FAULT DETECTION OF SATELLITE ANTENNA INSTALLATION USING MACHINE LEARNING

Gallage Yasanthi Vanodya Perera

(209363M)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

July 2022

FAULT DETECTION OF SATELLITE ANTENNA INSTALLATION USING MACHINE LEARNING

Gallage Yasanthi Vanodya Perera

(209363M)

Thesis submitted in partial fulfilment of the requirements for the degree Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

July 2022

Declaration of the candidate & Supervisor

I declare that this is my own work and this thesis does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: . UOM Verified Signature ……...…

Name: G. Y. Vanodya Perera

The above candidate has carried out research for the Master's thesis under my supervision.

Signature of the supervisor: . UOM Verified Signature ……

Name: Dr. Thanuja D. Ambegoda

Abstract

With rapidly growing customer expectations, satellite antenna has become an integral part of day-to-day life nowadays. Due to the growing demand, the number of installations increase rapidly. When the installation quantity increases, the installation quality decreases gradually. This leads to more customer complaints. Currently, a verification team does a second-round check on the installation and the signal level. Even though this help in identifying installation issues before customer complaints, this has become a failure due to operational cost with the growing number of installations. The proposed method is based on an image classification technique using Convolutional Neural Networks (CNN) along with a transfer learning technique using the well-known model, VGG16. The experimental data set was created by capturing images of correct and incorrect satellite installations. The correct installation positions are covered with a set of predefined image templates (3 test scenarios). Images of the satellite installations were captured according to the templates and they were tagged as correct or incorrect by eye-bowling the installation. A basic CNN model, a VGG16-CNN model, and a random forest classifier were compared by evaluating the model accuracy and the balanced accuracy. The VGG16-CNN model was selected with the best performance. The average accuracy and the average balanced accuracy of the final model are obtained as 85.8% and 86.1% respectively. This study was performed on a Windows 10 Pro 64-bit machine with an Intel i5-7200U CPU operating at 2.50GHz and 16 GB RAM. The prediction time was fast with a mean time of 0.5 seconds per image. Experimental verification was done in the field and an average accuracy of 90.56% was obtained. With these prediction models integrated, the operational cost can be reduced and the coverage can be increased significantly.

Keywords: Fault detection, Image classification, Satellite, Antenna, CNN, VGG16

Dedication

To Sankha

Acknowledgments

This project would not have been possible without the support of many people. Many thanks to my supervisor, Dr. Thanuja D. Ambegoda, Department of Computer Sciences, University of Moratuwa, for the valuable pieces of advice, guidance, and suggestions given to me throughout the research project.

I would like to thank everyone at Linear Squared (Pvt) Ltd who helped me to use the necessary data and successfully complete this dissertation. I really hope that the results of this study will pay you back.

I would like to thank my loving parents for everything they have done for me over the years. None of my academic achievements are possible if not for their support and affection all along.

My special thanks go to my husband, Dr. Sankha Perera for being with me through thick and thin. Thank you Sankha, for supporting me to balance my work life and work on this project during my pregnancy. You made it very easy for me.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Background

The need for satellite television antennas is constantly increasing. The number of installation requests for service providers rises as a result of this. This experiment was carried out with one of Sri Lanka's most well-known satellite television service providers. Customers typically receive a comprehensive solution from service providers, which includes both connectivity and infrastructure. The system's infrastructure will be set up at the customer's location once requested. Over 1500 installations are typically completed by a group of well-trained teams across the country in a month's time. The service provider's principal goal is to reduce the number of client complaints.

However, because infrastructure construction takes place in the field, there is a visibility gap for the service provider to assess installation quality. Furthermore, sending follow-up verification teams to each installation location to inspect and evaluate the quality of each physical installation is not practical. Due to the high operational costs of the installation verification teams, a sample of total installations is selected at random and only those are verified to ensure that the installation process is running up to standards. (about 20% of total monthly installations)

1.2 Research problem

When the operational cost of such quality control processes rises, sampling is a solution, according to the literature. However, in this specific use case, assessing the quality using a set of samples from the entire base of installations does not provide a reliable indicator of overall installation quality. Customer satisfaction will also be inconsistent if quality inspections are limited to a single sample.

The primary objective of this study is to provide a fault detection solution using Machine Learning that could perform real-time verification to help service teams to verify the infrastructure during the installation process itself. Taking this solution to

the installation teams in the field can improve the quality of the antenna installations and reduce the number of customer complaints by increasing the verification coverage significantly with a low operational cost.

As the initial step for this matter, this study would suggest the best Machine learning algorithms that could do quality checks by investigating the characteristics of images of the installations. In other words, this research outcome would be the best machine learning algorithm that could identify the incorrect installations by processing images of the installations. This research can be extended to a second step, where the algorithms are integrated into a mobile application for real-time processing. Then, the installation teams can take pictures themselves after completing an installation to get the real-time response from Machine Learning algorithms in the back-end.

1.3 Research Objectives

- To design the most critical test scenarios to verify the installation quality after it is installed in the field.
- To train a classification model to identify which installation images are correct and which are erroneous.
- To select an evaluation criterion that can be used to compare image classification models to select the best
- To come up with ways that can improve image classification model accuracies by using a smaller number of training image sets.
- To test the predictive models in the field

2. LITERATURE REVIEW

2.1 Image classification

Nowadays, image classification is very popular and has become a trend among data scientists and developers, especially since data volumes in various industries have increased. Image classification use cases can be found in various day-to-day applications such as cancer detection, quality control (fault detection), traffic control systems, animal monitoring, automatic harvesting, etc.

The most popular area for image classification using big data volumes is using deep learning. Deep learning is known in the real world as it has the ability to act like a person. It is a pretty basic process for people to classify items in order to determine whether or not an image is correct. Even if we try to teach a computer how to do it, it's difficult to expect machines to produce the same results as people. The system should be able to handle hundreds, if not thousands, of input data to improve the efficiency of the 'training' process. [1]

Image classification has also benefited from machine learning approaches. However, there are grey areas here and there in all these approaches that are still under the research phase. [2]

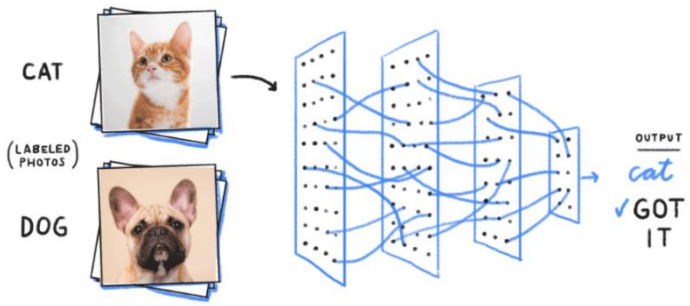


Figure 2.1 Image Classification

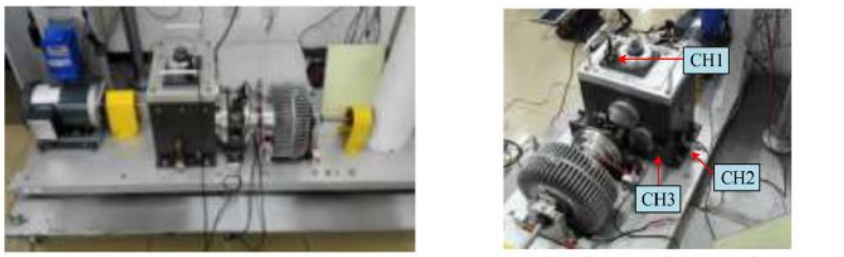
2.2 Similar fault detection applications

Even though various types of image classification [3] use cases can be found on fault detection of medical applications and other machinery installations, a specific use case for this kind of satellite antenna installation is rare to find.

Defect detection of bearing and gears

Bearings and gears have always been important components in intelligent manufacturing. When the rotating machine's transmission components fail, the accompanying vibrations have an impact on the overall process and safety in industrial manufacturing, making problem diagnostics critical. Many professionals and academics have worked on bearing and gear failure processes and defect diagnosis methods using vibration signals, resulting in the creation of equipment health monitoring.

A modified CNN model for defect detection was proposed in this study [4]. There are five types of layers in the proposed CNN model's structure including a bottleneck layer. The transformed feature maps are fed into the input layer, which then transposes the image. Two examples of experiments are used to test the recommended methodologies. The defects on the wind power test rig include various sophisticated bearing and gear failures. The cavitation, impeller imbalance, and misalignment issues on the centrifugal pump test rig. The suggested model's classification accuracy was compared to traditional ML models like ANN, SVM, and DBN. The suggested model's mean prediction accuracy is significantly higher than the compared models and they are stated as 99.47% and 97.32% for the two experiments explained above.



(a) Wind power test rig (b) Installation positions of sensors

(a) Outer bearing fault (b) Inner bearing fault (c) Gear tooth fracture (d) Gear tooth root fault

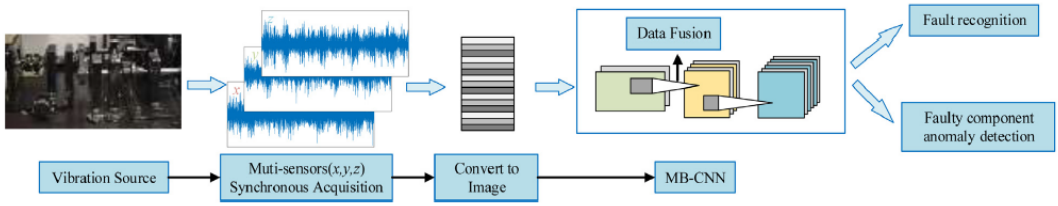


Figure 2.2 Fault components and the flow chart for the proposed method

Fault detection of vehicle pipes

Another use case [5] is a fault detection solution that can evaluate vehicle pipes and determine whether they are normal or not. As a second step, the solution extends to discard the defective pipe by removing the actuator. Using a modified Hough transform, the inspection method detects the segments in a side-view image and estimates measurements such as the angle of the pipe and the diameter. The program finds the sizes of inner circles and the location of their center using a front-view picture. The diameter of the pipe that is observed using the inspection algorithm is used to classify the pipe's condition as excellent or bad. The entire check is said to take less than a second.

Fault detection of antenna arrays

Furthermore, for mmWave wireless communication systems, a unique DL-based antenna array detection scheme is provided. [6] MmWave communication is a promising 5G system communication technology. The proposed technique can determine whether or not the antenna array is defective. It can also locate the antenna with a failure using distribution classification if certain assumptions are met. To detect and find flaws, two deep neural networks (DNN) of differing complexity are created. The basic network is intended to discover flaws at a minimal cost, whereas the precise network begins to find them once the former model has detected them. Because the data set split is unbalanced, the authors used 'recall' to test the model's performance using (faulty) samples. At high SNR (Signal to Noise Ratio), the accuracy of fault location within antenna blocks in all sizes achieved nearly 100% recall.

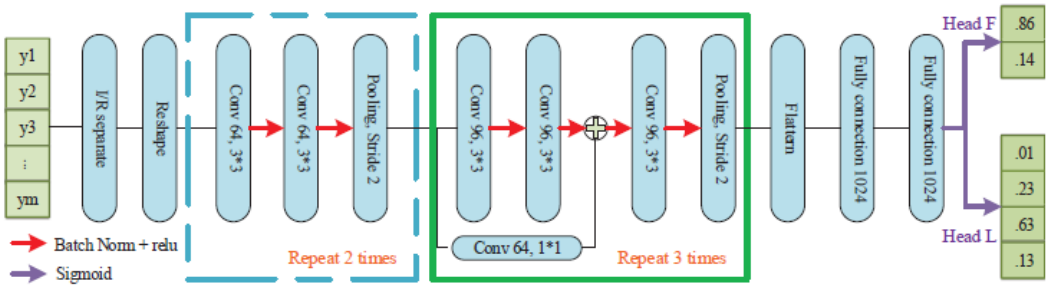


Figure 2.3 Structure of the proposed CNN-based fault location network

Fault detection of power plants

The majority of countries is now heavily investing in large-scale renewable energy installations. To continue generating the desired amount of energy on a regular basis, such installations require sophisticated maintenance and monitoring systems, primarily on the structural and surface parts. The proposed image-based monitoring approach [7] is said to be one of the most promising alternatives when compared to other inspection methods because of its effectiveness and low cost. Drone inspection photographs can be used to keep an eye on the power plants on a frequent basis.

Through comparing the outcomes of the validation set, different types of CNN models (EfficentDet D0 to D5, YOLOv3, YOLOv4, and YOLOv5) were trained. The YOLOv5 model has proven to be the most accurate, but it is also the most resource-intensive. As a result, the authors conclude that the YOLOv4 model is a preferable design choice for detecting faults in power plants. They also believe that this technology will greatly cut maintenance costs while also increasing the durability and lifetime of huge power plant systems.

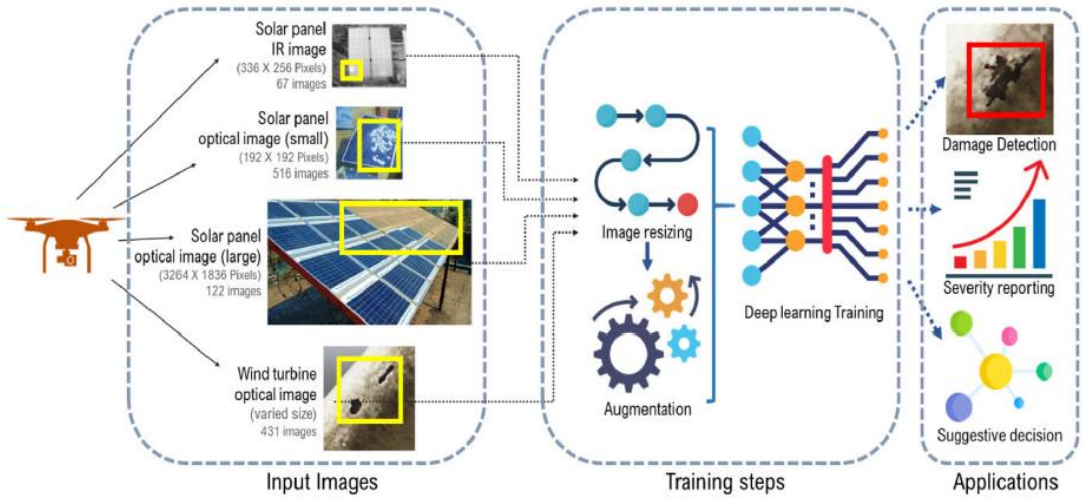


Figure 2.4 The proposed framework for surface damage detection

Fault detection of high-voltage insulators

Another use case [8] is the fault detection method for high-voltage insulators. The defect in this use case is a burn-mark. An insulator is one of the most important components of a high-voltage transmission line. The burn detection approach has been broken down into many steps by the authors. The recognition of insulators takes place in the first step, which is based on image processing. The main algorithms for this step are the symmetry detection and grabcut algorithms. They convert the RGB color image to grayscale and use basic filters to decrease noise. After recognizing insulators in an image, the burn-mark detection algorithm is used to determine the region of interest. The burn-mark method was developed using four main criteria. The color criteria

establish the color range. The shape criteria is used to eliminate blobs that are not circular or linear. To lessen the computational burden, the area around the connection is cropped using location criteria. To avoid erroneous detection, the size requirement filters off the number of pixels. The authors find that the detection rate of the proposed algorithm is 80.43% after evaluating the suggested method utilizing three primary criteria: hit, miss, and fail.

Damage detection of solar panels

As the surface of the solar panels are exposed, they are prone to failure. By processing solar panel photos collected using thermal cameras and RGB cameras, a fault detection solution is presented [9]. The RGB image is used to locate the solar panel, while the thermal image is used to identify the defective solar panel. The authors have used the thermal camera to locate the location of the photovoltaic panel based on the features of the solar panel problem. The RGB photos are transformed to grayscale images in the proposed system's initial step of pre-processing to determine the solar panel array's boundary. They have used information from the thermal image to detect a damaged solar panel. In the thermal image, the damaged solar panel cells exhibit certain peculiar patches that are highlighted or darkened. The authors conclude that they have put the solar power plant to the test to see if it can be used in the real world.

Disease detection of plants

Plant diseases can be defined as a fault in plants. A plant disease detection solution is proposed [10] using deep learning methods. The generated model was able to detect the presence of leaves with 13 different disorders. A new plant disease picture database was used, with over 3,000 original photographs gathered from various internet sources and enhanced with appropriate modifications. The CaffeNet architecture serves as a starting point, however, it has been updated to support the 15 classes in this study. The network is made up of 5 convolutional layers, 8 learning layers, and 3 fully connected

layers. The last layer was updated and the softmax layer was parameterized to meet the study's needs. The trained model's overall accuracy was 96.3% in the end.

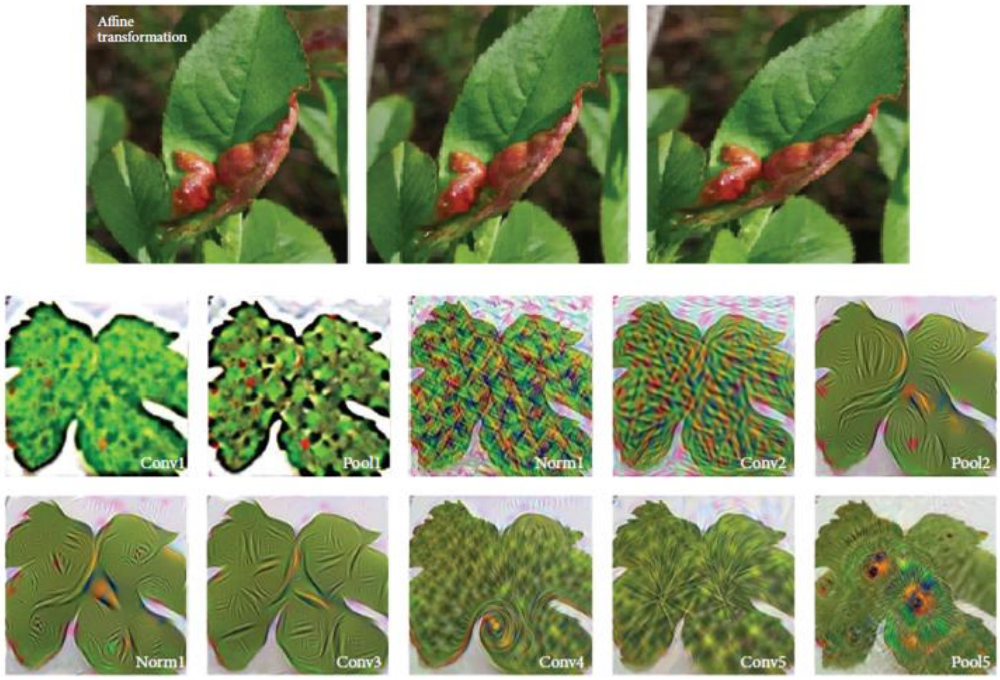


Figure 2.5 Input and output images to the CNN model

Fault detection of bearings

Condition monitoring is necessary to reduce operating expenses, extend the life of machines, and improve operational uptime. Condition monitoring is a technique for inspecting the state of a machine and detecting problematic parts. Vibration analysis is a well-established approach for condition monitoring of rotating machinery as vibration patterns differ based on the defect or machine condition. Automatic defect detection is primarily based on manually-engineered features. Unfortunately, designing and interpreting such characteristics requires a high level of human knowledge. The over-head of feature engineering for individual faults needs to be lowered as much as feasible to enable non-experts in vibration analysis to do condition monitoring.

Two approaches for detecting bearing faults are proposed in this study [11], [One is based on feature engineering, while the other is based on feature learning.] The authors collected high-frequency vibration signals as unbalanced bearings and used a logistic regression classifier to classify the situation in the feature engineering-based study.

The authors' goal in the feature learning-based analysis was to determine the exact bearing situation regardless of balance or imbalance. Four bearing conditions were discovered, and the bearings were classified into one of the four conditions using a random forecast classifier and a support vector machine. The authors presented a CNN classifier to compare the feature learning-based outcomes. The goal was to learn the patterns of change in the joint accelerometer signals in order to distinguish between the complex bearing conditions. The proposed design, which produced the best results in the experiment, comprises a 64-unit convolutional layer followed by a 200-unit fully-connected layer.

When compared to a traditional manual feature extraction strategy, the CNN-based method improves classification accuracy by about 6% without relying on considerable domain knowledge for problem detection.

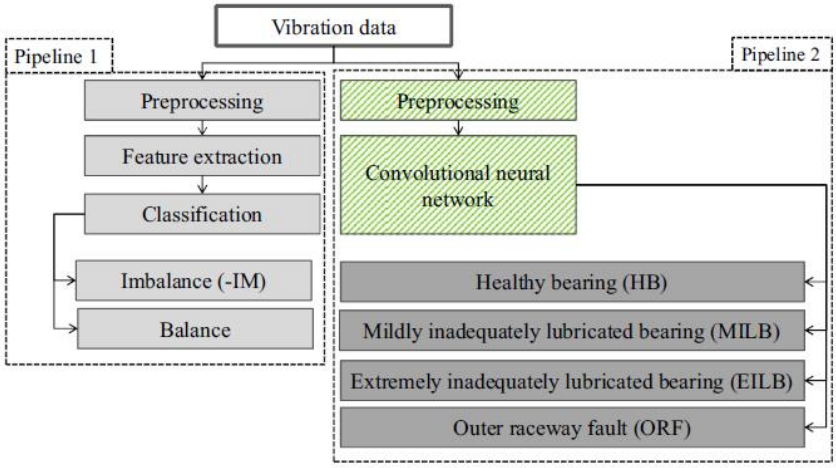


Figure 2.6 Architecture for the two pipelines used

Detection of vehicles at automatic charging systems

The use of vehicle type classification in everyday life is critical. It can be seen in multiple places such as in advanced monitoring systems, highway automatic charging systems, and illegal pre-emption of way detection.

Vehicle categorization approach based on CNN was proposed in this research [12]. They've done data augmentation that uses modifications on the input data to create new training instances synthetically. The authors used a strategy that combined photo flipping and cropping. Using the flipped and cropped process, a single image is stretched to ten images for use in model training.

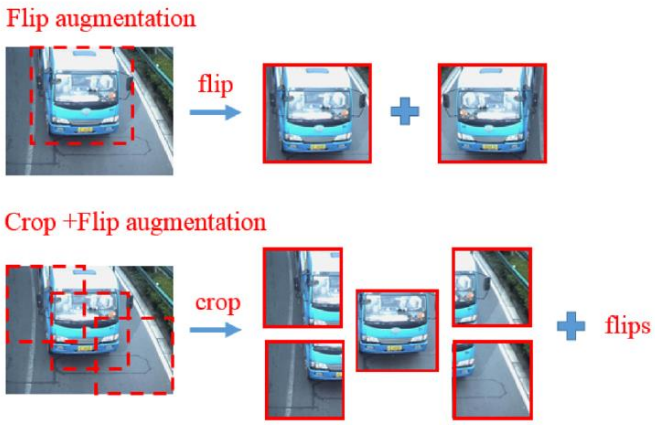


Figure 2.7 Crop Flip Augmentation

The 1st layer of the network uses 96 convolution kernels, and the 2nd layer uses 256 convolution kernels. And the convolutional stride is two. It uses 384 convolutional kernels in the 3rd, 4th, and 7th layers, with a convolutional stride of 1. It employs 256 convolutional kernels in the 5th and 6th layers, with a convolutional stride of 1. It uses max pooling in the 1st, 2nd, and 5th layers, with a sliding window size of 3 by 3 and a stride of 2. It minimizes the dimension of data and time taken for the computation

while effectively avoiding overfitting the network. The proposed approach, which is focused on autos and trucks, has a high accuracy rate of over 90%.

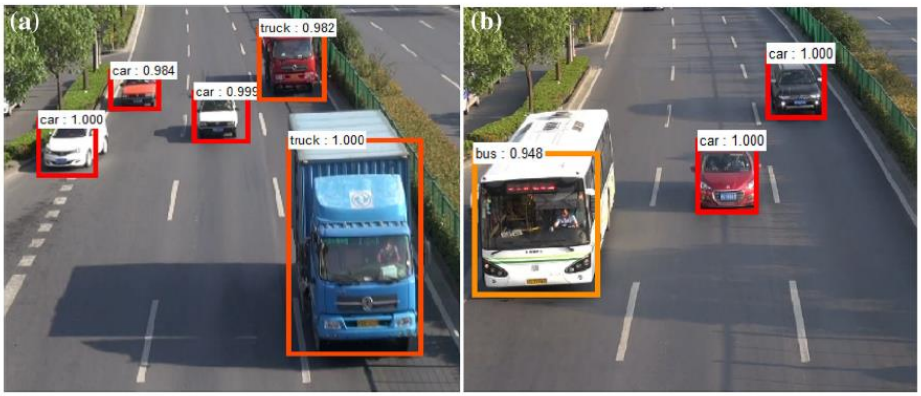


Figure 2.8 Selected examples of vehicles

Detection of fruits for labelling

To determine the type of fruit for labelling and weighing purposes, a deep learning framework for fruit categorization was presented [13]. Two deep learning architectures are investigated in the proposed framework. The first architecture is based on a six-layer recommended light model, while the second is based on a VGG-16 deep learning model that has been pre-trained.

The light CNN model consists of three sets of two CNN layers, a max-pooling layer, and a hidden fully connected layer with the output layer.

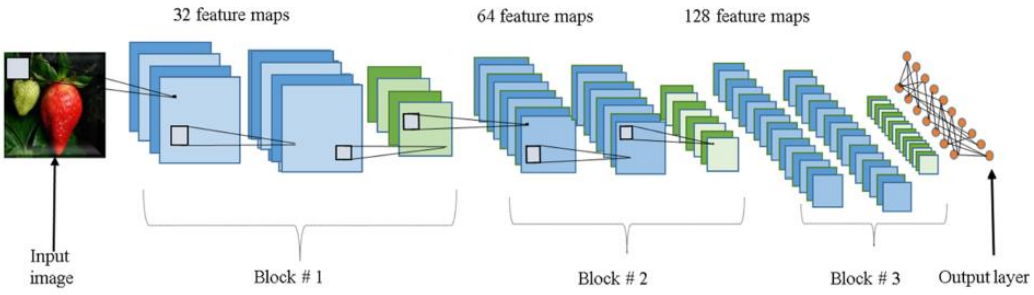


Figure 2.9 The structure the light CNN model

The suggested architecture's second model is VGG-16, a deeper CNN model. It is made out of five convolutional operations blocks. A maximum pooling layer connects adjacent blocks. Each block has three convolutional layers. Within each block, the number of convolution kernels remains constant, increasing from 64 in the first block to 512 in the last. There are a total of 16 layers that can be learned.

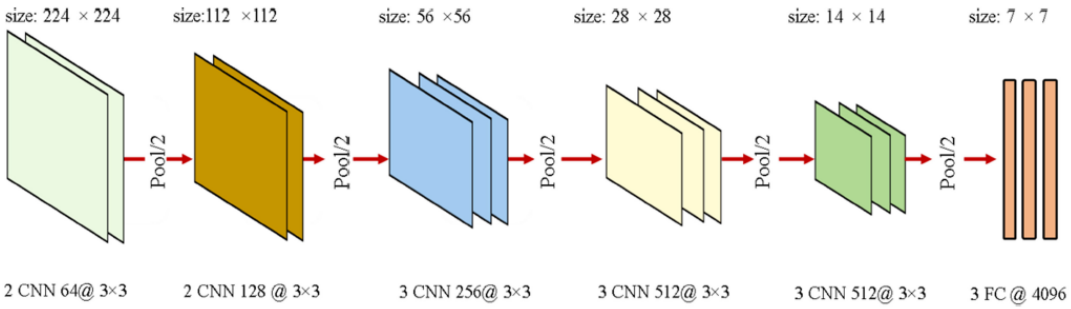


Figure 2.10 The structure of the VGG-16 based model

The suggested framework was evaluated using datasets of various sizes and complexity. Regardless of the number of images included for training, the VGG-16 fine-tuned model obtained outstanding accuracy on both datasets. With data augmentation, the light CNN model also obtained high accuracy on the smaller dataset.

2.3 Relevant related work on image classification techniques

Reference	Application	Basic techniques used
[01]	A novel convolutional neural network-based fault recognition method via image fusion of multi-vibration-signals	Convolutional Neural Networks
[03]	Deep Learning Based Antenna Array Fault Detection	
[04]	Image based surface damage detection of renewable energy installations using a unified deep learning approach	
[07]	Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification	
[08]	Convolutional Neural Network Based Fault Detection for Rotating Machinery	
[09]	Real-time vehicle type classification with deep convolutional neural networks	
[10]	Automatic Fruit Classification Using Deep Learning for Industrial Applications	
[02]	Real-Time Pipe Fault Detection System Using Computer Vision	Hough Transformations
[05]	Image Processing Based Insulator Fault Detection Method	Symmetry Detection and Grabcut Algorithm

Table 2.1 Summary of the related works of classification systems

According to the summary of the related works of image classification systems, the approaches Convolutional Neural Networks (CNN), Random Forest Classifier, and VGG16 CNN model for image classification are chosen to be further investigated for this study.

3. DESIGN

3.1 Pre-defined test scenarios

The data was collected using field-trained satellite antenna installation teams of a well-known service provider in Sri Lanka through a pre-built mobile application. As the first step, a set of test scenarios were designed in order to accurately cover the fundamental aspects of the installation. The instructions and the template image were embedded in the mobile application for the service teams (in the field) to capture images as expected. Once all the images of the satellite installations were captured according to the templates, they were tagged as correct or incorrect by eye-bowling the installation. This study was done only for the 3 above-mentioned test scenarios that cover the majority of the fundamental aspects of the installation accuracy.

Design details of QA tests for satellite antenna

3.1.1 Test Case 01: Mounting of the Satellite Dish

Figure 3.1 Template image for test case 01

The satellite dish is typically mounted using a GI pole. The GI pole at the point of connection to the satellite dish needs to be perpendicular to the ground. Hence this test case is designed to evaluate the angle between the GI pole (at the point of connection) and the ground. (*Figure 3.1*)

Test Objective

Under any given circumstance, the perpendicularity of the GI pole end needs to be preserved.

3.1.2 Test Case 02: Non-obstructive signal view

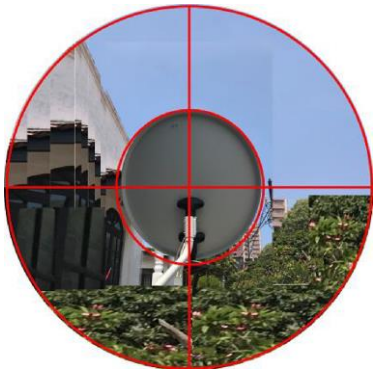


Figure 3.2 Template image for test case 02

The satellite dish needs to be installed in a way such that it has a non-obstructive signal reception. This is generally achieved by avoiding tall buildings and trees within the line of sight of the dish. However, in urban settings, it is difficult to achieve a 100% clean line of sight. Avoiding immediate obstructions in the near vicinity is adequate in most cases. Hence this is an extremely good test case for deep learning-based evaluation as it is very difficult to define any rules around the clearance. (*Figure 3.2*)

Test Objective

To validate that there are no obstructions such as trees and buildings within the close vicinity of the dish antenna.

3.1.3 Test Case 03: Connection to the TV

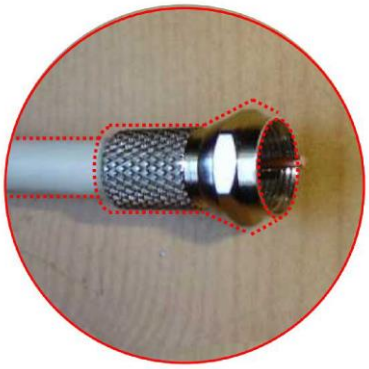


Figure 3.3 Template image for test case 03

The connection between the TV and the satellite dish is established through a coaxial cable via a setup box. The TV is ultimately connected to the setup box using a connector of type F (F-connector). This connection type cancels the external electromagnetic noises that can interfere with the signal while transmitting the satellite signal to the TV. (*Figure 3.3*)

However, the nature of this connection type is less stable compared to digital connectors such as HDMI. If not properly set up, it can get disrupted by moving the setup box or the TV. It can also be exposed to environmental factors such as corrosion if the correct setup guidelines are not followed.

Test Objective

There are two main setup guidelines associated with an F-connector installation.

First and foremost, it needs to have a coaxial cable at the center which is of correct exposed length. A short-exposed length would result in a week connection with the TV. A long-exposed length would mean it is unduly exposed to the external environment which can lead to signal noise and long-term decay.

Secondly, the screw nut and the cap should have tight coupling with the rest of the cable to make the connection robust against the movement of any of the devices.

3.2 Proposed method

The main focus of the proposed solution is to provide a backend solution [Machine Learning algorithm] for a quality control tool that can perform real-time verification on the installation process itself [in an extended step]. The image capturing part would be achieved by using a mobile device with the pre-defined set of installation guidelines mentioned before. Once an image of a pre-defined test case gets uploaded into storage, the image will be classified as correct or incorrect based on a pre-trained image classification model for that particular pre-defined test case.

The steps of the proposed method:

(a) The data will be collected using a mobile phone with a camera according to the pre-defined set of satellite installation guidelines. (Both correct and incorrect images are taken in order to train the image classification model)
(b) Train image classification models using Convolutional Neural Networks (CNN).
(c) Implement an algorithm to get the classification prediction for an image in specific storage.

3.2.1 Convolutional neural networks

The structure of the mammalian visual system serves as inspiration for convolutional neural network construction. The layer categories are divided into three main groups: convolutional layer, pooling layer, and fully-connected layer. [14]

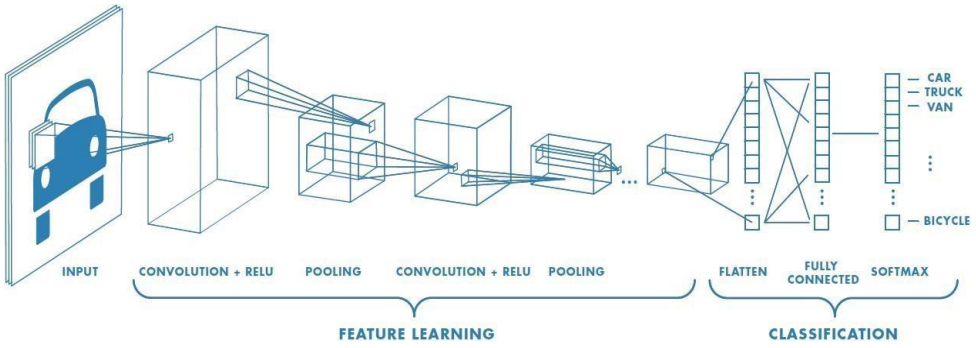


Figure 3.4 Convolutional neural networks

Convolutional Layer

The Convolutional layer, which has local connections and weights of shared properties, is the heart of the Convolutional neural network. The goal of the Convolutional layer is to learn input feature representations.

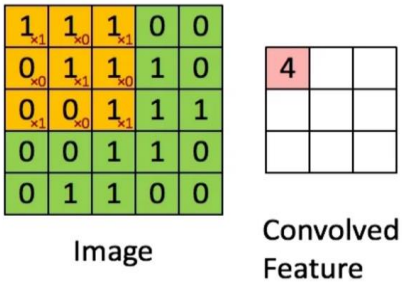


Figure 3.5 Convolutional layers

Pooling Layer

The sampling method is the same as fuzzy filtering. The pooling layer can minimize the dimensions of feature maps and improve the robustness of feature extraction by acting as a secondary feature extraction layer. It's commonly sandwiched between two Convolutional layers.

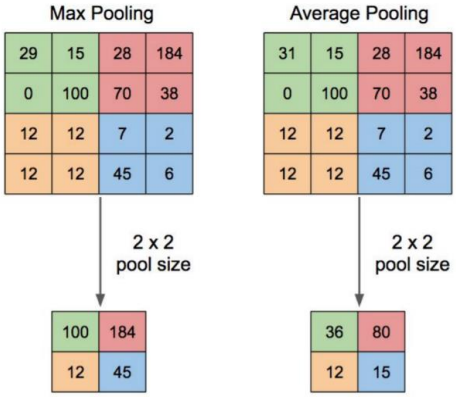


Figure 3.6 Pooling layer

Fully-connected Layer

A Convolutional neural network classifier is made up of one or more fully connected layers in general. They link every neuron in the current layer to every neuron in the previous layer. In fully-connected layers, no spatial information is kept. An output

layer follows the final fully connected layer. Softmax regression is often used for classification jobs since it generates a well-performed probability distribution of the results. [14]

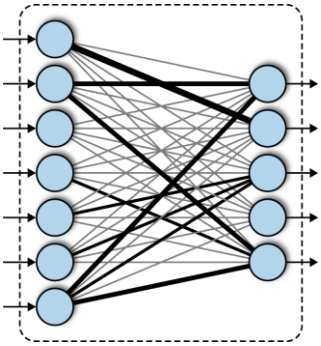


Figure 3.7 Fully connected layer

Transfer Learning on VGG-19

Without a large quantity of well-organized datasets, training an extremely deep Convolutional Neural Network model is difficult. For fault diagnosis, a transfer learning approach is proposed in this study [15]. This study is based on pre-trained VGG-19.

To begin, the features of transformed images are extracted using the VGG-19 model. The proposed retrained VGG-19 is put to the test on Case Western Reserve University's well-known dataset on the motor bearing. The ultimate accuracy of the trained VGG-19 is 99.175%, while the model training time is only around 200 seconds. Many DL and machine learning algorithms were outperformed by these findings.

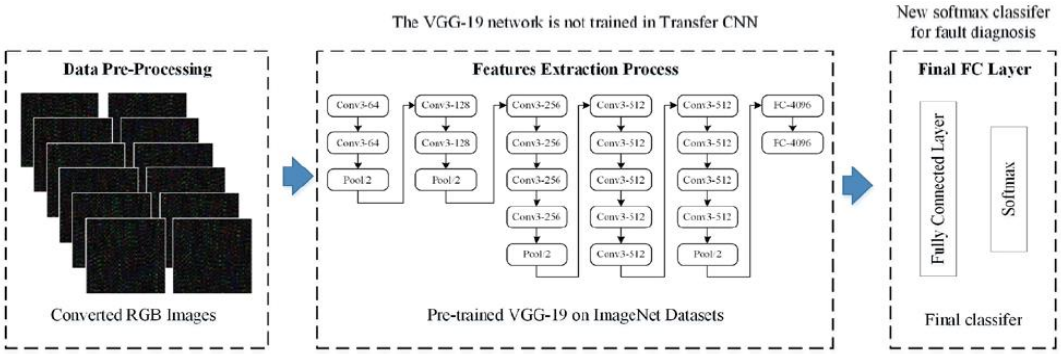


Figure 3.8 The flowchart of the proposed VGG-19 re-training

3.3 Proposed Solution Architecture Diagram

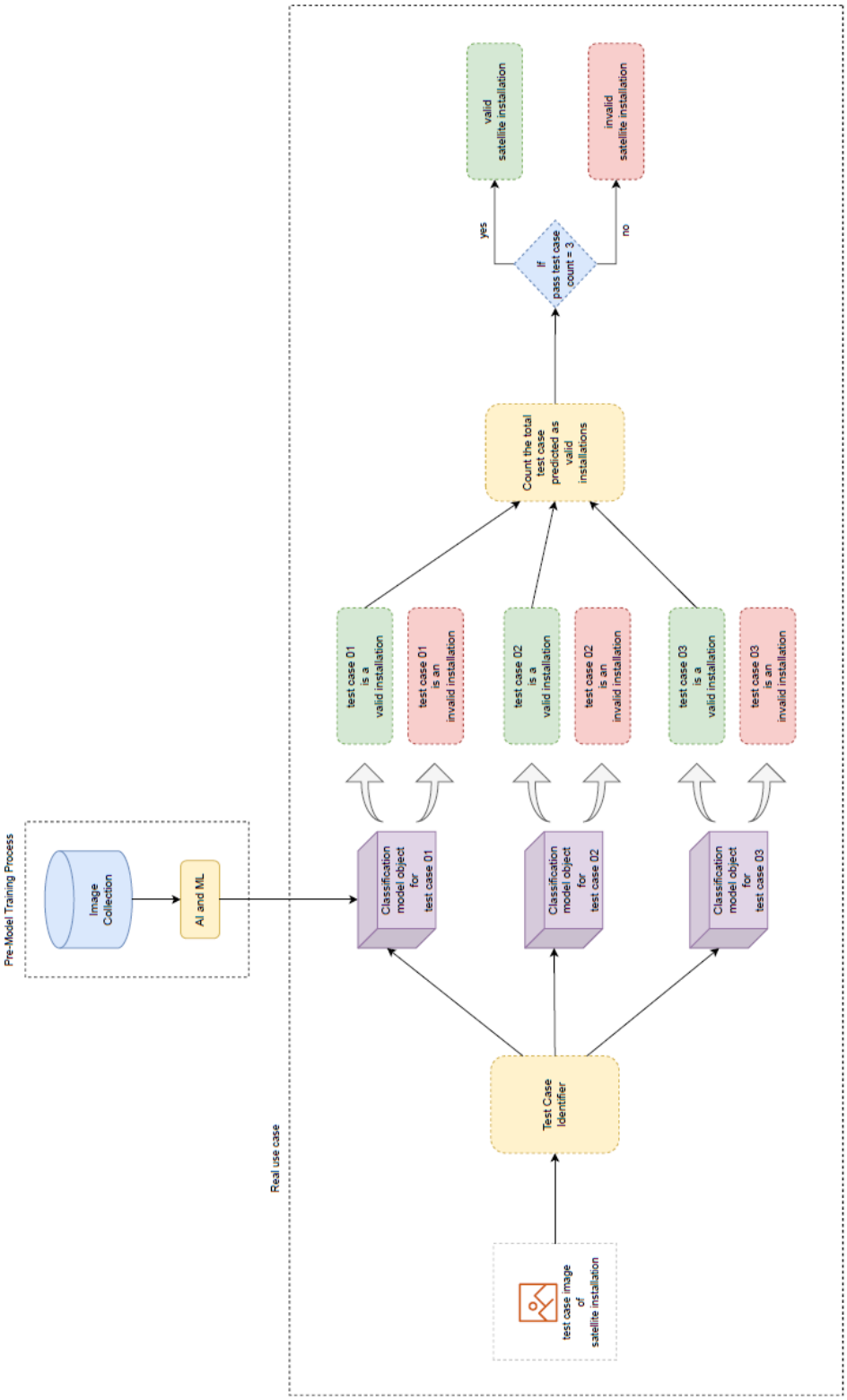


Figure 3.9 Proposed solution architecture diagram

4. IMPLEMENTATION

4.1 Data Collection

The data was collected using field-trained satellite antenna installation teams through a pre-built mobile application according to the template instructions. The collected images were filtered by removing the non-satellite images that were out of the template. The number of filtered images used for model training is given below. (*Table 4.1*)

Test Case	Total no of images	Total images	
		correct	incorrect
1	1241	707	534
2	1158	502	656
3	2117	967	1150

Table 4.1 No of images per test case

All the images of the installations were tagged manually whether correct or not by eye bowling. (*Figure 4.1*)

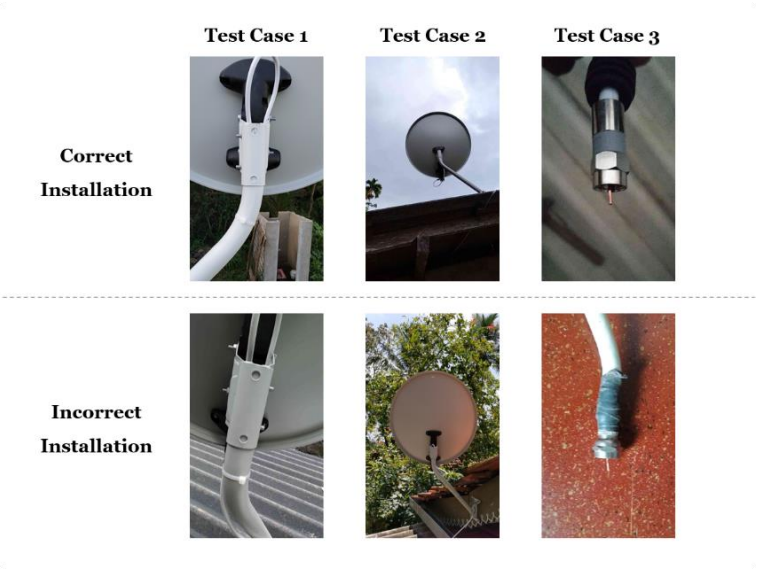


Figure 4.1 Tagged correct and incorrect installation images

4.2 Model Training and Testing

4.2.1 Hardware and software used

This study was performed on a Windows 10 Pro 64-bit machine with an Intel i5-7200U CPU operating at 2.50GHz and 16 GB RAM. The model training, testing, and application development scripts were written using Python 3.7. On a Tensorflow backend, CNN models were trained using the Python inbuilt package 'Keras'. The Python inbuilt package ‘Sklearn’ was used to train Random Forest models as benchmark models. Weights & biases (wandb.ai) [16] was used to generate intermediate results on training and validation sets during the model training process. The Python inbuilt package Streamlit [17] was used to build the application for demonstration purposes.

4.2.2 Train, Test, and Validation

70% of the total images were taken to train the model, 10% as the validation image set, and 20% of the images as the out-of-sample test set to evaluate the model performance.

The train, test, and validation splits are as below:

Test Case	Train Set		Validation Set		Test Set	
	correct	incorrect	correct	incorrect	correct	incorrect
1	496	374	71	53	140	107
2	358	459	51	66	93	131
3	677	805	97	115	193	230

Table 4.2 Train, Test and Validation splits per test case

4.2.3 Model Training

Three different machine learning models were evaluated. These are a basic CNN model with 2 convolutional layers, a random forest classifier, and a VGG16-CNN model. Three different classification models were built per each test case to compare.

Basic CNN model

A basic image classification model [18] is built. There are two convolutional layers, after each is a max-pooling layer. The fourth layer is a flatten layer. It is followed by a fully connected layer and an output layer with softmax activation function to classify the segmentation into the installation quality categories: correct and incorrect. The optimizer 'Adam' is used with a learning rate of 0.002. The number of epochs were varied in the range of [20, 30, 40, 50] and selected the best number of epochs for the models. The selected number of epochs of the best models were 30, 30, 50 for test scenarios 1, 2, and 3 respectively. The validation data set (*Table 4.2*) was used during the model training process to monitor the validation accuracy.

Random Forest Model

The input images were converted into numeric arrays in order to train the Random Forest classifier. The default parameters were used when training the models. The number of trees in the forest was used as 100 and the minimum samples split was used as 2.

Transfer learning on VGG16

As deep CNN models are difficult to train without a huge quantity of data, a transfer learning technique [19] was experimented with a commonly used VGG16 model (a CNN model that has been trained on 1.2 million images on 1000 different categories). The VGG16 model learned a robust representation of low-level features that may be shared between images to transfer knowledge and extract features from the new images because it was trained on such a huge dataset. The 3 fully-connected layers at the top of the VGG16 network were removed and the same VGG16 model architecture was

used as a feature extractor to recalculate the weights using target data (satellite images). A softmax layer was added as the output layer and it was adjusted to meet the requirement of the study to classify the input images into 2 classes.

4.2.4 Intermediate Results

Weights & biases (wandb.ai) [16] was used to generate intermediate results on training and validation sets during the model training process. Weights & Biases is a machine learning platform for developers to quickly track experiments, evaluate model performance and visualize results.

The intermediate results that were observed:

- The accuracy of the training data set by epoch
- The loss of the training data set by epoch
- The accuracy of the validation data set by epoch
- The loss of the validation data set by epoch

4.2.5 Evaluation Criteria

The performance of the image classification models for all 3 test scenarios were evaluated using quantitative matrices including accuracy (*Equation 1*), balanced accuracy (*Equation 2*), Precision (*Equation 3*), Recall (*Equation 4*), and F1 Score (*Equation 5*). After training the models with training data and validating with validation data, an independent test set was used only to report the final accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{Equation 1}$$

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \qquad \text{Equation 2}$$

$$Precision = \frac{TP}{TP+FP} \qquad \text{Equation 3}$$

$$Recall = \frac{TP}{TP+FN} \qquad \text{Equation 4}$$

$$F1\ Score = \frac{2\ (Recall \times Precision)}{(Recall + Precision)} \qquad \text{Equation 5}$$

Where;

- TP represents the number of true positives which is the number of incorrect installations that are predicted correctly as incorrect,
- TN represents the number of true negatives which is the number of correct installations that are predicted correctly as correct
- FP represents the number of false positives which is the number of correct installations that are predicted incorrectly as incorrect
- FN represents the number of false negatives which is the number of incorrect installations that are predicted incorrectly as correct

4.2.6 Image Pre-processing

The input images were resized into 224 × 224 pixels to fit the requirement of the adopted VGG16 model. The images were rescaled to greyscale to speed up the training process. (*Figure 4.2*)

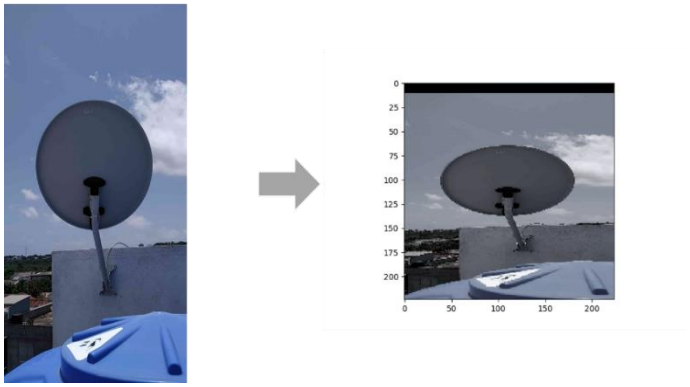


Figure 4.2 Pre-processed image

4.2 7 Error analysis

A manual error analysis was done after each iteration of model training to identify the misclassified images. It concluded that the misclassified images (*Figure 4.3*) have mixed features that are significantly difficult to categorize whether correct or not even with human intervention.

Figure 4.3 Error analysis examples for test case 2

4.3 Application

A simple application has been developed using Streamlit [17] for demoing purposes. The main functionalities that are included in the application:

i. Show the template image of the installation when the test case no is selected

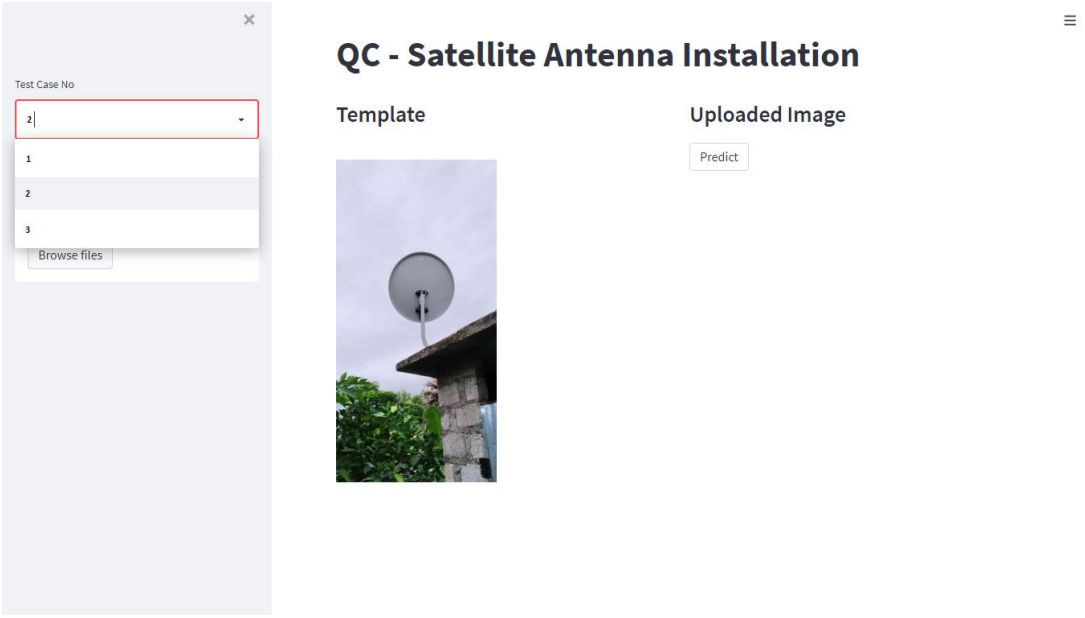


Figure 4.4 Screen to check the template image

ii. Upload the new satellite antenna installation image to get the prediction as correct or incorrect

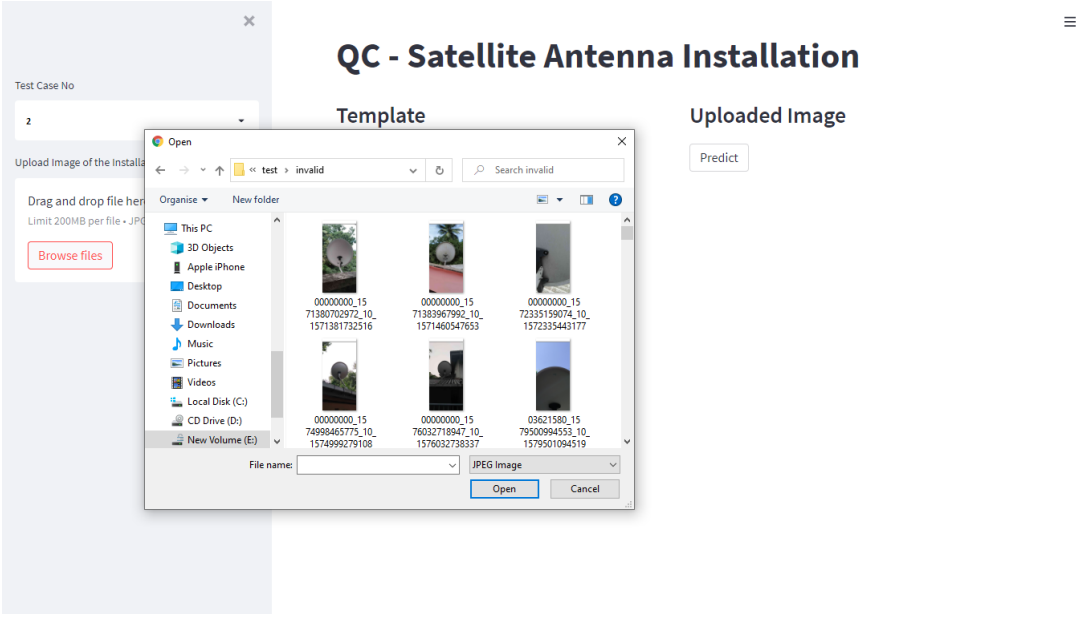


Figure 4.5 Screen to upload the new satellite image

iii. Get the prediction for the uploaded image using the already built model object

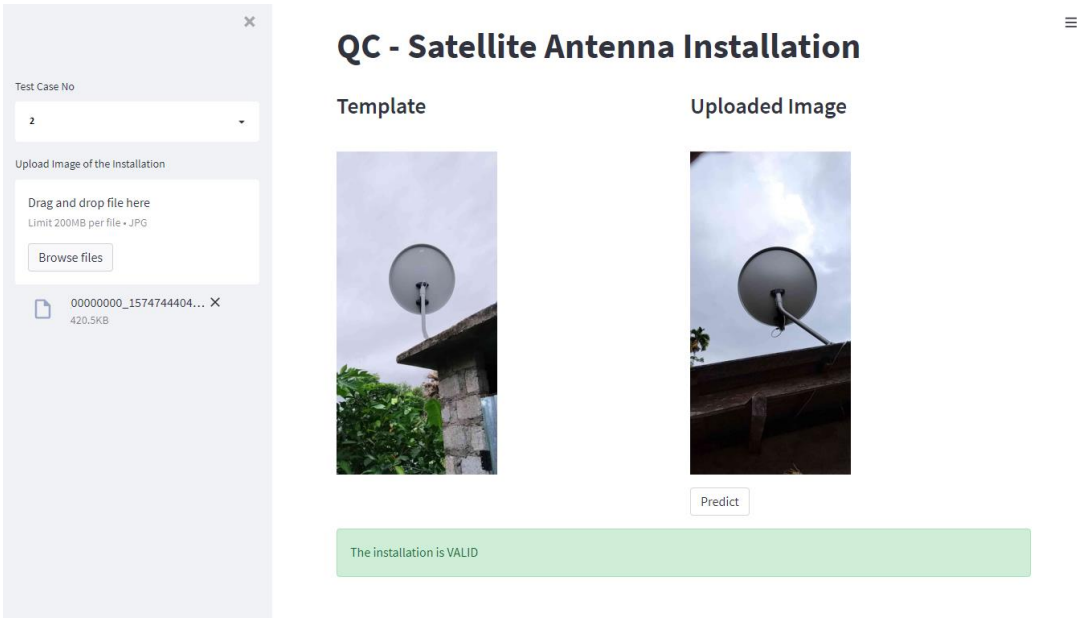


Figure 4.6 Prediction for a valid installation

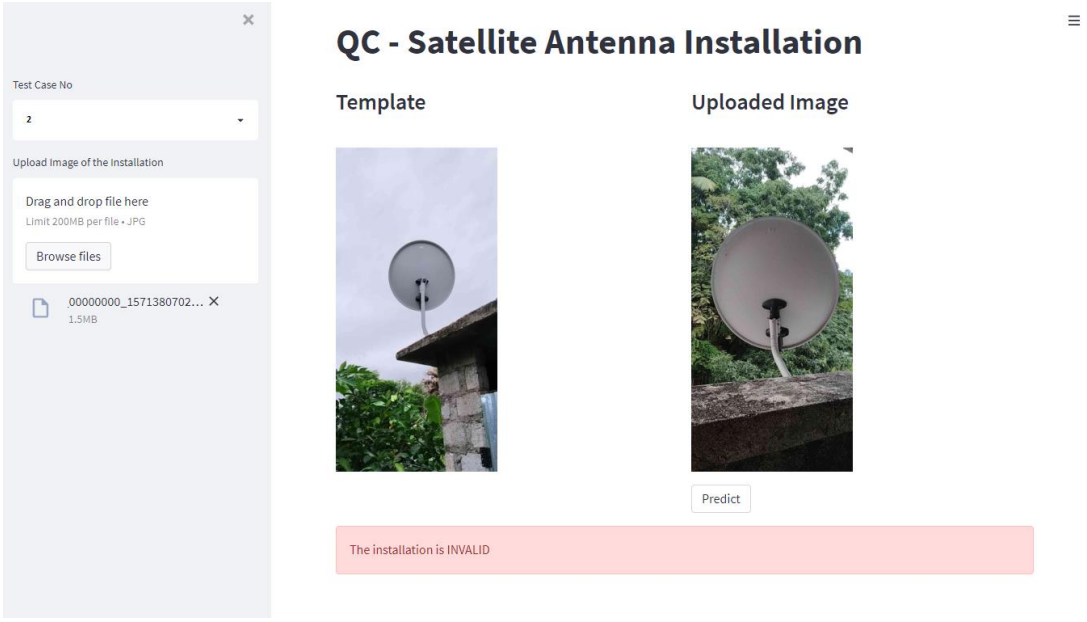


Figure 4.7 Prediction for an invalid installation

5. RESULTS AND DISCUSSION

5.1 Results of the basic CNN model

5.1.1 Test Case 01: Mounting of the Satellite Dish

The model training was done for 30 epochs and it took about 39 minutes to train the model. The intermediate results of the validation and training sets and the accuracy matrix based on the testing set are as follows.

Figure 5.1 Intermediate results of the validation and train sets test case 1

		Prediction		
		Valid	Invalid	Total
Actual	Valid	140	0	140
	Invalid	107	0	107
	Total	247	0	247

Figure 5.2 Confusion matrix for test case 1

The overall accuracy is 56.7% and the recall is 0%. The precision and F1 score are not applicable as the model prediction is biased towards ‘correct’. All the actual incorrect images were predicted as correct.

5.1.2 Test Case 02: Non-obstructive signal view

The model training was done for 30 epochs and it took about 55 minutes to train the model. The intermediate results of the validation and training sets and the accuracy matrix based on the testing set are as follows.

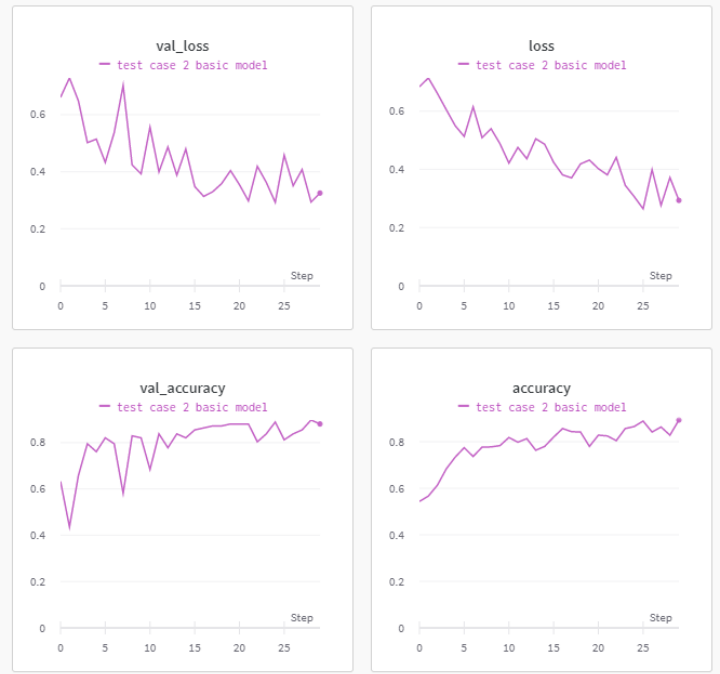


Figure 5.3 Intermediate results of the validation and train sets test case 2

		Prediction		
		Valid	Invalid	Total
Actual	Valid	64	29	93
	Invalid	20	111	131
	Total	84	140	224

Figure 5.4 Confusion matrix for test case 2

The overall accuracy, precision, recall, and F1 score are 78.1%, 79.3%, 84.7%, 81.9% respectively. The accuracy of the basic CNN was not up to the expected target of this study (above 80%).

5.1.3 Test Case 03: Connection to the TV

The model training was done for 30 epochs and it took about 55 minutes to train the model. The intermediate results of the validation and training sets and the accuracy matrix based on the testing set are as follows.

Figure 5.5 Intermediate results of the validation and train sets test case 3

		Prediction		
		Valid	Invalid	Total
Actual	Valid	73	120	193
	Invalid	37	193	230
	Total	110	313	423

Figure 5.6 Confusion matrix for test case 3

The overall accuracy, precision, recall, and F1 score are 62.9%, 61.7%, 83.9%, 71.1% respectively. The accuracy of the basic CNN was not up to the expected target of this study (above 80%).

5.2 Results of the Random Forest model

5.2.1 Test Case 01: Mounting of the Satellite Dish

		Prediction		
		Valid	Invalid	Total
Actual	Valid	109	31	140
	Invalid	57	50	107
	Total	166	81	247

Figure 5.7 Confusion matrix for test case 1

The overall accuracy, precision, recall, and F1 score are 64.4%, 61.7%, 46.7%, 53.2% respectively. The accuracy of the random forest classifier was not up to the expected target of this study (above 80%).

5.2.2 Test Case 02: Non-obstructive signal view

		Prediction		
		Valid	Invalid	Total
Actual	Valid	77	16	93
	Invalid	22	109	131
	Total	99	125	224

Figure 5.8 Confusion matrix for test case 2

The overall accuracy, precision, recall, and F1 score are 83.0%, 87.2%, 83.2%, 85.2% respectively. The accuracy of the random forest classifier was up to the expected target of this study (above 80%).

5.2.3 Test Case 03: Connection to the TV

		Prediction		
		Valid	Invalid	Total
Actual	Valid	115	78	193
	Invalid	71	159	230
	Total	186	237	423

Figure 5.9 Confusion matrix for test case 3

The overall accuracy, precision, recall, and F1 score are 64.8%, 67.1%, 69.1%, 68.1% respectively. The accuracy of the random forest classifier was not up to the expected target of this study (above 80%).

5.3 Results of the VGG16 - CNN model

5.3.1 Test Case 01: Mounting of the Satellite Dish

The model training was done for 30 epochs and it took about 190 minutes to train the model. The intermediate results of the validation and training sets and the accuracy matrix based on the testing set are as follows.

Figure 5.10 Epoch accuracy results of the validation and train sets test case 1

According to the intermediate results, validation accuracy increases with the training accuracy (*Figure 5.10*). Hence, the model is not overfitting.

		Prediction		
		Valid	Invalid	Total
Actual	Valid	110	30	140
	Invalid	12	95	107
	Total	122	125	247

Figure 5.11 Confusion matrix for test case 1

The overall accuracy, balanced accuracy, precision, recall, and F1 score are 83.0%, 83.7%, 76.0%, 88.8%, 81.9% respectively. The accuracy of the CNN using VGG16 is up to the expected target of this study (above 80%).

5.3.2 Test Case 02: Non-obstructive signal view

The model training was done for 30 epochs and it took about 168 minutes to train the model. The intermediate results of the validation and training sets and the accuracy matrix based on the testing set are as follows.

Figure 5.12 Epoch accuracy results of the validation and train sets test case 2

According to the intermediate results, validation accuracy increases with the training accuracy (*Figure 5.12*). Hence, the model is not overfitting.

		Prediction		
		Valid	Invalid	Total
Actual	Valid	83	10	93
	Invalid	17	114	131
	Total	100	124	224

Figure 5.13 Confusion matrix for test case 2

The overall accuracy, balanced accuracy, precision, recall, and F1 score are 87.9%, 88.1%, 91.9%, 87.0%, 89.4% respectively. The accuracy of the CNN using VGG16 is up to the expected target of this study (above 80%).

5.3.3 Test Case 03: Connection to the TV

The model training was done for 50 epochs and it took about 296 minutes to train the model. The intermediate results of the validation and training sets and the accuracy matrix based on the testing set are as follows.

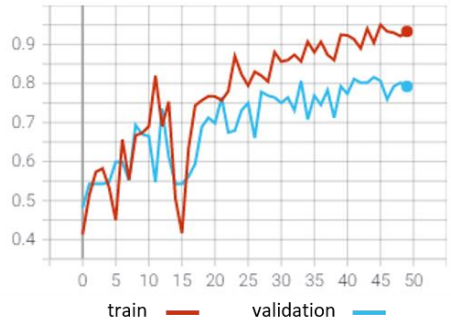


Figure 5.14 Epoch accuracy results of the validation and train sets test case 3

According to the intermediate results, validation accuracy increases with the training accuracy (*Figure 5.14*). Hence, the model is not overfitting.

		Prediction		
		Valid	Invalid	Total
Actual	Valid	167	26	193
	Invalid	31	199	230
	Total	198	225	423

Figure 5.15 Confusion matrix for test case 3

The overall accuracy, balanced accuracy, precision, recall, and F1 score are 86.5%, 86.5%, 88.4%, 86.5%, 87.5% respectively. The accuracy of the CNN using VGG16 is up to the expected target of this study (above 80%).

5.4 Model Performance Evaluation

The accuracy of the final image classification model is compared with the basic CNN model and Random Forest model accuracies. The comparison is as below.

Test Case No	Matric	Basic CNN	Random Forest	VGG16-CNN
1	Overall Accuracy	56.7%	64.4%	**83.0%**
	Balanced Accuracy	50.0%	62.3%	**83.7%**
	Precision	N/A	61.7%	76.0%
	Recall	0.0%	46.7%	88.8%
	F1 score	N/A	53.2%	81.9%
2	Overall Accuracy	78.1%	83.0%	**87.9%**
	Balanced Accuracy	76.8%	83.0%	**88.1%**
	Precision	79.3%	87.2%	91.9%
	Recall	84.7%	83.2%	87.0%
	F1 score	81.9%	85.2%	89.4%
3	Overall Accuracy	62.9%	64.8%	**86.5%**
	Balanced Accuracy	60.9%	64.4%	**86.5%**
	Precision	61.7%	67.1%	88.4%
	Recall	83.9%	69.1%	86.5%
	F1 score	71.1%	68.1%	87.5%

Table 5.1 Image classification results on test data

All the evaluation matrices of the VGG16-CNN model are greater than the other models tried; basic CNN [18] and random forest [21].

The correct and incorrect installation image split is not highly imbalanced in this use case. The random forest image classification gives a good accuracy of 83% only for test scenario no 2. Hence, it cannot be used as a generic modeling framework for this specific use case. The basic CNN model accuracies are not up to the expected target of this experiment. (*Table 5.1*)

5.5 Experimental verification in the field

The best performing predictive models (VGG16-CNN models) for the three test scenarios were embedded into a beta version of the mobile application and the proposed method was tested on a selected set of locations using a few service teams. The results of the experiment concluded that the service teams can benefit from the predictions on their installations to improve the installation quality. The confusion matrix along with the initial accuracy obtained from this field experiment is as below.

5.5.1 Test Case 01: Mounting of the Satellite Dish

		Prediction		
		Valid	Invalid	Total
Actual	Valid	53	3	56
	Invalid	3	57	60
	Total	56	60	116

Figure 5.16 Confusion matrix for experimental verification test case 01

The overall accuracy, balanced accuracy, precision, recall, and F1 score of the experiment for test scenario 01 are 94.8%, 94.8%, 95.0%, 95.0%, 95.0% respectively.

5.5.2 Test Case 02: Non-obstructive signal view

		Prediction		
		Valid	Invalid	Total
Actual	Valid	48	4	52
	Invalid	8	54	62
	Total	56	58	114

Figure 5.17 Confusion matrix for experimental verification test case 02

The overall accuracy, balanced accuracy, precision, recall, and F1 score of the experiment for test scenario 02 are 89.5%, 89.7%, 93.1%, 87.1%, 90.0% respectively.

5.5.3 Test Case 03: Connection to the TV

		Prediction		
		Valid	Invalid	Total
Actual	Valid	53	7	60
	Invalid	7	44	51
	Total	60	51	111

Figure 5.18 Confusion matrix for experimental verification test case 03

The overall accuracy, balanced accuracy, precision, recall, and F1 score of the experiment for test scenario 03 are 87.4%, 87.3%, 86.3%, 86.3%, 86.3% respectively.

6. CONCLUSION

The availability of training data was a major issue at the beginning of this study. Others have utilized image classification to provide quality control in a variety of ways. However, this application is both non-generalized and focused on a single, specific use case. As deep learning image classification model training requires a considerable amount of data, there was a complete reliance on starting the initial model training until all the data are collected.

The next task was to create test scenarios for the complete installations of the satellite antenna. Several brainstorming sessions with experts were held back and forth to finalize the test scenarios. Initially, ten test scenarios were designed to cover the entire antenna installation. However, a few test scenarios were impossible to capture using a mobile phone when it came to actual satellite antenna installation on-site. There were practical issues to capture some test scenarios once the antenna is installed on roofs. After doing a field test with the installation teams, five main test scenarios were selected to cover the entire installation quality. Only the three most essential test scenarios out of the final five were considered in this study.

Once the images are collected, tagging them manually took a considerable amount of time and it was not accurate enough as it is done by subjective eye bowling. The performance of the model can be further improved by correcting the incorrectly tagged images and by adding more images to the training data set.

The initial predictive models are built based on a very basic CNN architecture [18]. Optimizing the model architecture by tuning the number of layers has not been done in this study. That can be suggested as future work to increase the prediction accuracies as different layers can affect the accuracy.

In the proposed models using the VGG16 model, the accuracies of the test set were comparatively better and considerably faster. The average accuracy of the models is 85.8% and the average balanced accuracy is 86.1%. The prediction time was very fast with a mean time of 0.5 seconds per image.

Experimental verification was done in order to test the performance of the predictive models in the field and the results were promising with an average accuracy of 90.56%

and an average balanced accuracy of 90.61%.

As an extended step, it is suggested to integrate these prediction models into the mobile application, so that the service teams themselves can get the prediction right after capturing the image based on the template. This reduces the operational cost that goes to the verification teams and the coverage can be increased significantly.

Even though there is a marginal error percentage, the customer complaints can be reduced as there is a prediction and a person involved with installing it. Their expertise should probably cover the error percentage in this initial training step. A test application is developed along with this study for demonstrating purposes.

The code can be found at: https://github.com/Vanodya/MSc_research_CNN/blob/main/research_code.py

Limitations and future directions

- Data availability was one of the major limitations of this study and it is suggested to collect more data on the test scenarios and rebuild the predictive models, as having a considerable amount of data can impact deep learning model accuracy.
- Most of the misclassified images were tagged incorrectly and they were corrected during the error analysis process. It is suggested that the performance of the model can be improved by tagging the images more accurately.
- The basic CNN model architecture can be optimized by tuning the number of layers. It is suggested as future work to improve the prediction accuracy.
- Integrating the predictive models in the mobile application is the suggested next step to complete this study.
- Handling the scale once the predictive models are integrated into the application is another area that can be improved further.

References

[1] F. Faux and F. Luthon, 'Theory of evidence for face detection and tracking', *Int. J. Approx. Reason.*, vol. 53, no. 5, pp. 728–746, Jul. 2012, doi: 10.1016/j.ijar.2012.02.002.

[2] M. A. Abu, N. H. Indra, A. H. A. Rahman, N. A. Sapiee, and I. Ahmad, 'A study on Image Classification based on Deep Learning and Tensorflow', vol. 12, no. 4, p. 7, 2019.

[3] S. Shakya, 'Analysis of Artificial Intelligence based Image Classification Techniques', *J. Innov. Image Process.*, vol. 2, pp. 44–54, Mar. 2020, doi: 10.36548/jiip.2020.1.005.

[4] H. Wang, S. Li, L. Song, and L. Cui, 'A novel convolutional neural network based fault recognition method via image fusion of multi-vibration-signals', *Comput. Ind.*, vol. 105, pp. 182–190, Feb. 2019, doi: 10.1016/j.compind.2018.12.013.

[5] K. Hyoung-Seok and L. Byung-Ryong, 'Real-Time Pipe Fault Detection System Using Computer Vision', *Int. J. Precis. Eng. Manuf.*, vol. 7, no. 1, pp. 30–34, 2006.

[6] K. Chen, W. Wang, X. Chen, and H. Yin, 'Deep Learning Based Antenna Array Fault Detection', in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, Apr. 2019, pp. 1–5. doi: 10.1109/VTCSpring.2019.8746510.

[7] A. Shihavuddin *et al.*, 'Image based surface damage detection of renewable energy installations using a unified deep learning approach', *Energy Rep.*, vol. 7, pp. 4566–4576, Nov. 2021, doi: 10.1016/j.egyr.2021.07.045.

[8] U. Tudevdagva, B. Battseren, W. Hardt, and G. V. Troshina, 'Image Processing Based Insulator Fault Detection Method', in *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, Oct. 2018, pp. 579–583. doi: 10.1109/APEIE.2018.8545429.

[9] S. Lee, K. E. An, B. D. Jeon, K. Y. Cho, S. J. Lee, and D. Seo, 'Detecting faulty solar panels based on thermal image processing', in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2018, pp. 1–2. doi: 10.1109/ICCE.2018.8326228.

[10] 'Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification'. https://www.hindawi.com/journals/cin/2016/3289801/

[11] O. Janssens *et al.*, ‘Convolutional neural network based fault detection for rotating machinery’, *J. SOUND Vib.*, vol. 377, pp. 331–345, 2016, doi: 10.1016/j.jsv.2016.05.027.

[12] ‘Real-time vehicle type classification with deep convolutional neural networks | SpringerLink’. https://link.springer.com/article/10.1007/s11554-017-0712-5?shared-article-renderer

[13] M. S. Hossain, M. H. Al-Hammadi, and G. Muhammad, ‘Automatic Fruit Classification Using Deep Learning for Industrial Applications’, *IEEE Trans. Ind. Inform.*, 2019, doi: 10.1109/TII.2018.2875149.

[14] T. Guo, J. Dong, H. Li, and Y. Gao, ‘Simple convolutional neural network on image classification’, *2017 IEEE 2nd Int. Conf. Big Data Anal. ICBDA*, 2017, doi: 10.1109/ICBDA.2017.8078730.

[15] L. Wen, X. Li, X. Li, and L. Gao, ‘A New Transfer Learning Based on VGG-19 Network for Fault Diagnosis’, *2019 IEEE 23rd Int. Conf. Comput. Support. Coop. Work Des. CSCWD*, 2019, doi: 10.1109/CSCWD.2019.8791884.

[16] ‘Weights & Biases’. https://docs.wandb.ai/

[17] ‘Streamlit Docs’. https://docs.streamlit.io/

[18] ‘Building a Convolutional Neural Network | Build CNN using Keras’, *Analytics Vidhya*, Jun. 22, 2021. https://www.analyticsvidhya.com/blog/2021/06/building-a-convolutional-neural-network-using-tensorflow-keras/

[19] S. Tammina, ‘Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images’, *Int. J. Sci. Res. Publ. IJSRP*, vol. 9, p. p9420, Oct. 2019, doi: 10.29322/IJSRP.9.10.2019.p9420.

[20] A. Lee, ‘Choosing a Baseline Accuracy For a Classification Model’, *Medium*, Jun. 23, 2021. https://towardsdatascience.com/calculating-a-baseline-accuracy-for-a-classification-model-a4b342ceb88f

[21] ‘Image Classification using Machine Learning’, *Analytics Vidhya*, Jan. 20, 2022. https://www.analyticsvidhya.com/blog/2022/01/image-classification-using-machine-learning/

[22] A. Das, ‘Convolution Neural Network for Image Processing — Using Keras’, Medium, Jan. 11, 2021. https://towardsdatascience.com/convolution-neural-network-for-image-processing-using-keras-dc3429056306.