**PROJECT REPORT**

**Team#9**

**Pothole Detection based on Machine Learning and Deep Learning Models.**

***Abstract***—Roads are essential for daily transportation world- wide, but their aging and usage patterns can cause deterioration of the road surface, leading to a decline in quality. This deterioration often results in the formation of potholes and cracks on the roads, which can cause damage to vehicles or pose a physical danger to occupants, particularly in underdeveloped countries. Identifying potholes in real time can help drivers avoid them and prevent accidents. Furthermore, recording their locations and sharing them can assist other drivers and road maintenance organizations in taking prompt corrective measures.

In our attempt to address the issue of pothole detection, The paper aims to combine the latest advancements in technology. Our goal is to develop practical, reliable, adaptable, and modular solutions. To achieve this, we will compare the performance of Random Forest, a machine learning model, with CNN, a deep learning model, in detecting potholes. The experiments were conducted on both models using multiple datasets and a conclusion was drawn to bring out the benefits of the model with 99% accuracy using CNN and 95% accuracy using Random Forest.

Keywords:

Pothole detection, random forest, CNN

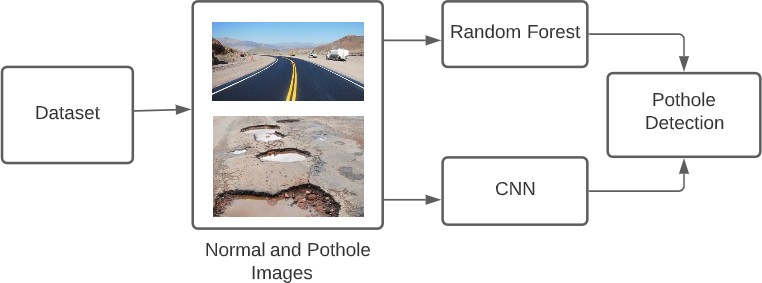
1. **INTRODUCTION**

The presence of potholes on roads is often an indication of poor maintenance, which may suggest underlying structural issues. Potholes can cause discomfort while driving and may result in expensive repair costs due to damage to the vehicle’s wheels, tires, and suspension system. A recent survey con- ducted by the Asphalt Industry Alliance in 2019 in England and Wales revealed a 24% increase in filling potholes. Potholes are responsible for about one-third of the mechanical problems experienced by drivers in the UK and cost them approximately £2.8 billion each year, according to a 2013 study by the Asphalt Industry Alliance. The compensation payments of £6.9 million provided to road users in England and Wales do not cover the people costs of resolving claims, which amount to 19.8 $million. Pothole damage is the primary reason for over 89% of claims made in England, marking an increase of 80% since 2018, as reported by the Asphalt Industry Alliance in 2019. While these statistics are worrying, implementing pothole detection systems can help to substantially decrease these figures by streamlining the repair process.

A 2018 study by Cycling UK indicates that 56% of people would cycle more frequently if the condition of roads, including potholes, were better. (Jones, 2018). So, it is essential to find and report potholes to let the proper authorities know how serious the problem is. This will guarantee that the necessary repairs are made on schedule. Due to recent rapid technological improvements, reliable sensors such as gyroscopes, accelerometers, GPS, electronic compasses, microphones, and cameras have been miniaturized and integrated into mobile devices Due to the prevalence of mobile phones and the lack of installation requirements for specialized hardware, this method of pothole detection is the least expensive and most effective. This study used real-time data gathered from mobile devices to identify potholes. The majority of current technologies to detecting potholes are less accurate, require expensive and specialized technology, or are not reliable enough to identify Potholes.

1. METHODOLOGY

Pothole detection devices help to prevent and mitigate various accidents that occur around the world. Many methods for detecting potholes have been proposed. The strategy proposed (Fig 1) here employs two approaches. CNN, a deep learning model, and Random Forest, a machine learning model. They are trained on multiple datasets and their performance is evaluated.

 Fig. 1

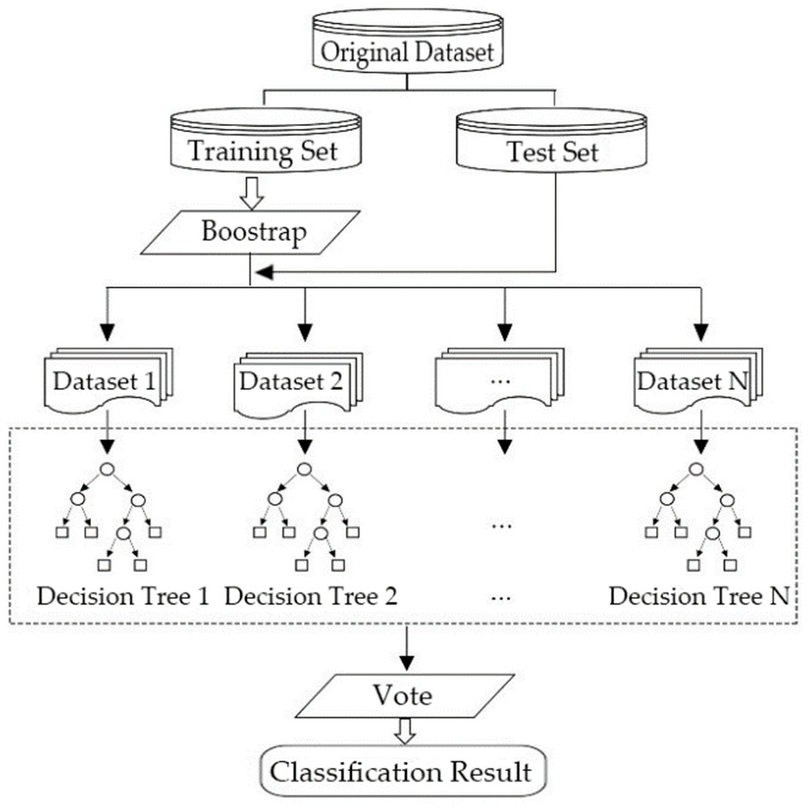
## Architecture of proposed method

The article compared pothole identification outcomes using two models, the random forest model being a machine learning model and the CNN model being a deep learning model. The article focuses on evaluating the accuracy and other performance metrics of random forest and CNN in the detection of potholes while working with various datasets to select the best model.

## RANDOMFOREST

Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. Here’s an overview of how Random Forest works and its architecture as in figure 2.

* 1. **Ensemble of Decision Trees:** Random Forest creates an ensemble of decision trees. Each decision tree is trained independently on a randomly sampled subset of the training data. This random sampling is done through a technique called bootstrapping, where each tree is trained on a different subset of the data.
  2. **Random Feature Selection:** During the training phase, a random subset of features is considered for splitting at each node of the decision tree. This feature selection helps to reduce tree correlation and enhances variety within the ensemble.
  3. **Tree Construction:** Each decision tree in the Random Forest is built using a recursive binary splitting mechanism. The goal is to locate ideal splitting points in the data that maximize separation between distinct classes or minimize impurity within each node. Depending on how the implementation is done, the splitting criteria could be based on Gini impurity or knowledge gain.
  4. **Prediction Voting:** Once the ensemble of decision trees has been trained, predictions are formed by aggregating the results from each tree. In classification tasks, the predicted class is selected via majority voting, in which each tree” votes” for the class label. For regression tasks, the predictions from each tree are averaged to give the final prediction.

Fig. 2. Random Forest Architecture

## CONVOLUTION NEURAL NETWORKS

Convolutional Neural Networks (CNNs) are deep learning algorithms specifically developed for image feature extraction which help in classifying images. The architecture of a CNN consists of various layers, as depicted in Figure 3. Each layer performs specific tasks, which will be described in detail below.

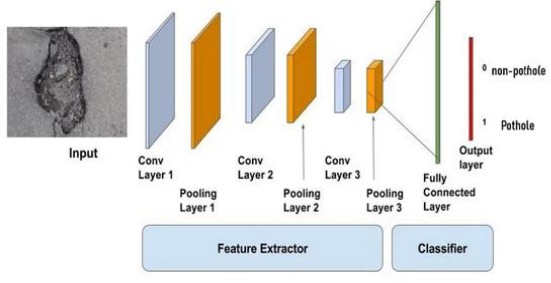
**The convulsion layer** of a CNN is critical for feature extraction as it is based on a kernel or filter and is applied to the image accordingly. The goal of this technique is to extract significant visual elements such as edges, gradients, and colors. The kernel passes the entire image in order to capture these features.

**Pooling Layer** is used to extract features that are not affected by rotation or position. Pooling is divided into two types. Specifically, Max pooling and Average pooling. When placed on a portion of an image, max pooling takes the maximum convoluted value of the kernel, whereas average pooling takes the average convoluted value. Following the convolution and pooling layers, the models can comprehend and extract image features.

**Fully Connected Layer** learns non-linear features from the output of a convolution layer. In each iteration of training, that output should be converted to a column vector and fed to a feed-forward neural network with back propagation. This aids the model in extracting the image’s prominent and low- level features. In turn, feature extraction from the model aids in the identification and classification of images.

**CNN with ReLu** (Rectified Linear Unit) activation function is commonly used after the convolutional layers. ReLU introduces non-linearity into the network, helping it learn complex relationships between features. The ReLU function returns the input value if it is positive, and zero otherwise. By applying ReLU activation to the outputs of the convolutional layers, CNNs can effectively capture and amplify important features while suppressing irrelevant or negative values.

ReLu(P)=0 if p≪ 0; *Pif P* ≫ 0

Fig. 3. CNN Architecture

**III. EXPERIMENTS AND RESULTS**

The experiments that were conducted to detect potholes are Random Forest and CNN algorithms as discussed. The potholes images in the dataset were varied in size and geo- metric shape. Detecting those potholes can help in improving road quality and prevent accidents. Python was used to implement algorithms. The detailed description of the architecture, dataset, and performance metrics of the proposed algorithm are discussed in detail.

## Methods and network training

Pothole detection is performed on both random forest and CNN. The datasets are trained in jupyter notebook for pothole detection. The platform and framework details are shown in Table 1. Parameters set to train potholes on random forest is seen in Table 2. The training is performed using CNN keras framework and parameters set for pothole classification is listed in Table 3. For one of the datasets, we resized the images to a size. As we resized the images the images got blurred so to make that images clear we have applied Cubic Convolution and sharpening methods to increase the image clarity and accuracy.

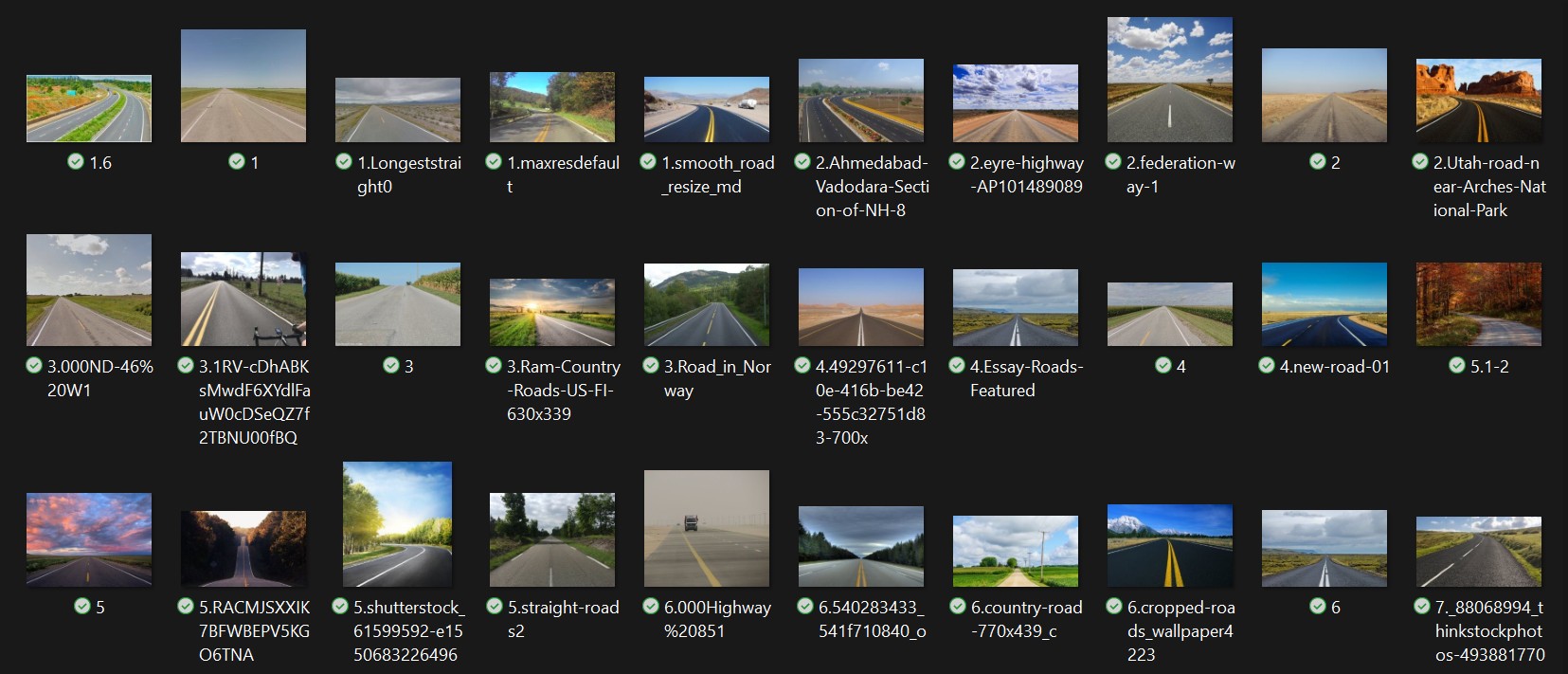
**Cubic Convolution:** Cubic convolution is a method for interpolating data in a two-dimensional space. In image processing, it is often used for resampling or upscaling images. The basic idea behind cubic convolution is to use a weighted average of neighboring pixels to determine the value of a pixel in the resampled image. The weights for each pixel are determined by a cubic function that is defined based on the position of the pixel relative to the target pixel.

**Sharpening Method:** Sharpening is a technique used to enhance the edges and details in an image. The goal of image sharpening is to increase the contrast along the edges of objects in the image, making them appear more defined and crisper.

## Dataset Collection

To get the best accuracy and analyze which model is giving the best accuracy, a total of 5 dataset have been taken where each dataset has subfolders named normal and potholes and each individual subfolder has the images of the potholes and normal images. Of these 5 datasets 3 of the datasets are different from each other and the other 2 datasets are combinations of any 2 datasets from the 3 datasets.

**Case 1:** Here the dataset has been taken from Kaggle website where the dataset has different dimensions for each image Fig 4. and Fig 5. and each subfolders had image of potholes and normal where normal has 367 images and pothole has total of 357 images.

Fig. 4. Normal images where each image is of different dimension.

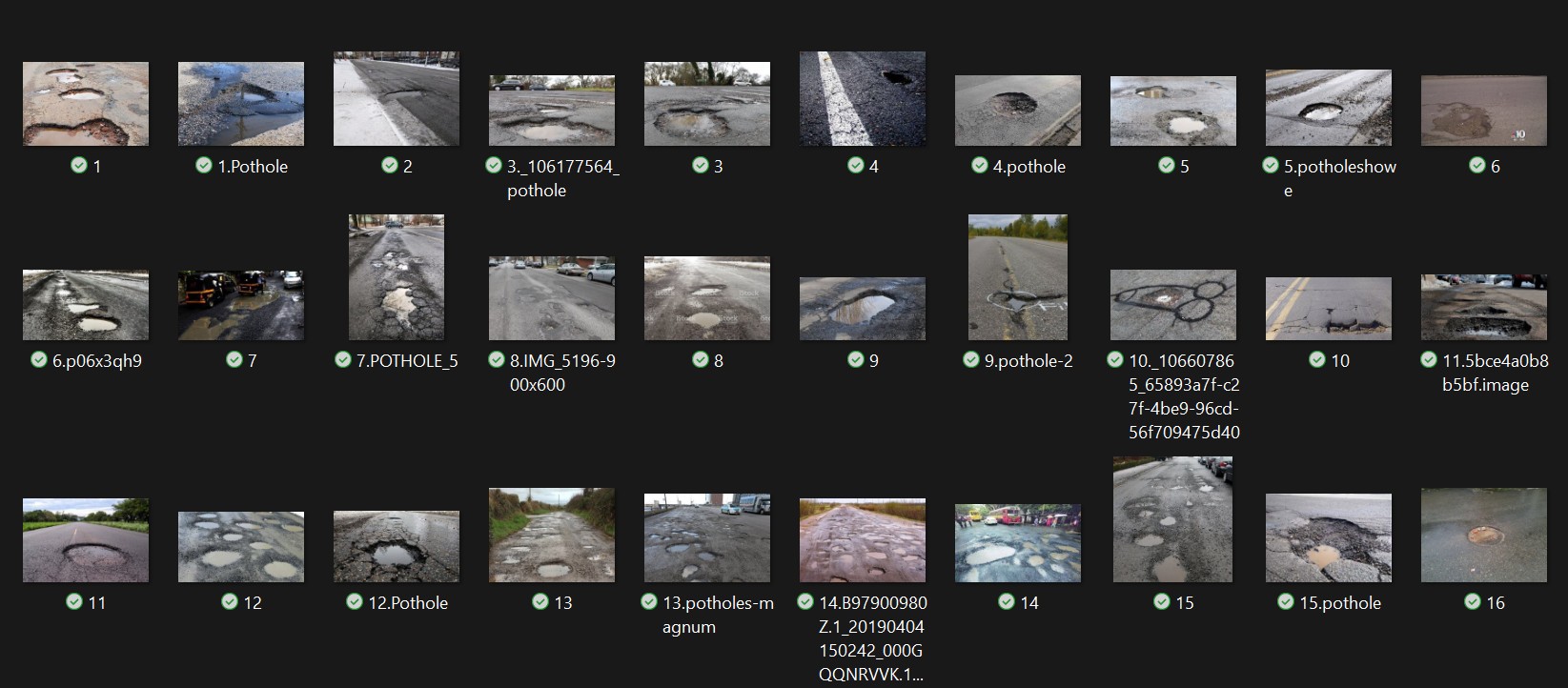


Fig. 5. Pothole images where each image is of different dimension

**Case 2:** This dataset has been taken from our friend’s project where the dimensions of the pothole and normal images are 64\*64 Fig 6. and Fig 7. in this dataset normal images are 339 and pothole images are 929.

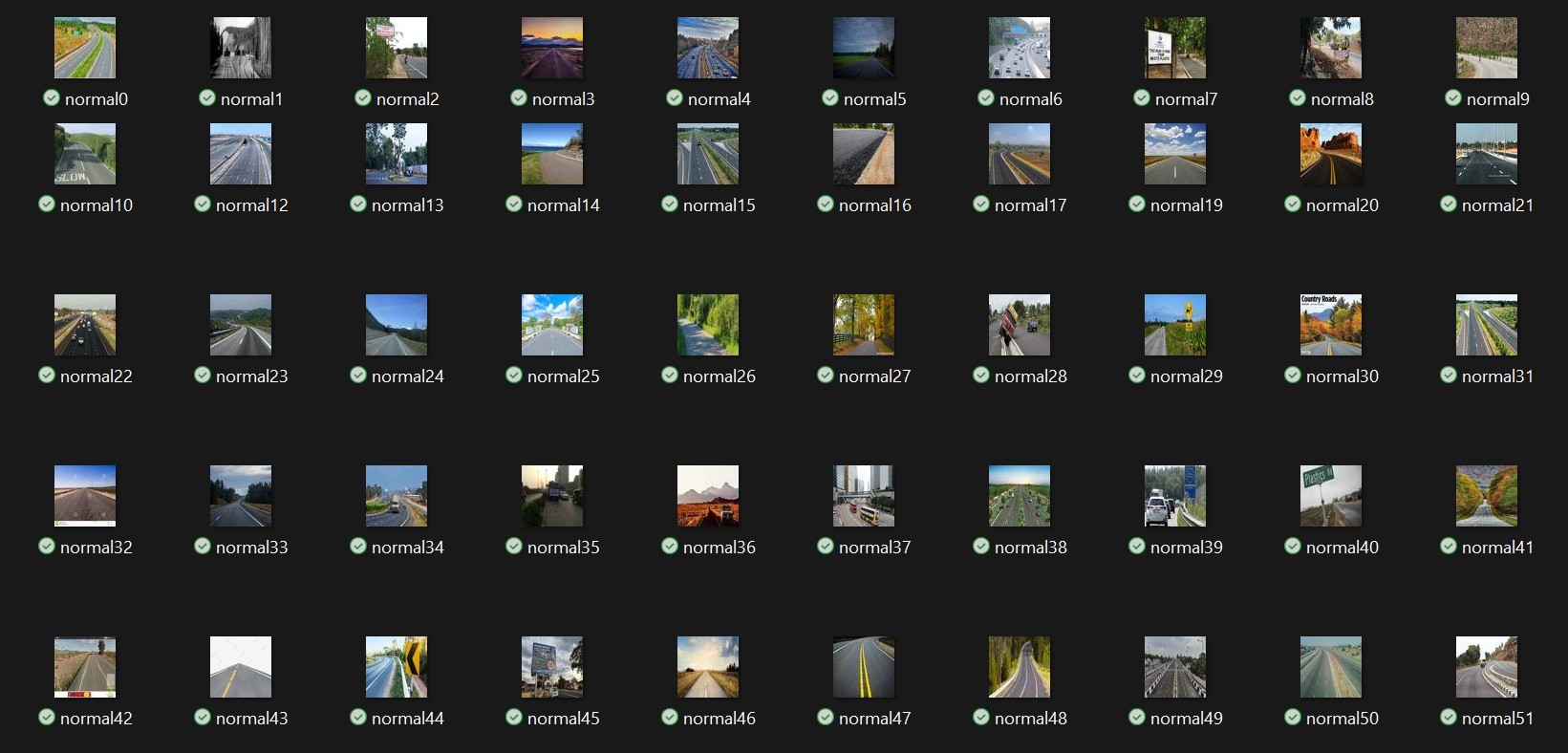


Fig. 6. Normal images of dimension 64\*64

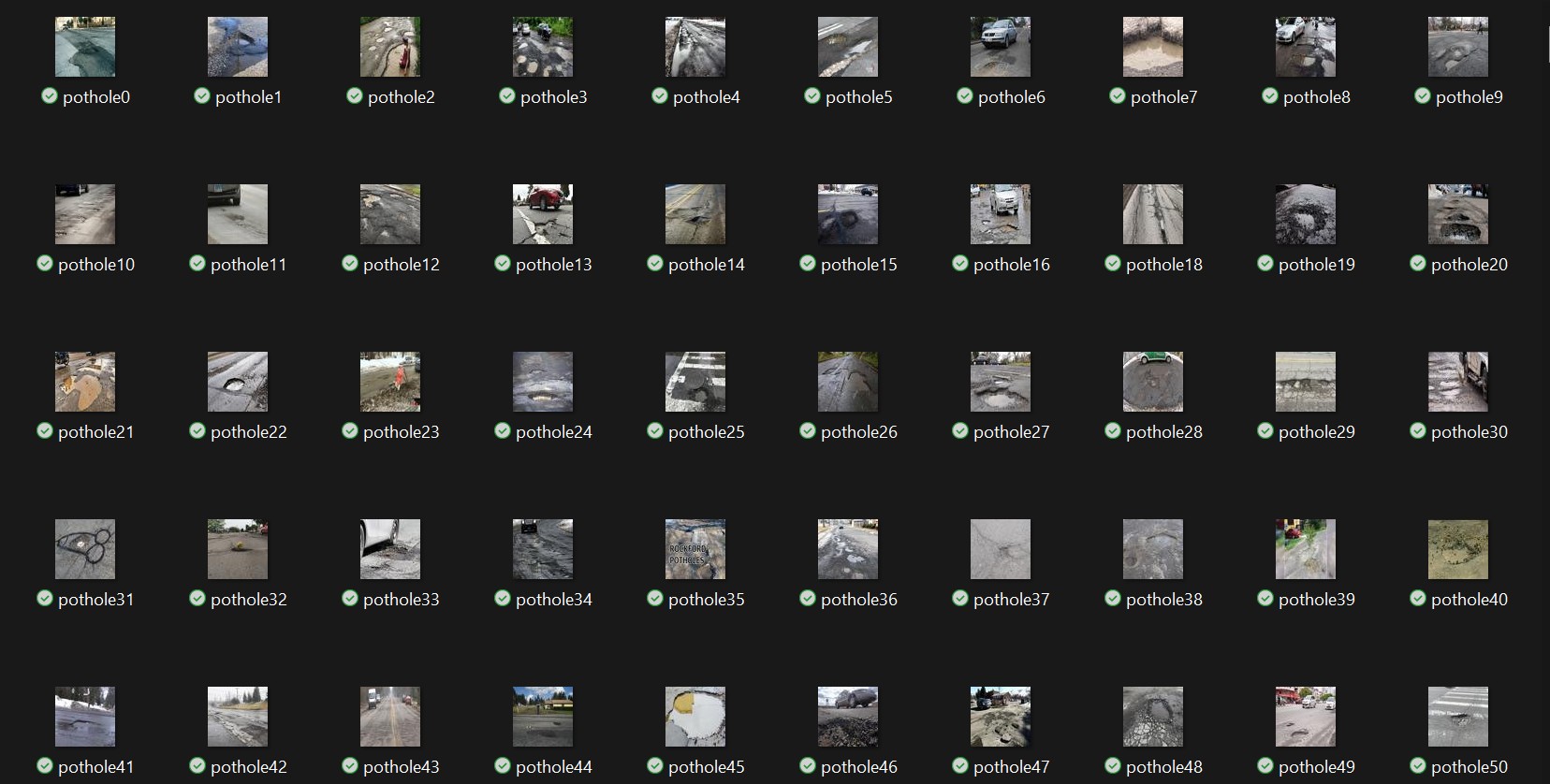
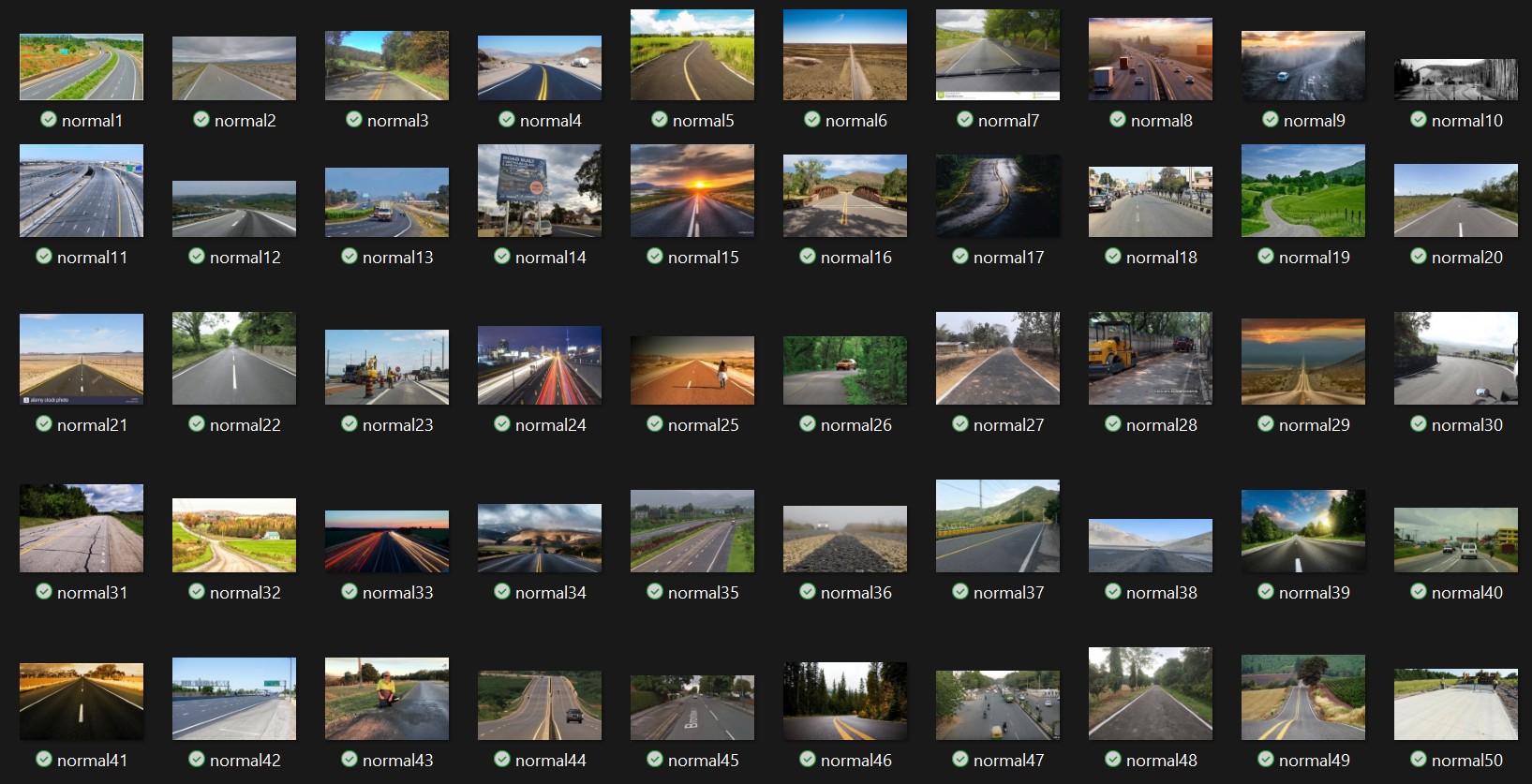


Fig. 7. Pothole images of dimension 64\*64

**Case 3:** This dataset is also taken from Kaggle where each image in this dataset consists of different dimensions Fig 8. and Fig 9. Here normal images are 351 and pothole images are 329.

**Case 4:** This dataset is a combination of the case 1 and case 2 dataset. Here the images are combination of different dimension images and 64\*64-dimension images. After combining the dataset, the normal images are 706 and pothole images are 1286.

**Case 5:** This dataset is a combination of case1 and case3. Where the dataset images are only different in dimension.



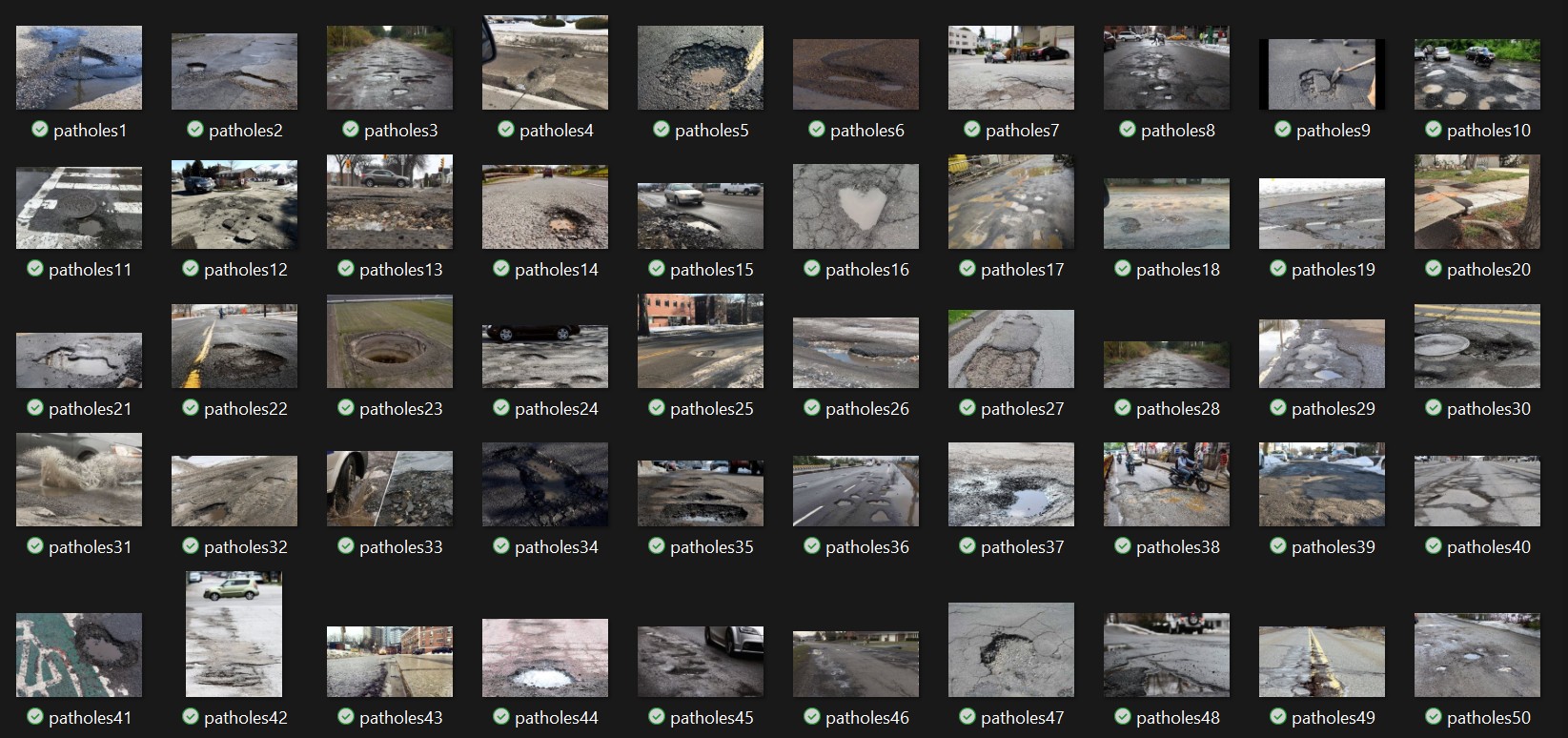
Fig. 8. Normal images where each image is of different dimension

Fig. 9. Pothole images where each image is of different dimension

Here the total count of normal images is 718 and pothole images are 686.

1. **Performance metrics**

Various metrics were employed to assess the performance of both Random Forest and CNN models.

TABLE I

FRAMEWORK DETAILS

|  |  |  |
| --- | --- | --- |
| Framework | Platform | Language |
| Random Forest | keras | Python |
| CNN | Keras | Python |

TABLE II

|  |  |
| --- | --- |
| Parameters | Values |
| Batch size | 256 |
| Learning rate | 0.2 |
| Nestimators | 100 |

TRAINING PARAMETERS ON RANDOM FOREST FOR POTHOLE CLASSIFICATION

* 1. ***Random Forest Classifier Performance Metrics****:* The metrics used to check performance here are 1. Accuracy 2. Precision 3. Recall 4. F1-score 5. Support

1. **Accuracy:** Accuracy is a statistical measure used to evaluate the performance of a machine learning model. It measures the percentage of correct predictions that the model

TABLE III

TRAINING PARAMETERS ON CNN FOR POTHOLE CLASSIFICATION

|  |  |
| --- | --- |
| Parameters | Values |
| Batch size | 256 |
| Learning rate | 0.2 |
| Epochs | 10 |

makes out of the total number of predictions. In other words, accuracy measures how well the model can correctly classify. The accuracy is given as

Accuracy = (number of correct predictions) / (total number of predictions)

1. **Precision:** The number of accurately identified potholes divided by number of detected potholes yields classification accuracy. Precision can be written as

Precision = TP / (TP + FP)

1. **Recall:** Recall is a performance metric that measures the proportion of true positive samples that are correctly identified by a classifier or model, out of all the actual positive samples. In other words, recall tells us how well a classifier can identify all positive samples in a dataset. It is the proportion of accurately observed potholes to a number of potholes in a dataset.

Recall = True Positives / (True Positives + False Negatives)

1. **F1-score:** F1-score is a measure of a classification model’s accuracy that combines precision and recall into a single metric. It provides a way to balance precision and recall, taking both false positives and false negatives into account. The F1-score is the harmonic mean of precision and recall, calculated as:

F1-score = 2 \* (precision \* recall) / (precision + recall) In pothole detection, the F1-score can be used to evaluate the performance of a pothole detection algorithm, taking into account both the accuracy of detecting potholes (recall) and the accuracy of identifying potholes specifically (precision). A high F1-score indicates that the algorithm can detect pot- holes accurately while minimizing false positives and false negatives.

1. **Support:** The support value is the number of true positive samples plus the number of false negative samples for a given class. In pothole detection, the support value for the pothole class would be the total number of potholes in the dataset, whether they were correctly or incorrectly classified. By taking the support value into account, evaluation metrics can provide a more accurate assessment of the model’s performance, especially when dealing with imbalanced datasets where some classes may have significantly fewer samples than others.

Here in the random forest, we find out the optimal number of n estimators that would yield the highest accuracy for the Random Forest classifier when applied to a given dataset. The output of n estimators and corresponding accuracies (fig 10) allows us to identify which value of n estimators produce the best model accuracy. Below fig is the graph of the case 1 dataset.

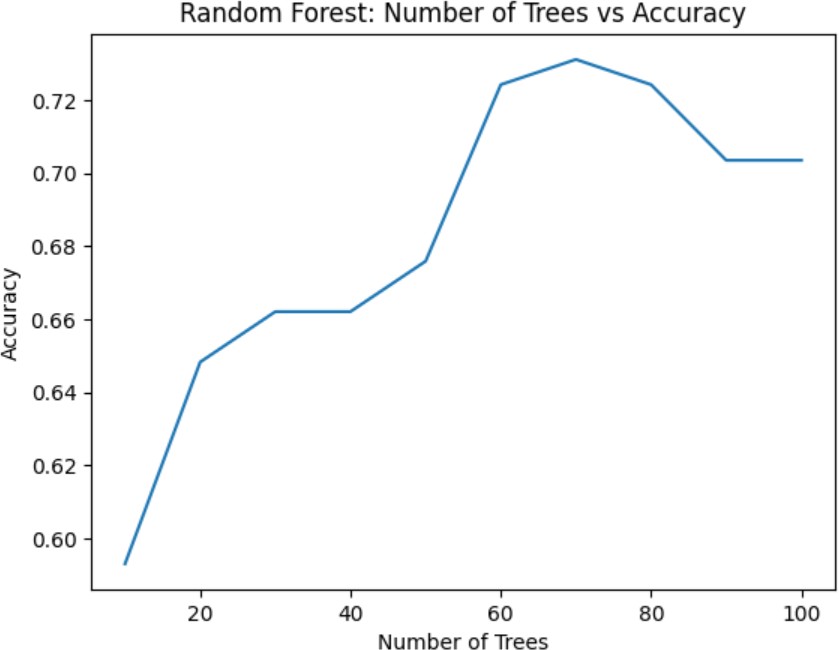


Fig. 10. Number of trees Vs Accuracy

* 1. ***Metrics for sequential CNN****:* To check the effectiveness of the CNN framework, different metrics are used along with those which are mentioned in the random forest classifier. Here we visualize the performance of a neural network during training and validation by plotting the loss and accuracy metrics over a given number of epochs. Fig 11 and Fig 12 shows the graphs of case 1 dataset.

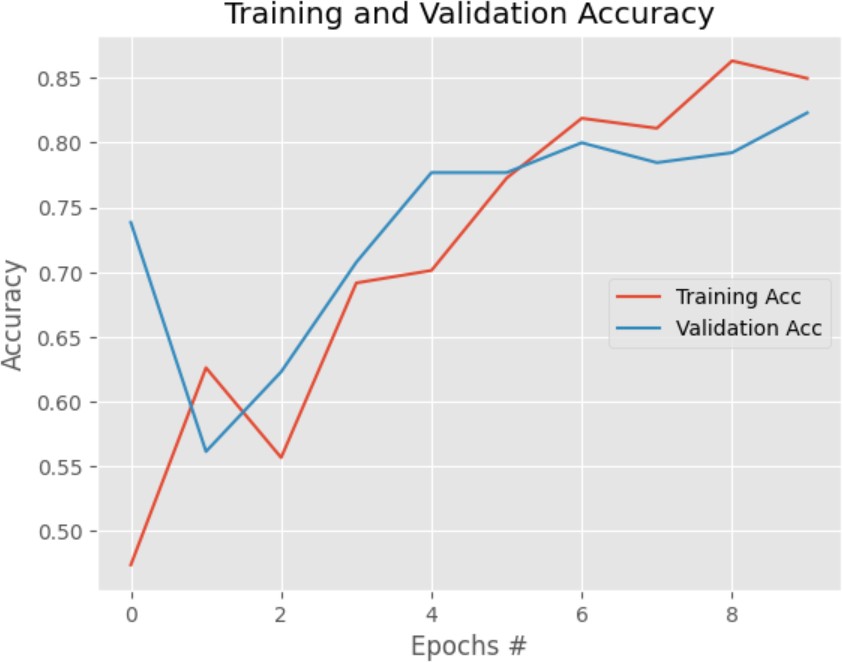


Fig. 11. Epochs Vs Accuracy

## Results and discussion

Here we have used two models Random Forest and CNN on the multiple datasets and here are the cases for each model.

1. *RANDOMFOREST:*

**Case 1:** The dataset used in this study was obtained from the Kaggle website, consisting of 367 normal images and 357 pothole images. We obtained the highest accuracy of 79% in our observations. We noticed that the performance was not as desired due to the small dataset size. However, compared to the total number of images, the results were relatively better. Additionally, we found that the accuracy decreased as we increased the test size, which can be attributed to the limited number of total images available.

**Case 2:** The dataset provided by our friend’s consists of 339 normal images and 929 pothole images. We achieved a maximum accuracy of 83.4% during our observations. Ad- additionally, we noticed that altering the parameters such as n estimators, random state, and test size had an impact on the results. Specifically, increasing the values of n estimators and test size led to better outcomes.

**Case 3:** This dataset is also taken from Kaggle where each image in this dataset consists of different dimensions fig [3.1]and fig [3.2]. Here normal images are 351 and pothole images are 329. We observed a maximum accuracy of 75% in our experiments. Based on our observations, we noticed that the accuracy varied when we changed the n estimators, and this variation was also dependent on different test sizes

**Case 4:** This dataset is a combination of the case 1 and case 2 dataset. Here the images dimension is combination of different dimension images and 64\*64- dimension images. After combining the dataset, the normal images are 706 and pothole images are 1286. We observed an accuracy of 87% and noticed that the results varied as we changed the test size. Interestingly, we observed an inverse relationship between the test size and the accuracy.

**Case 5:** This dataset is a combination of case1 and case

3. Where the dataset images are different in dimension. Here the total count of normal images is 718 and pothole images are 686. We achieved a highest accuracy of 95% in our observations. One possible reason for this high accuracy could be the quality of the images. Specifically, in case 4 of the dataset, we observed that different-sized images were combined, and when these images were enlarged, they became blurred, leading to lower accuracy compared to the rest of the dataset.

**Case 6:** Here the dataset is the same as in the case 2 but as the dataset images are too small to evaluate i.e. 64\*64 so to increase the image size we have resized the image properties to 256\*256 and then applied the sharpening and cubic convolution methods (4.1) to the resized dataset. Then the accuracy obtained now is 84.2%.

1. *CONVOLUTION NEURAL NETWORKS:*

**Case 1:** The dataset used in this study was obtained from the Kaggle website, consisting of 367 normal images and 357 pothole images. The observed results showed an accuracy of 94%. Interestingly, we noticed that as the test size increased, the accuracy decreased. This can be attributed to the fact that with a larger test size, the training dataset becomes smaller, potentially leading to a decrease in accuracy.

**Case 2:** The dataset provided by our friend’s consists of

339 normal images and 929 pothole images. During our analysis, we achieved an accuracy of 86.5Interestingly, we observed that the accuracy remained relatively stable even when we changed the test size. The variations in accuracy were minimal, indicating that the test size had a limited impact on the overall performance.

**Case 3:** The dataset used in this analysis was sourced from Kaggle and included 351 normal images and 329 pot- hole images. We achieved an accuracy of 93% during our evaluation. However, we observed that as we increased the test size, the accuracy began to decrease. This decrease can be attributed to the fact that a larger test size reduces the amount of available training data, potentially leading to a decline in overall accuracy.

**Case 4:** During our analysis of this combined dataset from case 1 and case 2, we observed a remarkable accuracy of 97%. This result suggests that having more images in the dataset positively impacts the accuracy of the model.

**Case 5:** This combined dataset consists of data from case 1 and case 2. During our analysis, we observed a remarkable accuracy of 99%, which is the highest among all the cases evaluated.

**Case 6:** Here the dataset is the same as in the case 2 but as the dataset images are too small to evaluate i.e. 64\*64 to increase the image size we have resized the image properties to 256\*256 and then applied the sharpening and cubic convolution methods (which is discussed in IV. A) to the resized dataset. Then the accuracy obtained now is 87.3%.

## Performance analysis between Random Forest and CNN

The performance of the two frameworks is described in Table 4. Table 4 demonstrates that CNN performs considerably more accurately than random forest, yet CNN still requires more training time than random forest. However, Random Forest performs well at classifying potholes and non-potholes while requiring less training time and resources. One can choose the model based on the availability of resources. Table 5 and Table 6 shows the Accuracy, Precision, Recall, F1- score for normal and potholes in random forest and CNN for case 1 which is dataset1.

TABLE IV

ACCURACY RESULTS OF RANDOM FOREST AND CNN

|  |  |  |
| --- | --- | --- |
| Dataset | RandomForest | CNN |
| Case 1 | 79% | 94% |
| Case 2 | 83.4% | 8  6.5% |
| Case 3 | 75% | 93% |
| Case 4 | 87% | 97% |
| Case 5 | 95% | 99% |
| Case 6 | 84.2% | 8  7.3% |

## Based on the performance analysis of both random forest and CNN models, it is evident that CNN yielded superior results compared to random forest. This indicates that the deep learning model CNN outperformed the machine learning model random forest. However, it’s important to note that these observations were made considering the small datasets used. It is possible that if the dataset size is increased, random forest might yield better results. As of now, based on the available information, CNN has been shown to be the better performing model.

TABLE V

VALIDATION RESULTS OF RANDOM FOREST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F-score | Support |
| Normal | 0.73 | 0.74 | 0.75 | 72 |
| Pothole | 0.75 | 0.73 | 0.74 | 73 |

TABLE VI

VALIDATION RESULTS OF CNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F-score | Support |
| Normal | 0.93 | 0.68 | 0.78 | 37 |
| Pothole | 0.74 | 0.94 | 0.83 | 36 |

## Comparative Study

Pothole detection is carried out using several methods, both classical and deep learning methods. Several methods are listed for comparison. As can be seen in Table 7

1. CONCLUSION

Pothole detection is crucial for preventing accidents and vehicle damages. Therefore, this paper focuses on conducting a comparative analysis between two machine learning models, Random Forest and CNN, to evaluate their effectiveness in detecting potholes. Multiple datasets were used to assess their performance, and it was observed that CNN exhibited faster speed and superior results compared to Random For- est. Additionally, the study noted variations in performance across different dataset types. Although the current results are promising, the paper acknowledges the potential for further improvement by increasing the dataset size to enhance the models’ performance.

REFERENCES

1. G. Srinidhi and Renuka Devi S M (2020). Pothole Detection using CNN and AlexNet . ICCIP
2. Anup Kumar Pandey, Rahat Iqbal,Tomasz Maniak, Charalampos Karyotis,Stephen Akuma,Vasile Palade (2022). Convolution neural networks for pothole detection of critical road infrastructure.
3. Mohan Prakash B1\* and Sriharipriya K.C1 (2022). Enhanced pothole detection system using YOLOX algorithm.
4. Bergstra, J., Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(2).
5. Shuai, L., Chenxi, Y., Donghai, L., Hubo, C. (2016). Integrated processing of image and GPR data for automated pothole detection. Journal of Computing in Civil Engineering, 30(6), 04016015. https://doi.org/10.1061/(ASCE)CP.1943- 5487.0000582