A Computer Vision-Based Approach for Automated Inspection of Cable Connections

Victor Lindvall
Abstract

A Computer Vision-Based Approach for Automated Inspection of Cable Connections

Victor Lindvall

The goal of the project was to develop an algorithm based on a Convolutional Neural Network (CNN) for automatically detecting exposed metal components on coaxial cable connections, a.k.a. the detector. We show that the performance of such a CNN trained to identify bad weatherproofings can be improved by applying an image post processing technique. This post processing technique utilizes specular features as an advantage when predicting exposed metal components. Such specular features are notorious for posing problems in computer vision algorithms and therefore typically removed. The results achieved by applying the standalone detector, without post processing, are compared with the image post processing approach to highlight the benefits of implementing such an algorithm.
Acknowledgements

A special thanks to my supervisor, Athanasios Karapantelakis, for this opportunity and always being available for discussing problems, ideas for viable solutions, and motivating me to perform at my best.

Thank you to Ericsson AB for providing a positive working environment allowing me to learn and experiment with the help of an abundance of resources available. Also, thanks to Maxim Teslenko and Yifei Jin for enthusiastically offering their expertise and opinions on creative ways to approach the project.

Thanks to my reviewer, Natasa Sladoje for helping shape the project in a scientific way from the start.

A big thank you to my parents, Claes Mikael and Chatarina Lindvall for continuously supporting me throughout the years; my big sisters and role models, Jenny and Malin Lindvall, for their endless help and inspiring me to pursue higher education. To my dog, Frisco, thanks for the unconditional love.

Thanks to Dr. Shiva Azadegan for influencing my decision to change my major to Computer Science.
Table of Contents

Abstract II

Acknowledgements III

List of Tables VI

List of Figures VII

List of Acronyms IX

1 Introduction 1
   1.1 Motivation and Goals . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   1.2 Methods and Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2 Literature Review 4
   2.1 Machine Learning . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   2.2 Artificial Neural Networks . . . . . . . . . . . . . . . . . . . . . . . . . 4
   2.3 Convolutional Neural Networks . . . . . . . . . . . . . . . . . . . . . . 5
      2.3.1 Convolutional Layer . . . . . . . . . . . . . . . . . . . . . . . . . 6
      2.3.2 Activation Functions . . . . . . . . . . . . . . . . . . . . . . . . . 7
      2.3.3 Pooling Layer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   2.4 Computer Vision . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
      2.4.1 Specularity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
      2.4.2 Maximally Stable Extremal Regions . . . . . . . . . . . . . . . . . 11
   2.5 Progressive Probabilistic Hough Transform . . . . . . . . . . . . . . . . 14
   2.6 Precision and Recall . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3 Data 16
   3.1 Labeled Training Data Acquisition . . . . . . . . . . . . . . . . . . . . . . 16
   3.2 Datasets for Training . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      3.2.1 Homogeneous dataset . . . . . . . . . . . . . . . . . . . . . . . . . 17
      3.2.2 Diverse Dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      3.2.3 Indoor and Outdoor Dataset . . . . . . . . . . . . . . . . . . . . . 18
   3.3 Datasets for Testing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
      3.3.1 Ericsson Test set . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
      3.3.2 Indoor Test set . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
      3.3.3 Outdoor Test set . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Implementation</td>
<td>22</td>
</tr>
<tr>
<td>4.1 Hardware</td>
<td>22</td>
</tr>
<tr>
<td>4.2 Darknet</td>
<td>22</td>
</tr>
<tr>
<td>4.3 OpenCV</td>
<td>22</td>
</tr>
<tr>
<td>5 Methodology</td>
<td>23</td>
</tr>
<tr>
<td>5.1 Approach</td>
<td>23</td>
</tr>
<tr>
<td>5.2 Baseline</td>
<td>24</td>
</tr>
<tr>
<td>5.3 Image Post Processing</td>
<td>25</td>
</tr>
<tr>
<td>6 Results</td>
<td>29</td>
</tr>
<tr>
<td>6.1 Evaluation Method</td>
<td>29</td>
</tr>
<tr>
<td>6.2 Ericsson Test Results</td>
<td>31</td>
</tr>
<tr>
<td>6.3 Indoor Test Results</td>
<td>35</td>
</tr>
<tr>
<td>6.4 Outdoor Test Results</td>
<td>39</td>
</tr>
<tr>
<td>7 Discussion &amp; Analysis</td>
<td>43</td>
</tr>
<tr>
<td>7.1 Ericsson Set</td>
<td>43</td>
</tr>
<tr>
<td>7.2 Indoor Set</td>
<td>46</td>
</tr>
<tr>
<td>7.3 Outdoor Set</td>
<td>47</td>
</tr>
<tr>
<td>8 Conclusion</td>
<td>50</td>
</tr>
<tr>
<td>8.1 Concluding Statements</td>
<td>50</td>
</tr>
<tr>
<td>8.2 Future Work</td>
<td>50</td>
</tr>
<tr>
<td>Literature</td>
<td>51</td>
</tr>
<tr>
<td>Appendices</td>
<td>54</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>3.1</td>
<td>Training Datasets Metadata</td>
</tr>
<tr>
<td>6.1</td>
<td>Ericsson Hex Detection Results</td>
</tr>
<tr>
<td>6.2</td>
<td>Ericsson WP3M Detection Results</td>
</tr>
<tr>
<td>6.3</td>
<td>Ericsson Bad Weatherproofing Detection Results</td>
</tr>
<tr>
<td>6.4</td>
<td>Ericsson Bad Weatherproofing Detection Results with Post Processing</td>
</tr>
<tr>
<td>6.5</td>
<td>Indoor Hex Detection Results</td>
</tr>
<tr>
<td>6.6</td>
<td>Indoor WP3M Detection Results</td>
</tr>
<tr>
<td>6.7</td>
<td>Indoor Bad Weatherproofing Detections</td>
</tr>
<tr>
<td>6.8</td>
<td>Indoor Bad Weatherproofing Results with Post Processing</td>
</tr>
<tr>
<td>6.9</td>
<td>Outdoor Hex Detection Results</td>
</tr>
<tr>
<td>6.10</td>
<td>Outdoor WP3M Detection Results</td>
</tr>
<tr>
<td>6.11</td>
<td>Outdoor Bad Weatherproofing Detection Results</td>
</tr>
<tr>
<td>6.12</td>
<td>Outdoor Bad Weatherproofing Detection Results with Post Processing</td>
</tr>
</tbody>
</table>
# List of Figures

Figure 1.1: Good vs Bad weatherproofing ........................................... 2  
Figure 2.1: Neuron Models ................................................................. 5  
Figure 2.2: Example of CNN Architecture ............................................. 6  
Figure 2.3: Basic Convolution Operation ............................................... 7  
Figure 2.4: ReLU vs Leaky ReLU ......................................................... 8  
Figure 2.5: 2x2 max pooling operation with stride 2 .............................. 9  
Figure 2.6: Types of reflections ............................................................ 10  
Figure 2.7: MSER definitions ............................................................. 12  
Figure 2.8: Visualization of extremal region ......................................... 13  
Figure 2.9: Image with various thresholding intensities .......................... 13  
Figure 2.10: Precision Recall Illustration [35] ...................................... 15  
Figure 3.1: Bounding boxes around objects of interest ........................... 16  
Figure 3.2: Examples of images found in old dataset ............................. 17  
Figure 3.3: Examples of images found in new dataset ......................... 18  
Figure 3.4: Examples from outside ....................................................... 18  
Figure 3.5: Bounding box encapsulates both exposed metal hex nut and weatherproofing case ......................................................... 20  
Figure 3.6: Example of images in Ericsson supplied test set .................... 20  
Figure 3.7: Example of images in indoor test set ................................... 21  
Figure 3.8: Example of images in outdoor test set ................................ 21  
Figure 5.1: Custom Built CNN Configuration ...................................... 24  
Figure 5.2: Pairing Diagram .............................................................. 24  
Figure 5.3: Applying MSER ............................................................... 27  
Figure 5.4: Example of Post Processing Finding True Positives Missed by Detector ................................................................. 28  
Figure 6.1: Mean Subtraction performance detecting hex nuts .................. 30  
Figure 6.2: Hex detector performance on Ericsson Test set .................... 31  
Figure 6.3: Weatherproofing case detector performance on Ericsson Test set ................................................................. 32  
Figure 6.4: Detector performance on Ericsson Test set .......................... 33  
Figure 6.5: Detector performance with post processing on Ericsson Test set ................................................................. 34  
Figure 6.6: Hex detector performance on Indoor test set ....................... 35  
Figure 6.7: Weatherproofing case detector performance on Indoor test set ................................................................. 36  
Figure 6.8: Detector performance on Indoor test set ............................ 37
List of Figures

Figure 6.9: Detector performance with post processing on Indoor test set . . . . 38
Figure 6.10: Hex detector performance on Outdoor test set . . . . . . . . . . . . . 39
Figure 6.11: Weatherproofing case detector performance on Outdoor test set . . 40
Figure 6.12: Detector performance on Outdoor test set . . . . . . . . . . . . . . . . 41
Figure 6.13: Detector performance with post processing on Outdoor test set . . 42

Figure 7.1: Hex Bounding Box Comparison . . . . . . . . . . . . . . . . . . . . . 43
Figure 7.2: Bad Weatherproofing Bounding Box Comparison . . . . . . . . . . . 44
Figure 7.3: Ericsson Test Set Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
Figure 7.4: Indoor Test Set Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
Figure 7.5: Outdoor Test Set Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
List of Acronyms

ANN  Artificial Neural Network  
CNN  Convolutional Neural Network  
FP  False Positive  
FN  False Negative  
IOU  Intersection Over Union  
MSER  Maximally Stable Extremal Regions  
PPHT  Progressive Probabilistic Hough Transform  
ReLU  Rectified Linear Units  
ROI  Region Of Interest  
TP  True Positive  
TN  True Negative  
WP3M  Weatherproofing 3M Case  
YOLO  You Only Look Once
1 Introduction

Humans are increasingly reliant on machines for automating redundant and repetitive tasks. The various applications for automation help alleviate the need for human intervention. Some benefits of implementing an automated system for radio tower inspection, which is the goal of this thesis, include a faster, cheaper inspection process, and a decrease in sometimes fatal accidents involving engineers. In order to effectively utilize machine intelligence for automation, machine perception must be developed such that the system is capable of interpreting data in a human-like manner. Humans rely on their senses to form their perception, whereas computers require input from hardware devices such as cameras or other sensors. The reduction in cost and increasing capabilities of computing hardware make machine perception much more accessible than ever before. Development in cognitive intelligence paves the way for innovative solutions to develop real life applications. Deep learning has been empirically demonstrated to be effective in solving difficult problems in application domains such as automatic speech recognition, image recognition, recommendation systems, natural language processing and much more [1], [2], [3], [4]. This project focuses on object detection and CNNs have been proven to be a scalable approach for such applications [5]. CNNs are modeled based on how the human brain processes information. With the help of these state-of-the-art technologies, developing a commercial application utilizing machine learning has become feasible. For inspection processes, such an application can save time, costs, and most importantly: lives.

1.1 Motivation and Goals

Ericsson AB located in Stockholm has a Cognitive Automation Lab group that conducts research in the area of Artificial Intelligence. One of the projects currently underway focuses on improving radio site inspection processes. A radio site is a place where infrastructure necessary for supporting telecommunication operations are located. Radio base stations are deployed at these sites containing a radio tower where the hardware is mounted, consisting of: radio units, antennas, and a ground cabinet. To ensure the hardware located at radio sites is properly configured, field technicians are deployed to the sites for maintenance inspections. This inspection process involves climbing up a radio tower and manually inspecting each component. Not only is this process time-consuming and expensive, it also puts field technicians at risk of injury in the event an accident occurs. To mitigate these safety concerns for field technicians, piloted drones
equipped with cameras are being utilized for radio site examinations. Flying drones alleviates the need for engineers to climb each base station for inspection and instead technicians will only climb radio towers if there is a problem that requires servicing. Currently, the process involves flying the drone up the radio tower to survey the site, so that the radio engineer can manually evaluate the results. Although the field technicians no longer have to climb the radio tower during inspection, their presence is still required along with a qualified pilot. The objective of this project was to develop an algorithm that automatically evaluates images of antennas to detect faults in cable connections. The long term goal is to deploy this algorithm in real-life on an Android Mobile device to inspect radio towers across the world. In order to achieve this goal, a proof-of-concept that is able to detect components in the tower and provide real-time diagnostics is needed. This master’s thesis targeted a particular subset of components on the radio tower, specifically the weatherproofing on coaxial cable connections. The primary focus is automatic detection of exposed metal on coaxial cables connected to a radio site antenna. Figure 1.1 provides an example of a radio antenna that is correctly weatherproofed (left), compared with an example showcasing faulty weatherproofing (right).

![Figure 1.1: Good vs Bad weatherproofing](image_url)

(a) No metal hex nuts exposed  
(b) Metal hex nuts exposed

1.2 Methods and Structure

This project applies a CNN to detect exposed metal hex nuts as well as plastic weatherproofings and pairing them to define instances of bad weatherproofing. To achieve this, the first step is to create a dataset for training and testing. Three different datasets were analyzed based on their performance under various real-life conditions. The first dataset was supplied by Ericsson at the start of the project, and the latter two datasets were constructed from scratch during the project. The CNN configuration used for this project was provided by Ericsson at the start of the thesis. The main contribution of this project is combining the proposed image post processing after object detection to result in better performance than a standalone CNN. By processing the image after the CNN detects objects, it is possible to find additional true positives that the CNN missed by detecting...
areas containing characteristics of specular highlights. This is accomplished by applying a computer-vision algorithm called Maximally Stable Extremal Regions (MSER) [6] and pairing hex nuts with plastic weatherproofings detected by the CNN.

This project report is organized as follows: A brief overview of relevant research is discussed in Chapter 2 (Literature Review). More information regarding data acquisition and labeling can be found in Chapter 3 (Data). The tools used and their configurations chosen for the project are presented in Chapter 4 (Implementation Framework). The details regarding the experimental methodology are outlined in Chapter 5 (Methodology). Results are presented in Chapter 6 (Results), followed by a discussion and analysis in Chapter 7, and conclusions drawn from the results in Chapter 8 (Conclusion).
2 Literature Review

2.1 Machine Learning

Machine Learning is a type of Artificial Intelligence within the field of Computer Science. A characteristic that distinguishes machine learning from other types of software is its ability to "learn without being explicitly programmed" [7]. There exist four popular methods for achieving this level of cognitive intelligence: supervised, unsupervised, semi-supervised, and reinforcement learning algorithms. Supervised learning algorithms learn by using large amounts of classified/labeled data to predict classes on new, previously unseen instances of similar data [8]. Unsupervised learning on the other hand learns from data that is unlabeled and learns by detecting hidden structures and making inferences within the dataset [9]. Similarly, semi-supervised learning utilizes both labeled and unlabeled data during training and is typically implemented for applications where the labeled data requires domain specific knowledge to achieve the ability to learn from it [10]. Finally, reinforcement learning uses actors to interact with the surrounding environment to learn from receiving rewards based on trial and error [11].

The goal of this project is to detect instances of exposed metal hex nuts in previously unseen images by training a supervised machine learning algorithm on a labeled dataset. We show that this can be achieved by training a neural network and applying computer vision techniques post object detection to improve the systems overall performance.

2.2 Artificial Neural Networks

The motivation for researching neural networks was initially to model how the human nervous system processes information. In [12], Anderson et al. state that "from an information-processing point of view, the most important components of the nervous system are the neurons". A human brain contains approximately 100 billion neurons which are connected via synapses. In Figure 2.1 a drawing of a biological neuron is compared with a mathematical model of a neuron. The way information is processed starts with neurons being recipients of signals outputted from dendrites and computing a signal based on that input, which is then communicated through the neuron’s axon. Connections to other dendrites are made by synapses from an axon. In the computer model of a neuron, the signals carried through the axons interact with the outputted signal from the dendrite multiplicatively and are summed in the cell body. The idea behind artificial neural networks is that the synaptic signal from the dendrite is learnable.
and influences the strength of connections to other neurons. The input to an artificial neuron is the weighted sum of all input signals, also referred to as the net input [13].

\[
net = \sum_{i=1}^{I} x_i w_i
\]  

(2.1)

Where I represents the number of input signals.

A neuron will be activated when a summed value in the cell body is greater than a certain threshold. How often this neuron fires is modeled by an activation function which influences the frequency of the outputted signal.

![Figure 2.1: Neuron Models](image)

(a) Biological Neuron [14]  (b) Artificial Neuron Model [15]

A desirable property of activation functions is that it introduces non-linearity to the network. In theory, this facilitates the ability for modeling any function in a multi-layer perceptron [16].

### 2.3 Convolutional Neural Networks

CNNs share similar characteristics with ordinary Artificial Neural Network (ANN)s in that both are composed of an aggregation of neurons which contain learnable weights and biases. Just like an ANN, a CNN receives an input and performs a dot product; then, on one’s own accord, the output can be followed by a non-linearity. The main difference between a CNN and a regular ANN is that the former will receive an image as input to the network [17], which has much higher dimension than what is typically inputted into an ANN. With this prior knowledge, the architecture is designed to optimize the forward function while considerably decreasing the number of parameters contained in the network by applying a convolutional operation on each input in the convolutional layer. A convolutional layer reduces the amount of data needed to represent an inputted image by highlighting important, learnable features needed to represent the object of interest. Traditionally, ANNs utilize fully connected layers in which every neuron accepts input from each element from the previous layer, and each connection has a weight value associated with it. A major difference in CNNs is the application of convolutional layers, in which each neuron receives input only from local neurons in the previous layer, and the same weight is applied to every neuron. As can be seen in figure 2.2, the three main layers used to achieve this are convolutional layers followed by an
activation function, pooling layers, and a fully-connected layer. A detailed explanation of these components can be found in the following sections.

![Example of CNN Architecture](image)

**Figure 2.2: Example of CNN Architecture [18]**

### 2.3.1 Convolutional Layer

The convolutional layer is fundamental to CNNs and performs the most computationally intensive operations within the network. If the input image has dimensions [32x32x3], then width and height are 32, and the color space has 3 channels: R (Red), G (Green), B (Blue). A convolutional layer will receive these raw pixel values of the image, and will perform a dot product operation on each of the local region connected with the weights in the input volume. The process starts with defining a filter with equal length and width. The filter length and width size is usually odd and of equal size because it provides a central pixel. Next, the filter is overlayed a section of the image one stride at a time. A stride is the number of pixels traversed in between each application of the filter. The stride length is a hyperparameter defined in the network configuration, with popular parameter values for strides being either 1, or 2. Outputted from the convolution performed is a feature map seen in Figure 2.3, which is representative of the responses that portion of the image activates in the spatial position [19]. It is worth noting that the size, stride, and padding parameters influence the size of the output feature map.
The first convolutional layers extract low level features such as edges, or patches of color in the initial layer. Higher layers in the network can produce more distinctive features for objects of interest. Instead of having computer vision researchers hand pick values for the filter, it is useful to be able to learn these parameters through backpropagation.

By backpropagation, the network can learn appropriate values to e.g. accentuate the edges optimally, and is invariant to orientation. This allows the CNN to learn how to detect low level features and relevant features more robustly than user defined, computer vision approaches. The size of the input image does not affect the convolution operation as long as the number of color channels is consistent, and the number of learnable parameters do not change either.

Padding involves adding a value of 0 to the border of an image to save edges in the output. Without it, the output is shrunk- potentially losing valuable information from the edges. Padding is useful for preserving dimension size in the output feature map when convolving an image.

### 2.3.2 Activation Functions

Non-linear activation functions are applied after convolutional operations and are used to better distinguish characteristics and apprehend features that linear models are not capable of capturing. There exist a variety of activation functions popularly used in CNN architectures, some are the following: Sigmoid, Tanh, ReLU, and Leaky ReLU. The motivation for using the sigmoid function initially stemmed from it closely representing the firing of a neuron in the brain, either being 0 or 1. It has not been used as much in recent years due to the saturation of gradients. In addition, values outputted by sigmoid functions are not zero-centered. Saturating gradients are a problem because the gradient at values 0 or 1 of the sigmoid function are near zero. When a gradient is small, it can affect other gradients during backpropagation causing signals to become depleted when...
passed to the neurons weights. On the other hand, Tanh functions are zero centered and transform real-valued numbers into a range of -1 to 1. Because of this, Tanh functions are typically preferred over the use of a sigmoid activation function, but are still prone to saturating gradients. Rectified Linear Units (ReLUs) mitigate the risk of saturating gradients that sigmoid and tanh function pose [22]. Implementing ReLUs as an activation function also includes the benefit of speeding up convergence of stochastic gradient descent [2].

![ReLU vs Leaky ReLU](image)

Figure 2.4: ReLU (left) vs Leaky ReLU (right)

A problem with ReLUs that can arise during training is dead or vanishing gradients, meaning affected neurons will not be activated again. This is a result of setting a learning rate that is too high, and the risk of it occurring can be decreased by instead using Leaky ReLU activation function. If $x < 0$, ReLU will set the output (and the slope) equal to 0, whereas a Leaky ReLU will instead have a small negative constant slope (alpha), usually 0.01. These two activation functions are defined as:

$$ReLU : f(x) = \max(0, x) \quad (2.2)$$

$$LeakyReLU : f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x) \quad (2.3)$$

### 2.3.3 Pooling Layer

Pooling layers are commonly used after the convolutional layer to subsequently reduce the number of parameters in the network. They mitigate the risk of overfitting by performing operations that decrease spatial size and extract features which are invariant to position. Dimensionality reduction also minimizes the amount of computational power needed for processing each representation.

Max pooling is the most popular implementation whilst utilizing pooling layers. It works by defining a filter size with equal dimensions that will iteratively apply the max operation on each stride. Stride lengths typically range from 1-2 steps per stride.
2.4 Computer Vision

2.4.1 Specularity

The prevalence of a specular reflection in real-life enhances the perception of an observed scene. Whether it is a beam of light bouncing off the frames of a friend’s sunglasses on a hot summer day, or a field covered in untouched snowfall sparkling from the sun’s persistently projected rays, they can be found in nature on a daily basis. In both of the examples mentioned above, what happens to light when it penetrates the surface of an object is telling of how it will appear to the viewer, which can also be captured when imaging such a scene. An image such as the one referenced above can be decomposed into component images based on their intrinsic properties— which define low-level features found at every point. Information encapsulated within an image has various forms of representation, often referred to as intrinsic image properties, defined by Barrow et al as being: range— which corresponds to the center projection of visible points, orientation— which portrays the direction of surfaces as a vector, reflectance— the amount of light an object reflects is dependent on material, and incident illumination— a way of measuring illuminations of an objects surface [23]. The reflective and incident illumination properties are important when dealing with specularity because they can be used to separate specular reflections from an input image. Recovering these two intrinsic image properties becomes an ill-posed problem since there are more unknown variables than there are to define them, but are useful in making inferences about the imaged scene. "Denoting by $I(x,y)$ (i.e the input image) and by $R(x,y)$ (i.e the reflectance image) and $L(x,y)$ (i.e. the illumination image), the three images are related by:" [24]

$$I(x,y) = L(x,y)R(x,y)$$

(2.4)

There are two reflective components when dealing with intrinsic properties, interface (specular) and body (diffuse). The incident light that penetrates an object’s surface can be separated into two categories of reflections, specular and diffuse. Figure 2.6 shows how
a specular reflection occurs when incident light is reflected in between the objects surface and the surrounding air at the interface. Conversely, diffuse reflections are caused by light scattering after subsequently penetrating an objects surface when light strikes the material interface [25]. There are only a few known materials which do not have diffuse reflections, one of which being metal. In this project, the object of interest that we would like to identify is a hex nut made out of metal. By applying an object detector, such as a CNN to find occurrences of black plastic weatherproofing cases and metal hex nuts, we can further utilize information regarding specular highlights to increase the performance of identifying instances of metal hex nuts.

Figure 2.6: Types of reflections [26]

Proposed techniques for localizing regions of specularity vary based on the method of capturing a scene. Examples on how they differ include using multiple images from altering viewing angles to detect specularity, and single image specularity detection, ‘there have been five principal approaches so far in solving the problem of specularity detection or separations: using (1) a single grey scale image, (2) different light directions, (3) polarization, (4) color, (5) color and polarization, and (6) color images and multiple viewpoints’ [27].

When conducting the initial research on the topic of specularity detection, a few constraints were considered. As mentioned above it is important that little-to-no apriori knowledge is used due to the fact that the observed scene is subject to change and will not be consistent in every instance. Another limitation on the solution is the desire for it to be able to perform in real-time on a mobile device. The end product will consist of a radio site engineer watching a video stream on a tablet or cellular phone from a drone. To achieve this, a single image approach is necessary for the fastest possible processing due to the fact that a multiple image approach would require capturing footage from
multiple perspectives, which is not feasible to do in real-time. For this reason, single grey scale image approaches for specularity detection were researched. A paper by Zhu et al. [28] on specularity detection based on MSER was found to be useful for this approach. The authors were able to reliably detect presence of specularity with no assumptions of the scene, and separate specularities regardless of the lighting conditions or surface material on a single image. Most importantly, they show that the method is useful for both metal and dielectric surfaces. The algorithm used in this approach is much less computationally expensive relative to a multiple image specularity detection solution, and can define maximally stable extremal regions in $O(n\log(n))$. The discovery of this research paper influenced the decision to implement MSER for validating the presence of exposed metal hex nuts in the proposed solution.

2.4.2 Maximally Stable Extremal Regions

Extremal Regions are connected areas characterized by almost uniform intensity, surrounded by contrasting background[6]. MSER is a feature extraction algorithm commonly used for blob detection in computer vision. Matas et. al proposed MSER as a solution for the wide baseline stereo problem- matching corresponding features in two images of a scene from different perspectives[6]. MSER is commonly used in applications for extracting text from images, but its methodology inspired Zhu et. al’s approach for detecting regions of specularity [28]. In this work, we are interested in detecting hex nuts, which contain a high degree of specularity, and therefore are a form of Extremal Regions. By adopting Zhu et. al’s approach, we can combine our object detector’s predictions of weatherproofing cases with regions of specularity detected by MSER to increase the system’s overall accuracy in detecting examples of bad weatherproofing. To introduce the process of identifying MSERs in an image, figure 2.7 was taken from [6] to formally define vocabulary used in the following explanation:
To find the maximally stable extremal regions, a grayscale image is used as input and its pixels are sorted by intensity. Then, each pixel is localized in increasing order while keeping track of each connected component found throughout the image in a list. Every connected component also has its area preserved by the union-find algorithm [29]. Union-find detects whether a connected components has an area overlapping with another connected component. If two connected components overlap during this process, they are merged by adding the pixels of the smaller component to the larger component and discarding the former. Matas et al. define extremal regions as having "highly desirable properties: the set is closed under 1. continuous (and thus projective) transformation of image coordinates and 2. monotonic transformation of image intensities "[6]. These extremal regions are detected by first identifying sets of contiguous pixels where surrounding boundary pixels have higher intensity values than those contained in the inner boundary. The algorithm then calculates the division of cardinalities to find extremal regions whose size over and iteration of thresholds are not significantly larger than the previous threshold. Figure 2.8 illustrates a blob and it’s extremal region’s, Q(i), as it iterates over various threshold values. Extremal regions that are smaller or larger than the minimum and maximum parameters set will not be considered as a maximally stable extremal region. These parameters are set by the user when instantiating the detector in OpenCV, and discussed in Chapter 5.
To visualize the process of thresholding and the regions which remain consistent over thresholds, an example image has been used for 2.9. A wide range of thresholds are applied to the image one iteration at a time, and pixels whose intensity value is less than the threshold will become black, whereas those greater than or equal to the threshold are white. As can be observed in Figure 2.9, starting from a low threshold intensity and gradually increasing the threshold iteratively causes regions of local intensity minima to grow and eventually merge. With enough increments in threshold intensity, the image will become completely black.

Regions that remain relatively close in size throughout the different threshold values are considered extremal regions. A parameter, $\Delta$, is used to define whether a region has changed in size over an iteration in threshold value. Maximal regions are sets of components that are connected throughout all threshold frames. Minimal regions can be found by performing the same process, but instead inverting the threshold intensity for each frame. This means that both consistently dark regions, as well as bright regions within an image can be found using MSER. As specular reflections are very bright areas that can be found on metal hex nuts, this project will focus on only maximal regions. By complementing MSER with an object detector, we can enhance our performance of detecting bad weatherproofings by suggesting regions with high specularity found by
MSER that are close to detected weatherproofing cases to contain a metal hex nut, which may have been missed by the detector.

### 2.5 Progressive Probabilistic Hough Transform

The Hough transform algorithm is a computer vision algorithm popularly used for detecting lines in images[31]. Lines are of particular interest for this project because weatherproofing cases are always located under an antenna, whose edge appears in an image as a horizontal line. The region of interest for detecting specularity with MSER is between the top of a weatherproofing case and the bottom of the hosting antenna. Therefore, we applied a Hough Transform to detect lines, specifically to find the bottom of the antenna.

A set of points \((x,y)\) contained within an image characterizing a straight line is defined by a relation, \(f\), such that

\[
f((\hat{m}, \hat{c}), (x, y)) = y - \hat{m}x - \hat{c} = 0,
\]

in which \(\hat{m}\) and \(\hat{c}\) are the parameters for slope and intercept defining a line. To help define lines within an image canny edge detector was used as a preprocessing step. The Canny edge detector first reduces noise in the image by applying a Gaussian filter, then computes intensity gradients using a Sobel filter, and concludes by applying non-max suppression and hysteresis thresholding to remove small, non edge pixels [32]. The resulting image is then used as input to Hough Transform which maps points from a Euclidean plane to a Hough plane. A straight line represented by \(y = mx + b\) in the Euclidean plane by \(x\) vs \(y\) and can be modeled in the Hough parameter space as \(x\cos\theta + y\sin\theta = p\) as \(m\) vs \(b\). The distinction between these representations is that a line in a Euclidean plane can be represented as a single point \((m, b)\), or \((p, \theta)\) in the Hough plane. The algorithm then iterates through each pixel in the image and finds edges where points align. Each line found corresponds to a Hough point and is incremented for each time a correlation is found, commonly referred to as a vote. A threshold value is then used to define whether a point in the Hough plane can be defined as a line.

A major drawback of using the standard Hough transform is its large storage and computational requirements [33]. In order to successfully detect the bottom of the radio antenna in a real-time application, an optimized version of the Hough transform has been utilized called the Progressive Probabilistic Hough Transform. Its computational speed is influenced on its ability to "exploit the difference in the fraction of votes needed to reliably detect lines with different numbers of supporting points"[34]. This allows less memory to be consumed as well as less time spent for finding lines in an image. However, lines outputted from the Progressive Probabilistic Hough Transform are more frequent and shorter than the original Hough implementation, where the lines extend infinitely.
2.6 Precision and Recall

When evaluating the results of a classification system it is common to measure the performance in terms of precision and recall. Precision represents the fraction of instances correctly identified out of all the predictions outputted by the model. Meanwhile, recall is the fraction of correctly identified instances out of all the instances that are relevant. For example, assume the object detector predicts three exposed hex nuts in an image which has four exposed hex nuts. Of the three predicted hex nuts only two are correctly identified (True Positive (TP)), and one is incorrectly classified as a hex nut (False Positive (FP)), as well as the one hex nut that went undetected (False Negative (FN)). This would result in a precision value of 2/3 and a recall of 2/4. If the detector did not find any object where there was none, it would be considered a True Negative (TN).

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

\[
AP = \int_{0}^{1} p(r)dr
\]

Figure 2.10: Precision Recall Illustration [35]
3 Data

3.1 Labeled Training Data Acquisition

The dataset was generated by capturing video recordings of radio antennas on an iPhone 6s, showcasing both good and bad examples of weatherproofing. The videos were then used as input to an open source labeling tool called OpenLabeling. This tool is particularly useful because it provides the software to convert each video into a sequence of images and track objects frame by frame. Tracking works by setting bounding boxes with labels around objects of interest, then by using OpenCV the tool automatically tracks the objects while traversing the video one frame at a time. This drastically reduces the nuances of gathering and manually labeling objects as the size of the dataset increases. A bounding box is first created by selecting two corners e.g. top left and bottom right, to fully encapsulate the object of interest, then the open sourced labeling tool will output coordinates for each bounding box in the image in .txt file in both YOLO and PASCOL VOC format. Two classes are of interest when labeling data for our set: the exposed metal hex nut, and w3pm plastic weatherproofing case as can be observed in Figure 3.1.

Figure 3.1: Bounding boxes around objects of interest
3.2 Datasets for Training

3.2.1 Homogeneous dataset

At the start of the thesis project an existing dataset created by Ericsson was available for training. Figure 3.2 contains examples of various images found in the dataset. It features multiple views of a radio antenna with a sky blue wallpaper as the background and a strip of lights switched on/off. Although the dataset included approximately 16,500 images, it contained a relatively small set of unique examples. This is a result of using sequences of images from multi-second videos that have little to no movement. When this is the case, images transformed from the video will appear very similar, if not identical. If a five second video with little movement is transformed into frames and an iPhone 6s records videos at 30 frames per second, the resulting sequence is 150 images that are nearly identical.

Figure 3.2: Examples of images found in old dataset

Another noticeable trait of the existing dataset is that it has images of one radio antenna. For a CNN to generalize well in new scenarios, there need to be examples in the training set of multiple radio antennas in various settings. After analyzing the contents of the provided dataset it was determined that a new dataset with more variation and less redundancy needed to be developed.

3.2.2 Diverse Dataset

The diverse dataset compiled in-house consists of approximately 20,000 images, varying by attenuating light, different viewing perspectives, as well as increasing distances between the camera and radio station. Three separate locations within the Ericsson HQ building with variation in setting were utilized for imaging: one being an antenna attached to a wall in a lab setting, and two radio base stations located in different parts of the campus.
Some measures of quality control include making sure there is sufficient lighting, and that the data capture source is always the same. Negative samples have also been captured to help the network generalize better. These samples showcase the radio site with similar backgrounds but no objects of interest to be found. In general, whenever a hex nut is exposed there will also be a plastic weatherproofing mechanism located directly below. To reduce the chances of our network learning correlations between the relationship of presence of metal hex nuts and weatherproofings, examples of only weatherproofings and only metal hex nuts have also been provided in the dataset. The training set has about 90% of the images, while the validation set makes up the remaining 10% to improve the network performance over time.

### 3.2.3 Indoor and Outdoor Dataset

The last dataset constructed has all the indoor examples described from Section 3.2.2, and additionally samples taken from outdoors. Figure 3.4 previews images found in the training set appended to the diverse set to create this new dataset.

Table 3.1 shows the number of instances of each object found within their respective dataset. The expose metal hex nut has been shortened to hex, and the plastic weatherproofing cases are named wp3m.
Table 3.1: Training Datasets Metadata

<table>
<thead>
<tr>
<th>Number</th>
<th>HomoHex</th>
<th>DiverseHex</th>
<th>InOutHex</th>
<th>HomoWp3m</th>
<th>DiverseWp3m</th>
<th>InOutWp3m</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>12015</td>
<td>1679</td>
<td>1919</td>
<td>2</td>
<td>1556</td>
<td>1845</td>
</tr>
<tr>
<td>1</td>
<td>1675</td>
<td>1043</td>
<td>5195</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>2007</td>
<td>5528</td>
<td>6087</td>
<td>2089</td>
<td>1037</td>
<td>5674</td>
</tr>
<tr>
<td>3</td>
<td>745</td>
<td>8876</td>
<td>8876</td>
<td>8824</td>
<td>5012</td>
<td>5012</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>2970</td>
<td>2970</td>
<td>143</td>
<td>9777</td>
<td>9777</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>79</td>
<td>79</td>
<td>0</td>
<td>894</td>
<td>894</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5384</td>
<td>1888</td>
<td>1888</td>
</tr>
</tbody>
</table>

3.3 Datasets for Testing

Three separate datasets were constructed from scratch to test the performance of the CNN detector and the post processing algorithm. Two methods were used when labeling these datasets. We were interested in how well the detector is able to identify and locate the two objects of interest: exposed metal hex nuts and black plastic weatherproofings, and how well the algorithm is able to detect instances of bad weatherproofing. In figure 3.5, one can observe three examples of bad weatherproofings. The reason for encapsulating the hex nut and weatherproofing in one bounding box stems from the fact that a hex nut detected by itself does not constitute a bad weatherproofing. For example, if an unrelated component, separate from the radio antenna of interest in this project contains a hex nut that is not meant to have a weatherproofing case then it should not be flagged as a bad weatherproofing. The detected hex nut is paired with a weatherproofing case and can then be considered a bad weatherproofing. In section 5.3 the method for pairing hex and weatherproofing objects is explained in further detail.
3.3.1 Ericsson Test set

The Ericsson supplied dataset has 94 images of the various radio antennas used for training the models. It features 24 indoor images with 4 views, and the rest being outdoor images from 6 different settings of varying lighting conditions.

3.3.2 Indoor Test set

The indoor test set was crafted independently and consists of samples of the mobile radio antenna in various settings that have not been trained on by any of the models. This test set has a total of 1083 different images. Figure 3.7 shows various examples of images found in the indoor test set.
3.3.3 Outdoor Test set

The second dataset individually created features the mobile radio antenna outdoors on a cloudy, slightly rainy day. It includes 361 images with 7 unique views.
4 Implementation

4.1 Hardware

Training the model is done on a dedicated machine running Ubuntu 16.04.6 Xenial equipped with a GTX GeForce 1080 GPU by Nvidia. Testing the model is done on a virtual machine running Ubuntu 18.04 bionic allocated 4GB of RAM, 1 processor, and 60GB of memory storage. The virtual machine is hosted on an HP EliteBook 840 G5 with an Intel(R) Core (TM) i7-8650U CPU @ 1.90GHz 2.11GHz, 32GB RAM, and Windows 10 x64.

4.2 Darknet

Darknet [36] is an open source framework developed by Joseph Redmon for neural networks, popularized by You Only Look Once (YOLO) [37]. It is written in C and CUDA, which allows developers to easily build custom networks on top of the framework. For this project we utilized the Darknet framework for creating a custom made Convolutional Neural Network, which was created and provided by Ericsson. The benefits for utilizing such a system are the speed of object detection and the lightweight computational costs for doing so. As this project is intended to be run on an Android mobile device, it is important that speed and computationally heavy processes are fully optimized. Some versions of the tablets do not have the hardware to support real-time implementations of various versions of YOLO. It was originally intended for the project to implement the real-time object detection CNN YOLO, but decided against it due to the hardware limitations within a mobile device.

4.3 OpenCV

OpenCV is an open source framework written in C/C++ for popular computer vision and machine learning algorithms. OpenCV supports multiple programming languages by providing wrapper classes, for this project version 4.0.1 was used to implement the solution in Java 12.0.1.
5 Methodology

Detecting presence of metal is a difficult computer vision task not only because it is a highly reflective material, but due to the appearance of a metal object being influenced by the illumination, angle of viewer, and surrounding environment. The majority of computer vision algorithms are designed with the assumption that all surfaces are perfectly Lambertian. In layman terms, this simply means that regardless of the angle of view from the observer’s perspective, the surfaces’ brightness will always be consistent[4]. These same algorithms will consider specular pixels as noisy outliers within the image, making stereo matching, photo consistency, segmentation, recognition, and tracking a problem. In our case, specularity should not be considered as noise, but instead thought of as containing useful information about the scene. However, instances where reflective metal exists within a region of interest but no metal hex nut is exposed can lead to false positive detections of bad weatherproofing.

This means as little apriori knowledge as possible is to be assumed when developing the algorithm, as the lighting conditions, angle of view, and occluding environments are variable to change. With this in mind, the following approach has been proposed.

5.1 Approach

The process begins with the CNN architecture chosen for training the model. It is fed images of radio antennas in a variety of scenarios with an associated .txt file describing the relative locations of both plastic weatherproofing cases and metal hex nuts as bounding boxes. The detector is trained on examples showcasing good weatherproofing (no exposed hex nuts), bad weatherproofing (one or more exposed hex nuts), no weatherproofing (only hex nuts), and negative samples (no weatherproofings or hex nuts).

The configuration for the CNN was developed by Ericsson and provided at the start of the thesis project, which can be observed in Figure 5.1.
Methodology

OpenCV allows us to compile the CNN in Java for the application and takes each frame as input to the model. After sending the image to the detector, the objects found by the detector are outputted with their absolute coordinates. For this project, only objects with a confidence value greater than 0.7 were considered.

5.2 Baseline

At the start of the thesis project, Ericsson already had a solution for pairing the detected hex objects with the black plastic WP3M weatherproofing cases. This process involves traversing each detected hex nut and weatherproofing case to compare their respective bounding box dimensions on the top left, top right, and central point. In figure 5.2, one can observe the points used for deciding if a hex nut detected is paired with a weatherproofing case.

![Pairing Diagram](image)

Figure 5.2: Pairing Diagram
Methodology

In order for a hex nut object to be paired to a weatherproofing case, the following requirements must be met:

\[(\text{HEX}_{\text{CP}}_X < \text{WP}_{\text{TR}}_X) \& (\text{HEX}_{\text{CP}}_X > \text{WP}_{\text{TL}}_X) \& (\text{HEX}_{\text{TOP}}_Y < \text{WP}_{\text{TOP}}_Y)\]

where \(\text{HEX}_{\text{CP}}_X\) is the hex nuts central point \(x\) value, \(\text{WP}_{\text{TR}}_X\) is the weatherproofings top right \(x\) value, \(\text{WP}_{\text{TL}}_X\) is the weatherproofings top left \(x\) value, \(\text{HEX}_{\text{TOP}}_Y\) is the hex nuts top \(y\) value, and \(\text{WP}_{\text{TOP}}_Y\) is the weatherproofings top \(y\) value.

The reason for choosing such an approach instead of calculating the Euclidean distance between the bottom of the hex nut and top of the weatherproofing is because the unknown distance between the exposed hex nut and weatherproofing case in practice. The weatherproofing case is typically just below the exposed hex nut, but there are some instances in which the weatherproofing case falls far below the hex nut, in which case a fixed Euclidean distance for pairing would not be suitable. So, with this baseline in mind, a hex nut must be paired with a black plastic weatherproofing case to be considered a bad weatherproofing, a detected hex nut alone is not enough.

5.3 Image Post Processing

As previously mentioned, the metal hex nuts pose a problem for computer vision tasks due to the high degree of specularity. Often times researchers will mitigate this problem by preprocessing the image to remove the specular regions. Instead of removing the specular highlights from the image, they can be used to our advantage by applying image post processing. This post processing technique focuses on improving the results of bad weatherproofing detections. Given a list of hex nuts and weatherproofing cases outputted by the detector, the algorithm developed will first attempt to create pairs of potentially bad weatherproofings using the baseline approach mentioned in section 5.2. For all weatherproofing cases that are not paired, the following algorithm will be applied.

First, it is known that if an exposed hex nut is present within the image it will be located above a plastic weatherproofing. The distance from which the camera is capturing images of the cable connections may change. But, the size of weatherproofing cases are fixed as well as the size of the metal hex nuts, so a Region Of Interest (ROI) can be made based on these assumptions. A ROI is defined above as all unpaired weatherproofings with the same width as the Weatherproofing 3M Case (WP3M) bounding box, and the height equal to 1/8 of the WP3M bounding box height. Another assumption that can be made is the fact that weatherproofing cases are always located underneath a radio antenna. These antennas have a straight horizontal edge that can be detected to aid in locating additional hex nuts missed by the object detector. The Progressive Probabilistic Hough Transform (PPHT) is applied to the image to find lines. Ideally, horizontal lines are the only ones of interest, as the bottom of the antenna is always apparent across the image. Due to slight variations of the antenna angle, the lines kept for further analysis...
Algorithm 1: Bad weatherproofing detection

\[
\text{mserRegionsList} \leftarrow \text{mser(image)}
\]
\[
\text{for } wp \text{ in unpairedWpList do}
\]
\[
\text{for } houghLine \text{ in houghLineList do}
\]
\[
\begin{align*}
\text{if } & \ wp \text{.isBelow(houghLine)} \quad \& \quad (\text{houghLine\text{.startX} < wp\text{.endX} } \quad \& \\
& \text{houghLine\text{.endX} > wp\text{.startX}) then}
\end{align*}
\]
\[
\begin{align*}
\text{hBottom} & \leftarrow \text{houghLine\text{.getBottomLineCoords()}} \\
\text{wpTop} & \leftarrow \text{wp\text{.getTopLineCoords()}} \\
\text{roi} & \leftarrow \text{newRect(hBottom, wpTop)}
\end{align*}
\]
\[
\text{for } \text{boundingBox in mserRegionsList do}
\]
\[
\begin{align*}
\text{if } & \ \text{middleCoordInsideRoi(boundingBox, roi) then}
\end{align*}
\]
\[
\text{roi\text{.setSpecualr(True)}}
\]
\[
\text{end if}
\]
\[
\text{end for}
\]
\[
\text{end if}
\]
\[
\text{end for}
\]

are within 30 degrees. In order to filter lines found by PPHT that are not of interest, the follow conditions must be met: First, the line must be above the top of an unpaired weatherproofing. Next, the \( x \) value of the startpoint coordinate for the Hough line found must be less than the the endpoint of the unpaired weatherproofing, as well as the \( x \) value of the endpoint coordinate of the Hough line must be greater than the \( x \) value of the startpoint of the unpaired weatherproofing.

Next, the top of each region of interest defined for unpaired weatherproofing cases is compared to the Hough lines found by the detector. To ensure the Hough line is in the correct vicinity to the defined ROI, it must satisfy the following criteria: The Hough line must be above the bottom of the ROI (equivalent to the top of the weatherproofing bounding box), the starting point of the Hough line must be less than the top right \( x \) dimension of the ROI, and the ending point of the Hough line must be greater than the top left \( x \) dimension of the ROI. The Hough line that passes all these requirements and is the minimum distance to the top of the ROI, will become the new \( y \) value for the top of that ROI. By applying this method, we are able to fit the ROI to the bottom of the antenna, as that is where the exposed metal hex nut will reside, if present.

Lastly, the image is downsized to half of the original dimension and inputted to the MSER detector. This is to detect areas of specularity within the image as an indication of where hex nuts are located. Downsizing the image helps speed up the detection process. The outputted contours and bounding boxes of MSERs are resized to the original image size dimensions. The list of bounding boxes containing MSERs outputted by the detector is traversed and compared to the ROI for each unpaired weatherproofing. If the middle point of the MSER bounding box is within the ROI, it is considered a bad weatherproofing.
Figure 5.3: Applying MSER
The reason for doing the image post processing is to help find true positive instances of hex nuts (thereby deeming it a bad weatherproofing) missed by the detector. Figure 5.4 shows an example where the detector successfully found three weatherproofing cases, but was unable to locate the two exposed metal hex nuts.

Since three weatherproofings were found and subsequently unpaired due to the lack of hex nut detections found by the detector, the algorithm was run and returned with two new bounding boxes signifying bad weatherproofings. In 5.4, one can observe the middle weatherproofing case outlined by a blue line. The color blue symbolizes black weatherproofing detections, and the red bad weatherproofings. Towards the top of the red bounding boxes there is a visible blue line near the hex nut, this also represents a weatherproofing detection. One can observe that the top of both bad weatherproofing bounding boxes is a light blue line which is the Hough line found by Progressive Probabilistic Hough Transform. Each of the weatherproofing detections searches above the top of their bounding box for a Hough line. If one is not found, which was the case with the middle weatherproofing, the algorithm will not consider it when searching for the presence of MSERs. This example was successful because neither of the exposed hex nuts were detected, so above the top of the blue line and below the bottom of the radio antenna is where the region of interest is defined. Two of the three regions of interest contain exposed hex nuts, which appear as green in the image, representing areas of specularity.
6 Results

6.1 Evaluation Method

The process of evaluating the results retrieved from the proposed method is as follows. A desktop version of the Android mobile app has been developed as a standalone Java application for testing purposes. This will facilitate the process of applying the algorithm to an entire test dataset made for testing this project. Three separate test sets were used to get a deeper understanding of the limitations and capabilities of the detector and post processing solution. The first test set provided by Ericsson AB, which will now be referred to as the Ericsson test set, include images that resemble real life scenarios. The Ericsson test set contains footage of the radio antennae never before seen by the detector in new areas of the Ericsson building as well as outdoors. The second test, created by the Author, features only images of the radio antenna inside the Ericsson building in scenes previously unseen by the detector. This test set will be referred to as the Indoor test set. The last test set used for testing the performance of the detector and post processing method is the Outdoor test set, which was also constructed by the Author, and features the radio antenna in various positions and angles in only outdoor settings.

The program reads an image one at a time, applies the algorithm, and outputs: class, confidence, and absolute coordinates of each bad weatherproofing into a .txt file named after the image file. By saving the coordinates for every bad weatherproofing detected by the standalone detector and the post processing method, we are able to make comparisons of the performance for both approaches. This is done by calculating the precision and recall of each detected object and plotting the results as a line graph with precision represented on the y-axis, and recall on the x-axis. For this project, the threshold chosen for a true positive Intersection Over Union (IOU) was 0.3 instead of the industry standard of 0.5. The reason behind this decision results from the end product being visually inspected by a user on a radio site in real time, and the difficulty for a human to distinguish the difference between an 0.3 IOU and 0.5 IOU \[37\], \[38\]. However, it is important to acknowledge that lowering the threshold for a true positive detection also increases success rate. In order to show the benefit of applying this algorithm in a real-life case scenario, the results have been compared to a standalone object detector, versus the combination of object detection with computer vision techniques. Since the beginning of the thesis project, many changes have been made to the already existing solution, which consisted of using just a CNN to detect hex nuts. The original implementation will be used as a baseline.
Before delving into results, it is worth noting that approximately mid-way through the thesis project a critical bug, inserted before this project started, was found in the code being used as a baseline, which significantly affected network performance. At the time of discovery, each frame was being preprocessed by subtracting the mean-value intensity on each of the RGB channels. This preprocessing step is not unusual in practice as there exist detector frameworks that expect this normalized input [2]. With the Darknet framework however, this preprocessing is done automatically and is not expected as an input, thus resulting in worse performance. To fix this bug, mean-value subtraction was removed altogether from the data preparation phase for network inputs. Figure 6.1 plots the precision and recall performance of the homogeneous dataset on each test set with preprocessing and without preprocessing. It is evident that the detector is significantly better at detecting hex nuts without mean value subtraction. With this in mind, it is assumed that mean value subtraction is not conducted as preprocessing for each of the following tests.

![Figure 6.1: Mean Subtraction performance detecting hex nuts](image)

Figure 6.1: Mean Subtraction performance detecting hex nuts
6.2 Ericsson Test Results

In order to effectively evaluate the performance of the bad weatherproofing detection and post processing method proposed, it is important to first see how the detector performs on identifying hex nuts and weatherproofing individually. Figures 6.2 and 6.3 showcase performance of each of the three models' performance on locating hex nuts and weatherproofings on the Ericsson test set. Furthermore, figure 6.4 displays the results for bad weatherproofing detections on each model, while figure 6.5 visualizes the post processing results. On this test set, the post processing helped on all of the models, however the In and Out model containing images from both indoor and outdoor settings, the increase was not significant. This can be attributed to the low number of missed hex nut detections.

![Hex Detection on Ericsson Test Set](image_url)

**Figure 6.2: Hex detector performance on Ericsson Test set**
Table 6.1: Ericsson Hex Results
Total Positives: 136

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor</th>
<th>Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>98</td>
<td>96</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>22</td>
<td>27</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td>38</td>
<td>40</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.3: Weatherproofing case detector performance on Ericsson Test set

Table 6.2: Ericsson WP3M Detection Results
Total Positives: 232

<table>
<thead>
<tr>
<th></th>
<th>Homogenious</th>
<th>Diverse</th>
<th>Indoor</th>
<th>Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>220</td>
<td>225</td>
<td>231</td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>150</td>
<td>18</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td>12</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Figure 6.4: Detector performance on Ericsson Test set

Table 6.3: Ericsson Bad Weatherproofing Results
Total Positives: 136

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>106</td>
<td>115</td>
<td>133</td>
</tr>
<tr>
<td>FP</td>
<td>9</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>FN</td>
<td>30</td>
<td>21</td>
<td>3</td>
</tr>
</tbody>
</table>
Figure 6.5: Detector performance with post processing on Ericsson Test set

Table 6.4: Ericsson Bad Weatherproofing Detection Results with Post Processing

Total Positives: 136

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>117</td>
<td>128</td>
<td>134</td>
</tr>
<tr>
<td>FP</td>
<td>27</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>FN</td>
<td>19</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>
6.3 Indoor Test Results

Figures 6.6 and 6.7 contain results for individual detections of hex and wp3m objects, while figures 6.8 and 6.9 show the baseline and post processing results.

![Hex Detection on Indoor Test Set]

Table 6.5: Indoor Hex Detection Results

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>1071</td>
<td>1934</td>
<td>1910</td>
</tr>
<tr>
<td>FP</td>
<td>372</td>
<td>92</td>
<td>253</td>
</tr>
<tr>
<td>FN</td>
<td>999</td>
<td>136</td>
<td>160</td>
</tr>
</tbody>
</table>

Figure 6.6: Hex detector performance on Indoor test set
Figure 6.7: Weatherproofing case detector performance on Indoor test set

Table 6.6: Indoor WP3M Detection Results
Total Positives: 2070

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>1785</td>
<td>2025</td>
<td>1996</td>
</tr>
<tr>
<td>FP</td>
<td>1603</td>
<td>165</td>
<td>248</td>
</tr>
<tr>
<td>FN</td>
<td>285</td>
<td>45</td>
<td>79</td>
</tr>
</tbody>
</table>
Figure 6.8: Detector performance on Indoor test set

Table 6.7: Indoor Bad Weatherproofing Detections
Total Positives: 2070

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor / Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>1296</td>
<td>1950</td>
<td>1954</td>
</tr>
<tr>
<td>FP</td>
<td>49</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>FN</td>
<td>774</td>
<td>120</td>
<td>116</td>
</tr>
</tbody>
</table>
Figure 6.9: Detector performance with post processing on Indoor test set

Table 6.8: Indoor Bad Weatherproofing Results with Post Processing
Total Positives: 2070

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>1610</td>
<td>1976</td>
<td>1981</td>
</tr>
<tr>
<td>FP</td>
<td>168</td>
<td>7</td>
<td>24</td>
</tr>
<tr>
<td>FN</td>
<td>460</td>
<td>94</td>
<td>89</td>
</tr>
</tbody>
</table>
6.4 Outdoor Test Results

Figures 6.10 and 6.11 contain results for individual detections of hex and wp3m objects, while figures 6.12 and 6.13 show the baseline and post processing results.

![Hex Detection on Outdoor Test Set](image_url)

Figure 6.10: Hex detector performance on Outdoor test set

Table 6.9: Outdoor Hex Detection Results
Total Positives: 353

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>47</td>
<td>255</td>
<td>329</td>
</tr>
<tr>
<td>FP</td>
<td>284</td>
<td>38</td>
<td>5</td>
</tr>
<tr>
<td>FN</td>
<td>306</td>
<td>98</td>
<td>24</td>
</tr>
</tbody>
</table>
Figure 6.11: Weatherproofing case detector performance on Outdoor test set

Table 6.10: Outdoor WP3M Detection Results

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor/Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>502</td>
<td>441</td>
<td>618</td>
</tr>
<tr>
<td>FP</td>
<td>980</td>
<td>192</td>
<td>86</td>
</tr>
<tr>
<td>FN</td>
<td>162</td>
<td>223</td>
<td>46</td>
</tr>
</tbody>
</table>
Figure 6.12: Detector performance on Outdoor test set

Table 6.11: Outdoor Bad Weatherproofing Detection Results
Total Positives: 353

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>176</td>
<td>262</td>
<td>330</td>
</tr>
<tr>
<td>FP</td>
<td>87</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>FN</td>
<td>177</td>
<td>91</td>
<td>23</td>
</tr>
</tbody>
</table>
Figure 6.13: Detector performance with post processing on Outdoor test set

Table 6.12: Outdoor Bad Weatherproofing Detection Results with Post Processing

<table>
<thead>
<tr>
<th></th>
<th>Homogenous</th>
<th>Diverse</th>
<th>Indoor Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>204</td>
<td>288</td>
<td>347</td>
</tr>
<tr>
<td>FP</td>
<td>218</td>
<td>4</td>
<td>19</td>
</tr>
<tr>
<td>FN</td>
<td>149</td>
<td>65</td>
<td>6</td>
</tr>
</tbody>
</table>
7 Discussion & Analysis

7.1 Ericsson Set

The hex detection performance of homogeneous (66.86% AP) and diverse (67.95% AP) models are comparable on the Ericsson test set. Upon further inspection, the bounding boxes outputted by the diverse model are typically much smaller than the homogeneous predictions. Both tight and loose bounding boxes that are slightly offset the ground truth are prone to being considered a false positive, since the IOU yields less than 0.3. This can be seen in figure 7.1, as both of these hex detections were labeled as false positives (IOU < 0.3). Both homogeneous and diverse models showed limitations on detecting hex objects in outdoor settings, whereas the indoor and outdoor model performed well on both settings but missed a couple indoor hex detections.

![Hex Bounding Box Comparison](image)

Figure 7.1: Hex Bounding Box Comparison

An interesting observation can be made when comparing the individual hex nut detection performance with the baseline bad weatherproofing detection performance. Since the same detected objects from the image are used to create the bad weatherproofing bounding box, it is intriguing to see that the baseline outperforms the hex nut detection. How could we be ‘finding’ more true positives before the post processing is applied? It is because of the pairings made between each detected hex nut and weatherproofing case. The bounding box for a bad weatherproofing extends from the bottom of a detected paired weatherproofing to the top of the paired hex nut for that weatherproofing. While the hex nuts in figure 7.1 were deemed false positives, they were paired with the weatherproofing detected beneath them to create a bad weatherproofing bounding box. In figure 7.2, the red bounding boxes symbolize bad weatherproofings and blue symbolizes detected weatherproofing cases that were left unpaired.
Overall, the post processing applied to the images after being run through the detector increase the average precision on each of the models. The homogeneous model increased 4.9%, finding 9 additional instances of bad weatherproofing while introducing 18 false positives. The diverse model had the greatest increase in average precision, being 9.57%. With the post processing techniques, we were able to find 13 additional hex nuts while introducing only 1 false positive. The Indoor and Outdoor model increased only 0.02% in average precision, as it had exceptional performance on the Ericsson test set at 97.65% without any post processing. The post processing identified 1 additional hex nut while introducing 2 false positives.

Below in figure 7.3, instances where the post processing on the Ericsson test set did and did not work are included. All of the different models had difficulty detecting the hex nut in the image used for ‘Diverse TP’ and ‘In and Out TP’, which features the antenna with the sky in the background and sun on the lefthand side. The post processing was able to find a Hough line corresponding with the antenna and an area of specularity within the ROI to identify a bad weatherproofing true positive missed by the detector. ‘Diverse TN’ shows an example where the bad weatherproofings were correctly paired by the detector, and the middle unpaired weatherproofing. For this middle unpaired weatherproofing, the bottom of the antenna was found by the PPHT but no region of specularity was identify by the MSER detector, therefore it was left unaltered. On the top right, ‘Diverse FN’ shows an instance where the post processing was not effective; the bottom of the antenna was found but the exposed hex nut is too dark for the MSER to detect any specular characteristics. In the middle of the figure, ‘In and Out FN’ presents an image where three bad weatherproofings exist. The two on the far right of the image were found by the detector, but the middle unpaired weatherproofing was not detected as a bad weatherproofing although the Hough line was found and there appears to be a specular region in between the top of the weatherproofing bounding box and Hough line. The reason this did not return as a bad weatherproofing is due to the middle point of the MSER being outside of the ROI. On the bottom row, the homogeneous results are presented with the first being ‘Homogeneous TP’s’. This same image was used in
figure 5.4, but is a good example of how the detector benefitted from the post processing, as it was able to find three weatherproofing cases but unable to find an instance of an exposed hex nut. In the middle image on the bottom row, ‘Homogeneous FN’ presents an instance where the weatherproofing case is found beneath an undetected exposed metal hex nut, but the post processing method was not effective. The Hough line shown in the image corresponds to the overlapping weatherproofing case, whose top $y$ value is slightly higher than the far left bounding box. Due to the limitation set on how high to look for a Hough line, the weatherproofing case did not make a match for this particular line representing the bottom of the antenna.

On the bottom right (‘Homogeneous FP’) and middle right (‘In and Out FP’) image, instances of a false positive is presented to show the importance of having a good weatherproofing case detector. In ‘In and Out FP’, the weatherproofing bounding box outputted on the far right is slightly below the top of the weatherproofing case itself. In this example, part of the case had a reflection which was detected by the MSER. On the other hand, the ‘Homogeneous FP’ example shows a taped cable connection being detected as a weatherproofing, and part of that tape also being reflected, thereby deeming it as a false positive of a bad weatherproofing.
7.2 Indoor Set

As previously mentioned in section 7.1, the bad weatherproofing detection results can outperform the hex detection alone. By simply pairing hex nuts with weatherproofing cases, the average precision on the homogeneous model increased 15.43% with the amount of true positives increased by 225. The hex nut detection performance compared to the baseline bad weatherproofing performance for both the Diverse and Indoor and Outdoor models decreased due to the weatherproofing detections not performing as well. In order for the detection to be considered a bad weatherproofing, there must be both a weatherproofing case and exposed hex nut detection. On this test set, there were instances where the hex nut was detected by the Diverse model but the weatherproofing case below was not found.

The Homogeneous model benefitted the most from the post processing applied after detection on the Indoor test set. The baseline resulted in 62.53% average precision, while the post processing increased the average precision to 76.95%. An additional 314 bad weatherproofings were identified by the post processing technique, while introducing 119 false positives. Both the Diverse and Indoor and Outdoor models performed well on the indoor test set. While the average precision only increased by 1.25% on the Diverse model, 26 instances of bad weatherproofings were found without introducing any false positives. Additionally, the Indoor and Outdoor model found 27 true positives missed by the detector, but also increased false positives by 22.

Figure 7.4: Indoor Test Set Results

Figure 7.4 contains a collection of images showing instances where the post processing does well, and where it did not work as expected. The top row has examples from the Diverse model. Starting with the top left and middle examples, ‘Diverse TP’, we can observe that the detector has some difficulties finding objects from a further distance.
This does not pose a problem for either the PPHT or MSER algorithms, but can interfere with how well we are able to define an accurate ROI to search for areas of specularity. On the top right hand side, ‘Diverse FNs’ shows an example where two exposed metal hex nuts are missed by both the detector and post processing. The hex nut on the right does show some specularity, but its middle point is not within the ROI defined by the Hough line and top of WP3M bounding box. The hex nut on the left side did not get detected due to the lack of Hough line found above the bounding box.

7.3 Outdoor Set

The outdoor test set yielded varying results for the models used. The hex detection on the Homogeneous model was 7.01% average precision with 47 true positives and 284 false positives. Here we see another instance of the Homogeneous detector predicting bounding boxes too large as the bad weatherproofing detection results are significantly better at 46.53% average precision, 176 true positives, and 87 false positives. The Diverse and Indoor and Outdoor also see an increase in their average precision for bad weatherproofings compared to the individual hex and WP3M detections, but not as drastic. Figure 7.5 displays examples outputted by the algorithm; situations where the post processing performs well and instances where it produced false positives. On the top left hand side we can observe the diverse models performance, the algorithm was sent two unpaired weatherproofings and was able to detect one of which that was a bad instance and not falsely display a red bounding box for the good weatherproofing. The middle image and right hand images show an instance where only one of the weatherproofing cases were detected and no hex nuts were detected in the image. In both, the bad weatherproofing was correctly detected, but the right image finds the Hough line in a region below the ideal bottom of the antenna. The diverse model was successfully able to improve average precision performance with post processing from 73.56% to 80.86% by increasing true positives found by 26, while introducing only one false positive.
The first image in the second row shows another instance where a good weatherproofing is able to find the bottom of the antenna through PPHT and does not find a region of specularity between. The second and third images show the post processing falsely labeling good weatherproofings as bad weatherproofings. In the middle image, the bounding box for the WP3M case is slightly below the top of the actual case. The algorithm then successfully finds the bottom of the antenna, but since the top of the weatherproofing case has a reflection from the sun it gets detected by MSER. Similarly, the third image in the second row outputted a bounding box for the WP3M case that is above the bottom of the antenna. This caused the Hough line to be found also to be above the bottom of the antenna, in this case it is part of the white stand to the right of the antenna. The metal screws above the antenna have specular characteristics that get detected by MSER as well, which can cause a false positive. The indoor and outdoor model went from 92.86% average precision to 94.60% with post processing, finding 17 new true positives, at the expense of 16 false positives, causing detections to become less precise.

In the third row one can observe an instance where the detector falsely classified a WP3M as a bad weatherproofing by pairing it with a falsely detected hex nut above in the first image. To the left of that false positive is a true positive bad weatherproofing found by the post processing. The middle image shows a false positive outputted by the post processing algorithm. This can be attributed to the horizontal line being found in a background object and a region with specular characteristics existing within that area. The rightmost image has a true positive found by the post processing and a false positive. The false positive has the same problem as the middle image in the middle row. The bounding box outputted by the detector as a wp3m ended below the top of the case. With the case having reflective properties, it is prone to being detected by MSER.
as well which can lead to false positives. The Homogeneous model on the outdoor test set was the first to suffer from post processing. This is due to the fact that a majority of detected WP3M cases were false positives, Hough lines were found in background regions, as well as regions containing specular properties in the background as well. The post processing decreased the average precision from 46.53% to 40.80%. While 28 additional true positives were able to be detected, 131 false positives were outputted.
8 Conclusion

8.1 Concluding Statements

In conclusion, the post processing method improved average precision in 2/3 tests on the Homogeneous model, 3/3 tests using the Diverse model, and 3/3 tests using the Indoor and Outdoor model. The success of the post processing method stems from having a detector that can reliably output precise bounding boxes for detected WP3M cases. For the majority of the tests, false positives were introduced when increasing the number of true positive detections. This not ideal, but still provides value to Ericsson, as they can visually inspect false positive images to ultimately decide if a human needs to scale the radio tower for maintenance. This project highlights the importance of using a good dataset to train a CNN and the potential for improving detection results with post processing.

8.2 Future Work

There are a multitude of improvements/changes that can be made to the overall process of finding exposed hex nuts missed by the detector. First, the weatherproofing cases tend to be around the same y axis, depending on far down a bad one has fallen. A desirable feature is to reliably find the bottom of the antenna as one long line, and using that line to use as the ceiling of the ROI for specularity. Another improvement is finding a way to look downward from a unpaired detected hex nut. There were a number of instances where the black weatherproofing case was not detected, but a true positive hex nut was. Currently, this would not be considered a bad weatherproofing by the algorithm. Furthermore, dynamically changing the parameters of MSER based on size of weatherproofings could help reduce false positive regions of specularity. Some changes that could made to testing that would be interesting include changing the training set bounding boxes to bad and good weatherproofings. For this project, two separate objects were labeled and paired later. Finally, based on the similar hex performance of the homogeneous and diverse dataset on the Ericsson dataset, it would be interesting to add the additional outdoor images to the homogeneous dataset to see how that affects performance.
Literature


