Transfer Learning Based Industrial Steel Plates Fault Diagnosis using Industrial Fault Signals

by

Mubasshira Mizan
17301074
Laila Sumiya Khan Nilo
17301022
Mosrika Momin Tuli
17301037

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
September 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mubasshira

Mubasshira Mizan
17301074

Laila Sumiya Khan Nilo
17301022

Mosrika Momin Tuli

Mosrika Momin Tuli
17301037

Approval

The thesis/project titled “Transfer Learning Based Industrial Steel Plates Fault Diagnosis using Industrial Fault Signals” submitted by

1. Mubasshira Mizan (17301074)

2. Laila Sumiya Khan Nilo (17301022)

3. Mosrika Momin Tuli (17301037)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 26, 2021.

Examining Committee:

Supervisor:
(Member)

____

Dr. Jia Uddin
Associate Professor
Department of Computer Science and Engineering
BRAC University (On Leave)
Assistant Professor(Research Track)
Technology Studies Department
Woosong University, Daejeon, South Korea

Co Supervisor:
(Member)

____

Nabuat Zaman Nahim
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

____

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

____

Sadia Hamid Kazi, Ph.D
Associate Professor
Department of Computer Science and Engineering
BRAC University

Ethics Statement

For the demonstration of our proposed model, we have researched a variety of articles, took help from websites, journals and publications. Our data has been taken from Kaggle.

Abstract

Transfer learning (TL) has shown its great advantage to solve small-training-sample issues utilizing information learned from existing large data with deep learning techniques. Transfer learning has been effectively applied in many deep learning networks where sufficient training samples are not accessible; it still experiences essential problems for image processing. Image processing technology has become an interesting field in medics as image processing plays avital role in the discovery of the diseases in the early stages, which facilitates the treatment of these diseases. Image processing divides into numerous scopes. For case, image classification, image segmentation, image enhancement and image assessment. In this thesis, we will review the existing industrial fault diagnosis models and will propose an image-based deep learning model to detect or predict industrial faults. In order to do that we will convert 1D sensor's fault signals to 2D images. After that, we will extract deep features using a deep learning model for training and testing the classifier. To validate our model, we will use an industrial fault dataset. As programming tools, we will use Python and MATLAB.

Keywords: Transfer Learning; Image Processing, Image Classification; Image Segmentation; Image Enhancement; Image Assessment; Deep Learning.

Acknowledgement

First and foremost, we are grateful to God for allowing us to complete our thesis without experiencing any major obstacles. Then we wanted to thank our supervisor, Dr. Jia Uddin, for bearing with our mistakes and providing frequent input to assist us improves our research. We would like to thank all of the helpful faculty members, as well as our co-supervisor, Nabuat Zaman Nahim, for their assistance. We would also like to thank our parents and teammates for their constant support over the entire semester. This journey was not successful without them. Thank you.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ANN	Artificial Neural Network
CAM	Class Activation Map
CNN	Convolutional Neural Network
GA	Genetic Algorithm
HSI	Hyper Spectral Images
KNN	K-nearest Neighbors
LR	Logistic Regression
ML	Machine Learning
MMD	Maximum Mean Discrepancy
TL	Transfer Learning
TPU	Tensor Processing Unit

Chapter 1

Introduction

1.1 Overview

For condition-based maintenance, knowing the cause of the out-of-control status, or determining which fault happened, is critical. Deep convolution transfer learning models, which display exceptional performance, are introduced as a hierarchical representation from raw data instead of using hand-built features. Machine learning algorithms such as support vector machines, knearest neighbor, random forest, and neural networks are used to learn the relationship between the health state and retrieved characteristics. Signal processing or statistical projection are used to obtain the retrieved characteristics, which needs a lot of human effort. Deep learning models with significant scalability and generalization ability, such as convolutional neural networks, auto-encoders, limited Boltzmann machines, and recurrent neural networks, have recently been used to handle raw data without the use of hand-made features. The training and testing datasets have the same feature distribution, and there is enough labeled data with defect information to make current deep learning successful. The trained model created on the source domain, however, cannot be simply applied to the target domain because the machine operates in a non-stationary working situation. Furthermore, there is insufficient labeled or even unlabeled data in the target domain. It takes time and money to collect the labeled data and create the model from scratch. Convolutional neural networks are generalized to transfer learning scenarios in transfer learning.Two layers with regard to task-specific features are adapted in a layer-wise way to regularize the parameters of CNN. When contrasted to a single kernel, the domain loss is determined by a linear combination of many Gaussian kernels, which improves the ability to adapt. Distribution discrepancy is reduced and the transferable features are learned through these two meant. Model weights from pre-trained models designed for typical computer vision benchmark datasets, such as the ImageNet image recognition tasks, can be reused.Transfer learning is versatile, allowing the use of pre-trained models directly and entails using models trained on one problem as a starting point on a related topic for feature extraction pre- processing and integration into totally new models. We will convert 1D sensor fault signals to 2D images and use an image-based deep learning model to detect or identify industrial faults.Given that they were trained on over 1,000,000 images for 1,000 categories, the models have learnt how to recognize generic features from photographs and have attained state-of-the-art performance and remain successful on the specific image recognition job for which they were designed. The model weights are available as free downloaded files, and many libraries offer APIs that make it simple to download and use the models. . The primary idea behind DTL is to construct a feature representation with a modest difference between multiple domains while maintaining good prediction performance for the source domain. Fault diagnosis is crucial and beneficial in a variety of situations, including academia and industry. Our goal of this work is to identify a better way to apply transfer learning methods based on image processing.

1.2 Problem Statement

In recent years, Industrial fault diagnosis have been improving a lot. Fault diagnosis is a method by which fault is being detected, isolated and measures the failure condition. Previously, machine learning was widely used in terms of detecting industrial faults. Machine learning is a method of data analysis as it is a branch of artificial intelligence that can automate analytical model building. Moreover, by this algorithm, systems can learn from the data, and then identify the pattern of the data and then make an effective decision with the minimum help of humans. Furthermore, it is a theory by which computer can learn from the data by observations or instruction, direct experience and make the better decisions in the new problems that we provide. In machine learning, there are three layers by which data is processed.Firstly,take some data which is input. Then train the data as per we need and then the desired output, we gain by the use of the trained data to make predictions on new data. However, machine learning does not need big amount of data for training. As a result, the result of the data become biased and the quality is not so good. Moreover, it needs enough time to understand the algorithm and also needs massive resources to give the accurate and relevant result. This results in the requirement of additional computational power for the programmer. Another main problem of this algorithm is that the programmer needs to choose the algorithm for his purpose. If he chooses the wrong algorithm, it will not generate the accurate result.

To overcome these problems of machine learning, deep learning is being used. It is a broader field of machine learning. Deep learning is being modeled after the neurons of the human brain. This means, a deep learning system can learn information over time. It also divides the problems in parts and then apply the algorithm for performing. There are three layers in deep neural learning: the input layer, the hidden layer and the output layer. Moreover, the accuracy and relevancy of the data is high and it follows the pattern of the data to perform the prediction as the pre-specified pattern. Deep algorithm needs GPU to train data properly. In machine learning, there are limited tuning capabilities of data. Deep learning can be tuned in various ways. Moreover, it is used to predict the output as well as perform the classification of the given data for the training. On the other hand, it has some disadvantages also. Deep learning needs big chunks of data to train and it takes longer time to train and get accurate and unbiased results. So it is not suitable for a small chunk of data. In addition, deep learning depends on high-end machines while the traditional learning depends on low-end machines for training and testing data. It uses feature extractors to reduce the complexity of data and makes patterns more visible. Though it is a complex process, it takes more time than the traditional one.

1.3 Research Objective

We are aiming to find more efficient uses of Transfer Learning to do Industrial Fault Diagnosis using Industrial Fault Signals. We have multiple options to choose from when it comes to Machine Learning. Transfer learning has several benefits, but the main advantages are saving training time, better performance of neural networks (in most cases), and not needing a lot of data. Our proposed system categorizes the industrial fault signals by pattern matching, neural nets and algorithms. By the end of the research, we hope to achieve a more effective system for finding industrial fault signals. This research will also enable us to understand the matter at hand with minute detail and further assist in solving similar problems or updating the system in the future.

Chapter 2

Literature Review

The main challenges of wind turbine fault detection lie in its nonlinearity and unknown disturbances (such as the wind). Expensive condition monitoring systems are often used to monitor the wind turbine, and typically also use additional, often expensive sensor signals. Most wind farms, on the other hand, have supervisory control and data acquisition (SCADA) systems in place for system control and data logging. However, the collected data are not used effectively. This paper proposes a method for early fault identification for the main components of wind turbines based on the existing data collected by commercial SCADA systems, and therefore no new installation of specific sensors or diagnostic equipment is required [1].

The proposed method uses Support Vector Machines, and it performs a non-linear classification by projecting the data points into higher dimensional hyper planes through Kernel Function. The simplest but most successful type of classifier is the linear classifier. Linear classifiers can only deal with the linear separable sample data. If the sample data is nonlinear, the solution of the linear classifier will be an infinite loop. In order to use it, we must make the non-separable data inseparable [2].

In FeiyaLv's paper, an improved algorithm, a weighted time series fault diagnosis based on an SSAE, is proposed to learn the high-order correlations of faults. Due to its strong learning ability, the weighted time series fault diagnosis can not only make use of the correlations among the monitoring variables but also use the time correlations among samples. The proposed method is particularly effective for detecting incipient errors due to the SSAE network's great expressive capacities. From a mathematical standpoint, another contribution of the study is the mathematical framework of the SSAE-based fault diagnostics, which is capable of multiple nonlinear mapping and complicated function approximation. The experimental results on the TEP benchmark showed that the proposed method outperforms other state-of-the-art approaches regarding its efficiency not only for distinguishing between faults and normal processes but also for accurately classifying faults. In addition, the experiments also revealed the importance of the time correlations and classifier performance. Better fault diagnosis performance is expected by adding the structures of the network and by combining it with some corresponding constraints for complex systems [3].

The experimental results of Magda Ruiz's paper show that the statistical features based on GLCM of wavelet time– frequency images can effectively identify the fault patterns of bearings and the proposed fault diagnosis method has satisfactory classification accuracy. Other classification methods based on sparse representation and dictionary learning, such as SRC, K-SVD without classifier, and D-K SVD, are also tested for comparison, and the results show that label consistent K-SVD has higher classification accuracy than the other methods when the same parameters are used. Possible future work includes combining the classification methods with other time–frequency image acquisition methods and image feature extraction methods in order to better extract the fault characteristics. Moreover, fault diagnosis of rotating machines including gear and rotor can be implemented to verify the effectiveness of the proposed method [4].

Transfer learning (TL) has shown its great advantage to solve small-training-sample issues utilizing information learned from existing large data with deep learning techniques. Transfer learning has been effectively applied in many deep learning networks where sufficient training samples are not accessible, it still experiences essential problems for hyperspectral image processing. the mismatch of hyperspectral sensors comes about in parts of trouble for transfer learning to be utilized in hyperspectral image (HSI) processing. Sensor-specific based transfer learning is proposed for hyperspectral images procured from same sensors, in which information learn from hyperspectral image, the network structure and parameters of a deep neural network are limited to transfer to images of the same sensor only. The validity of sensor-specific transfer learning is assessed using three deep learning-based tasks, including feature learning, super-resolution, and image de-noising; it is impossible to apply TL to images acquired from different hyperspectral sensors. In order to achieve sensor- specific spatial-spectral include transfer learning, the CNN for hyperspectral feature learning is trained to learn the separability of a sensor using a specific data set under the classification task.Consequently, the CNN for hyperspectral feature learning can be directly applied for other datasets acquired by the same sensor. the proposed CNN is first trained on a source dataset and then directly applied to a target dataset obtained from the same sensor without varying the parameters involved in the network. The parameters in the network trained under unsupervised sensor-specific mode are fine-tuned using a small number of samples from the target dataset obtained by the same sensor. Through such fine-tuning, the performance of unsupervised sensor-specific feature learning can be further improved. For hyperspectral image processing we can use sensor-specific based transfer learning [5].

Medical image segmentation methods generally perform well when provided with a training set that is representative of the test images to segment. Voxel wise brain tissue, white matter lesion (WML), and hippocampus segmentation for these three-application image weighting greatly improved performance overweighting all training images equally. Numerous medical image segmentation methods are based on supervised classification of voxels. Problems may grow when training and test data follow different distributions, like due to differences in scanners, scanning protocols, or patient groups. Under such conditions, weighting training images according to distribution similarity has been shown to significantly improve performance. As a way to reduce differences between training and test data and explore the added value of kernel learning for image weighting, we investigate kernel learning. propose a new image weighting method that minimizes maximum mean discrepancy (MMD) between training and test data, which enables the joint optimization of image weights and kernel, it is the new proposed image weighting method. When used separately, experiments on brain tissue, white matter lesion, and hippocampus segmentation show that both kernel learning and image weighting, greatly improve performance on heterogeneous data. either individually or jointly Combining image weighting and kernel learning, optimized can give a small additional improvement in performance [6].

The classification experiments for distinguishing CAKUT patients from normal controls based on their kidney US images have demonstrated that integrating the transfer learning features and conventional image features could improve the classification of US kidney images. For children it is a really challenging task that classification of ultrasound (US) kidney images for diagnosis of congenital abnormalities of the kidney and urinary tract (CAKUT). Improving existing pattern classification models that are built upon conventional image features is desirable. Transfer learning-based method to extract imaging features from US kidney images to raise the CAKUT diagnosis in children. Especially, a pre-trained deep learning model is obtained for transfer learning-based feature extraction from 3-channel feature maps computed from US images, including original images, gradient features, and distanced transform features. The transfer learning features, conventional imaging features, and their combination these are the support vector machine classifiers are then built upon different sets of features [7]. This research demonstrated how thermography may be used to perform intelligent fault detection and condition monitoring on a cooling radiator. Though fault diagnosis to detect problem types exactly is highly complicated and challenging, combining image processing, genetic algorithm (GA), and artificial neural network (ANN) approaches improves diagnostic efficiency and accuracy.A revolutionary intelligent fault diagnosis and condition monitoring technology that provides various heat signatures for different fault circumstances. Radiator tubes clogged, radiator fins clogged, fins and tubes loosely connected, radiator door failure, coolant leaking, and normal operating circumstances These are the six different types of radiator cooling conditions.To breakdown the thermal images, the 2D-DWT is used, and statistical texture characteristics from the original pictures are retrieved and deconstructed into thermal pictures. In the following stage, the significant chosen characteristics are employed to improve the performance of the constructed ANN classifier for the six different cooling radiator settings.The classification findings showed that this approach might be used for intelligent condition monitoring and problem diagnostics of mechanical equipment with a significant thermal signature that indicates its state of operation. Although the system requires some initial training, it is better suited to the continuous and periodic maintenance of a large number of comparable pieces of equipment. In such cases, there should be a significant positive return on the investment [8] [9].

The research [10] suggested an automated feature extraction defect diagnostics system for rotating machinery. The SIFT descriptor is used to extract fault features from the converted recurrence plot and subsequently the feature vector automatically. In their proposed method, they trusted that the average classification accuracy is as high as 98.33% and thus they verified the effectiveness of their proposed method.

They introduced a second-order technique to defect diagnostics of rotating machinery in [11]. Thermogram statistical characteristics in relation to mRMR and SFAM. To improve picture quality, the thermograms of diverse machine settings are first preprocessed. Then, using bi-dimensional empirical mode decomposition, an improved approach is built.After that, GLCM is used to extract second-order statistical characteristics. They discovered that BPNN and PNN take substantially longer to train than SFAM. As a result, it appears that SFAM can provide higher performance at a lower computational cost, which is particularly valuable in real-world applications. Furthermore, massive volumes of data might be exploited in this technique. They suggested a technique of problem diagnostics for induction motors based on binary pattern-based texture analysis, according to research Paper [12]. Because other machines create some background information, it is exceedingly difficult to discover true defects in industrial settings. They developed a two-dimensional texture analysis based on local binary patterns to reduce the problem's severity.The vibration data is transformed into twodimensional gray-scale visuals in this approach, then features are extracted into LBP and the descriptors are utilized to exploit induction motor flaws. They've shown that even when there's a lot of noise, the best classification performance takes the least amount of computational time.

In another paper, researcher [13] created a methodology for bearing problem identification under variable speed settings that does not rely on manual characteristics or typical machine learning methodologies such as artificial neural networks (ANN) and super vector machines (SVM). They employed AE signals to determine a specific health state, which was subsequently employed in conjunction with transfer learning.Furthermore, it collaborates with CNN to provide accurate diagnosis under varied speed settings. Furthermore, ASI transforms AE signals into spectrum imaging before displaying AE spectral properties in images. Last but not least, these photos are used for teaching and testing purposes. It gives a stable classifier technique with excellent accuracy after testing.

They presented a fault diagnostics methodology for rail surface anomalies utilizing FPGA in railways to prevent railway line failures in [14]. They also offered surface-based examination as one of the numerous inspection approaches. The method entails using a camera with an FPGA board to capture an image of the railway. The proposed technique also monitors and diagnoses issues in real time.

Chapter 3

Background

Steel plays an important role in the construction materials sector in this era. Steel structures are resistant to natural and man-made wear, making them commonplace all over the world. This competition will aid in the identification of flaws, which will aid in the more efficient manufacture of steel. Severstal is at the forefront of steel mining and production efficiency. They think that the future of metallurgy demands progress in all elements of the industry—economic, ecological, and social— They are also concerned about corporate social responsibility.With petabytes of previously abandoned data, the firm just established the country's largest industrial data lake. Severstal is currently utilizing machine learning to improve automation, make it more efficient, and maintain high quality in its production processes [15]. Flat sheet steel manufacturing is a particularly sensitive operation. By the time flat steel is ready to ship, numerous machinery has touched it, heating, rolling, drying, and cutting are all steps in the process. Images from high-frequency cameras are being used by Severstal to power a fault detection system. If we are successful, we will help keep steel production standards high and enable Severstal to keep innovating, resulting in a stronger, more efficient world for everyone. In the industrial sectors, there may be many types of faults. The main 8 types of faults are given below:

3.1 Raw materials porosity

Following an acid etching test on steel, it was discovered that some parts of the sample's surface are not dense. There are some obvious voids as well. Porosity is a term used to describe dark patches with inconsistent color tones when contrasted to other portions of the space. Central porosity is defined as porosity that is concentrated in the sample's center: General porosity refers to porosity that is more uniformly dispersed across the sample's surface. The porosity of steel is clearly regulated in both GB/T9943-2008 High- Speed Tool Steel and GB/T1299-2014 Tool Steel, however it frequently exceeds the standard supply.

- Porosity weakens the steel's strength and makes it prone to crack during hot activities such as welding. During heat treatment, it is also easy to produce cracks in the porosity.

- Because of the porosity of the material, the produced tools are easily worn and the surface is not smooth. Because porosity affects steel performance, tool steel has strict porosity requirements. The maximum amount of porosity that can be tolerated.

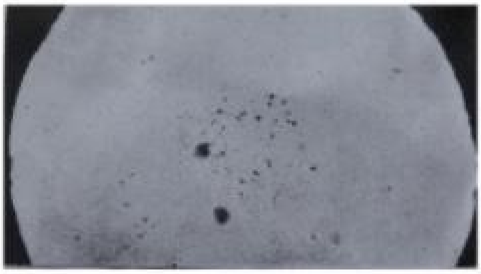


Figure 3.1: Central porosity [16]

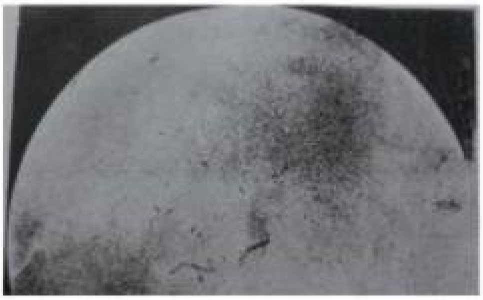


Figure 3.2: Cracks of Central porosity steel during forging billets [16]

In above Figure 3.1 and 3.2 showing forging porosity and porosity cracking patterns on 990mm W18Cr4V.

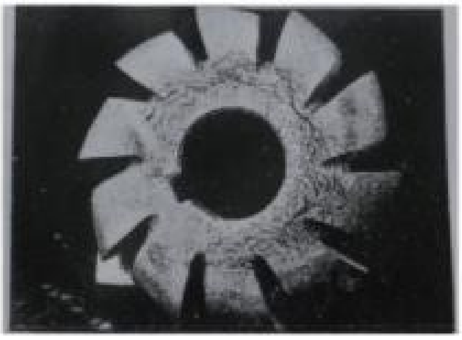


Figure 3.3: Cracks of slotting cutter material due to porosity during heat treatment [16].

Here, figure 3.3 depicts a slotted milling culler made of WI 8Cr4V steel that has been severely cracked due to a lack of heat treatment.

3.2 Shrinkage Residue

When the ingot is cast, the liquid steel shrinks in the middle step of the precipitation process to produce a tubular hole, which is known as shrinkage. Shrinkage is most commonly seen around the feeder in the ingot's head and should be removed once the ingot is done. A billet is created. Shrinkage residue refers to the residual piece that cannot be totally removed. Although it is reasonable to eliminate all shrinkage, steel mills frequently pursue the success rate at the expense of quality. A residue, resulting in an irreparable disaster for the next step Figure 3.4 shows a photo of 470mm W18 steel shrinkage residue and severe porosity, Figure 3.5 shows the depreciation residue of p70mm W18 steel after it has been rolled to generate cracks. When a company sawed p75mm M2 steel a few years ago, it was discovered that there was also depreciation.

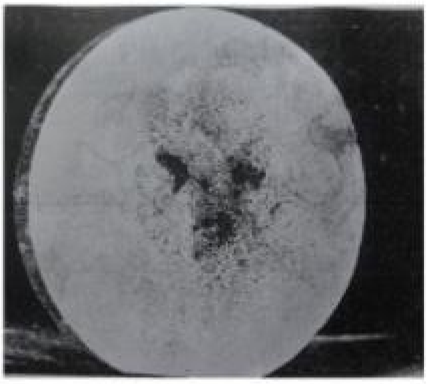


Figure 3.4: Steel shrinkage residue [16].

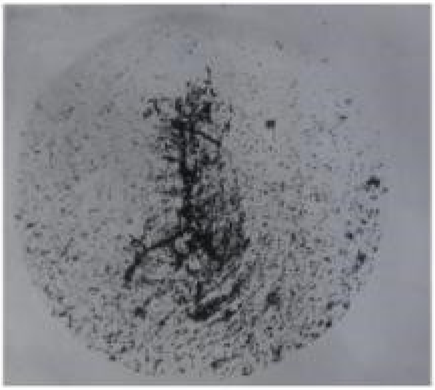


Figure 3.5: Cracks caused by W18 steel shrinkage [16].

3.3 Surface crack

Lengthwise cracks are quite widespread on the surface of raw materials for high-speed steel. The following are some possible explanations:

- During the cooling phase of hot rolling, stress concentration occurs, and cracks form because surface fractures are not completely removed, or the surface is scratched by the die, along the scratch line.

- Folds caused by inadequate die holes or high feed rates during hot rolling will result in cracks along the fold lines during future processing.

- Cracks may form during hot rolling if the cooling rate is too fast or the rolling stop temperature is too low.
- In the cool winter, cracks on the surface prevalent on W18 steel 13mm-4.5mm flat steel rolled, implying that the fractures have climatic consequences, but any cracks are not discovered when the same steel grade and specification is rolled at other times.

Figure 3.6 shows the depth of a surface crack in p30mm W18 steel Bibliography.

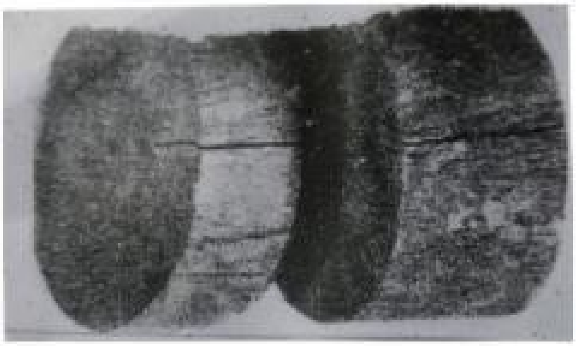


Figure 3.6: Surface crack [16].

3.4 Cracks in the center of raw material

Due to significant deformation during the high-speed steel hot rolling process, the central temperature rises rather than decreases. The crack appears in the material center as a result of thermal stress. Figure 3.7 shows a p35mm W18 steel center crack.It is common to detect the center fracture when cutting high-speed steel raw materials in tool factories.

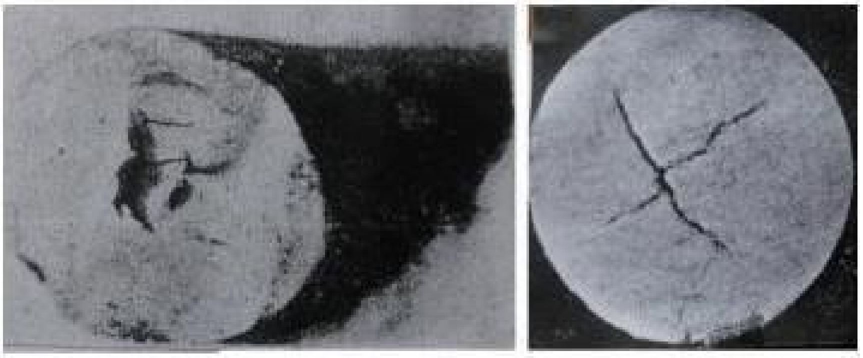


Figure 3.7: Central crack [16].

3.5 Segregation

Segregation refers to the unequal the chemical make-up of the alloy formed during the hardening process, particularly the impurities in carbon and steel are distributed unevenly, which has a major influence. The steel's performance is affected. There are three types of segregation: Micro segregation, Segregation by density, Segregation by region. The alloy's constituent phases have highly varied densities. The heavier sinks while the lighter floats during the solidification process. There is a cross-shaped pattern, which is discovered. The carbon content of the carbon part is higher. Figure 3.8 depicts a quenched W18 steel metallographic sample. There is a cross-shaped pattern, which is discovered .The carbon content of the matrix portion is decreased after the chemical composition study, while the carbon content of the carbon part is higher. The cross-shaped section has a higher content. It's the square segregation generated by carbon and alloy component segregation that results in a cross shape after rolling. The steel's strength will be diminished if there is severe regional segregation, and it will be easy to crack at the segregation point during hot working.

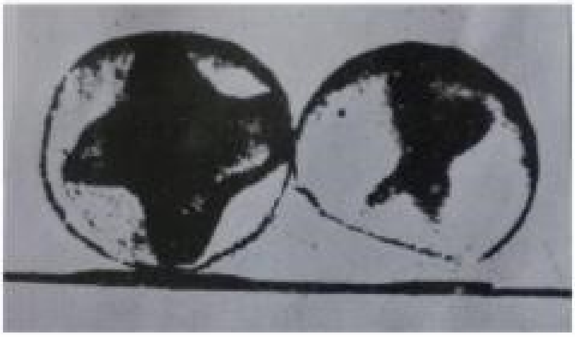


Figure 3.8: Cross-shaped segregation (3x) [16].

3.6 Carbide nouniform

The eutectic carbide breakdown rate refers to how much the eutectic carbides in HSS are broken down during the hot press operation.The stronger the deformation, the greater the degree of carbide fracture and the lower the carbide level. Severe carbides in steel, such as coarse ribbon, mesh, and huge carbide buildup, have a substantial impact on the quality of the steel.Severe carbides in steel, such as coarse ribbon, mesh, and huge carbide buildup, have a substantial impact on the quality of the steel. As a result, strict control is essential to ensure that HSS tools are of high quality. Figure 3.9 shows the effect of carbide no uniformity on the bending strength of W18 steel.

Grades 7-8 with nonuniformity have only 40% – 50% of the bending strength of grades 1–2, as shown in the figure.Uneven quenched grains and a band-like distribution of carbides will also result from the concentration and dispersion of carbides.When working with heated metals, severe carbide nonuniformity can easily lead to cracking and overheating, both of which are potentially hazardous. When the tool was finished, it was tipped into use.Figure 3.10 illustrates a quenching fracture in coarse zonal carbides of W18 steel.

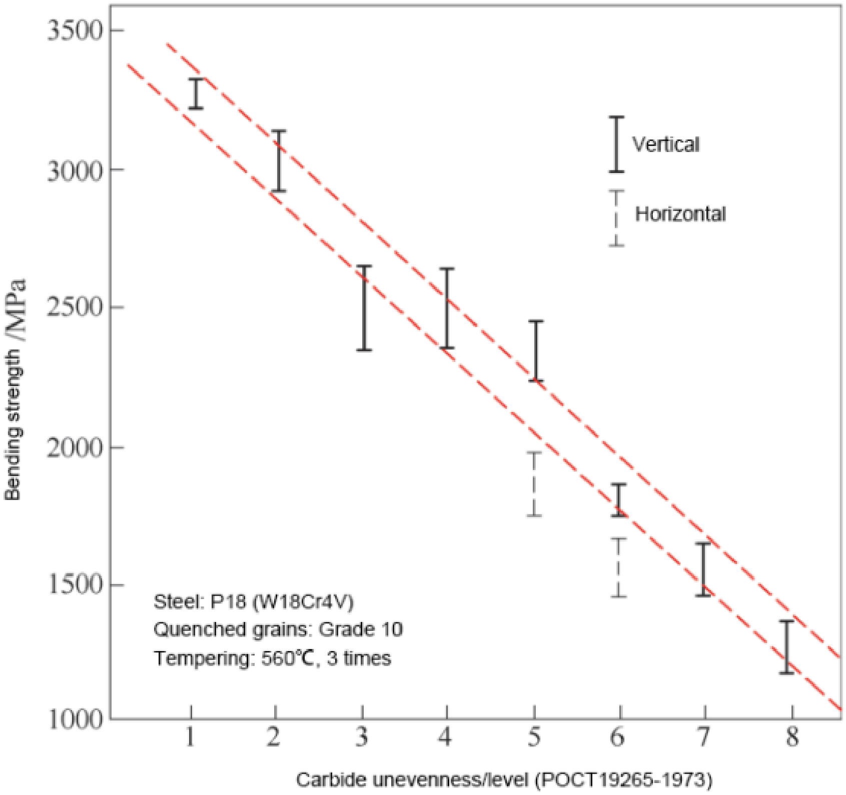


Figure 3.9: The influence of carbide nonuniformity on the bending strength of HSS (W18Cr4V) [16].

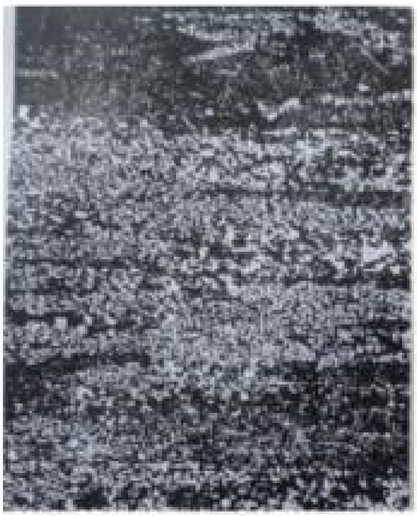


Figure 3.10: Coarse zonal carbide [16].

3.7 Network carbide

Due to the high heating temperature, long holding time, and delayed cooling process of carbide precipitation at grain boundaries, steel in hot rolling or annealing produces a network. It's time to make the carbide. The tool's brittleness is increased by the network carbide, making it susceptible to chipping. In general, the existence of full network carbide is prohibited in steel. After they've been quenched and tempered, network carbides should be inspected. Figure 3.11 illustrates the morphology of T12A steel network carbides , The morphology of 9SiCr steel network carbides is seen in Figure 12. There is a lot of overheating during annealing, as can be seen.

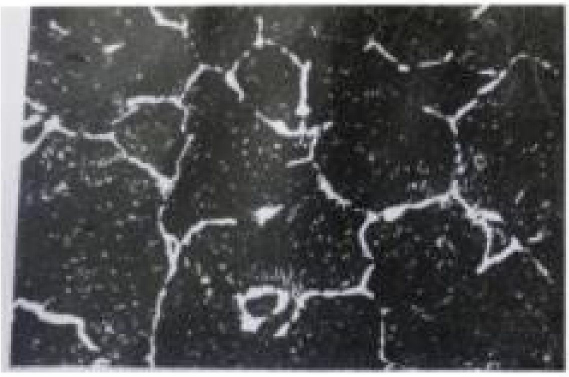


Figure 3.11: T12A Steel Mesh carbide [16].

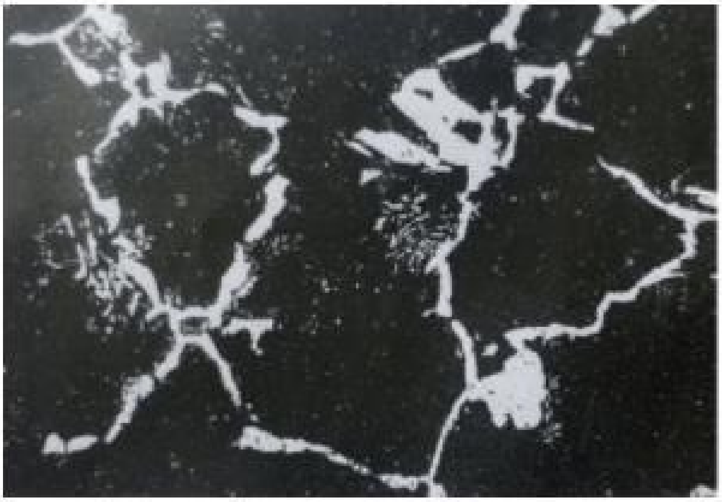


Figure 3.12: 9SiCr steel Mesh Carbide (500x) [16].

3.8 Failure of mix and composition

Material mixing is common in tool and mold manufacturing companies, and it is both a managerial fault and a low-level issue. Mixed furnace, mixed steel, and mixed specifications number are three elements of mixed material. Particularly because a confused furnace number is fairly prevalent, resulting in a lot of incorrect heat treatment with no recourse. From time to time, unqualified tool material components appear. Some high-speed steel components do not satisfy the GB/T9943-2008 High-speed Tool Steel standard because the carbon content is lower than the normal lower limit, notably for the high or low amount of carbon W6Mo5Cr4V2C05 belongs to the HSS-E type..Steel mills must verify that the steel can reach a hardness of greater than 67HRC because it is HSS-E. It is a tool factory internal matter, nothing to do with the steel mill, whether the tool has such a high hardness or not, However, up to 67HRC, the steel mill is responsible. There are also several examples of unqualified die steel composition, and there are still debates.

3.9 Machine Learning

We used a dataset from UCI machine learning repository: steel plate fault dataset on that we wish to work on initially.For our Study, we have used a dataset which has 1941 lines of text. There are seven types of steel plate's fault independent variable. Pastry, Z_Scratch, K_scatch, stains, dirtiness, bumps,Otherfaults. And there are 27 independent variables for attributes. Our preferred algorithms take inputs from the dataset and train 70% of the data for the prediction of accurate output and test the other 30% data [17]. We have also scaled the data so that the features can take a value between 0 or 1. From our research, we will be using CNN, KNN, Logistic Regression and Random Forest algorithms to compare the percentage of accuracy of our model.

3.10 Convolutional Neural Networks (CNN)

CNN is widely used for the detection of faults by analyzing images from a dataset because of its high accuracy. Feature extraction and classification are the two sections of conventional CNN. CNN has several properties for dealing with images that initiates the faster training along with less training parameters.

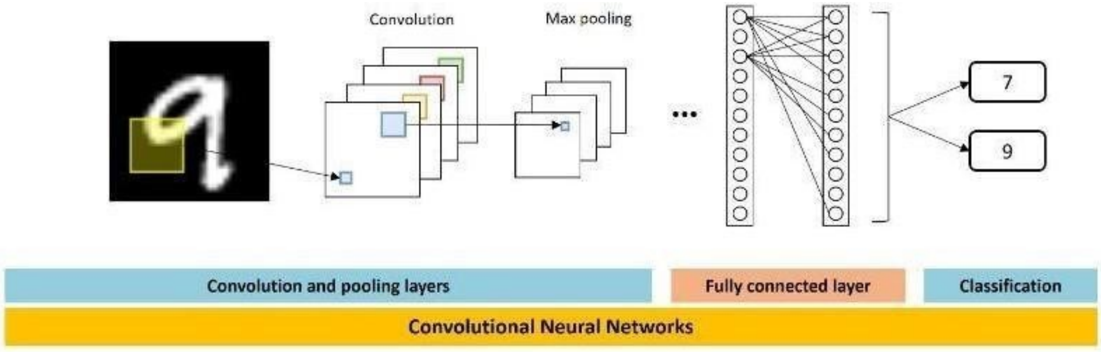


Figure 3.13: An Architecture of conventional CNN [15]

From the figure 3.13, firstly the faulty image is inserted and identifies the area which it will be working with. Secondly, the convolution area is being implemented by the weights in CNN extracting features from a weight sharing feature map. After this, max pooling is done by ensuring the spatial size of the feature maps and controlling the over-fitting of the pixels. Then CNN goes through the fully connected layer for the classification of the extracted features.

Moreover, from the figure 3.14, CAM (Class activation map) method is used to help the CNN predict the particular class. CAM performs both binary and multi-class classification problems. It estimates which part of an image is being focused and analyzed [26]. Furthermore, CAM is a modified convolution layer as it is placed after the final convolution layer for the better accuracy of the prediction.

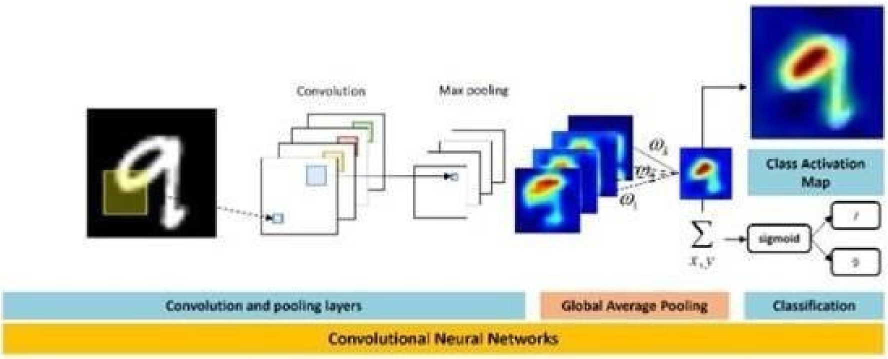


Figure 3.14: CNN architecture with CAM [15]

3.11 KNN(k-nearest neighbors)

By far the simplest basic machine image classification algorithm is the k-Nearest Neighbor classifier. This approach is based on the distance between feature vectors, similar to how an image search engine works, but we have labels connected with each image, allowing us to forecast and return a category for the image [18]. To calculate this, we utilize the Euclidean distance between two points equation 3.1, which is defined as follows:

$$d\left(p_1, p_2\right) = \sqrt{\left(p_1^1 - p_2^1\right)^2 + \left(p_1^2 - p_2^2\right)^2} \tag{3.1}$$

If every point on a 2D image is a vector, the Euclidean distance (scalar) may be calculated using the following diagram:

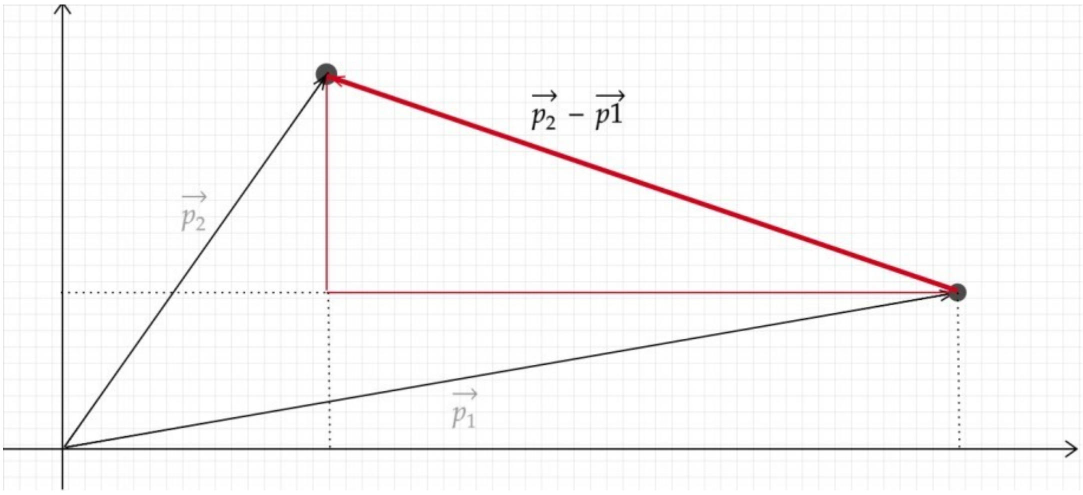


Figure 3.15: The Euclidean distance [17]

We send test data to the model after training to anticipate the defect, exactly like we supplied a new point in 2D space to identify its pixel.

3.12 Logistic regression

Logistic regression is a statistical model for binary or multiclass classification which is used for the probability of categorical outcome rather than a continuous output. It provides a logistic function which gives a sigmoid “S” shaped output.

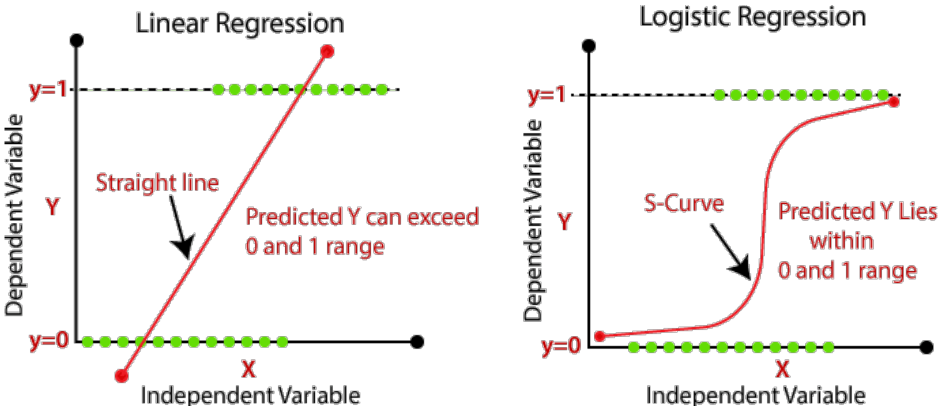


Figure 3.16: Logistic regression with predicted “S” shaped curve [18]

From this figure, we can see that logistic regression is a part of linear regression algorithm. But logistic regression is more accurate than linear regression. We will use a solver that supports logistic regression with no regulation penalty. Moreover, it works with a large number of data samples and it is best for multiple classes logistic regression. We can also use the coefficient values for determining which pixels have been chosen or what class the sample belongs to. From the coefficient matrix we can easily get the per class probability for any sample data [19].

3.13 Random forest

Random forests become more popular but it is not so commonly used Because many remote sensing practitioners are unfamiliar with random forest techniques, they are used for image classification and regression [20]. A random forest is a supervised learning method. The essential concept of the bagging approach is that combining several learning models increases total output. The random forest algorithm mixes many decision trees to provide a more accurate and reliable forecast. When the data set grows larger, random forest is preferable. To handle issues involving classification and regression it works like a decision tree. An network method is random forests, which means it calculates a response by combining the results of several separate models. In the vast majority of circumstances, the result of the hybrid model will be superior to the outcome of any of the individual models.In our model random forests calculate a response variable by creating many several decision trees. After that, it runs each of the decision trees with each item to be modeled. The response is determined by analyzing the replies from all of the trees. The class that is expected or assigned for that item is the class that is predicted or assigned the most.

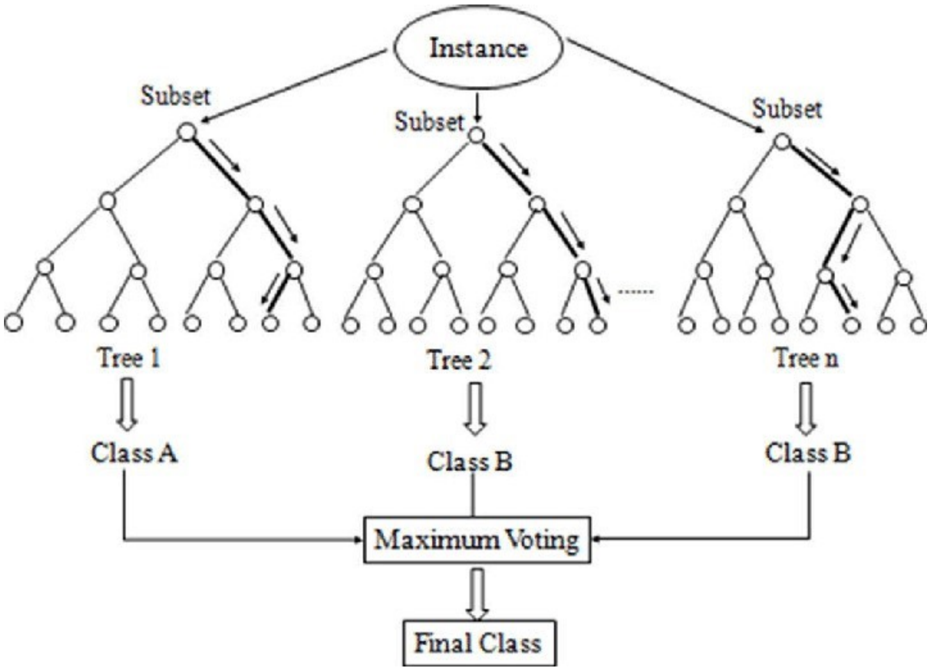


Figure 3.17: Classification of classes in random forest [19]

There are several parameters that must be provided. The most often used parameters are [19]: image bands and digital elevation models; These are input training data, which include predictor and response variables like land cover type and biomass. The number of trees that need to be created in order to create a binary rule for each split of the predictor variable is then determined. In addition, there are parameters for calculating information about inaccuracy and variable importance.

Chapter 4

Methodology

4.1 Workflow

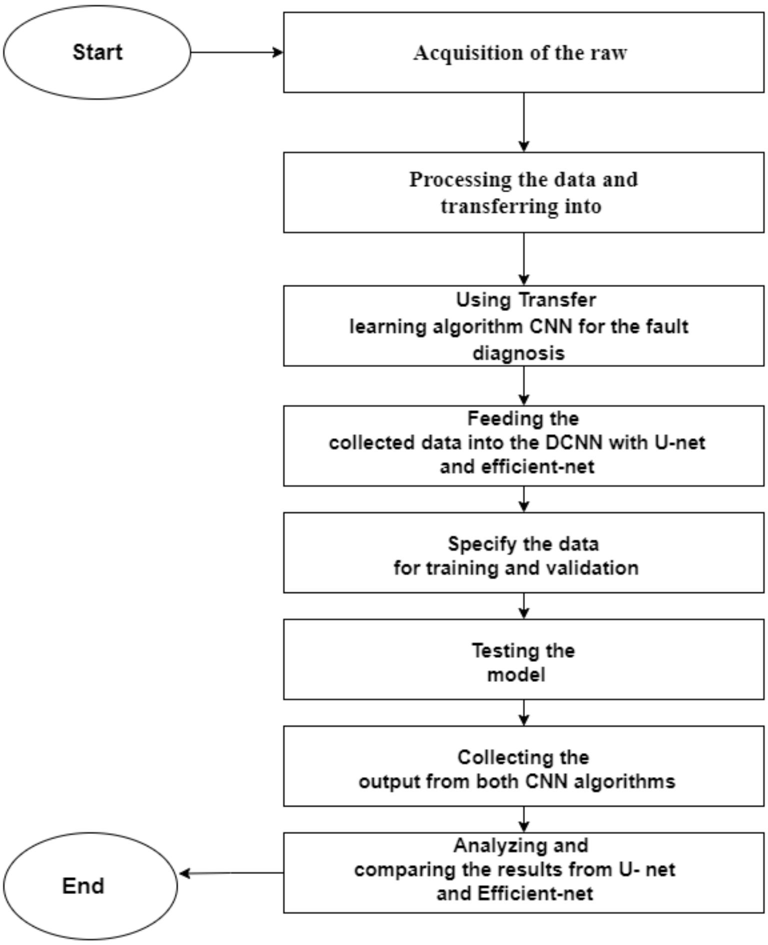

Start
Acquisition of the raw
Processing the data and transferring into
Using Transfer learning algorithm CNN for the fault diagnosis
Feeding the collected data into the DCNN with U-net and efficient-net
Specify the data for training and validation
Testing the model
Collecting the output from both CNN algorithms
Analyzing and comparing the results from U- net and Efficient-net
End

Figure 4.1: Workflow diagram [20].

For our proposed model, we have collected datasets of Steel faulty plates from Kaggle. Then we processed the data and transferred it into 2D image. After that, we have used transfer learning algorithm CNN for the fault diagnosis. We use unet and efficientnet models for comparing the accuracy of our given data. Then we specified the training data and tested the model. At the end, we have analyzed and compared the results and chose a model which works better to detect steel plate faults effectively [21].

4.2 Architecture Used

4.2.1 UNet

Olaf Ronneberger et al. [22] developed the UNET which is a convolutional network design for image segmentation that is quick and precise. It appears to be in the shape of a "U." The architecture is symmetric and is mostly made up of two routes. The first is an encoder path, whereas the second is a decoder path. The encoder path records the image's context for creating feature maps. The encoder path is nothing more than a stack of convolution and maximum pooling layers. Using transposed convolutions, a decoder path was employed to achieve exact localization. Because U-net only has Convolutional layers and no Dense layers, it can accept images of any size.It can detect and localize boundaries since it conducts pixel categorization, resulting in the input and output can be the same size. The network's basic structure is as follows in figure 4.2.

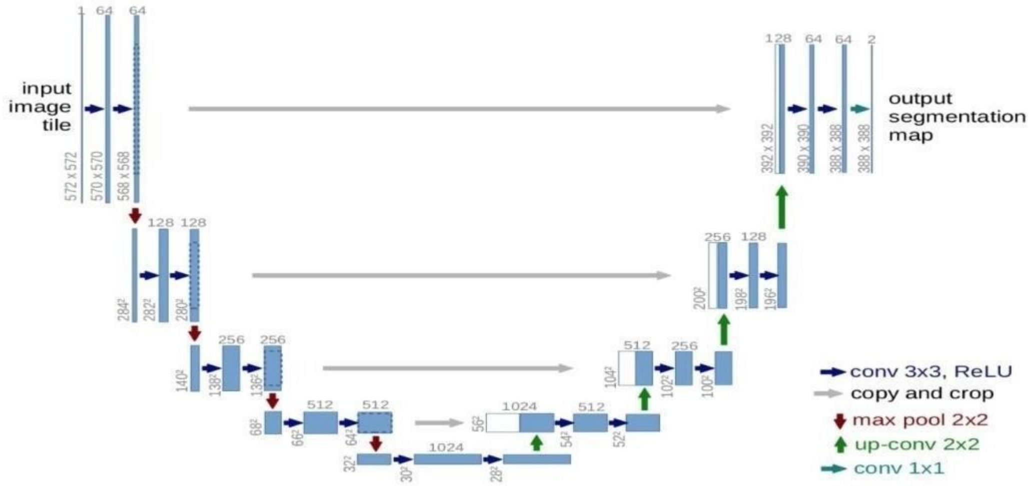


Figure 4.2: UNet Architecture [21].

4.2.2 Contraction/down sampling path(Encoder Path)

There are four blocks in the encoding path. It is made up of two 3x3 convolutions that are applied repeatedly. After each conv, there is a ReLU and batch normalization. The spatial dimensions are then reduced using a 2x2 max pooling operation. We double the number of feature channels while halving the spatial dimensions at each down sampling step. In the first part of the figure 4.3 it is shown.

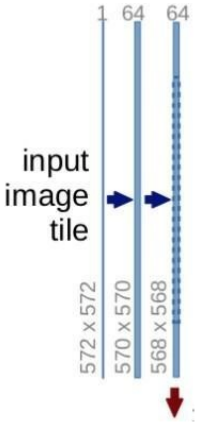


Figure 4.3: Input image tile conv 3*3 ReLU [21].

Each procedure has two convolutional layers, and the number of channels changes from 1 to 64 as the depth of the picture is increased via the convolution process. The max pooling process (shown by the red arrow going down) reduces the image size by half (the size reduction from 572x572 to 568x568 is due to padding difficulties, however the solution here uses padding= "same"). The procedure is done three times more, in the figure 4.4.

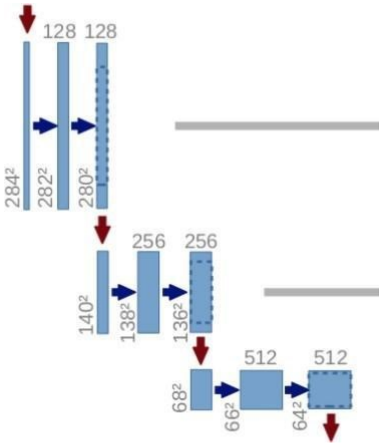


Figure 4.4: Conv 3*3 ReLU with Maxpooling [21]

and now we've arrived at the bottom (figure 4.5),

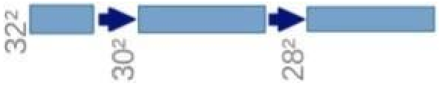


Figure 4.5: Conv 3*3 ReLU without Maxpooling

There are still two convolutional layers generated, however there is no max pooling. At this time, the image has been scaled to 28x28x1024.

4.2.3 Expansion/Up sampling path (Decoder Path)

Every step in the expanding route starts with an upsampling of the feature map and then a 2x2 transpose convolution, which reduces the number of feature channels in half. We also have a concatenation with the contracting path's matching feature map, as well as a 3x3 convolutional neural network (each followed by a ReLU). A 1x1 convolution is used to map the channels to the desired number of classes in the final layer.

The image will be upsized to its original size in the expanding route. The following is the formula (figure 4.6):

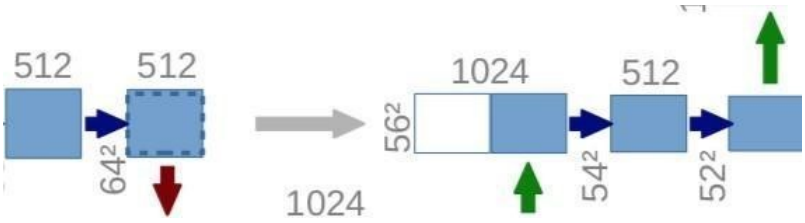


Figure 4.6: Upsampling [21]

Transposed convolution is an up sampling technique that extends the size of pictures. It basically pads the original picture before doing a convolution process. The picture is upsized from 28x28x1024 to 56x56x512 after transposed convolution, and then concatenated with the equivalent image from the contracting route to create an image of size 56x56x1024. The goal here is to integrate the data from the previous layers to provide a more exact prediction. This method is repeated three times more, as before. The final stage is to modify the picture to meet our prediction criteria now that we've reached the top of the architecture, from figure 4.7.

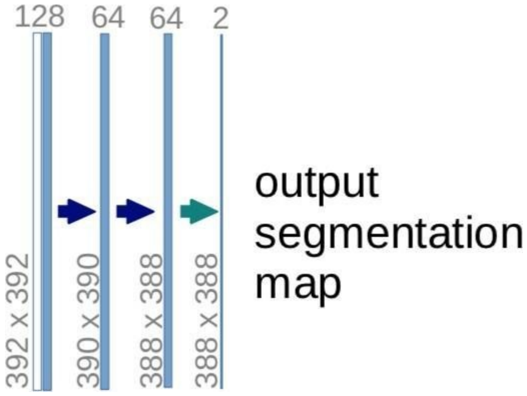


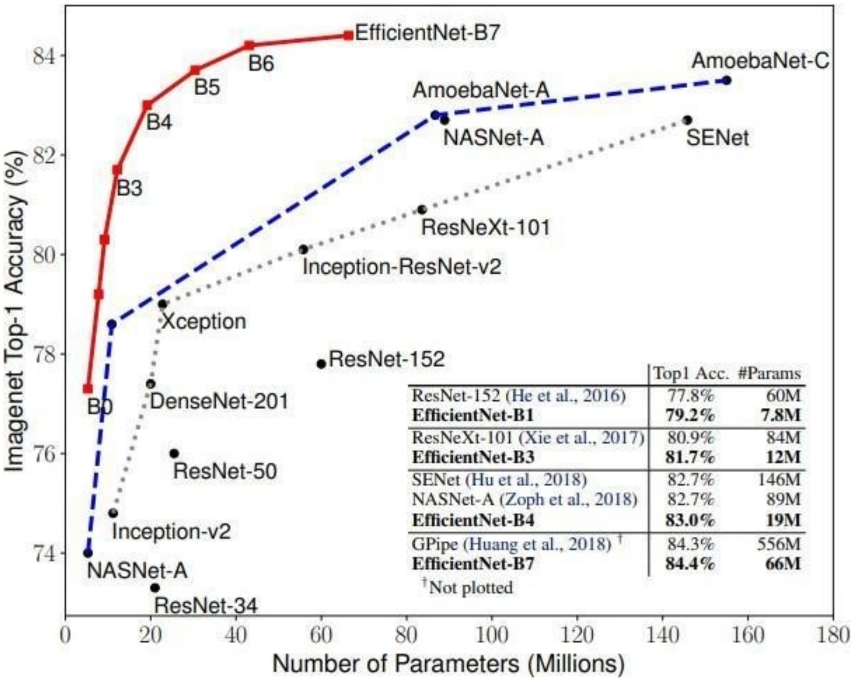
Figure 4.7: Output segmentation map, conv 1*1 [21]

The last layer is a convolution layer with one 1x1 filter (notice that there is no dense layer in the whole network). The remainder of the training for neural networks is the same.

4.2.4 EfficientNet

EfficientNet is a convolutional neural network design and scaling method that scales all depth, breadth, and resolution dimensions evenly using a compound coefficient. If we need to employ 2N times more computer resources, we can simply raise the network depth by N, the breadth by N, and the image size by N, where,,, are constant coefficients discovered using a brief grid search on the original tiny model.

To scale network breadth, depth, and resolution evenly and consistently, EfficientNet uses a compound coefficient. EfficientNet refers to a set of convolutional neural network models as a whole. However, due to several of its characteristics, it is more efficient than most of its predecessors. The principle behind compound scaling is that if the input image is bigger, the network needs additional layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger picture. Instead of Darknet53, the EfficientNet feature extraction network was employed as the backbone feature extraction network. For network scaling, it examines the balance between network depth, width, and picture resolution to enhance the algorithm's accuracy with limited computational resources, resulting in considerably higher accuracy and efficiency than prior ConvNets. EfficientNet-B7 is 8.4 times smaller and 6.1 times quicker than the current ConvNet on ImageNet, with state-of-the-art 84.3 percent top-1 accuracy. EfficientNets transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and three more transfer learning datasets with an order of magnitude less parameters, is shown in figure 4.8.



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Figure 4.8: Computed parameter vs. ImageNet Top-1 accuracy comparison [23]

The EfficientNet model family consists of eight models with numbers ranging from B0 to B7, with each model number suggesting a variant with more parameters and higher accuracy. Since 2012, we can observe that model success on the ImageNet dataset has risen as models have become more sophisticated. However, in terms of processing capacity, most of them are ineffective. Smaller models have been used in more efficient ways in recent years. So much so that when scaling down the model, the focus is on depth, breadth, and resolution, rather than focusing on all three separately. It works the way that divides the original convolution into two phases to minimize calculating time and cost while maintaining accuracy. It's called depth wise Convolution and Point WiseConvolution (figure 4.9).

Inverse Res: The original ResNet blocks are made up of two layers: one squeezes the channels and the other widens them. In this way, it links skip connections to

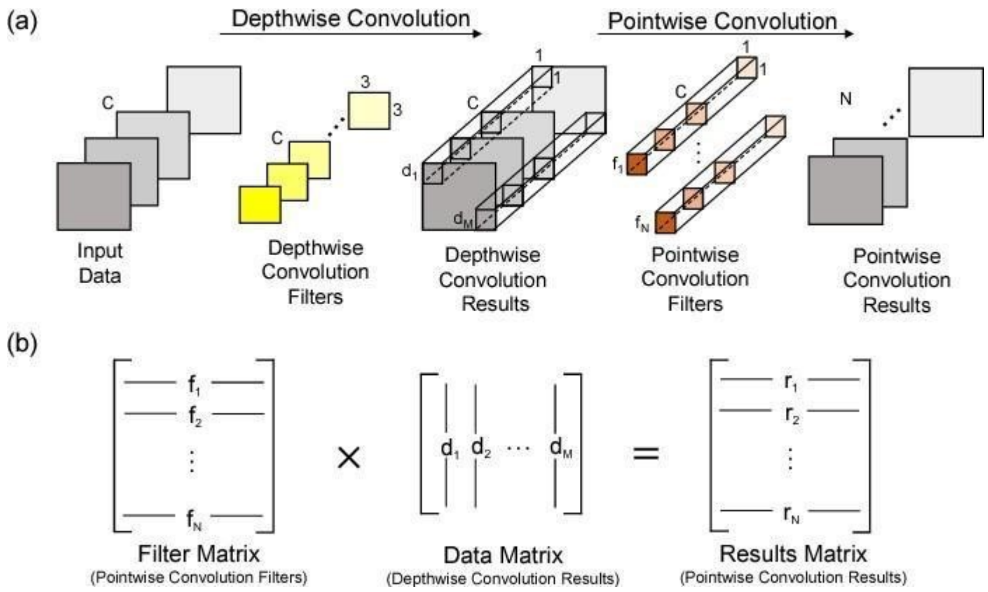

(a)
Depthwise Convolution
Pointwise Convolution
C
3
3
C
1
1
C
d1
dM
C
1
1
f1
fN
N
Input Data
Depthwise Convolution Filters
Depthwise Convolution Results
Pointwise Convolution Filters
Pointwise Convolution Results
(b)
f1
f2
fN
×
d1 d2 ··· dM
=
r1
r2
rN
Filter Matrix
(Pointwise Convolution Filters)
Data Matrix
(Depthwise Convolution Results)
Results Matrix
(Pointwise Convolution Results)

Figure 4.9: Depthwise convolution and Pointwise Convolution [23]

rich channel layers. In MBConv, blocks are made up of a layer that first stretches and then compresses channels, enabling levels with fewer channels to be skipped.

Linear bottleneck: Uses linear activation in the last layer of each block to prevent information loss via ReLU.

Because they decrease computing costs, energy consumption, and training and inference durations, models like EfficientNet are ideal for deploying transfer learning on the edge. Transfer learning may now be utilized on smartphones and other edge devices thanks to this degree of model efficiency. Simple and smart mathematical changes, in my opinion, may be used to arrive at these successful answers.

Chapter 5

Implementation of proposed models

5.1 UNet Based 2D CNN

In our proposed model, our image sizes are 512 x 512 as an input and output mask produces 512 x 512. Moreover, we have used ReLU in the algorithm for performing the activation function in all layers. As our images are two types such as defective and defect free, we have applied the loss function as follows,

$$L(y, \hat{y}) = \sum_{i}^{n} [y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i] \tag{5.1}$$

From this equation 5.1, when there is no number of our samples, yi is the ground truth level and the predicted level is yi ' . For the training of our efficiency, we have applied real time augmentation.Data augmentation is required for reducing the over fitting of the same data. Moreover, this method generates new images artificially by cropping, zooming, rotating etc.Then , all the epochs were run during the training. For the automatically augmenting data, we have used Keras Deep learning library from python during the training phase.

Moreover, we have used “batch normalization” which allows learning independently to every layer of the network.As a result , the artificial neural networks work faster and are more stable. For the two dimensional maps, max pooling is applied to 2x2 patches with the stride (2,2). We have used “global average pooling 2D” to replace the fully connected layers.The output of global average pooling is 1x 1x512. Moreover, we have used “dropout” for dropping out units during the training phase. Additionally, we have used a dense layer that receives all the data from previous neurons.It is a widely used layer in CNN models [24].In order to implement our model, we have imported a UNet model using keras module. We have made the final connected layer false and imported the pre-trained model first. Moreover, we have used the basic unet algorithm .We have tested our model with a test dataset.Therefore, we have transferred the weights from unet to our model.

5.2 EfficientNet Based 2D CNN

For our proposed model, we have used an efficient net for steel plates fault detection. Firstly, we had the 2D picture of our faulted steel plates and our first step is to run the images through Image neural network classifiers so that we can understand that the image of plates has the defect of not. Our second step is to run the same image through segmentation neural networks to identify the location and type of the fault.

For the training purpose we have used K=5 and K-Fold cross validation for the data distribution.Moreover, all the pictures are the same size.In the binary classification step , we used EfiicientNet Classifiers version B0-B5 for the ensemble of data and the classifiers were selected according to their performance.

binary_ensemble = assemble_ensemble(bin_keys) − = eval_temp_fix(binary_ensemble, dsets_bin['valid']) binary_ensemble.summary()

Then we did the binary model evaluation and calculated the accuracy of the single best classifier. After the ensemble of binary model we got our required accuracy. For the second step, we have used UNet++ along with EfficientNet Classifiers version B0B5 for. Furthermore, we have used the metric Dice Coefficient, equation 5.2.

$$\text{Dice coefficient } = \frac{2|\hat{Y} \cap Y|}{|\hat{Y}| + |Y|} \tag{5.2}$$

During our training phase, we have calculated dice coefficients in three ways such as batchwise average, image-wise average and channel-wise average.In addition, images without and faults are only used in the training phase and those images were not being used for oversampling. Then we have selected segmentation ensemble weights. After that we have a mask prediction loop for post processing the data with or without faults for ensemble. A Tensor processing unit (TPU) would be able to make all the predictions and temporarily store data in numpy arrays. After that we have retrieved demo Images and plot ground truth versus prediction function.

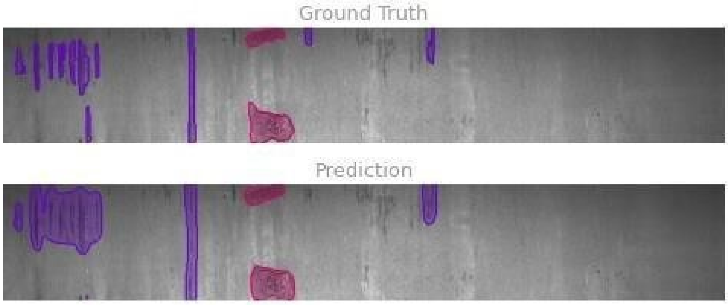


Figure 5.1: Ground Truth VS Prediction

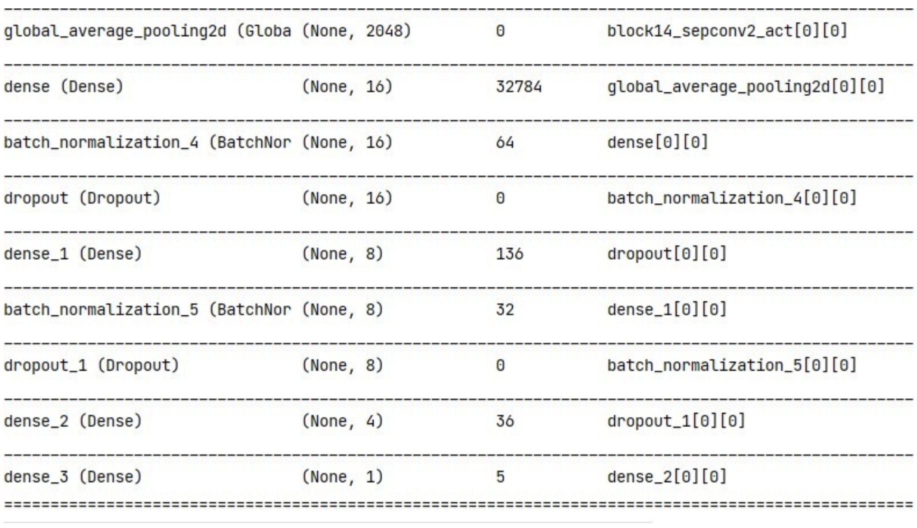
Chapter 6

Result

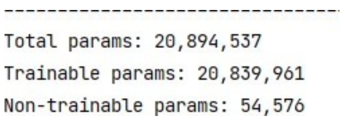
In our model, we have used 5000+ data for training and 650+ data for training and validating. After that, the epochs are executed and we get our accuracy, loss , validation accuracy and validation loss. Moreover, we get the total parameters, trainable parameters and non-trainable parameters. Moreover, we have plotted the graphs for indicating more accuracy for both of our models. As a result, we can easily compare among our proposed models which one would be giving more accuracy and be the best model of detecting steel plates fault.

6.1 UNet model summary Model

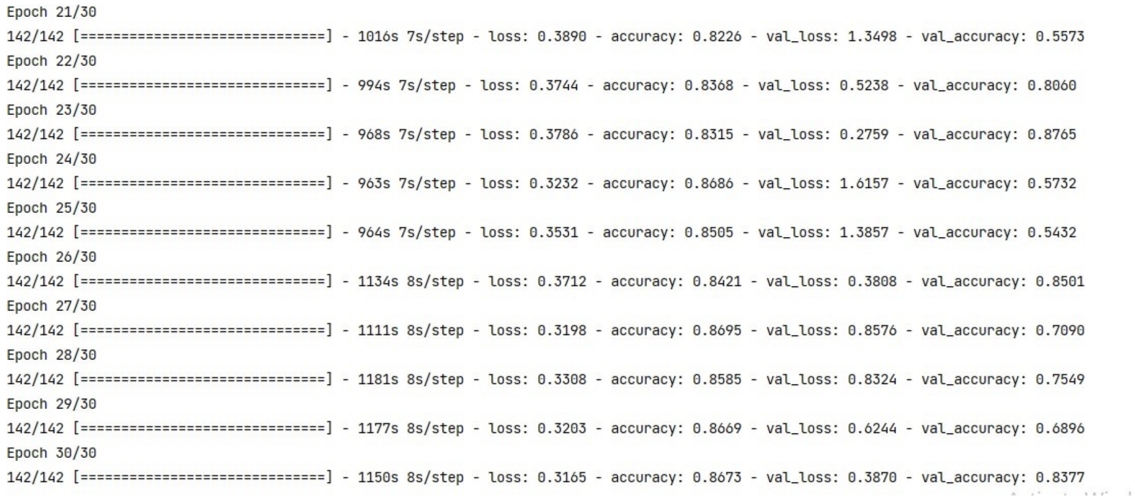
Summary:



```
__________________________________________________________________________________________________
global_average_pooling2d (Globa (None, 2048)         0           block14_sepconv2_act[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 16)           32784       global_average_pooling2d[0][0]
__________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 16)           64          dense[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 16)           0           batch_normalization_4[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 8)            136         dropout[0][0]
__________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 8)            32          dense_1[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 8)            0           batch_normalization_5[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 4)            36          dropout_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 1)            5           dense_2[0][0]
==================================================================================================
```



```
________________________________
Total params: 20,894,537
Trainable params: 20,839,961
Non-trainable params: 54,576
```



```
Epoch 21/30
142/142 [==============================] - 1016s 7s/step - loss: 0.3890 - accuracy: 0.8226 - val_loss: 1.3498 - val_accuracy: 0.5573
Epoch 22/30
142/142 [==============================] - 994s 7s/step - loss: 0.3744 - accuracy: 0.8368 - val_loss: 0.5238 - val_accuracy: 0.8060
Epoch 23/30
142/142 [==============================] - 968s 7s/step - loss: 0.3786 - accuracy: 0.8315 - val_loss: 0.2759 - val_accuracy: 0.8765
Epoch 24/30
142/142 [==============================] - 963s 7s/step - loss: 0.3232 - accuracy: 0.8686 - val_loss: 1.6157 - val_accuracy: 0.5732
Epoch 25/30
142/142 [==============================] - 964s 7s/step - loss: 0.3531 - accuracy: 0.8505 - val_loss: 1.3857 - val_accuracy: 0.5432
Epoch 26/30
142/142 [==============================] - 1134s 8s/step - loss: 0.3712 - accuracy: 0.8421 - val_loss: 0.3808 - val_accuracy: 0.8501
Epoch 27/30
142/142 [==============================] - 1111s 8s/step - loss: 0.3198 - accuracy: 0.8695 - val_loss: 0.8576 - val_accuracy: 0.7090
Epoch 28/30
142/142 [==============================] - 1181s 8s/step - loss: 0.3308 - accuracy: 0.8585 - val_loss: 0.8324 - val_accuracy: 0.7549
Epoch 29/30
142/142 [==============================] - 1177s 8s/step - loss: 0.3203 - accuracy: 0.8669 - val_loss: 0.6244 - val_accuracy: 0.6896
Epoch 30/30
142/142 [==============================] - 1150s 8s/step - loss: 0.3165 - accuracy: 0.8673 - val_loss: 0.3870 - val_accuracy: 0.8377
```

Accuracy : 86.73 %
Loss : 31.65%
Validation Accuracy : 83.77%
Validation Loss: 38.70%

Validation Loss graph:

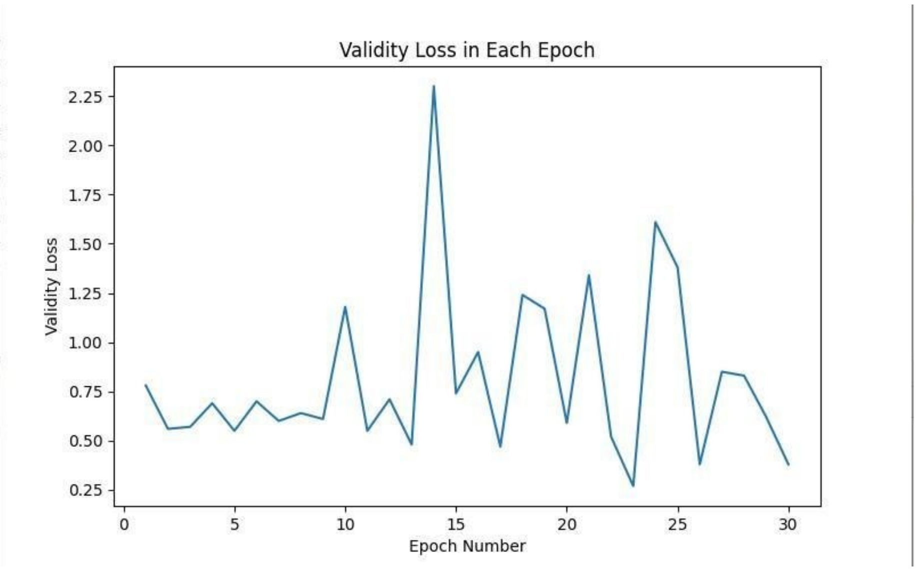


Figure 6.1: Validity Loss

Validation Accuracy graph:

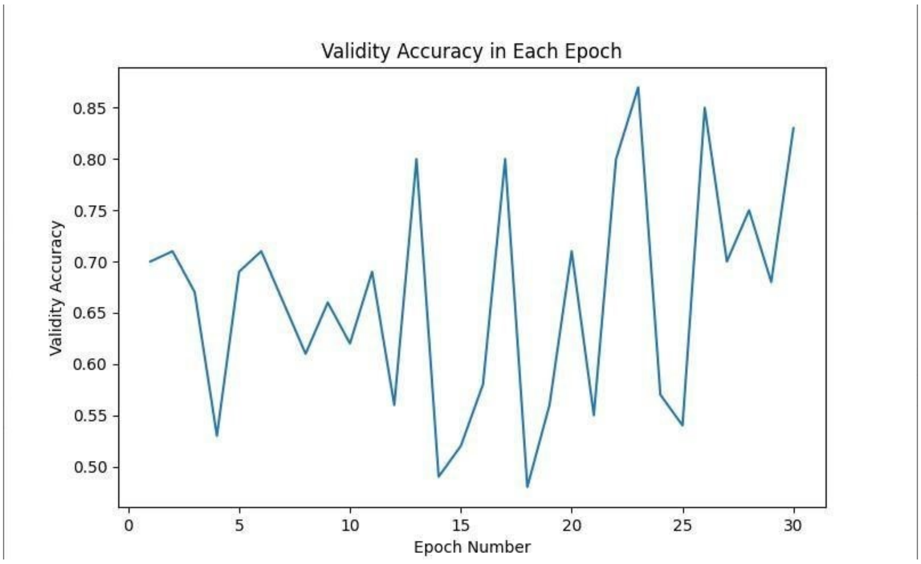
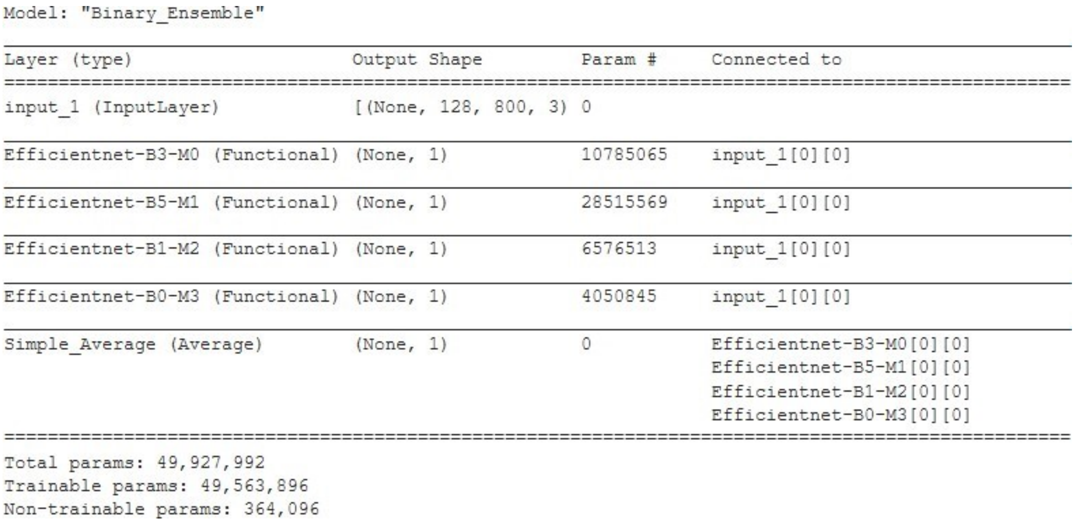


Figure 6.2: Validity Accuracy

6.2 Model summary of EfficientNet

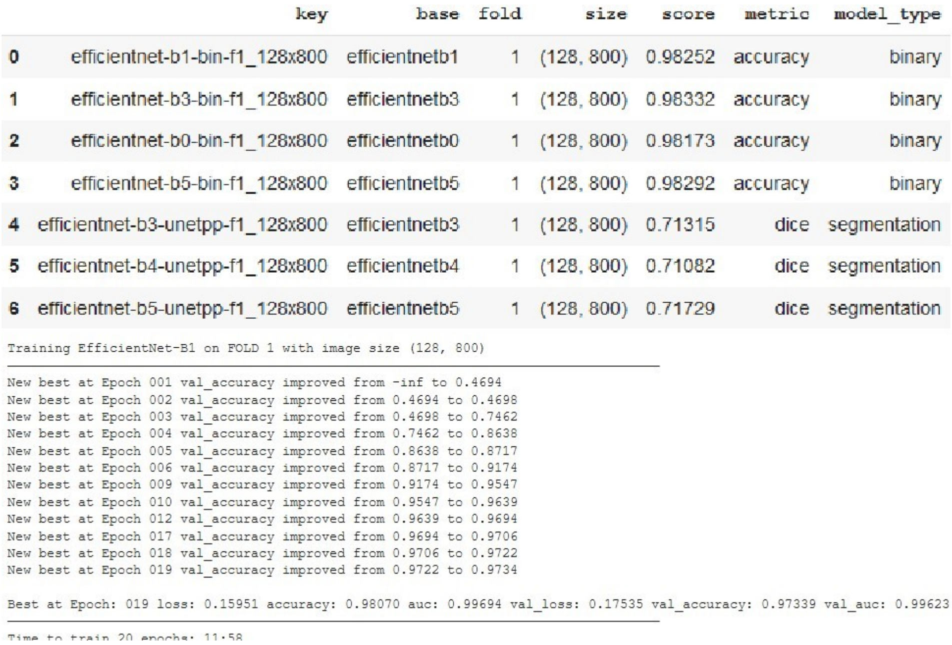
6.2.1 Binary Ensemble



```
Model: "Binary_Ensemble"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 128, 800, 3) 0

__________________________________________________________________________________________________
Efficientnet-B3-M0 (Functional) (None, 1)            10785065    input_1[0][0]
__________________________________________________________________________________________________
Efficientnet-B5-M1 (Functional) (None, 1)            28515569    input_1[0][0]
__________________________________________________________________________________________________
Efficientnet-B1-M2 (Functional) (None, 1)            6576513     input_1[0][0]
__________________________________________________________________________________________________
Efficientnet-B0-M3 (Functional) (None, 1)            4050845     input_1[0][0]
__________________________________________________________________________________________________
Simple_Average (Average)        (None, 1)            0           Efficientnet-B3-M0[0][0]
                                                                 Efficientnet-B5-M1[0][0]
                                                                 Efficientnet-B1-M2[0][0]
                                                                 Efficientnet-B0-M3[0][0]
==================================================================================================
Total params: 49,927,992
Trainable params: 49,563,896
Non-trainable params: 364,096
```

6.2.2 Training EfficientNet-B1



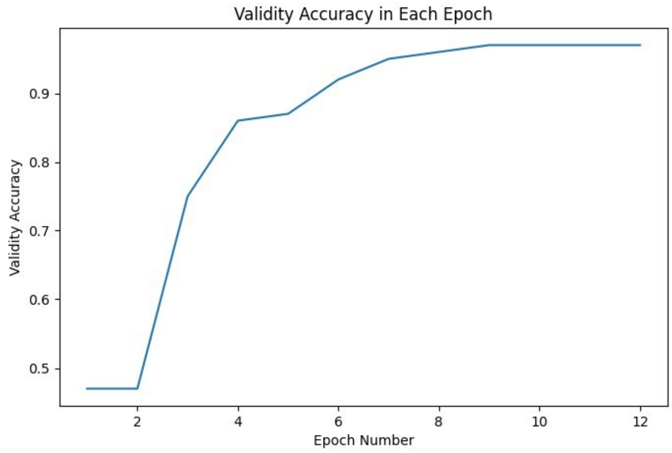
	key	base	fold	size	score	metric	model_type
0	efficientnet-b1-bin-f1_128x800	efficientnetb1	1	(128, 800)	0.98252	accuracy	binary
1	efficientnet-b3-bin-f1_128x800	efficientnetb3	1	(128, 800)	0.98332	accuracy	binary
2	efficientnet-b0-bin-f1_128x800	efficientnetb0	1	(128, 800)	0.98173	accuracy	binary
3	efficientnet-b5-bin-f1_128x800	efficientnetb5	1	(128, 800)	0.98292	accuracy	binary
4	efficientnet-b3-unetpp-f1_128x800	efficientnetb3	1	(128, 800)	0.71315	dice	segmentation
5	efficientnet-b4-unetpp-f1_128x800	efficientnetb4	1	(128, 800)	0.71082	dice	segmentation
6	efficientnet-b5-unetpp-f1_128x800	efficientnetb5	1	(128, 800)	0.71729	dice	segmentation

```
Training EfficientNet-B1 on FOLD 1 with image size (128, 800)
________________________________________________________________________
New best at Epoch 001 val_accuracy improved from -inf to 0.4694
New best at Epoch 002 val_accuracy improved from 0.4694 to 0.4698
New best at Epoch 003 val_accuracy improved from 0.4698 to 0.7462
New best at Epoch 004 val_accuracy improved from 0.7462 to 0.8638
New best at Epoch 005 val_accuracy improved from 0.8638 to 0.8717
New best at Epoch 006 val_accuracy improved from 0.8717 to 0.9174
New best at Epoch 009 val_accuracy improved from 0.9174 to 0.9547
New best at Epoch 010 val_accuracy improved from 0.9547 to 0.9639
New best at Epoch 012 val_accuracy improved from 0.9639 to 0.9694
New best at Epoch 017 val_accuracy improved from 0.9694 to 0.9706
New best at Epoch 018 val_accuracy improved from 0.9706 to 0.9722
New best at Epoch 019 val_accuracy improved from 0.9722 to 0.9734

Best at Epoch: 019 loss: 0.15951 accuracy: 0.98070 auc: 0.99694 val_loss: 0.17535 val_accuracy: 0.97339 val_auc: 0.99623
________________________________________________________________________
Time to train 20 epochs: 11:58
```

Accuracy: 98.07%
Loss: 15.95%
Validity Accuracy: 97.33%
Validity Loss: 17.54%

6.2.3 Validity Accuracy graph



When we look at the model summaries and validation losses and accuracies, we can clearly see the problem with UNet or other state of the imagenet classifier based transfer learning frameworks, despite the fact that the overall accuracy of detecting the target classes has been quite similar between both transfer learning methods. The imagenet dataset contains millions of data samples divided across four classifications, and efficientNet is trained on it. When we consider the amount of total

Table 6.1: Comparison between UNet and EfficientNet

	UNet	**EfficientNet**
Accuracy	86.73%	98.07%
Loss	31.65%	15.95%
Validity Accuracy	83.77%	97.33%
Validity Loss	38.70%	17.54%

parameters and trainable parameters in the UNet-based architecture, it is clear that we are extracting and performing numerous resource-intensive operations on a large number of inefficient features. Although U-Net is one of the most commonly used fully convolutional neural networks, it has limitations in extracting some of the more sophisticated characteristics that might aid picture segmentation. In this paper, we look at some of the constraints of the U-Net in terms of locating and segmenting the object of interest (OOI) in a synthetic dataset. When we look at our transfer learning-based architecture efficientNet, on the other hand, we can see that the model we selected was ideal for the job. Our architecture has far fewer total and trainable parameters than the efficientNet-based model, depending on the sample size of our data. In addition, the total to trainable parameter ratio is 1:1. This implies that our architecture has no unnecessary or additional computation. As a result, compared to using transfer learning with other models pre-trained on imagenet, our model is significantly more optimized and efficient in identifying industrial steel defects.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

We use a transfer learning technique to fill the gap between simulation and physical worlds. This method allows the computer simulation's diagnostic information to be applied to the actual process. With minimal learnable parameters, transfer learning model produces better outcomes than comparable architectures as it also showed great promise in process fault diagnosis. For classification and fault diagnosis,Picture representations of the raw signal are created using time-frequency representations of the raw data and put into a Deep convolutional neural network (CNN) architecture. To detect industrial steel faults, we utilized a transfer learning model in our paper.The model was 92.5% accurate in detecting problems, but it was unable to identify them.For the never-ending population and demand growth, industrial manufacturing has become the most serious issue of the twenty-first century. For industrial manufacturing to function safely and effectively, fault detection is essential. So we focused on Industrial Steel Fault Diagnosis Using Transfer Learning and industrial Fault Signals. We used EfficientNet, a convolutional neural network, to improve accuracy. It's a scaling method that scales all depth, breadth, and resolution parameters evenly using a compound coefficient. The procedure for training the neural network with extracted features from each character 's test image provides detection precision comparable to something significant on the other hand,aside from the present characteristics, more may be added to improve the neural network's performance.Our suggested model is able to identify data with an average accuracy of 98%, according to experimental results.The proposed model improves previous fault detection approaches and we believe it has the potential to become a new basis for future computer vision applications.

7.2 Future Works

We described the CNN model and how a transfer learning model based on images to detect or forecast industrial problems in our article. We will achieve this by converting 1D sensor fault signals to 2D images. After that, we'll use a deep learning model to extract deep features for training and testing the classifier. We classified the data with 98% accuracy using two architectures: unet and efficientnet. Despite the fact that our model is reliable, there are still some defects in it. As a result, we would like to improve on these faults by employing more effective methods. We would also like to make the prediction process more usable by speeding it up. We can see from our model's outputs that there are a few classes that cannot be classified consistently due to a lack of data. If we can obtain more consistent and structured data in the future, we will be able to improve our model's accuracy.

Bibliography

[1] Magda Ruiz et al. “Wind turbine fault detection and classification by means of image texture analysis”. In: *Mechanical Systems and Signal Processing* 107 (2018), pp. 149–167.

[2] Xueqian Bai et al. “A Method of Printing Press Fault Diagnosis Based on Image Texture Information”. In: *China Academic Conference on Printing & Packaging and Media Technology.* Springer. 2016, pp. 837–843.

[3] Feiya Lv et al. “Weighted time series fault diagnosis based on a stacked sparse autoencoder”. In: *Journal of Chemometrics* 31.9 (2017), e2912.

[4] HD Yuan, J Chen, and GM Dong. “Machinery fault diagnosis based on time–frequency images and label consistent K-SVD”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 232.7 (2018), pp. 1317–1330.

[5] Shaohui Mei et al. “Sensor-specific Transfer Learning for Hyperspectral Image Processing”. In: *2019 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp).* IEEE. 2019, pp. 1–4.

[6] Annegreet Van Opbroek et al. “Transfer learning for image segmentation by combining image weighting and kernel learning”. In: *IEEE transactions on medical imaging* 38.1 (2018), pp. 213–224.

[7] Qiang Zheng, Gregory Tastan, and Yong Fan. “Transfer learning for diagnosis of congenital abnormalities of the kidney and urinary tract in children based on ultrasound imaging data”. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018).* IEEE. 2018, pp. 1487–1490.

[8] Amin Taheri-Garavand et al. “An intelligent approach for cooling radiator fault diagnosis based on infrared thermal image processing technique”. In: *Applied Thermal Engineering* 87 (2015), pp. 434–443.

[9] Amin Taheri-Garavanda et al. “An Intelligent Approach for Cooling Radiator Fault Diagnosis Based on Infrared Thermal Image Processing”. In: ().

[10] Yang Wang et al. “A Fault Diagnosis Scheme for Rotating Machinery Using Recurrence Plot and Scale Invariant Feature Transform”. In: *3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019).* Atlantis Press. 2019, pp. 675–681.

[11] Faisal Al Thobiani, Tiedo Tinga, et al. “An approach to fault diagnosis of rotating machinery using the second-order statistical features of thermal images and simplified fuzzy ARTMAP”. In: *Engineering* 9.06 (2017), p. 524.

[12] Md Rifat Shahriar, Tanveer Ahsan, and UiPil Chong. “Fault diagnosis of induction motors utilizing local binary pattern-based texture analysis”. In: *EURASIP Journal on Image and Video Processing* 2013.1 (2013), pp. 1–11.

[13] Md Junayed Hasan, MM Manjurul Islam, and Jong-Myon Kim. “Acoustic spectral imaging and transfer learning for reliable bearing fault diagnosis under variable speed conditions”. In: *Measurement* 138 (2019), pp. 620–631.

[14] Orhan Yaman, Mehmet Karakose, and Erhan Akin. “A Fault Diagnosis Approach for Rail Surface Anomalies Using FPGA in Railways”. In: *International Journal of Applied Mathematics Electronics and Computers* Special Issue-1 (2017), pp. 42–46.

[15] Kyung Ho Sun et al. “Vision-based fault diagnostics using explainable deep learning with class activation maps”. In: *IEEE Access* 8 (2020), pp. 129169–129179.

[16] MachineMFG. *16 Types of Steel Defects*. https://www.machinemfg.com/types-of-steel-defects/. [Online; accessed 19-May-2021]. 2021.

[17] Yang Tian, Mengyu Fu, and Fang Wu. “Steel plates fault diagnosis on the basis of support vector machines”. In: *Neurocomputing* 151 (2015), pp. 296–303.

[18] Kunshan Huang et al. “Spectral–spatial hyperspectral image classification based on KNN”. In: *Sensing and Imaging* 17.1 (2016), pp. 1–13.

[19] Jun Li, José M Bioucas-Dias, and Antonio Plaza. “Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression”. In: *IEEE Geoscience and Remote Sensing Letters* 10.2 (2012), pp. 318–322.

[20] Ned Horning et al. “Random Forests: An algorithm for image classification and generation of continuous fields data sets”. In: *Proceedings of the International Conference on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences, Osaka, Japan*. Vol. 911. 2010.

[21] Nabil Ibtehaz and M Sohel Rahman. “MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation”. In: *Neural Networks* 121 (2020), pp. 74–87.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[23] EfficientNet. *EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling*. https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html/. [Online; accessed 25-May-2021]. 2019.

[24] Robert W Donaldson and Jiaze He. “Instantaneous ultrasound computed tomography using deep convolutional neural networks”. In: *Health Monitoring of Structural and Biological Systems XV*. Vol. 11593. International Society for Optics and Photonics. 2021, p. 1159325.