onset. Detecting diseases early can help prevent large-scale pandemics and enable appropriate management practices. Traditional methods of detecting diseases are time-consuming and expensive, especially when the farm is extensive, making it challenging to monitor each plant. Thus, there is a need for a more efficient and cost-effective solution. Image processing techniques can automate the detection of diseases in leaves, thereby saving time, money, and effort. The dataset used in this study consists of 16021 images of ten types of diseased tomato leaf images. All images are resized to 256 x 256 and divided into three parts, namely, training, testing, and validation datasets. By analyzing the features of diseased leaves, image processing technology can accurately diagnose illnesses quickly. Deep learning techniques, specifically convolutional neural networks (CNNs), have proven to be effective in image classification tasks, including plant disease detection

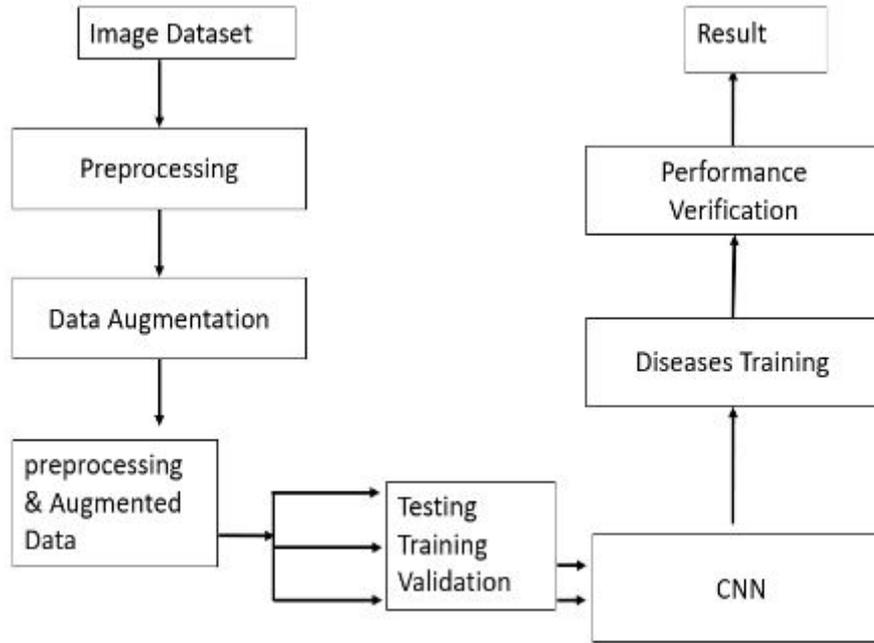
II. OBJECTIVES

The main objective of this study is to develop an automated system for detecting and classifying tomato leaf diseases using CNN. Specifically, this study aims to:

- 1 Develop a CNN model that can accurately detect and classify common tomato leaf diseases, such as early blight, late blight mold, bacteria spot, leaf mold, target spot, yellow leaf curl virus, two spotted spider mite, mosaic virus and septoria leaf spot.
- 2 Compare the performance of the developed CNN model with at different epochs.
- 3 Contribute to sustainable agriculture by providing a cost-effective, automated solution to identify tomato leaf diseases at an early stage, thereby enabling farmers to take preventive measures and reduce crop losses.

III. METHODOLOGY FLOW CHART

The following figure shows the methodology flow chart, it describes the way of approached to detect the tomato leaf diseases



Dataset Collection

The dataset used for this project was obtained from Kaggle. The dataset consists of 16,021 tomato leaf images with ten classes:

Tomato_Bacterial_spot, Tomato_Early_blight, Tomato_Late_blight, Tomato_Leaf_Mold, Tomato_Septoria_leaf_spot, Tomato_Spider_mites_Two_spotted_spider_mite, Tomato__Target_Spot, Tomato__Tomato_YellowLeaf__Curl_Virus, Tomato__Tomato_mosaic_virus and Tomato_healthy. The images were captured from different locations, seasons, and under different lighting conditions.

Dataset Statistics

The dataset consists of 16,021 images with ten different classes representing different types of tomato leaf diseases and a healthy class. Table 1 presents the class distribution of the dataset.

S.No	Type of tomato leaf disease	No. of images
1	Tomato_Bacterial_spot	2,127
2	Tomato_Early_blight	1,000
3	Tomato_Late_blight	1,909
4	Tomato_Leaf_Mold	952
5	Tomato_Septoria_leaf_spot	1,771
6	Tomato_Spider_mites_Two_spotted_spider_mite	1,676
7	Tomato__Target_Spot	1,404
8	Tomato__Tomato_YellowLeaf__Curl_Virus	3,209
9	Tomato__Tomato_mosaic_virus	373
10	Tomato_healthy	1,591

Preprocessing

Before training the CNN model, the dataset was preprocessed using several techniques such as data augmentation, normalization, and resizing. Data Augmentation is a very popular technique in image processing, especially computer vision to increase the diversity and amount of training data by applying random (but realistic) transformations. Data augmentation techniques such as rotation, flipping (horizontal and vertical), and random cropping were applied to increase the size of the dataset and introduce more variability in the data.

Normalization was applied to scale the pixel values between 0 and 255 to improve the convergence of the model during training.

Resizing was also applied to standardize the image size to 256x256 pixels to reduce the computational cost of training the mode and also send the images as batches as the batch size is 32.

Dataset Split

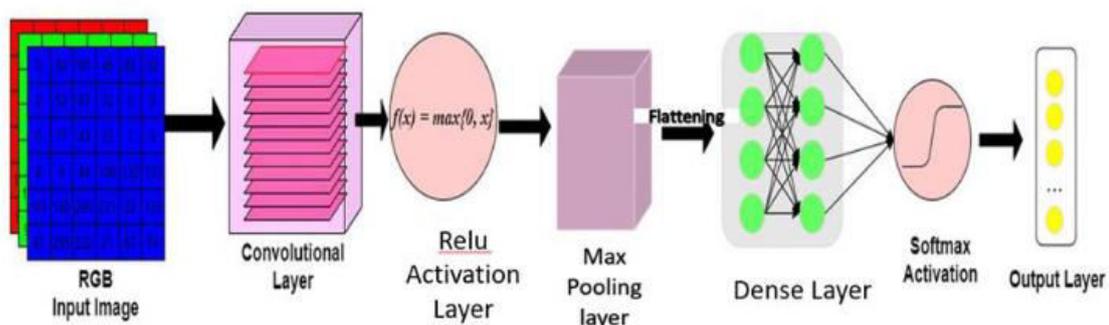
To evaluate the performance of the CNN model, the dataset was split into three subsets, namely the training dataset, validation dataset, and test dataset. The training dataset was used to train the model, the validation dataset was used to tune the hyperparameters, and the test dataset was used to evaluate the performance of the model on unseen data. The dataset was split randomly into the three subsets, with the training dataset containing 80% of the images, the validation dataset containing 10% of the images, and the test dataset containing 10% of the images. Table 2 shows the dataset lengths for the train, validation, and test datasets.

Train dataset length	400
Valid dataset length	50
Test dataset length	51
Total dataset length	(400+50+51)=501

The dataset split ensures that the model is trained on a sufficiently large dataset while also allowing for a fair evaluation of its performance on unseen data.

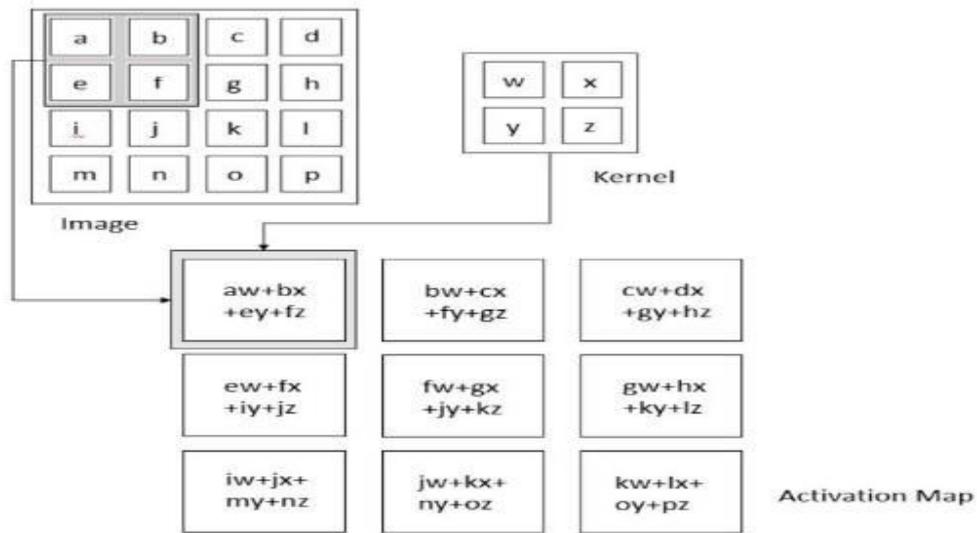
CNN Model Architecture

A Convolutional Neural Network (CNN) is a type of artificial neural network commonly used for image and video analysis, recognition, and processing. It is designed to automatically extract meaningful features from raw pixel data of an image, enabling it to recognize objects, faces, shapes, and patterns. CNNs are inspired by the structure and function of the visual cortex in the brain. The network is made up of a series of interconnected layers, each consisting of several neurons that perform simple computations on the input data. The layers are typically arranged in a specific order, including convolutional layers, pooling layers, and fully connected layers. The following fig shows the CNN model architecture with properly connected layers



Convolution Layer

Convolutional layers are the core building blocks of a CNN. They apply filters or kernels to the input image, sliding over the entire image and performing a dot product between the filter and the input pixels. This process generates a feature map, highlighting the regions of the input image that are most important for recognizing a particular pattern or object.

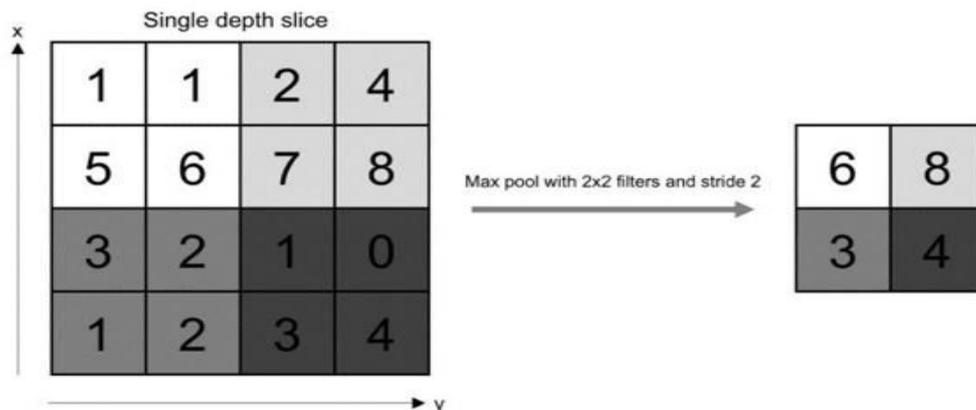


Relu Activation Function

The Rectified Linear Unit (ReLU) activation function is a widely used activation function in CNNs. It introduces nonlinearity into the network and improves its ability to model complex relationships between the input and output data. The ReLU function only allows positive values to pass through the neuron. When the input to the neuron is positive, the output is equal to the input value, and when the input is negative, the output is equal to zero. The ReLU function is defined as $f(x) = \max\{0,x\}$.

Pooling Layers

Pooling layers are an essential component of Convolutional Neural Networks (CNNs) used in computer vision applications. They are used to reduce the spatial dimensionality (width and height) of the input data while retaining its essential features. Max pooling is a commonly used type of pooling layer in which the maximum value within a defined region of the input feature map is selected and then passed on to the next layer. The size of this defined region (often referred to as the pooling window or kernel size) is typically set by the user.

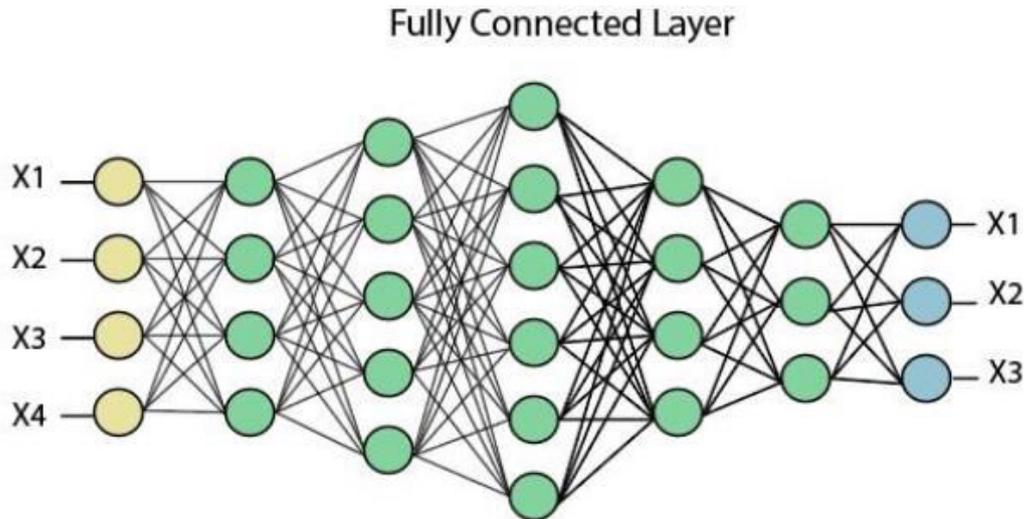


Fully Connected Layer

Fully connected input layer – The preceding layers' output is "flattened" and turned into a single vector which is used as an input for the next stage.

The first fully connected layer – adds weights to the inputs from the feature analysis to anticipate the proper label.

Fully connected output layer – offers the probability for each label in the end.



Softmax Activation Function

The softmax activation function is a mathematical function that is commonly used in artificial neural networks, particularly in multi-class classification problems. It is used to produce a probability distribution over the possible classes in the output layer of a neural network.

Sparse Categorical Cross entropy

The sparse categorical cross-entropy loss function calculates the cross-entropy loss between the predicted probability distribution and the target label tensor, taking into account the sparsity of the target label tensor.

Adam Optimizer

Adam (Adaptive Moment Estimation) optimizer is a stochastic gradient descent optimization algorithm used to optimize the parameters of a neural network during training. In other terms Adam is an optimizer used to update the weights and biases of a neural network during training.

Summary of The CNN Model

The construction of a Convolutional Neural Network (CNN) model with multiple levels, each consisting of convolution layers and max pooling layers. The model has six levels with decreasing output shape reduction, and the input data is sent as batches with a batch size of 32. The convolution layers have 32, 64, 64, 64, 64, and 64 filters, respectively, with a kernel size of 3x3 and ReLU activation. Each convolution layer is followed by a max pooling layer with a pool size of 2x2. The output shape of each layer is calculated based on the input size, kernel size, padding, and stride. The output parameters of each convolution layer are calculated based on the kernel filter size, number of filters, and number of previous filters, while the output parameters of the dense layer are based on the input and output channel numbers.

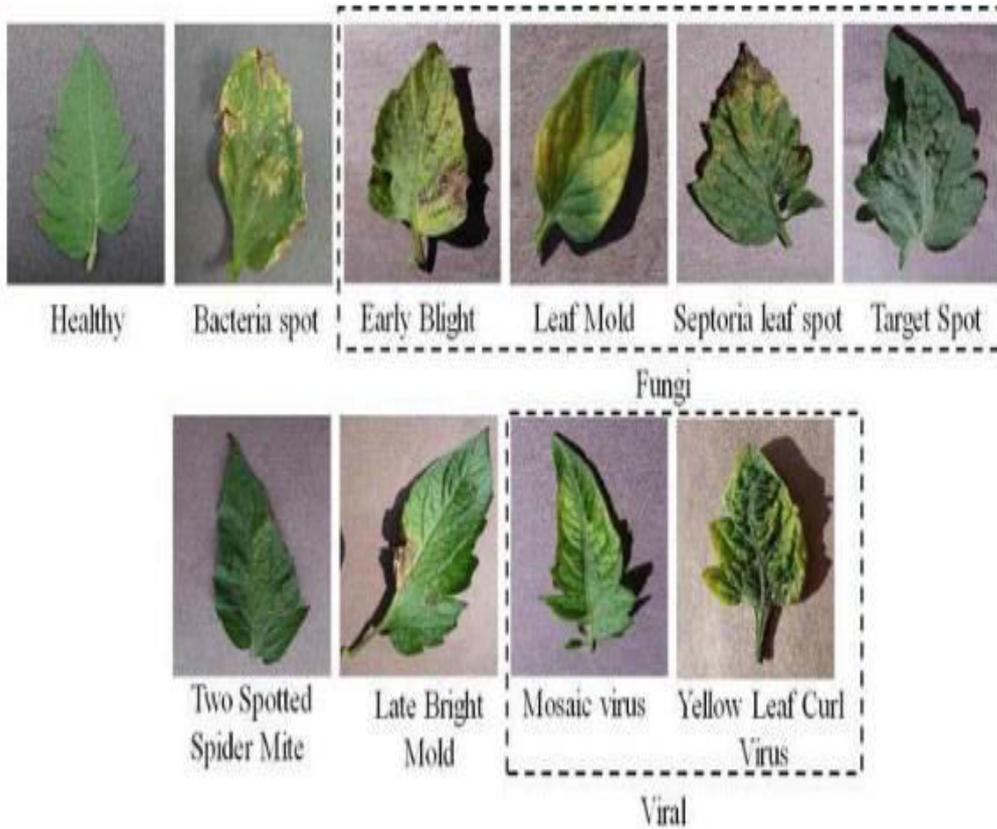
Algorithm Process

Step. 1 - Import the necessary libraries.

- Step. 2 - Set the input parameters, such as image size, batch size, and number of classes.
- Step. 3 – Load the dataset and preprocess the images
- Step. 4 -Define the CNN model architecture.
- Step. 5 - To train the CNN model at different epochs.
- Step. 6 – Evaluate the performance of the model and save the model with .h5 format
- Step. 7 - Reload the model and predict the tomato leaf images.

IV. RESULTS

The aim of this study is to detect the different types of tomato leaf diseases using convolutional neural networks (CNNs). The dataset used for this study is Plant Village, it taken from the Kaggle website, which contains ten types of tomato leaf images with labels indicating healthy and unhealthy leaves



Training results

The training results for a CNN typically include the loss and accuracy metrics during the training process. The loss metric is a measure of how well the model is performing on the training data and is typically calculated as the difference between the predicted output and the actual output.



V. CONCLUSION

In this project, we have presented a approach for tomato leaf disease detection using convolutional neural networks (CNN). We trained a deep learning model using a dataset of tomato leaf images, which was collected from various sources. The trained model was able to accurately detect the presence of ten common tomato leaf diseases, namely, bacterial spot, early blight, late blight, leaf mold, spectorial leaf spot, spider mites two spotted spider mite, target spot, yello leaf curl virus, mosaic virus and healthy. The proposed system is designed to provide an easy-to-use and efficient solution for detecting tomato leaf diseases. It uses a web interface page that allows end-users to upload images of tomato leaves and get real-time predictions on the presence of diseases. The system is capable of processing a large number of images quickly, making it ideal for use in agricultural applications..

VI. FUTURE SCOPE

Tomato leaf disease detection using CNN has great potential for future applications. Here are some possible future scopes for this technology:

- Real-time disease detection: The current project used pre-captured images of tomato leaves for disease detection. In the future, the system can be designed to detect diseases in real-time using a camera attached to a robotic arm that moves around the tomato plants. This would enable early detection and treatment of diseases, thus improving crop yields and reducing losses.
- Transfer learning: The current project used a CNN model. In the future, transfer learning can be used to improve the accuracy of the model. This would involve using pre-trained CNN models that have been trained on a large dataset and fine-tuning them on the tomato leaf disease dataset.

- Deployment on mobile devices: The current project was implemented on a desktop computer. In the future, the system can be optimized for deployment on mobile devices such as smartphones and tablets. This would enable farmers to use the system in the field for real-time disease detection and treatment.

REFERENCES

- [1] D. T. Mane and U. V. Kulkarni, "A survey on supervised convolutional neural network and its major applications," *International Journal of Rough Sets and Data Analysis*, vol. 4, no. 3, pp. 71–82, 2017.
- [2] Aravind KR, Raja P, Anirudh R. Tomato crop disease classification Using A Pre-Trained Deep Learning Algorithm, *Procedia Comput Sci.* 2018; 133:1040–7.
- [3] Karthik R, Hariharan M, Anand Sundar, Mathikshara Priyanka, Johnson Annie, Menaka R. Attention embedded residual CNN for disease detection in tomato leaves. *Applied Soft Comput.* 2020.
- [4] Zhong Yong, Zhao Ming. Research on deep learning in apple leaf disease recognition. *Comput Electron Agric.* 2020; 168:105146.
- [5] Selvaraj MG, Vergara A, Ruiz H, et al. AI-powered banana diseases and pest detection. *Plant Methods.* 2019; 15:92.
- [6] Girshick, R., J. Donahue, T. Darrell, and J. Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition.* Pages 580-587. 2014.
- [7] Fuentes A, Yoon S, Kim SC, Park DS. A robust deep-learning-based detector for real-time tomato plant diseases and pest's recognition. *Sensors.* 2022; 2017:17.
- [8] Picon A, Alvarez-Gila A, Seitz M, Ortiz-Barredo A, Echazarra J, Johannes A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Comput Electron Agric.* 2019;1(161):280–
- [9] Lowlesh Yadav and Asha Ambhaikar, "IOHT based Tele-Healthcare Support System for Feasibility and performance analysis," *Journal of Electrical Systems*, vol. 20, no. 3s, pp. 844–850, Apr. 2024, doi: 10.52783/jes.1382.
- [10] L. Yadav and A. Ambhaikar, "Feasibility and Deployment Challenges of Data Analysis in Tele-Healthcare System," 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-5, doi: 10.1109/ICAIIHI57871.2023.10489389.
- [11] L. Yadav and A. Ambhaikar, "Approach Towards Development of Portable Multi-Model Tele-Healthcare System," 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-6, doi: 10.1109/ICAIIHI57871.2023.10489468.
- [12] Lowlesh Yadav and Asha Ambhaikar, Exploring Portable Multi-Modal Telehealth Solutions: A Development Approach. *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, vol. 11, no. 10, pp. 873–879, Mar. 2024.11(10), 873–879, DOI: 10.13140/RG.2.2.15400.99846.
- [13] Lowlesh Yadav, Predictive Acknowledgement using TRE System to reduce cost and Bandwidth, March 2019. *International Journal of Research in Electronics and Computer Engineering (IJRECE)*, VOL. 7 ISSUE 1 (JANUARY-MARCH 2019) ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE).



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details