Faculty of
Mathematics
and Informatics

VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS
MASTER'S STUDY PROGRAM
MODELLING AND DATA ANALYSIS STUDY PROGRAM

Corrosion Detection on Steel Panels using Semantic Segmentation Models

Korozijos aptikimas ant plieninių plokščių naudojant semantinės segmentacijos modelius

Master's Thesis

Author: Mantas Micikevičius
VU email address: mantas.micikevicius@mif.stud.vu.lt
Supervisor: Assist. Dr. Tomas Plankis

Vilnius
2023

Abstract

The effectiveness of several semantic segmentation networks, including U-Net, FPN, PSPNet, and LinkNet, is examined in this study for the purpose of corrosion detection on various steel panels that have been affected by various chemicals that cause corrosion to emerge. Since these types of images lack publically accessible ground truth datasets, an image prepossessing algorithm was developed and is now employed both fully automatically and semi-manually. Both approaches are compared in order to determine the significance of image masking accuracy and human involvement throughout the process. The aim is to examine the performance of neural networks by comparing their performance metrics not only between the models used in this study, but also with those from earlier studies that used fully automated corrosion detection algorithms or various deep learning architectures. The findings indicate that manually creating ground truth datasets has a significant impact on model accuracy metrics, and that only models trained with these types of datasets may be employed successfully in real-world applications. Additionally, compared to various corrosion detection techniques used in other reviewed researches, all architectures tested in this thesis worked well and demonstrated superior results in majority of the used indicators.

Keywords: Corrosion detection, Semantic Segmentation, Convolutional Neural Networks, Image prepossessing algorithm, Images masking

Santrauka

Šiame tyrime yra nagrinėjamas keletos semantinio segmentavimo tinklų, įskaitant U-Net, FPN, PSPNet ir LinkNet efektyvumas. Šie modeliai yra naudojami siekiant aptikti koroziją ant įvairių plieno plokščių, kurios buvo paveiktos įvairių cheminių medžiagų, sukeliančių koroziją. Kadangi tokio tipo paveikslėlių duomenų su sužymėtomis kiekvieno pikselio klasėmis viešai prieinamų nėra, taigi buvo sukurtas pikselių klasifikavimo algoritmas, kuris šiame darbe yra naudojamas tiek visiškai automatizuotai, tiek pusiau rankiniu būdu. Abu metodai yra palyginti, siekiant nustatyti paveikslėlių pikselių klasifikavimo tikslumo ir žmogaus dalyvavimo visame procese reikšmę. Tikslas yra ištirti neuroninių tinklų veikimą, lyginant jų našumo rodiklius ne tik tarp šiame tyrime naudotų modelių, bet ir su ankstesnių tyrimų modeliais, kuriuose buvo naudojami pilnai automatizuoti korozijos aptikimo algoritmai arba įvairios gilaus mokymosi architektūros. Išvados rodo, kad pusiau rankinis pikselių klasifikavimas turi didelę įtaką modelio tikslumo rodikliams. Taip pat, tik su tokio tipo duomenimis parengti modeliai gali būti sėkmingai naudojami realiame pasaulyje. Be to, modelius palyginus su įvairiais korozijos aptikimo metodais, naudojamais kituose apžvelgtuose tyrimuose, visos šiame darbe išbandytos architektūros veikė gerai ir parodė geresnius rezultatus pagal daugumą naudojamų rodiklių.

Raktiniai žodžiai: Korozijos aptikimas, semantinė segmentacija, konvoliuciniai neuroniniai tinklai, paveikslėlių apdorojimo algoritmas, paveikslėlių maskavimas

Contents

Introduction

Numerous industries face the costly and persistent issue of corrosion. The annual cost of corrosion is commonly estimated to be between 3 and 4 percent of the gross domestic product [1]. Additionally, between 15% and 35% of this sum is believed to be preventable, with a sizeable chunk related to the expense of inspection [2]. Cost reductions and risk reduction are the driving forces behind research into automated corrosion detection.

The process of image processing has drawn a lot of attention as computer technology and vision studies have advanced [3]. An autonomous image processing algorithm, for instance, was developed by Bonnin Pascual and Ortiz [9]. Utilizing convolutional neural networks (CNN) is yet another widely used and successful strategy. The degree of automation in the CNN based method is considerable, and features do not need to be manually extracted. To further industrial automation, the deep learning approach has also been gradually applied to defect detection in recent years. In paper [4], it's claiming that fully convolutional network (FCN), U-Net, and Mask R-CNN architectures are the most well-liked and effective image segmentation models for corrosion detection. In a research which used all three models in semantic segmentation of rust detection on a private dataset [11], U-Net performs slightly better in terms of precision than the rest ones, while for F1-score all deep models yield almost the same performance. But by a slight margin the best performance was reported for the Mask R-CNN model, with an average F1-Score of 0.71. While investigating the time complexity between the same models, U-Net has the fastest approach as well. However, U-Net wasn't compared with models like Feature pyramid network (FPN), Pyramid Scene Parsing Network (PSPNet) and LinkNet in rust detection application, even when these models are showing great performance regarding accuracy and effectiveness metrics in other semantic segmentation tasks [23][24][26]. In this paper, a U-Net, FPN, PSPNet and LinkNet architectures will be used for pixel level corrosion detection on various steel panels.

Currently there are no publicly available ground truth (corrosion labelled pixels) image datasets of rusted steel structures. Therefore, in this paper, image processing algorithm will be used, consisting of 7 different functions by which results it's determined if a pixel represents rust or not. The process also requires a human intervention. Each image of steel panels will have different manually specified algorithms' thresholds to create most accurate ground truth images dataset. In addition, second ground truth dataset will be created with prespecified thresholds for all images to compare and analyse the importance of manual part in masks creation.

Both masks creation methods are compared visually and by calculating models performance, while trained on both datasets. In all of the experiments, deep learning networks trained on semi-manually generated masks were superior, taking into account all performance measures. Additionally, compared to alternative methods from previous reviewed studies [12][11], these networks produced greater accuracy metrics. All developed python scripts can be found in here: `https://github.com/mmicikevicius/rust_detection_2022`

The aim of the thesis

The thesis' objective: analysis of multiple semantic segmentation models for corrosion detection trained on semi-manually and automatically created ground truth datasets of various images of steel with rust areas.

The goals to achieve this purpose are:

1. Exploration of relevant articles in the scientific literature
2. To gather various images of steel panels with corrosion affected areas
3. To create a ground truth datasets from steel panels' images using image processing algorithm
4. To train semantic segmentation models for corrosion detection on steel objects and compare their performance

1 Related Work

Topics such as corrosion detection on steel items, object recognition and semantic segmentation, image processing for pixel-level classification and deep learning models in computer vision were used to choose scientific papers for the literature review. The most important factors were recent publications, journal popularity, and additional examination of the approaches presented.

1.1 Computer Vision

A type of input method using various imaging systems to replace the organ of vision is computer vision, which is utilized to replace the brain to process and explain. The ultimate goal of computer vision is to enable machines to perceive and interpret the world similarly to humans and to possess the autonomy necessary for environment adaptation. Utilizing computer vision technology, corrosion defect information of coating material will be converted into digital and quantitative information to enable analysis and application, as well as to increase the level of uniformity and accuracy [5].

A fundamental issue in computer vision is color representation. The effectiveness of many systems depends on a proper color representation. Despite the fact that the majority of color images are captured in red, green and blue (RGB) color space, computer vision applications rarely utilise this space. The fundamental cause is that RGB does not distinguish between color and intensity, resulting in channels that are highly linked [6]. Therefore, one of the most frequently chosen color spaces for image segmentation tasks is hue, saturation, value (HSV) [7]. In the study [8], HSV was tested for image segmentation tasks and compared with other color spaces. With HSV, used methods showed the best results and generally came to the conclusion that it's especially effective while dealing with segmentation of noisy color images.

1.1.1 Data Augmentation

On a variety of computer vision tasks, deep convolutional neural networks have exhibited astounding performance. To prevent over-fitting, these networks, significantly rely on big data. The validation error must drop along with the training error in order to create useful Deep Learning models. This can be done very well with data augmentation. Simple adjustments like horizontal flipping, color space augmentations, rotations, image translations, and random cropping are what make Data Augmentations effective. The distance between the training and validation sets, as well as any upcoming testing sets, will be minimized as a result of these changes, which will represent data with a wider range of potential data points [21].

1.2 Corrosion Detection

For structural steel members and components, corrosion is a common cause of failure. According to Zoran C. Petrović [10], when it comes to the frequency of failure mechanisms in engineering structures, corrosion, in all of its manifestations, leads with 42%.

1.2.1 Automated Image Processing

Visual inspection is the first step in preventing these problems or at the very least, in maintaining structures. When inspected visually, corroded areas appear to be between red and brown in color and have a rougher surface than non corroded ones. Therefore, the automatic image processing algorithm created by Bonnin Pascual and Ortiz [9] for corrosion detection quantifies these two visual features to locate the rust in a given image. Based on the same algorithm automated image processing process was developed in paper [12]. Since the classification technique was quite simple with predetermined thresholds for various pixel features, so the results wasn't perfect. This method doesn't take into account any different image conditions like lighting, background, etc., compered to other deep learning approaches. In average it's performance metrics like precision and recall reached 59.5% and 77.3% respectively.

1.2.2 Deep Learning Approach

Semantic segmentation is one of the deep learning model types. It describes the process of predicting the relevant class for each pixel of the image. Rust detection is well suited task for this type of a model. In paper [11], semantic segmentation approach was applied for corrosion detection using most popular semantic segmentation models like FCN, U-Net, and Mask R-CNN. This method not only uses the exact pixel values, but also can take into account the surroundings of it depending of models structure. Results of this research was that the models were able to achieve between 71% and 81% of precision. Specifically U-Net model performed slightly better in terms of precision than the rest ones, while for F1-score all deep models yield almost the same performance of about 70%. But by a slight margin the best F1-score with 72% was reported for the Mask R-CNN model.

2 Methodology

We will discuss the relevant procedures and definitions associated with our research in this section. The modeling technique involves pre-processing the data and semi-manually creating a ground truth dataset using an image processing algorithm which is combined with multiple deep learning networks. Therefore, we begin with information about corrosion detection algorithms before moving on to the fundamental ideas and properties of semantic segmentation models.

2.1 Corrosion Detection Algorithm

The first key step before moving to the algorithm is resizing huge, high-quality images into smaller ones since a smaller image that still contains essential details requires less computation time [12].

The algorithms' primary section then begins with a roughness analysis. The discovered rough area is moved to the second stage, or the color step, for additional research as a prospective corroded zone. The candidate areas' color is compared to a set of corrosion colors in the color stage. The final result of this method is a map that displays the sites of corrosion that has been found. The structure of the algorithm can be seen in Figure 1.

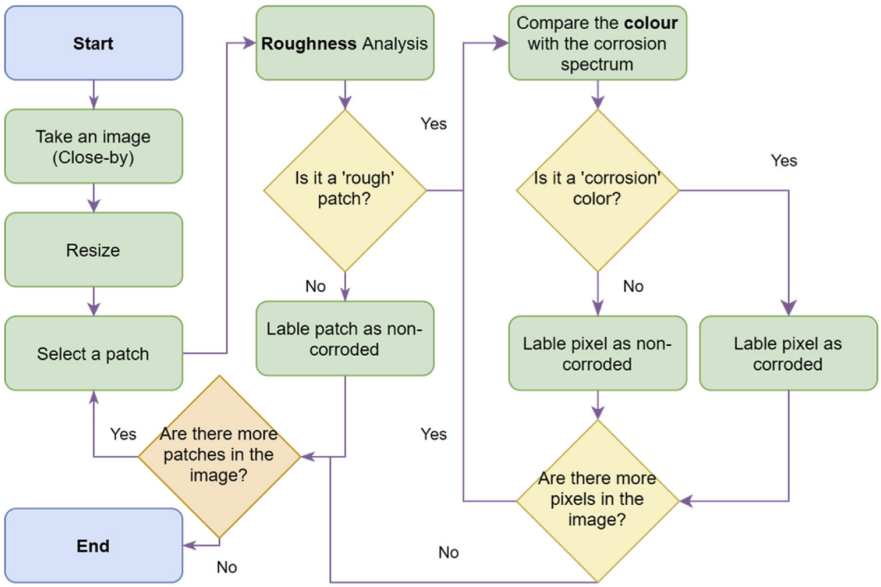

Figure 1: The corrosion detection algorithm [12]

2.1.1 Roughness Analysis

A corroded surface has a non-uniform distribution of corrosion colors, whereas a non-corroded surface has a fairly consistent color distribution. Measuring the uniformity of a patch, often known as a portion of an image, is one technique to quantify the color distribution of that area [13]. The value of uniformity ranges from 0 to 1. A value of 0 indicates that the patch has a non-uniform distribution of colors, which could suggest the existence of corrosion, while a value of 1 indicates that the inspected patch has a uniform color distribution that is interpreted as a non-corroded patch [12]. Equation (1) shows uniformity:

$$\text{uniformity} = \sum_{i,j} p(i,j)^2 \tag{1}$$

where p is the Gray Level Co-occurrence Matrix (GLCM) which is described below.

In computer vision related applications, the method of examining the spatial distribution of images using grayscale pixels is frequently utilized [14]. The white and black in a color image should be changed to white and black, respectively, and the remaining colors should be transformed to various shades of gray, in order to create a grayscale image for this purpose. The GLCM is built after a color image is converted to a grayscale image with fixed gray levels. Element $p(i,j)$ quantifies how often the gray level of i is in the neighborhood of the gray level of j. Two factors, namely direction and distance, should be taken into account while establishing the neighborhood. In the roughness step, each patch uniformity is determined and then compared to a threshold. The patch under investigation is regarded as corroded if the computed uniformity falls below the threshold [12].

2.1.2 Color Analysis

Atmospheric circumstances cause steel to corrode in shades of red, yellow, and red-brown. So one may create a classifier for corrosion detection by quantifying corrosion colors and contrasting them with a reference color.

The right color space must be chosen as the initial step in the quantification process. Based on [12], it appears that HSV color space is the best choice for representing colors associated with corrosion. The color spectrum in the RGB brings some complication in the sense that applying thresholds on it would include a lot more colors than the wanted rust color spectrum. Although the placement of the spectrum in the HSV color space does make it possible to apply thresholds for hue, saturation, and value more effectively.

Value (V) can be used to prevent the well-known instabilities in the computation of hue and saturation when a color is close to white or black. The pixel is then categorized as not having corrosion [9]. Regarding H and S, we can apply histogram method which makes use of a normalized histogram of H and S values of corrosion colors and then applies a two-dimensional Gaussian filter (2).

$$G(H, S, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\left(\frac{(H-\mu H)^2}{\sigma^2}\frac{(S-\mu S)^2}{\sigma^2}\right)} \tag{2}$$

Applying a threshold allows you to eliminate the low-probability H and S combinations that indicate the corrosion color because this histogram is normalized. σ here is a parameter that can be used to calculate the probability of each H-S combination.

2.2 Convolutional Neural Network (CNN)

Over the past couple of decades, Convolutional Neural Networks (CNN) have produced ground breaking findings in a range of pattern recognition related domains, including image processing. The reduction of Artificial Neural Networks (ANN) parameter count is CNNs' most advantageous feature. The most crucial presumption regarding problems that CNN solves is that there shouldn't be any spatially dependent features [15].

LeNet, AlexNet, VggNet, GoogleNet, ResNet, and DenseNet are examples of advancements in CNN based image classification. Semantic segmentation, also known as pixel-level classification, is one of the available objectives for CNNs. This task entails prediction of the relevant category for each pixel in a digital image and production of a pixelwise mask for each object in the image [17].

Recently, with the advancement of CNN architectures, variations of CNN models using RGB data are used for automated visual evaluation of metal structures for structural health monitoring. Since ground truth data are used throughout the learning process, these techniques have the main advantage of increasing detection and classification accuracy when compared to color distribution modeling. This results in CNN structures that are better at identifying defective regions [18].

As was already mentioned, CNN places a strong emphasis on the use of images in their input. This concentrates the architectures' setup to best meet the requirements for handling that particular type of data. CNNs are comprised of three types of layers. These are convolutional layers, pooling layers and fully connected layers.

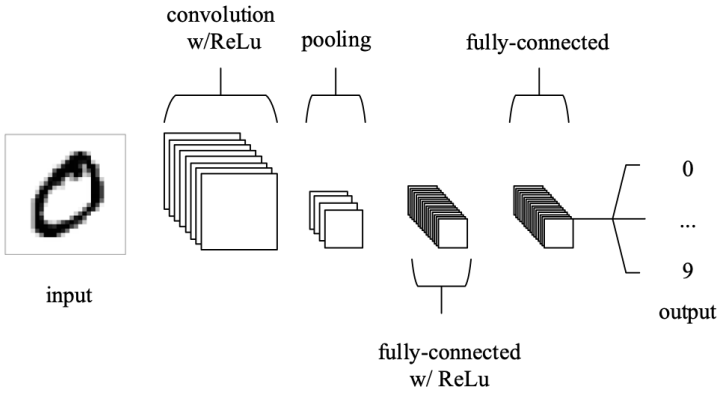

Figure 2: A straightforward CNN structure with only five layers [16]

There are four main areas where from the given Figure 2 of CNNs fundamental functionality can be divided [16].

1. The input layer will store the images' pixel values, as with other ANN variants.

2. The convolutional layer will calculate the scalar product between the weights of the input volume-connected region and the neurons whose output is related to particular regions of the input. The goal of the rectified linear unit (ReLu) function is to trigger activation function such as sigmoid to the output of the activation produced by the previous layer.

3. To further reduce the amount of parameters in that activation, the pooling layer will then simply downsample along the spatial dimensionality of the input.

4. The fully-connected layers will next carry out the identical tasks as in traditional ANNs and make an attempt to derive class scores from the activations, which can then be applied to classification. Additionally, it is proposed that ReLu be applied in between these layers to enhance performance.

2.2.1 Convolutional Layer

The convolutional layer is crucial to how CNNs work, as its name suggests. The layers parameters focus around the use of usually small in spatial dimensionality learnable kernels. Each filter is convolved across the spatial dimensions of the input by the convolutional layer as the data reaches it, creating a 2D activation map.

The scalar product is calculated for each value in that kernel as we move through the input. The model will learn from this how to create kernels that

activates when they spot a certain feature at a particular spatial position in the input.

Convolutional layer neurons are not entirely linked, in contrast to standard ANNs. Instead, it makes use of a small input region that is connected to each neuron. The receptive field size of the neuron is a common name for this region dimensionality. The depth of the input is almost always equal to the magnitude of the connectivity through the depth.

Through the optimization of their output, convolutional layers are also able to considerably lower the model's complexity. The three hyperparameters of depth, stride, and setting zero-padding are used to optimize these.

By manually adjusting the number of neurons in convolutional layers with respect to the same region of the input, the depth of the output volume will be set.

The stride is a measurement that positions the receptive field by determining the depth around the spatial dimensions of the input.

Zero-padding, which is the straightforward process of padding the inputs boundary, is an efficient way to provide further control over the dimensionality of the output volumes [16].

2.2.2 Pooling Layer

In order to decrease the number of parameters and the computational complexity for further layers, pooling layers peforms downsampling. It might be compared to lowering the resolution when it comes to image processing. Pooling has no impact on the quantity of filters. One of the most popular kinds of pooling techniques is max-pooling. It divides the image into rectangular subregions and only returns the highest value found inside each. 2x2 is one of the most typical max-pooling sizes. It should be noted that downsampling does not maintain the informations location. Therefore, it should only be used when the availability of information is crucial [15].

2.2.3 Fully-connected Layer

Neurons in the fully connected layer have direct connections to the neurons in the two adjacent layers, but they are not connected to any neurons within them. This is comparable to how neurons are placed in standard ANN models [16].

A fully connected layer main disadvantage is that it has many parameters that require expensive computation in training sets. As a result, we attempt to reduce the quantity of nodes and connections [15].

2.3 U-Net: Convolutional Network

Convolutional networks are frequently implemented for classification tasks in where the output to an image is a single class label. The desired output, or the assignment of a class label to each pixel, should incorporate localization in

many visual tasks, particularly in rust detection image processing. Thousands of training images are typically out of reach for rust detection jobs as well [19].

Another fundamental drawbacks of all typical deep learning techniques are that they adhere to a framework for local data processing. As a result, their performance suffers, particularly when the corroded and uncorroded parts exhibit identical color or texture characteristics at the local processing level. Recently, a U-Net model for rust defect identification that conducts global-local data processing has been used to address these issues [18].

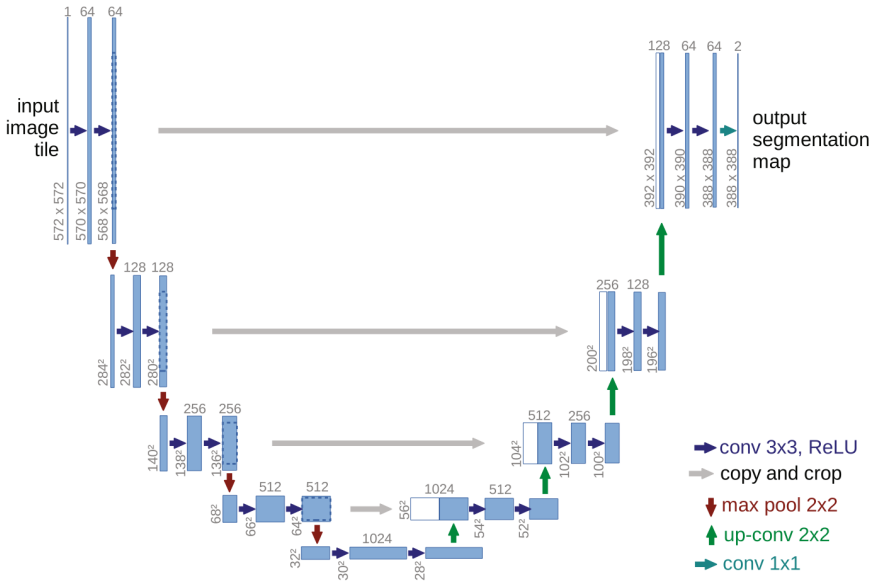

Figure 3: U-Net architecture [19]

Hence, in this paper [19] was build so-called U-Net Convolutional Network in a sliding-window setup to predict the class label of each pixel by providing a local region (patch) around that pixel as input. First, this network can localize. Secondly, the training data in terms of patches is much larger than the number of training images. Model is built as the fully convolutional network (FCN). This modified architecture of FCN works with small number of images and produces more accurate segmentations. The layers in Figure 3 enhance the output resolution. In order to localize, high resolution features from the contracting path are combined with the upsampled output. A successive convolution layer can then learn to assemble a more precise output based on this information.

Figure 3 shows the network architecture in detail. It consists of a contracting path on the left side and an expanding path on the right side. The contracting path adheres to the standard convolutional network architecture. Two 3x3 convolutions (unpadded convolutions) are applied repeatedly, and after each

one, a rectified linear unit (ReLU) function and a 2x2 max pooling operation with stride 2 are applied for downsampling. The number of feature channels are doubled with each downsampling step. Every step in the expanding path consists of an upsampling of the feature map followed by a 2x2 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU function. Due to the loss of border pixels in each convolution, cropping is required. Then each 64 component feature vector is mapped to the desired number of classes by a final 1x1 convolution layer. The U-Net architecture includes 23 convolutional layers in total [19].

In paper [20] U-Net Convolutional Network was used to detected corroded areas in images of electric poles. The model was also compared to Efficient Neural Network (E-Net) and regural FCN model. The models training was conducted with augmented data. After that, models were evaluated on test sets according to pixel calculation formulas such as pixel accuracy, intersection, union, IoU accuracy, and average IoU tests. Results showed that U-Net has higher values in three metrics of semantic segmentation accuracy on the rust and mean factors.

2.4 Feature Pyramid Network (FPN)

The Fully Convolutional Networks (FCN) are now known as U-Net and Feature Pyramid (FPN) neural networks after being significantly enhanced. FPN builds feature pyramids with a negligible additional cost using a pyramidal structure of deep convolutional networks. It includes top-down and bottom-up paths. Building high-level semantic feature maps at all scales requires the development of a top-down, lateral-connected pathway. When used as a generic feature extractor in a variety of applications, including object detection and instance object segmentation, this architecture outperforms FCN significantly [22].

These pyramids are scale-invariant in a way that a change in an object scale is compensated by a change in the object level inside the pyramid. This feature, by scanning the model over both positions and pyramid levels, enables a model to detect objects across a wide variety of scales. This technique produces proportionally scaled feature maps at several levels in a fully convolutional manner from a single-scale image of any size as the input. The underlying convolutional designs have no bearing on this procedure [23].

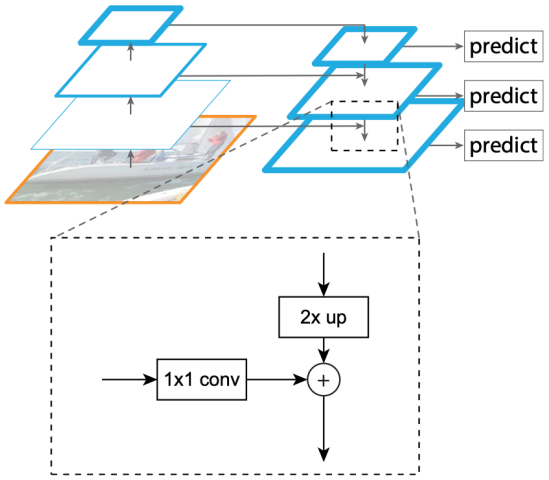

Figure 4: Feature Pyramid Network demonstrating the top-down pathway and lateral connections [23]

The feed-forward computation, which computes a feature hierarchy made up of feature maps at various scales with a scaling step of two, is the bottom-up method. For this model, each step has a single pyramid level. The final layer of each stage output is selected as a reference set of feature maps, which are then enhanced to form the pyramid. This decision makes sense because each stage deepest layer need to have the strongest features [23].

By upsampling geographically coarser but semantically stronger feature mappings from higher pyramid levels, the top-down pathway creates the illusion of greater resolution features. Through lateral connections, these features are then improved with features from the bottom-up pathway. Each lateral connection combines feature maps from the top-down and bottom-up pathways that are the same size [23].

2.5 LinkNet Architecture

Encoder-decoder pairs serve as the foundation of the network architecture for the majority of the semantic segmentation approaches now in use. Information is encoded into feature space by the encoder, and segmentation is performed out by the decoder by mapping this information into spatial categorization [24]. As a result, spatial information may be lost at the encoder and impossible to recover at the decoder. Additionally, despite the fact that semantic segmentation focuses on applications that need to operate in real-time, the majority of current deep networks have excessively long processing times. LinkNet directly propagates spatial information from the encoder to the decoder at a corresponding level to overcome the issues. As a result, processing times are significantly reduced due to the time and operations needed to relearn lost features [25].

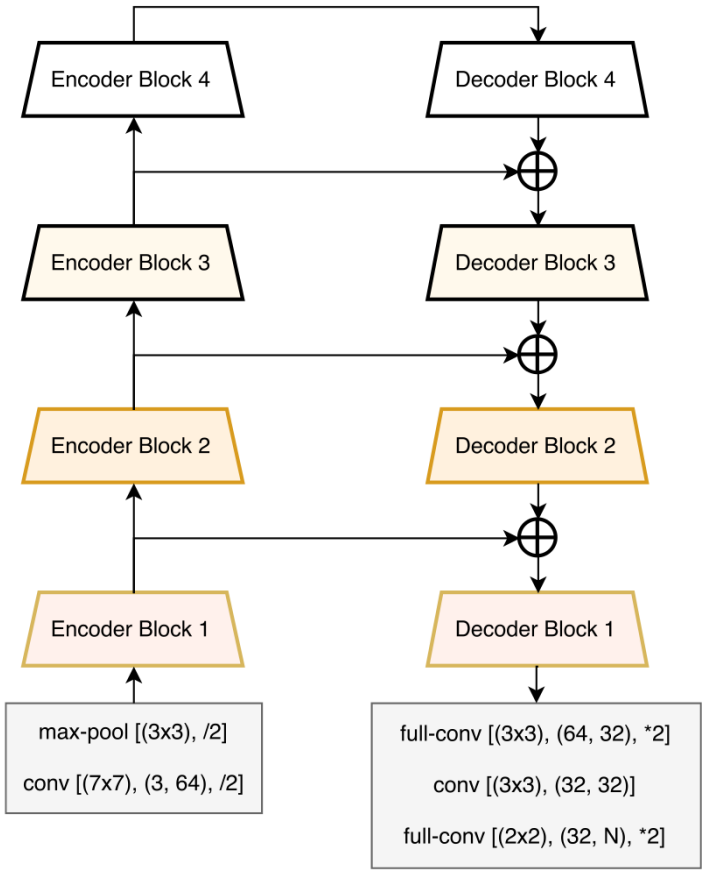

Figure 5: LinkNet Architecture [24]

Figure 5 presents the LinkNet architecture. Here the decoder is on the right side of the network and the encoder is on the left. Each convolutional layer is separated by batch normalization, which is followed by ReLU non-linearity. The initial block of the encoder performs convolution and spatial max-pooling on the input image. The decoder can utilize fewer parameters since it uses information that the encoder has learned at every layer. When compared to alternative segmentation networks, this leads to an overall more efficient network [24].

2.6 Pyramid Scene Parsing Network (PSPNet)

PSPNet is a neural network for semantic image segmentation that is fully convolutional. It carries over the pixel-level feature to the specifically created global pyramid pooling feature. The final prediction is more accurate when local and global clues are combined [26].

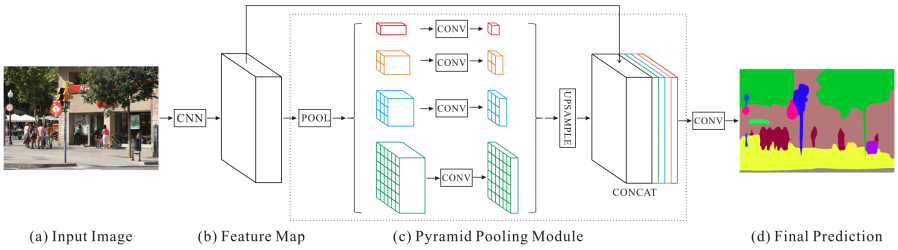

Figure 6: PSPNet Architecture [26]

In Figure 6 is shown PSPNet network architecture that combines convolutional layer and pyramidal pooling modules to create multiscale scenes. The gradual abstraction from the low-level features to the high-level features is realized by the convolutional layer. Then second, the multiscale pooling and convolution of the last layer of abstract features in the convolutional layer module were carried out by the pyramid pooling module. The multiscale pooled and convolved features were then upsampled in order to preserve the scale of the final convolutional layer, which was used to combine and convolve the two. Finally the attribution of each image element type was established, by repeating the deconvolution process until the scale matched the input image [27].

3 Practical Part

In this section data preparation, actual models training, experiments and model results will be described. As a result, we begin with information about data preparation which involves introduction of selected dataset, it's preprocessing for masks creation and models training.

3.1 Introduction of Corrosion Affected Panels Dataset

For this thesis unique dataset of images was selected from Yin, Biao and et al. research [28]. There where total of 600 images collected of material panels used for standardized corrosion tests for the use of discovering new materials. As shown in Figure 7, each image of material panels has one or two lines of corrosion affected areas with multiple different backgrounds. Material panels have been affected with different hazardous chemicals, so corroded areas are different in color and has different level of noisy areas around actual corrosion.

For corrosion tests, there were two experimental approaches applied. The first experiment is ASTM B117, a static salt-fog corrosion test that involves continuously atomizing 5% salt-fog (NaCl) into the test chamber that contains the panel at 35C. The second laboratory experiment, called cyclic corrosion, is a continuation of the static salt-fog experiment and is thought to be a more accurate representation of environmental conditions seen outside. Each trial cycle consists of a dwell period at ambient temperature, a high humidity event at the higher temperature, and a dry cycle event at the higher temperature. A solution of NaCl, NaHCO3, and CaCl2 (which is similar to seawater) is sprayed across the surface of the panels four times during the ambient phase of the test [28].

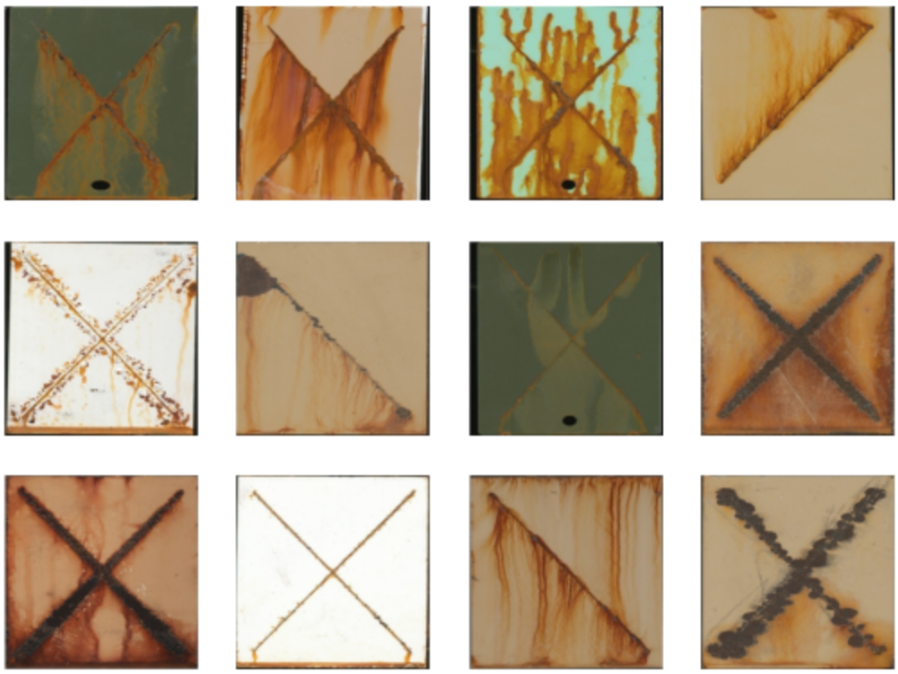

Figure 7: Sample images of material panels after corrosion tests

3.2 Data Selection

In this research, two image datasets were created. For the first one, all 600 images of steel panels are used. This dataset will be used with fully automated image processing algorithm for masks creation. For the second dataset only 150 images were selected for this thesis from whole dataset because a lot of images were removed due to lack of rust areas in them and repetitiveness, since there was a lot almost identical corroded areas with the same background. In addition, for the second dataset, masks creation for each image was done with a process which needs quite a lot manual input to create most accurate ground truth masks. Due to that it's heavily time consuming process and it's been decided that most diverse dataset of 150 images with most accurate ground truth masks is enough to train the models for corrosion detection.

3.3 Data Preprocessing

Each image from selected datasets has a resolution of 512x512 pixels. Images for model training and validation will be converted into digital format, having 3 layers: blue, green and red (BGR color space). Each layer represents model feature having number between 1 and 256. All three values represents a color of a pixel, so in total $256 * 256 * 256 \approx 16.7$ million possible colors.

A strong method to improve the quantity of data you have and avoid model overfitting is data augmentation. Since in this case our dataset is quite small, strong data augmentation part is necessary. A large number of different augmentations will be used:

- Horizontal flip with a probability of 50% that the transform will flip the image horizontally, and with a probability of 50% that the transform won't modify the input image
- Affine transformations like scaling, rotating or shifting the image
- Perspective transformation
- One of the brightness, contrast or colors manipulations are performed for each image
- Either image bluring or sharpening are applied for each image.
- Gaussian noise with a probability of 20%
- Random crops will take an input image, extract a random patch with size 256 by 256 pixels from it

All these transformations are performed using Albumentations - fast augmentation library in python. Every image from training set was augmented once, so the total number of images used in the training process did not change. The examples of images after the augmentations are applied as shown in Figure 8.

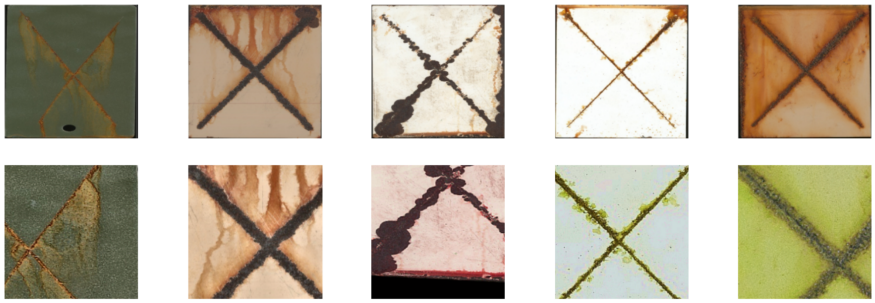

Figure 8: Sample images of training dataset before and after the augmentations

3.4 Ground Truth Dataset Creation

Semantic segmentation models are classifying each pixel of an image. Because of that each image used for training and validation has to have masks. Images' masks are the label data for models. Since at the current moment there are no publicly available ground truth image datasets of rusted steel structures, it is essential part of this work to create one. There are already quite some

research papers which tried to automate this process by creating algorithms with specified thresholds of color values and other features of an image to identify if a pixel should be labeled as corrosion or not. Similar strategy is also used in this study for the bigger image dataset. However, every image has a different corrosion occurrence reasons, different materials getting damaged of it and even different shooting conditions of an image like lighting, shadows, etc. Because of that corroded area pixels can have a very dissimilar feature values. Therefore in advance specified and applied to every image the same threshold values wouldn't give an accurate results.

Creation of a ground truth datasets in this paper is done by algorithm consisting of 7 different functions. Every function takes different features in consideration to determine if a pixel represents rust or not. Then by the majority of outputs algorithm determines actual label value. Before calculations all images are converted to hue, saturation, value (HSV) color space, since based on paper [12] research, this color space appears to be the most suitable one for defining corrosion-related colors. In addition Laplace smoothness value is also calculated to use as a feature of a pixel. It's used to recognise patterns of interests in an image and ignore noisy areas. It's calculated by using python package OpenCV functions GaussianBlur and Laplacian.

The Gaussian blurring technique basically scans every pixel in the image and recalculates its value based on the pixels around it. The kernel is the region that is scanned around each pixel. The number of pixels surrounding the center pixel is scanned by a larger kernel. Equation (2) shows calculations behind the function. In this case the input is an image in HSV color space, kernel in size of 7 and $\sigma = 0$. Since standard deviation is 0, it will be determined automatically using the kernel size:

$$\sigma = 0.3 * ((ksize - 1) * 0.5 - 1) + 0.8 \tag{3}$$

The results of Gaussian blur functions then are passed to Laplacian Operator in order to receive Laplacian derivatives. The detection of image edges makes extensive use of image derivatives. Image derivatives identify the regions of an image where the pixel intensity shifts noticeably. This aids in mapping any image edges. In most circumstances where the edges are detected, the estimated value of the second derivative of an image turns out to be zero. This is the idea underlying Laplacian derivatives. It's crucial to remember that zeros wouldn't only show up on the edges. Because of that Gaussian blur is applied before passing image to Laplacian Operator. It reduces zero occurrences in other meaningless locations of an image. The Laplacian operator is defined by:

$$\text{Laplace}(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{4}$$

where f is the input, x and y are the standard Cartesian coordinates. Equation 4 result is Laplace smoothness value which is used as one of the input for image preprocessing algorithm.

All the same functions were used to determine pixel label for complete dataset (600 images) and for the filtered one (150 images). Here are the descriptions of functions with pre-defined thresholds for complete dataset mask creation:

- Labeling function 1 marks pixel as rust if:

  $H \geq 175$ and $205 > S \geq 30$ and $135 \geq V > 30$

- Labeling function 2 marks pixel as rust if:

  $15 \geq H > 4$ and $205 > S \geq 30$ and $135 \geq V > 30$

- Labeling function 3 marks pixel as rust if:

  $Laplace\ Smoothness < 10$ and $15 \geq H > 4$ or $H \geq 175$ and $205 > S \geq 30$ and $135 \geq V > 30$

- Labeling function 4 clears pixels with high smoothness rate:

  $Laplace\ Smoothness > 4000$

- Labeling function 5 marks near black color pixels as not rust:

  $V < 30$

- Labeling function 6 marks near white color pixels as not rust:

  $V > 230$ and $S > 35$

- Labeling function 7 marks pixel as rust if:

  $256 > H > 230$ and $S > 230$

where H is hue, S is saturation and V is value of a pixel in HSV color space. Thresholds were chosen based on [12] and after the observation of corroded image areas pixel value patterns.

But as mentioned above, masks still wouldn't be highly accurate with these pre-determined values. Therefore, the thresholds were modified for each of the 150 images individually in the smaller dataset by visually analysing the results. It's a highly time consuming process, but it makes ground truth dataset much more accurate visually and moreover semantic segmentation models has a much better training material. Figure 9 confirms that, because even both masks approaches visually represents rust areas quite accurately, but still the first row of ground truth images labels the corrosion a bit denser and is not that much impacted by the leaks of corrosion.

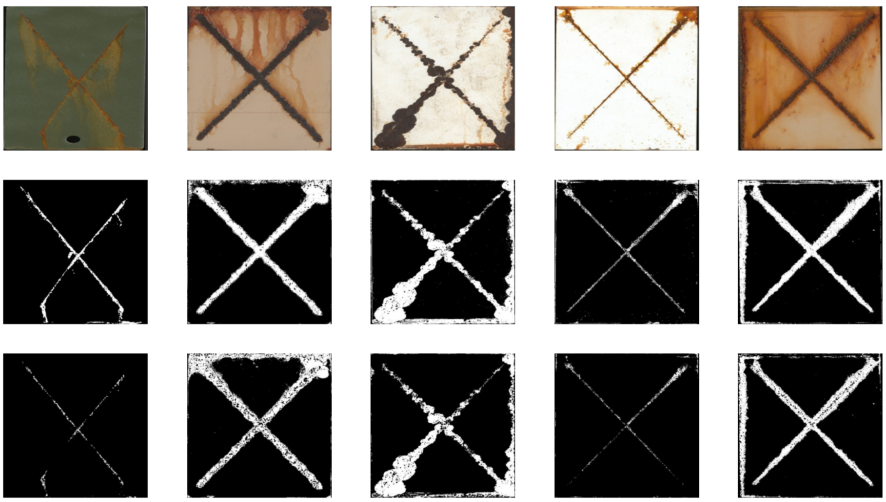

Figure 9: Example images and ground truths from the filtered dataset and complete dataset. Images (first row), with their corresponding ground truth label maps from filtered dataset (second row) and from complete dataset (third row), white = corrosion, black = background.

3.5 Semantic Segmentation Models

Our aim is to train a semantic segmentation models for corrosion detection of various images of steel with rust areas. Both images datasets were divided into training, validation and testing datasets. From the filtered set 100 images were used for training, 20 images for validating each epoch of a model and 30 images for testing model metrics. The same proportions were used for the complete dataset, 400 images used for training, 80 images for validation and 120 images for testing. In total 4 different models were trained for rust detection. For each model distinct backbones were chosen depending on other researches of experimenting with these models and the testing done in this study:

- FPN with ResNet18 backbone
- U-Net with VGG16 backbone
- LinkNet with Inception-v3 backbone
- PSPNet with EfficientNetB3 backbone

Two components make up the broad semantic segmentation network: an encoder and a decoder. Encoder is a pre-trained convolutional neural network, including ResNet, VGG-Net, MobileNet, etc. To produce the intensive classification, the decoder projects the distinguishable attributes into the pixel space

[29]. In this case each backbone model weights trained on 2012 ILSVRC ImageNet dataset. It helps to build faster and better convergence having models. Using the Python packages Keras, TensorFlow, and Segmentation Models, all deep learning networks were created.

Models with a smaller dataset were trained utilizing augmented data over the course of 60 total epochs for each deep learning network, with Adam optimizer as optimization setting with $1e-5$ learning rate. Same optimization settings were also used for the models trained with complete dataset, but only 20 epochs were specified, due to significant training set size change. Sigmoid function is used for every model as the activation function in the output layer:

$$S(x) = \frac{1}{1 + \exp^{(-x)}} \tag{5}$$

On the validation dataset, the loss was estimated using binary cross entropy after each epoch which is defined as:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i)) + (1 - y_i)\log(1 - p(y_i)) \tag{6}$$

where $p(y_i)$ is probability produced by the model that class is equal to y_i, while y_i is the actual pixel label and N is the output size.

Only weights with the lowest loss value were saved to be used later. All four model architectures are also described in Table 1. Based on paper [30], in most circumstances, neural networks perform better as the number of parameters increases. In Table 1 is shown that all models total parameters count differs significantly, especially LinkNet model which has more than 26 millions parameters. But that doesn't mean that in all cases this model will be superior.

Table 1: The properties of each model

Models	FPN	U-Net	LinkNet	PSPNet
Total params	13,815,370	23,752,273	26,268,401	1,985,343
Trainable params	13,805,124	23,748,241	26,227,953	1,973,679
Non-trainable params	10,246	4,032	40,448	11,664

As shown in computational time comparison Table 2, PSPNet and LinkNet networks computational costs are significantly lower compared to the rest ones. LinkNet model is the most computationally efficient with an average cost of only 0.11s $\pm$ 18ms. While U-Net and FPN networks shows to be heavier architectures with an average computational costs of 0.36s $\pm$ 21ms and 0.37s $\pm$ 25ms per sample respectively.

Table 2: Comparison of average computational time of semantic segmentation models on one sample and the full test dataset

Time (s)	FPN	U-Net	LinkNet	PSPNet
One image	0.37	0.36	0.11	0.16
All images	11.1	10.8	3.26	4.85

3.6 Experiments and Results

Every trained model were tested on the same test sets containing 30 images or 120 images and their labelled masks depending on the used dataset. To compare models performance this variety of evaluation indicators were selected:

- Accuracy, which is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TN + FN + TP + FP} \tag{7}$$

- Precision, which measures the number of corrosion class forecasts that really fall within the corrosion class and is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

- Recall, which quantifies the proportion of actual rust pixels that were accurately classified and is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

- F1-score, which measures weighted mean of recall and precision metrics. Hence, both false positives and false negatives are considered while calculating this indicator.

$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

- Intersection over Union (IoU), which is one of the the most often used evaluation measure for semantic segmentation and object detection applications. It calculates the ratio of the area of union between the predicted segmentation and the ground truth and the region where the prediction matches the ground truth pixels. IoU is defined as:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{11}$$

where A is the ground truth, B is predicted segmentation, TP defines that the predicted pixel is rust and the real label is also rust. TN denotes that the model predicted not rust and the actual label is also not rust. FP indicates that the predicted is corrosion, however the real label not corroded pixel. FN means that prediction was that pixel is rust free, but the real label is corrosion.

3.6.1 Models Performance

Table 3 compares the performance of the FPN, U-Net, LinkNet, and PSPNet deep learning models based on the aforementioned metrics. It details the evaluation of models trained and tested with filtered dataset and semi-manually created ground truth images. Very similar accuracy across models is typical for semantic segmentation tasks, but U-Net clearly outperforms the others with a 96.3% accuracy rate. Likewise, similar patterns can be observed with respect to F1-score, Recall, and most importantly, IoU. However, LinkNet has the biggest precision percentage. FPN architecture also shows quite good results, even if it is behind U-Net in almost all metrics except precision, but it is really near in every other metrics to it. The lowest performance in all metrics except recall has PSPNet model with only 56.1% IoU score, but this score is still considered as a tolerable performance.

Table 3: Comparative performance metrics for the different semantic segmentation models trained and tested with semi-manual created masks

Models	Accurracy	F1 score	Precision	Recall	IoU
FPN	96.2	75.9	78.8	73.2	61.1
U-Net	96.3	77.5	77.1	77.8	63.2
LinkNet	96	73.6	80.3	67.8	58.2
PSPNet	95	71.8	72.7	71	56.1

An illustration of a visual comparison of neural networks utilised for corrosion detection against ground truth and genuine images can be seen in Figure 10. The findings correspond to the percentages presented in Table 3, and aesthetically, FPN, LinkNet, and U-Net appear to be the most accurate of the three. Nevertheless, there are some circumstances in which the segmentations predicted by PSPNet can appear to be more accurate. This is especially the case with the second image from the top. Although it had the worst performance across almost all of the indicators, the network was able to accurately forecast the greatest amount of actual corrosion spots, whereas other models failed to capture a significant number of pixels that are indicative of rust.

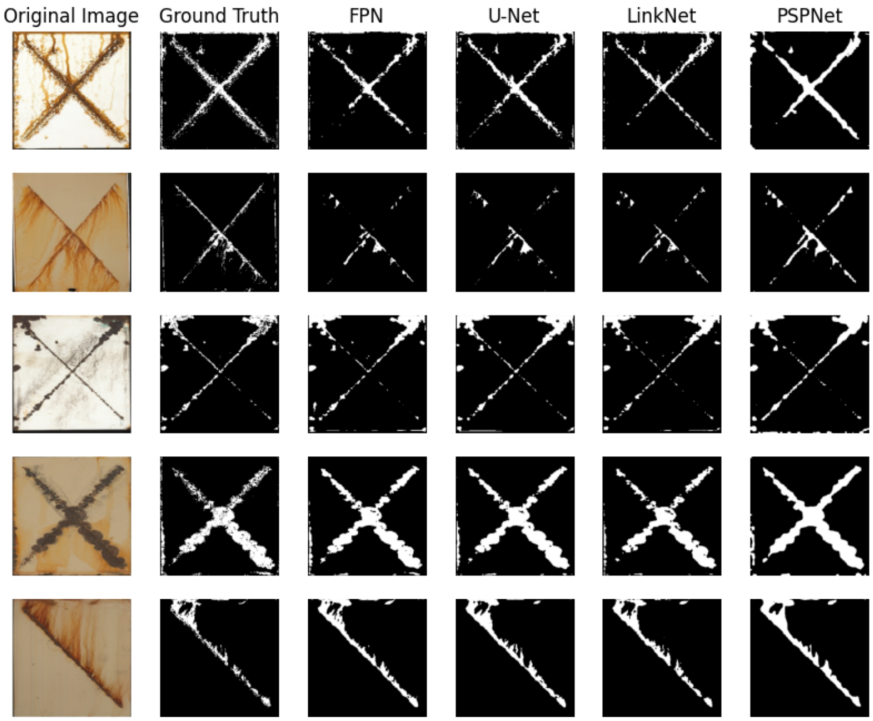

Figure 10: Qualitative comparison between FPN, U-Net, LinkNet and PSPNet showing segmentation results for corrosion detection datasets.

It also appears, according to the Figure referenced as 10 that all of the deep learning models have a tendency to combine all of the nearby existing pixels into the same class, which causes it to lose its precision. This is especially true for the PSPNet architecture, in which the area of projected corrosion in the images has a tendency to always be a little bit more spread out.

3.6.2 Ground Truths Creation Methods Comparison

Models trained and tested with fully automated mask creation process metrics are shown in Table 4. The accuracy scores for all models are fairly good, with LinkNet having the highest percentage of 98.2 percent. In comparison to other models, it possesses the greatest F1 score as well as the highest precision and IoU scores by a significant margin. Except for LinkNet, all of the models that were trained and tested with automated image processing algorithms had low IoU scores. IoU is widely regarded as the most important semantic segmentation accuracy indicator.

Table 4: Comparative performance metrics for the different semantic segmentation models trained and tested with complete dataset

Models	Accurracy	F1 score	Precision	Recall	IoU
FPN	97	66	62.7	69.6	49.2
U-Net	97.8	67.7	84.8	56.4	51.2
LinkNet	98.2	76.3	85	69.2	61.6
PSPNet	97.2	64.2	70.1	59.3	47.3

Due to the fact that semi-manual image processing algorithms produce significantly more accurate masks, a set of 30 ground truths images that were made semi-manually was used in the comparison of the significance of masks creation method. Both models that were trained with the complete dataset and ones that were trained with a filtered dataset were evaluated using the same 30 labelled visuals. Table 5 displays the findings obtained from deep learning architectures trained with 400 autonomously generated masks. It appears from the accuracy percentages that the models have fairly similar performance, and it is not dependent on the procedure of mask development. This was discovered while comparing these findings with the scores from Table 3.

However, when other factors are considered it becomes abundantly clear that a totally automated masks creation process is not well suited for the training of accurate models. In particular, it is observed from recall and IoU scores, which indicate that the models are not able to reliably categorise rust pixels. These measurements and the outcomes they produce for models that have been trained using the entire dataset are regarded as having low performance. However, the average precision percentages are significantly greater, which demonstrates that these models were able to categorise the background with a better degree of accuracy, particularly the U-Net model with 86.7% accuracy when compared to models trained with a smaller dataset. Even though it lags behind in almost all performance metrics, the FPN model trained with automatically formed masks is the only one that can compete with models trained with semi-manually created masks. This is due to the fact that the margins of error in the FPN model are not nearly as large as compared with other used architectures.

Table 5: Comparative performance metrics for the different semantic segmentation models trained with complete dataset and tested with semi-manual created masks

Models	Accurracy	F1 score	Precision	Recall	IoU
FPN	95.6	70.4	77.6	64.3	54.3
U-Net	95.2	62.2	86.7	48.5	45.1
LinkNet	95.4	65.9	82.3	54.9	49.1
PSPNet	94.5	62.6	78.7	52	45.6

3.6.3 Performance Comparison Against Alternative Methods

When compared to a single completely automated technique proposed in [12], the performance of the proposed semi-automatic image processing algorithm and training semantic segmentation models for corrosion detection appears to be much superior. Performance metrics of this method can be found in section 1.2.1. Even the PSPNet model which had the worst performance in almost all metrics from Table 3 , still had a notably higher precision score. Moreover, U-Net network was able to get better recall percentage compared to automated algorithm.

Models used in this study also performs well while compared to other deep learning approaches for corrosion detection. PSPNet, FPN, LinkNet and U-Net architectures during the testing showed better precision output compared to FCN and Mask R-CNN models in paper [11]. Performance metrics of these networks can be found in section 1.2.2. However, only U-Net network trained in this thesis was able to reach similar result compared to the observed U-Net architecture within the same performance metric. When compared F1-scores, bigger differences are visible. All deep learning networks from this study got higher scores compared to the models from [11].

4 Conclusions

In order to accurately create ground truth datasets for corrosion detection, this research proposed a semi-manual image processing approach. In addition semantic segmentation models like U-Net, FPN, LinkNet and PSPNet were presented for automatic corrosion recognition and comparison of these models effectiveness were studied. All deep learning methods were trained using 100 images pre-processed with several random data augmentation approaches, validated with additional 20 images and tested with a new set of 30 images. In addition, ground truth creation method was compared to fully automated masks creation algorithm in which all 600 images were pre-processed and resulted in creating training, validation and testing sets with 400, 80, and 120 images respectively. Networks performance also was compared with other researches which used fully automated image processing algorithm and different deep learning networks. The following conclusions were obtained:

1. Image processing algorithm was created containing 7 separate functions and by majority of it label for each image is determined. Since human intervention was included into the process, the masks are significantly more accurate.

2. Fully automated image processing algorithm for masks creation showed inferior results compared with semi-manual approach. That can be stated after the both, visual and performance analysis. All networks trained with automatically labelled images performed lesser within almost all indicators. Only U-Net and LinkNet networks trained in that way were able to get better precision scores compared to models trained with more time consuming method. It confirms the idea that human intervention still has a huge positive effect for creating usable models in real life scenarios.

3. In the comparison of models used in this study, all networks were generally accurate, but U-Net has the highest accuracy. The same is with F1-score, Recall and most importantly IoU. U-Net is designed to learn from a fewer training samples and since the main training set used in this study has only 150 images, this feature of a model could have been crucial in the results. Best precision score had LinkNet model while U-Net and FPN were still very near.

4. Most of the networks used in this study performed well regarding the performance metrics compared to other researches [12][11]. Based on it, deep learning techniques are more well suited for these type of tasks than image processing algorithms. And not only U-Net, but also LinkNet and FPN models had a better results in most of the comparisons. In previous researches these models weren't mentioned for corrosion detection.

5. Performance and visual evaluation demonstrates that the used techniques are promising instruments for automated rust detection. However, neither

the performance metrics, neither the visual analysis shows a state of art results.

4.1 Remarks

Even if the used methods wouldn't deliver a performance of 100% on neither of the chosen metrics, but when considering the wider picture, such as corrosion inspection of vast industrial assets, the goal is to quickly screen structures and highlight deteriorated areas rather than accurately classifying every pixel.

References

[1] Hou, Baorong, et al. "The cost of corrosion in China." npj Materials Degradation 1.1 (2017): 1-10.

[2] Koch, G. et al. International Measures of Prevention, Application, and Economics of Corrosion Technologies Study. NACE Int. 1–3 (2016).

[3] Cheng, Zhiyu, Jun Liu, and Jinfeng Zhang. "An Improved Mobilenetv2 for Rust Detection of Angle Steel Tower Bolts Based on Small Sample Transfer Learning." International Conference on Intelligent Computing. Springer, Cham, 2022.

[4] Nash, Will, Liang Zheng, and Nick Birbilis. "Deep learning corrosion detection with confidence." npj Materials Degradation 6.1 (2022): 1-13.

[5] Ji, Gang, Yehua Zhu, and Yongzhi Zhang. "The corroded defect rating system of coating material based on computer vision." Transactions on edutainment VIII. Springer, Berlin, Heidelberg, 2012. 210-220.

[6] Omer, Ido, and Michael Werman. "Color lines: Image specific color representation." Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.. Vol. 2. IEEE, 2004.

[7] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," , Proceedings of IEEE International Conference on Image Processing, Sep. 2002, pp. 589-592.

[8] Bora, Dibya Jyoti, Anil Kumar Gupta, and Fayaz Ahmad Khan. "Comparing the performance of L* A* B* and HSV color spaces with respect to color image segmentation." arXiv preprint arXiv:1506.01472 (2015).

[9] Bonnin-Pascual, Francisco, and Alberto Ortiz. "Chapter Corrosion Detection for Automated Visual Inspection." (2014).

[10] Petrović, Zoran C. "Catastrophes caused by corrosion." Vojnotehnički glasnik 64.4 (2016): 1048-1064.

[11] Katsamenis, Iason, et al. "Pixel-level corrosion detection on metal constructions by fusion of deep learning semantic and contour segmentation." International Symposium on Visual Computing. Springer, Cham, 2020.

[12] Khayatazad, Mojtaba, Laura De Pue, and Wim De Waele. "Detection of corrosion on steel structures using automated image processing." Developments in the Built Environment 3 (2020): 100022.

[13] Baraldi, Andrea, and Flavio Panniggiani. "An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters." IEEE transactions on geoscience and remote sensing 33.2 (1995): 293-304.

[14] Feliciano, Flávio Felix, Fabiana Rodrigues Leta, and Fernando Benedicto Mainier. "Texture digital analysis for corrosion monitoring." Corrosion Science 93 (2015): 138-147.

[15] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." 2017 international conference on engineering and technology (ICET). Ieee, 2017.

[16] O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." arXiv preprint arXiv:1511.08458 (2015).

[17] Weng, Weihao, and Xin Zhu. "INet: convolutional networks for biomedical image segmentation." IEEE Access 9 (2021): 16591-16603.

[18] Katsamenis, Iason, et al. "Simultaneous Precise Localization and Classification of metal rust defects for robotic-driven maintenance and prefabrication using residual attention U-Net." Automation in Construction 137 (2022): 104182.

[19] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[20] Duy, Le Dinh, et al. "Deep learning in semantic segmentation of rust in images." Proceedings of the 2020 9th International Conference on Software and Computer Applications. 2020.

[21] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of big data 6.1 (2019): 1-48.

[22] Seferbekov, Selim, et al. "Feature pyramid network for multi-class land segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.

[23] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[24] Chaurasia, Abhishek, and Eugenio Culurciello. "Linknet: Exploiting encoder representations for efficient semantic segmentation." 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE, 2017.

[25] Zhu, Qingtian, et al. "Efficient multi-class semantic segmentation of high resolution aerial imagery with dilated LinkNet." IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2019.

[26] Zhao, Hengshuang, et al. "Pyramid scene parsing network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[27] Pan, Qian, et al. "A deep-learning-based approach for wheat yellow rust disease recognition from unmanned aerial vehicle images." Sensors 21.19 (2021): 6540.

[28] Yin, Biao, et al. "Corrosion Image Data Set for Automating Scientific Assessment of Materials." (2021).

[29] Zhang, Rongyu, et al. "Comparison of backbones for semantic segmentation network." Journal of Physics: Conference Series. Vol. 1544. No. 1. IOP Publishing, 2020.

[30] Golubeva, Anna, Behnam Neyshabur, and Guy Gur-Ari. "Are wider nets better given the same number of parameters?." arXiv preprint arXiv:2010.14495 (2020).