

Sandpaper Defect Detection

by

Imran Ahmad Shahid

1902008

Master's thesis in Computer Engineering

Academic Supervisor: Johan Lilius

Åbo Akademi University

Faculty of Science and Engineering

Department of Information Technologies

May 2021

CONTENTS

ABSTRACT.....	7
ACKNOWLEDGEMENTS.....	8
CHAPTER 1: INTRODUCTION	9
STRUCTURE OF THE THESIS.....	12
CHAPTER 2: THEORETICAL BACKGROUND.....	13
CURRENT QUALITY CONTROL	16
<i>Advantages of Line scan camera</i>	17
<i>Issues with Line Scan Camera</i>	17
PROPOSAL OF NEW SYSTEM	18
<i>Proposed System Architecture</i>	19
<i>Advantages of New System</i>	19
<i>Issues with Proposed System</i>	20
CHAPTER 3: RESEARCH METHODS	21
TRADITIONAL APPROACH.....	22
DEEP LEARNING	24
<i>Convolutional neural network</i>	26
<i>VGG16</i>	28
CHAPTER 4: DATASET CREATION	29
IMAGES CATEGORIES	30
ISSUES WITH DATASET	31
<i>First batch image issues</i>	32
<i>Second batch image issues</i>	32
PREPROCESSING.....	32
DATASET LABELING.....	33
DATASET SELECTION	34
<i>Dataset Generation</i>	34
CHAPTER 5: IMPLEMENTATION	36
HARDWARE INFORMATION	36
TRADITIONAL APPROACH.....	36
DEEP LEARNING	39
<i>Convolutional neural network</i>	39
<i>VGG16</i>	40
CHAPTER 6: RESULTS AND DISCUSSION	44

TRADITIONAL APPROACH.....	44
DEEP LEARNING	46
<i>Convolutional neural network</i>	46
<i>VGG16</i>	51
CHAPTER 7: CONCLUSION.....	55
REFERENCES.....	57

List of Tables

Table 1 First batch images count	30
Table 2 Hardware specification	37
Table 3 Convolutional neural network hyperparameters	40
Table 4 VGG16 hyperparameters	41
Table 5 Maximum and minimum tile score difference	45
Table 6 Maximum and average tile score difference	45

List of Figures

Figure 1: Cognex In-Sight 9920L	10
Figure 2 Sandpaper grits.....	14
Figure 3 Coating types	14
Figure 4 Sandpaper production line	15
Figure 5 Proposed system architecture.....	19
Figure 6 Traditional approach pipeline	23
Figure 7 Canny hysteresis thresholding	24
Figure 8 Deep learning approach.....	25
Figure 9 Convolution neural network	27
Figure 10 VGG16 architecture.....	28
Figure 11 Sandpaper without faults	30
Figure 12 Sandpaper with faults	30
Figure 13 Side dirt marks	31
Figure 14 Side margins.....	31
Figure 15 Labeling application	33
Figure 16 Score calculation	37
Figure 17 Text edges detection.....	37
Figure 18 All edges	37
Figure 19 Image subtraction operation response	38
Figure 20 Convolutional neural network source code	40
Figure 21 Keras VGG16.....	42
Figure 22 Custom VGG16 source code	42
Figure 23 CNN model loss for dataset v1	47
Figure 24 CNN accuracy for dataset v1	47
Figure 25 CNN model loss for dataset v2	48
Figure 26 CNN accuracy for dataset v2	48

Figure 27 CNN model loss for dataset v3	49
Figure 28 CNN accuracy for dataset v3	49
Figure 29 CNN model loss for dataset v4	50
Figure 30 CNN accuracy for dataset V4	50
Figure 31 Keras VGG16 loss for dataset v2.....	52
Figure 32 Keras VGG16 accuracy for dataset v2	52
Figure 33 Keras VGG16 loss for dataset v4.....	53
Figure 34 Keras VGG16 accuracy for dataset v4	53
Figure 35 Custom VGG16 loss for dataset v4.....	54
Figure 36 Custom VGG16 accuracy for dataset v4	54

ABSTRACT

Sandpaper is an abrasive material used to smoothen, clean, or cut a surface.

Sandpaper is made by gluing grains such as garnet or emery with a coating paper.

Sometimes grains are not glued adequately with a coating paper due to uneven sheet of stearate layer or other production errors that cause markings on the surface of sandpaper. These marks reduce the efficiency of sandpaper. The goal of this research study is to find a method to classify faulty sandpapers.

Currently, a line scan vision camera is deployed on the production line for quality control. This camera has an algorithm to classify sandpaper. However, this approach is not accurate enough for the industrial standards. The goal is to apply computer vision approaches that include deep learning and a traditional method for detecting defects in the sandpaper. Moreover, this study compares the accuracy of the line vision camera, deep learning, and traditional method.

Keywords: Sandpaper, Deep Learning, Machine Vision, Defect Detection, CNN, VGG16, Edge Detection

ACKNOWLEDGEMENTS

I want to thank Johan Lilius for supervising my thesis and for his guidance along the way. I would also like to thank Luca Zelioli and Valentine Soloviev for their help and support during my research studies.

Imran Ahmad Shahid

Turku, May 2021

CHAPTER 1: INTRODUCTION

This chapter will cover the following points:

1. Overview of sandpaper and its manufacturing problems
2. Summary of the thesis
3. Purpose of the thesis
4. Structure of the thesis

Sandpaper is a type of abrasive material used for smoothing, polishing, cleaning, or cutting a surface. It is used individually and commercially in different applications. Sandpaper is made by gluing grains such as garnet or emery with a coating paper. Sometimes, grains are not glued adequately with coating paper due to irregular stearate layer or other production errors. It causes part of sandpaper paper to remain uncoated, which is visible as markings on the surface. These markings reduce the efficiency of sandpaper. Hence the product becomes unusable.

Markings reduce the quality of the sandpaper and create an economic problem for the manufacturer. When a customer finds faulty sandpaper in a roll, they complain, and the manufacturer replaces it with a roll of new sandpaper and discards the roll of faulty sandpaper. It causes manufacturer losses in both monetary and reliability values if they do not detect the errors in sandpaper during the production process.

Since these markings are visually identifiable, computer vision can be a solution to determine sandpaper quality and classification. The production line of sandpaper is continuous; thus, a line scan vision camera is a solution to start. It has a single row of light-sensitive pixels that constantly scan moving objects at a high frequency. Line scan vision cameras also have an algorithm to identify faulty products.

The sandpaper manufacturer deployed a Cognex In-Sight 9902L line vision camera on the production line of sandpaper for quality control. This camera has an algorithm to

detect faulty sandpapers. Figure 1 shows an image of a line camera currently being used in the production line. According to Cognex, an In-Sight 9902L camera has the following characteristics: (In-Sight 9902L Series Features, 2021)



Figure 1: Cognex In-Sight 9920L

1. Cognex In-Sight 9902L is a high-resolution self-contained system suitable for a thorough analysis of long, cylindrical, or continuous moving objects.
2. It produces high-definition 32mp images to detect even the most minor features and defects of the product.
3. The Cognex In-Sight 9920L system only requires a small view of the target part and can also be suitable for the restricted field of view.

Currently deployed quality control classifies sandpaper into acceptable and non-acceptable categories, but it has several issues:

1. Firstly, it is not accurate, and it needs to improve as it misclassifies images.
2. Secondly, its algorithm is immutable and cannot be modified.
3. Lastly, it is expensive and not feasible to deploy it in the whole factory.

This research study aims to find a more reliable solution that can be upgradeable and cheaper than a line scan camera. Initially, the first task was to obtain images of sandpaper from the production line. The manufacturer provided the first dataset of sandpaper images that were classified using a line vision camera. There were several issues with the dataset:

1. Firstly, the classification of images was inaccurate.
2. Secondly, some images have markings on the borders, which are irrelevant to determine sandpaper quality.
3. Lastly, the manufacturer provided images in two batches, and the exposure and contrast of images of both were different, making it a biased dataset.

Later the company provided the second dataset; there were also some issues with it. Firstly, the light condition was different from the images of the first dataset, which caused different contrast and exposure. Besides this, images were not labeled into acceptable and non-acceptable sandpaper.

To start research, preprocessing and classification of data were needed. For ease, an application was developed for image classification. Apart from this, contrast and exposure settings were made on images to make them consistent; along with it, the unnecessary borders were removed because they were causing issues in determining the faulty sandpaper.

In this thesis, two approaches have been used to predict sandpaper quality, the traditional computer vision approach and the deep learning approach.

1. In the traditional approach, edge detection is used to classify sandpaper. Edge detection comprises several mathematical techniques that try to identify points in a digital image at which the image illumination turns distinctly or has discontinuities. The points at which image brightness changes sharply are typically organized into curved line segments termed edges.
2. Deep learning is a subset of machine learning. It learns directly from raw data and improves its predictive accuracy when provided with further data to the algorithm. In deep learning, algorithms try to process the data and find patterns in it.

Observing the results concluded that a deep-learning convolution neural network performs better than VGG16, edge detection, and line scan camera.

Structure of the thesis

The thesis is structured into seven chapters.

1. Chapter 1 presents the purpose, summary, and structure of the thesis.
2. Chapter 2 discusses the theoretical background of sandpaper, the current setup of quality control in the sandpaper production line, its advantages and disadvantages, the proposal of the new system and its architecture, and the benefits of having a proposed design.
3. Chapter 3 describes the research methods used in the thesis.
4. Chapter 4 presents information about preprocessing of the dataset and dataset creation.
5. Chapter 5 includes an implementation of the research methods, including CNN, VGG16, and edge detection.
6. Chapter 6 contains the results and compares accuracy between line camera, edge detection, and CNN models.
7. Chapter 7 concludes the research and provides information regarding possible future work.

CHAPTER 2: THEORETICAL BACKGROUND

The last chapter covered the introduction of sandpaper, the problem in manufacturing sandpaper, a summary of this thesis, and the structure of the document. This chapter will cover the following topics:

1. Theoretical background and application of the sandpaper
2. Current quality control and its advantages and disadvantages
3. The proposed system, its architecture, and benefit over current quality control

Despite the name, sandpaper sheets do not contain sand; it is made from abrasive minerals such as aluminum oxide or garnet glued on the backside of the paper. These minerals have sharp points or edges; thus, sandpaper can be considered a cutting tool like a saw or a chisel. The only difference between sandpaper and chisel is that sandpaper cannot be sharpened. Sandpaper production involves several steps:

1. Firstly, a paper is printed with the brand information and dried with ultraviolet heat.
2. Secondly, it is dipped in resin, which acts as glue for the abrasive.
3. Lastly, grit is conveyed beneath the paper and statically charged, which causes the grit to jump up and embed in the resin glue, creating a sharp edge on the sandpaper.

When the sandpaper is rubbed or pushed across a piece of wood, these abrasive grains cut the shavings out of the surface. It is like dust particles but magnified; it is like a shaving produced by other cutting tools. (Danit Brown, 2020)

The different types of sandpaper have three primary characteristics: its grit, the abrasive material, and the coating. (Types of Sandpaper and Abrasives, 2021)



Figure 2 Sandpaper grits



OPEN COATED



CLOSE COATED

Figure 3 Coating types

1. The grits on sandpaper determine the coarseness on it. Figure 2 shows what grit looks like, and a grit number points to the number of holes per square inch in the screens used to sieve the grains during production: coarser paper has a lower grit number.
2. The coating determines the density of abrasive particles on sandpaper. There are two types of coatings, open-coated and close-coated. Figure 3 shows the structure of these coatings.
 - Sandpaper containing about 70% abrasive grains is known as open-coated, used with power tools and softwoods.

- Sandpaper which is fully covered with abrasive material, is known as close-coated. It works with hand-sanding, hardwood, and metal surfaces.
3. Sandpaper is made from different materials, including aluminum oxide, ceramic, flint, garnet, and silicon carbide.



Figure 4 Sandpaper production line

Mirka is a sandpaper manufacturing company, and they market their product as a new generation of sandpaper for surface finishing professionals. This abrasive Mirka Iridium results from years of development, and it is developed with the latest cutting-edge technology. Iridium is a premium paper abrasive for regular sanding made up of a mixture of ceramic and aluminum oxide grains on a flexible paper with precision coating. It gives excellent results on both hard and soft surfaces and is an ideal choice for professionals in any industry. There are different types of product groups developed by a company for various applications. These abrasives have the following advantages over other sandpapers:

1. It cuts faster, and it takes fewer steps to make the rough surface smooth.
2. It practically repels dust particles; thus, grains stay sharp for a longer time.
3. Iridium works well on both soft and hard surfaces.

During the production of sandpaper, different types of errors occur. These errors reduce the efficiency of sandpaper, which makes it unusable. If these flaws are not

detected at the time of production, it causes reputational and financial loss, as the company will need to replace the faulty sandpaper with the new sandpaper. Multiple reasons can cause markings on the surface of the sandpaper at the time of production: (Common Wide Belt Sander Issues & Solutions, 2021)

1. The sandpaper markings result from the uneven coating of the stearate layer, which is a protective layer that enhances the lifespan of the product.
2. A damaged roller can cause wavy or consistent peaks to appear as valleys on the sandpaper surface. The unsuitable grit combination also causes a wavy texture.
3. Ridges on the surface of the sandpaper are caused due to low belt speed, high pressure of sanding, or impurities in the workpiece that have damaged the abrasive grains.
4. Chatter marks are consistent even lines that can appear on a workpiece after running through a wide belt sander. A worn or incorrect roller, worn or sticky conveyor belt, worn bearing, or a wrong tension pressure can cause chatter marks.

Detection of markings on the surface of the sandpaper is necessary for the manufacturer. The manufacturer has several production lines producing different types of sandpapers. The goal of the manufacturer is to identify whether the product is acceptable or non-acceptable. Currently, a line scan camera is installed in one of the production lines for quality control purposes. It has a built-in algorithm to detect the faults of the product. Figure 3 shows the sandpaper production line. Initially, a line scan camera was installed for quality control because it is simpler to use when having a continuous moving product. Apart from this, an ordinary camera requires configuration to set a reference point to take an image of a continuous object.

Current Quality Control

The company wants to determine the quality of sandpaper, whether produced sandpaper is acceptable or non-acceptable. Currently, the Cognex In-Sight 9902L line scan vision system is installed on one of the production lines, and it has a built-in

algorithm to classify faulty sandpaper. The goal of a company is to deploy quality control on all the production lines. The mechanism of taking a photo in the line camera is different from an ordinary camera and can be described as:

"A line scan camera has a single line of pixels. To build up a two-dimensional image of the object, either the camera or object is moved perpendicular to the pixel line. It might seem like a complicated way to image an object compared with "frame cameras" that take two-dimensional images. However, when the product is enormous, continuously moving, or the task needs perfect or high-resolution imaging, a line scan camera is often a better choice than a frame camera." (Teledyne DALSA, 2015)

Advantages of Line scan camera

The advantages of using a line scan vision system are as follows:

1. An In-Sight 9902L line scan vision system camera has an algorithm to find faulty products, thus there is no need to invest and develop a custom algorithm to determine product quality.
2. A line-scan vision camera is well suited for products that are in continuous motion.
3. Generally, the line scan vision cameras have a higher resolution and image quality than ordinary cameras.
4. The line camera is not required to set the configuration to determine the reference point for taking the images.

Issues with Line Scan Camera

Although the In-Sight 9902L line scan vision system is a good choice as our object is large and continuously moving, this system has the following issues:

1. Line scan vision systems are expensive, and implementing them on all the production lines is not practical.
2. The current quality control using the In-Sight 9902L camera is immutable, and the algorithm of the camera cannot be modified.

3. The accuracy of the line scan vision system is not 100%, and it misclassifies sandpaper.

Proposal of New System

Line scan cameras or manual quality control methods are expensive and inaccurate. The goal is to deploy an alternative solution for quality control and improve the accuracy of current quality control. Thus, it is required to develop an efficient, reliable, upgradeable system that is cheap and better than a line scan camera. By creating this system, a company can deploy it in all the production lines, and it can also be used and adjusted for different types of sandpaper.

For this purpose, a new system is proposed which requires an ordinary camera to install on the production line to take continuous images. Along with it, a custom-build algorithm is needed to process these images and determine whether the produced sandpaper is acceptable or non-acceptable.

This thesis aims to develop an algorithm for the classification of faulty sandpaper using computer vision. For this purpose, two computer vision approaches are used to predict sandpaper quality, the traditional computer vision and the deep learning approach.

1. In the traditional approach, edge detection is used to find the faulty sandpaper. There are various algorithms available for edge detection. In this thesis, Canny edge detection is used to find edges because it is more efficient than other edge detection algorithms.
2. In deep learning computer vision, various neural networks are available to find a pattern of faults in the sandpaper. In this research study, the convolution neural network and VGG16 are used to find flaws in sandpaper.

The proposed system aims to deploy ordinary cameras on the production line, and by using either of these two approaches, an algorithm will run on an edge computing device for classifying sandpaper.

Proposed System Architecture

For the detection of faulty sandpaper, the aim is to place two cameras on a production line that will continuously take images of the finished product. The system will have an edge computing device with an algorithm to detect the faulty sandpaper. The system will process the captured images of both cameras and feed them to the computer vision algorithm, and it will classify the image into acceptable and not-acceptable. Figure 5 imitates the new system architectures.

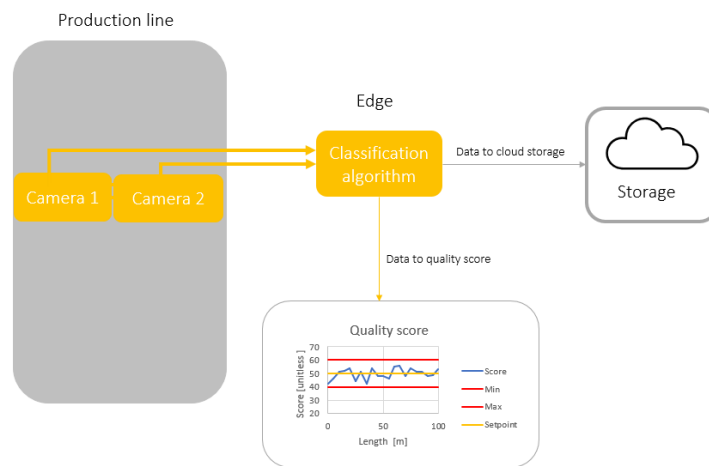


Figure 5 Proposed system architecture

For the quality check of the produced product and making decisions about the future, the system also aims to store images of sandpaper with errors and without errors on the cloud storage. It will help determine the percentage of acceptable and non-acceptable sandpaper, and these insights will be helpful in future decision-making and analyzing the performance of production lines.

Advantages of New System

1. The aim is to classify ten images per second.
2. It will be ten times cheaper than the line scan camera system.
3. It will be more accurate in classifying faulty sandpaper.
4. It will adapt to changes in the requirements.

Issues with Proposed System

1. Ordinary cameras require a configuration or a point of reference to capture the images.
2. Ordinary cameras might not be able to produce high-quality images as line scan vision systems.

CHAPTER 3: RESEARCH METHODS

The previous chapter covered information about sandpaper, its manufacturing material, the current quality control used in the production line of the sandpaper, issues with current quality control, the goal to implement quality control in all the production lines, and a new system proposal and its architecture.

This chapter will discuss computer vision and its two approaches used in this thesis to predict the quality of the sandpaper: the traditional approach and a deep learning computer vision approach.

Computer vision is a field of computer science that deals with understanding digital media, which includes images and videos. The purpose of computer vision is to automate human vision tasks. Computer vision is anything that the human eye can see and identify and even beyond that. In computer vision, computers try to solve a visual problem by making a statistical model of the digital media and finding a pattern.

Computer vision tasks include obtaining the images from the camera, processing the image, such as adjusting exposure, contrast, or cropping the image if required. Later, understanding the digital image by making a statistical model of it. Lastly, analyzing that image data and finding the desired information from the image.

Computer vision has a significant impact on all industries. Moreover, it becomes part of daily lives, as knowingly or unknowingly we are using it daily. Following are some areas of computer vision:

1. In the automotive industry, self-driving cars use computer vision, traffic sign detection, traffic flow analysis, pedestrian detection, and collision avoidance systems.
2. In healthcare, computer vision is helping doctors in the diagnosis of cancer cells, Covid-19, and other illnesses.

3. In agriculture, it is helping farmers in monitoring the crops from insects and diagnosing plant diseases.
4. In banking, it is helping customers to open accounts by taking a selfie and a video call.
5. In retail and manufacturing, it is used in theft detection by analyzing off-limit areas, counting people in the area, and quality management of the product.

In this thesis, two approaches of computer vision are used to classify the defective sandpaper. First is a traditional approach, mainly the color, edges, or corners detection based on image processing algorithms and methods. The second approach is deep learning, in which a neural network learns to mimic human behavior, and it tries to learn the patterns in the data.

The deep learning technique is the latest approach in computer vision, and it has increased performance and efficiency and achieved results that were not possible before with the traditional systems. However, this does not imply that now traditional methods have become obsolete. Various applications still perform better using traditional techniques. Apart from this, today, a hybrid approach is also being used, which improves computer vision performance and solves problems that are not suited for deep learning. A hybrid approach is used in 3D vision systems such as video classification, computer graphics, and robotics. (Niall O'Mahony, 2019)

Traditional Approach

In computer vision, the traditional approach aims to understand the context of the image data. Objects can be recognized, detected, or this approach can resolve other computer vision problems based on the requirements. Several algorithms and techniques are used to extract desired knowledge from the images. Figure 6 shows the traditional computer vision pipeline. (Richmond Alake, 2020)

Features in computer vision are measurable and quantifiable forms of data that represent specific aspects of observation. The traditional approach includes obtaining

the images, extracting desired features, and running a computer vision algorithm to obtain the expected output.

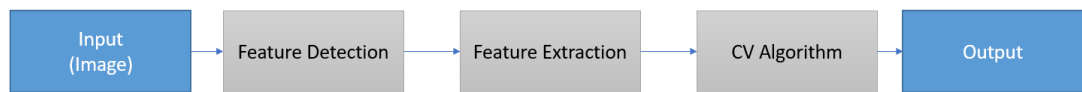


Figure 6 Traditional approach pipeline

Feature detection means finding relevant information or features in the image, such as corners, edges, isolated points, continuous curves, or connected regions. When the input data is too large to be processed, analyzing the data is necessary to remove the redundant and irrelevant data. The selection of the relevant data from input data is called feature extraction.

After the extraction of related features, these are fed to the computer vision algorithm for processing and obtaining the desired output. In this thesis, edge detection is used to find the faulty sandpaper.

Edges are points of the image that are used to represent the information in it. Edge detection is one of the algorithms that can provide some scenic understanding of the images. With the help of combining detected edges, information can be fetched from the image. Various edge detection algorithms are available, such as the Canny edge detection, the Sobel method, and the Fuzzy logic method. In this thesis, the Canny edge detection (J. Canny, 1986) method is used to detect defective sandpaper.

The Canny edge detection is a method to obtain valuable architectural information from different vision objects, and it reduces the amount of data to be processed. It is widely used in many computer vision applications. John F. Canny developed it in 1986. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. The stages of finding out edges are the following: (Concept of Canny edge detection, 2021)

1. First, use a gaussian filter to smooth the image for the removal of noise.

- In the second step, it performs non-maximum suppression, which is applying gradient magnitude thresholding to remove unwanted pixels that may not generate the edge.

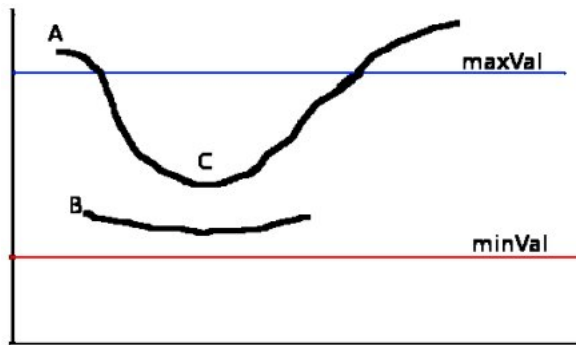


Figure 7 Canny hysteresis thresholding

- The last step is to perform hysteresis thresholding, in which two threshold values, minimum and maximum, are defined. Any edge intensity gradient higher than the maximum threshold is an edge, and lower than the minimum threshold is a non-edge. A value that lies in between is a determined edge or non-edge based on its connectivity. In figure 7, edge A is undoubtedly an edge as it is higher than the maximum threshold. Although edge C is below maximum thresholding, it relates to edge A. Thus it is also an edge. However, edge B is between the maximum and minimum threshold, but it does not connect with any edge that is higher than the maximum threshold. Thus, edge B will be discarded and not considered as an edge.

Deep Learning

Deep learning is a subset of machine learning that imitates the behavior of the human brain, in which it processes the data, tries to create and find patterns in the information for decision making. Its capability of learning patterns is achieving results that were not possible before using traditional methods.

A computer model learns to perform classification and identification tasks directly from image, text, or sound in deep learning. Deep learning models can achieve

accuracy, sometimes surpassing human-level performance. For the training and finding patterns in the dataset, deep learning requires an extensive labeled dataset for training and substantial computing resources such as high-performance GPU to process the data. Figure 8 shows a pipeline of a deep learning algorithm.

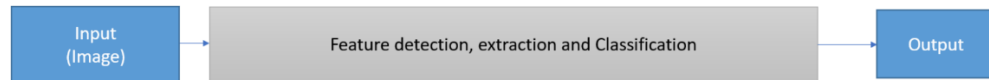


Figure 8 Deep learning approach

Deep learning applications have now become a vital part of our lives. They are used in the following daily life applications:

1. Data from cameras, geo-mapping, and other sensors are used to develop a model for self-driving cars.
2. The internet has now become a primary source of information. Deep learning models help to identify fraud detection and determine phishing websites that might take credit card details.
3. One of the popular applications of deep learning is virtual assistance. Siri, Google Assistant, and Cortana are examples of virtual assistants. Deep learning helps to understand human commands using natural language processing.
4. Deep learning also helps doctors in health care by an immediate, reliable, and quick diagnosis of life-threatening diseases.
5. One standard application of deep learning is visual recognition, which is used by smartphone companies to implement face lock in mobile phones. Apart from this, deep neural networks also help identify faces, events, and backgrounds of images.

Deep learning algorithms require a massive amount of computation resources and information to solve complex issues. In this thesis, CNN and VGG16 models are used to detect defective sandpaper.

In the deep learning models, there are various parameters whose value change improves the learning of the model. These values are determined based on the initial

intuition, and later they are changed according to the model performance. These parameters are called hyperparameters which are as follows:

1. In the deep learning models, a filter is used to slide through the image to determine the features from the image; it is called a kernel. The kernel size defines the field of view of the convolution.
2. Stride is the step taken by a kernel while sliding through the image. Stride one means that a kernel slides through the image, pixel by pixel. Stride two means that a kernel slides through the image by moving two pixels per step.
3. Padding is used to add an extra frame on the boundary of the image to facilitate the kernel.
4. Batch size is a parameter that defines the number of samples processed before the model is updated.
5. Epoch parameters define the number of times the model will work through the whole dataset.
6. An activation function is a parameter that determines the output of the neuron given a set of inputs.
7. Optimizers are the algorithms used to adjust the characteristics of the neural networks, such as the weight of the model and the learning rate, to reduce the loss.

These hyperparameters are essential in determining the performance of the neural network. The accuracy of the model can be improved by altering the value of these hyperparameters and monitoring the behavior change in the model by change in hyperparameters.

Convolutional neural network

A convolutional neural network (CNN) is a class of deep neural networks applied to analyze visual imagery. CNN is a tool for solving the problem of pattern recognition. Figure 3.4 shows an architecture of CNN consisting of convolutional layers, fully connected, and an output layer.

CNN (Mohamed Eelgandy, 2020) obtains images as an input; convolutional layers apply filters on the images to extract features and then feeds features into fully connected layers for classification. CNN (CS231n Convolutional Neural Networks for Visual Recognition, 2021) works as follows:

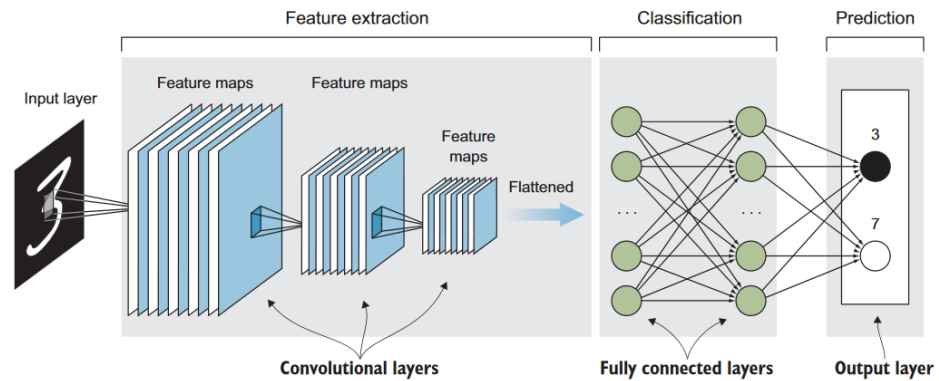


Figure 9 Convolution neural network

1. Input layers hold the raw pixel values of the image.
2. In the convolutional layers, a model tries to find features in the image. This layer consists of a set of learnable filters. Each of these filters is small spatially with width and height. Every filter slides through the pixels of the image and computes a dot product between the entries in the filter and the input at a particular position on the image. At the end of sliding through the whole image, it produces a two-dimensional activation map.

After that, the pooling layer is applied to the feature map to reduce the size. Thus, it reduces the number of parameters to learn and computation performed in the network. Pooling is just like a filter on a feature map. There are two types of pooling:

- Max pooling operation selects a maximum value from the region of the feature map covered by the filter.
- The average pooling operation calculates the average of the elements present in the region of the feature map covered by the filter.

Later, these features are flattened to a vector for inputting it to a fully connected layer. In each convolutional layer, the feature dimension shrinks, and the depth

of features increases until a model has a long array of features in the last layer of feature extraction.

3. The flattened feature vector is fed to the fully connected layers to classify the extracted features of the image. Each fully connected layer is made up of neurons, where each neuron is fully connected with the neurons of the previous layer of the network. Neuron value is computed by a dot product of their weights with the input followed by a bias offset.
4. The last fully-connected layer of the network is the output layer, and in classification settings, it represents the class score. The neural network fires the node that represents the correct prediction of the image.

VGG16

VGG16 is a variant of convolution neural network proposed by K. Simonyan and A. Zisserman from the University of Oxford (Karen Simonyan, Andrew Zisserman, 2015). This architecture was runner-up in ILSVRC 2014. Its main contribution was in showing that the depth of the network is a critical component for good performance.

VGG16 contains 16 CONV/FC layers and, appealingly, features a highly consistent architecture that only performs 3x3 convolutions and 2x2 pooling from the beginning to the end. This architecture consists of 13 convolution layers, 2 FC (fully connected layers), followed by an output layer (Rohit Thakur, 2019). Figure 3.5 shows the architecture of VGG16. The model loads a set of weights pre-trained on ImageNet.

The model achieves 92.7% test accuracy in ImageNet, a dataset of over 14 million images belonging to 1000 classes. The default input size for the VGG16 model is 224 x 224 pixels with three channels for RGB images.

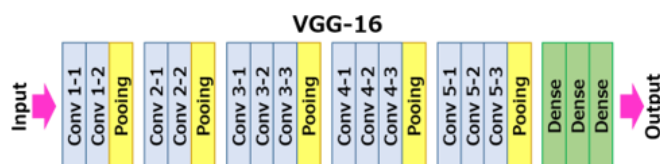


Figure 10 VGG16 architecture

CHAPTER 4: DATASET CREATION

The previous chapter explained the working principles of computer vision and its applications in daily life. Moreover, computer vision approaches used in this research study to predict sandpaper quality were discussed.

This chapter highlights the following information:

1. Size and structure of images
2. Preprocessing performed on the images to make them consistent
3. A custom application for image labeling

The sandpaper manufacturing company provided a dataset of images in multiple batches. The first batch of images was taken from a line scan vision system installed on the production line. Classifications of acceptable and non-acceptable images of the first batch were done by the built-in algorithm of the line vision system. Images from the first batch were RGB images. Classifications of the images were not accurate.

The production line of the sandpaper is long, and one camera cannot capture the whole viewpoint. Thus, two cameras were installed on the production line to capture the complete view. The first batch consisted of labeled images from both cameras. Table 1 represents the count of the first batch of images from both cameras.

The second batch included greyscaled images of multiple product types. These images were not classified into acceptable and non-acceptable groups. As the images from the second batch were taken from an ordinary camera, it was required to classify these images into acceptable and non-acceptable groups. This classification was performed manually.

Images Categories

There are two categories of images:

1. Images have markings
2. Images that do not have any markings

Table 1 First batch images count

Camera	Number of good images	Number of bad images
Camera 1	2344	1364
Camera 2	2533	909

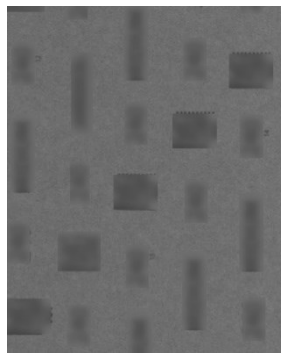


Figure 11 Sandpaper without faults

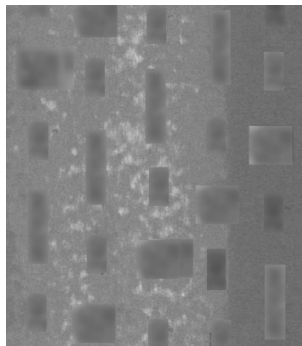


Figure 12 Sandpaper with faults

Faulty sandpaper is the one that has markings on it. Figure 12 shows faulty sandpaper that has markings on the surface; because grains are not glued adequately with coating.

Figure 11 shows sandpaper that does not have any markings on it; hence, it is acceptable.

Issues with Dataset

During the analysis of the dataset of images, several issues were found in both batches of the dataset. Issues of the first batch and second batch are as follows:

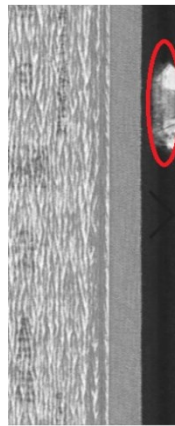


Figure 13 Side dirt marks

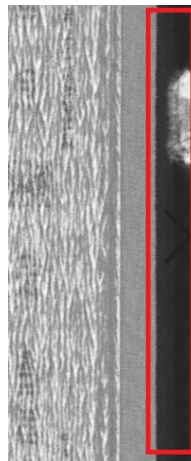


Figure 14 Side margins

First batch image issues

1. An algorithm of a line scan camera is used to classify sandpaper images, but it was not accurate.
2. The images from the first camera contained side margins, which were irrelevant for classifying the images. Figure 14 shows an image from the first camera that has a margin on it.
3. The images from the first camera also contained some dirt marks near the margins, and these dirt marks were also irrelevant to determine faulty sandpaper. Figure 13 shows an image of these dirt marks.

Computer vision models did not achieve high accuracy with the first batch of images. After analysis of the dataset, it turned out that these margins and marks were causing the issues. Moreover, misclassified images were also reducing the performance of the computer vision models.

Second batch image issues

1. Ordinary cameras were used to capture the images from the production line. These cameras do not have an algorithm to classify sandpaper. Thus, images from the second batch were not labeled.
2. Images from the second batch were shot in different light conditions. Thus, they had different exposure and contrast as compared to the first batch.

Preprocessing

As discussed in the last section, there were issues with both batches of images. To make data consistent, preprocessing was required on the images. The steps that have been performed on the images to remove the inconsistencies in the dataset are as follows:

1. The side margins and dirt marks were irrelevant to determine the quality of the sandpaper. Thus, 148 pixels were removed from the left side of the images.
2. Images from the first batch were RGB. These images were converted to greyscale to reduce the size of the dataset and make it consistent with the second batch of images.
3. Due to the difference in light conditions, the second batch had different exposure and contrast from the first batch. Thus, the exposure and contrast of the second batch were changed to make them identical to the first batch of images.

Dataset labeling

After preprocessing, the next task was to classify the images into acceptable and non-acceptable groups. Although the images from the first batch were labeled using a line scan camera, they were not accurate, and relabeling was required.

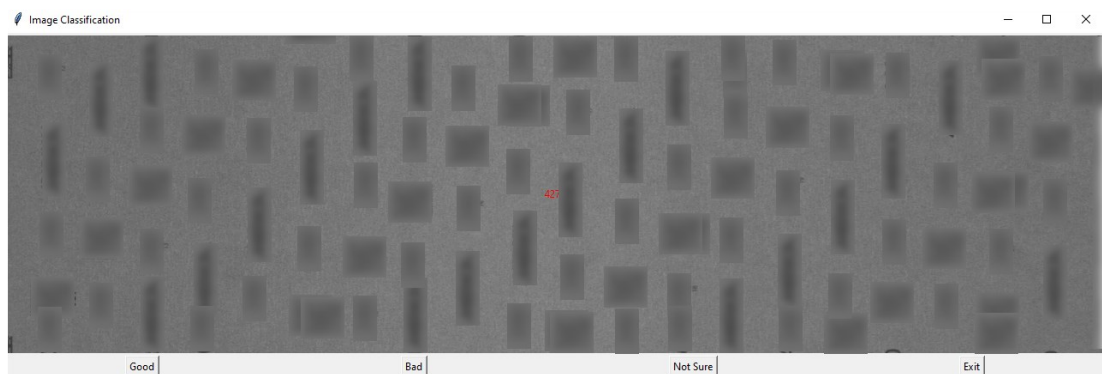


Figure 15 Labeling application

An application was developed to classify and copy images into their respective group folder. Figure 4.5 shows an interface of the application.

The following fields are required in the application:

1. Input directory path of sandpaper images
2. Output directory path of acceptable sandpaper images
3. Output directory path of non-acceptable sandpaper images

It loads all the images from the input directory and the existing images from the output directories. Application only processes images that are not available in the output directories.

The application loads all the images and displays them one by one for classification. The user is required to press the number key "1" for assigning an acceptable label to the image and the number key "2" for setting an unacceptable label to the image, followed by the "enter" key for confirmation. Upon confirmation, the application moves the current image into the chosen class directory and displays the next image.

This application helped label 4000 images from both batches, half of which were acceptable and the other half of which were non-acceptable.

Dataset Selection

Image selection for the training, validation, and testing was made by two experiments:

1. In experiment one, data selection for testing, validation, and training was made by selecting consecutive images. Experiment one gave less than 50% accuracy in model prediction.
2. In experiment two, data selection for testing and validation was made by selecting every 10th image from the dataset, and the remaining images were used for training. This experiment removed biased data and created diversity in the training, validation, and testing datasets.

Dataset Generation

There were four datasets made within the framework of this thesis for training the models. These datasets were created from the images of the first and second batches. Details of these datasets are as follows:

1. Dataset v1 consisted of images provided by the manufacturer before the actual dataset containing 100 images.
2. Dataset v2 consisted of images classified by a line vision camera.
3. Dataset v3 consisted of images that were classified using an edge detector.
4. Dataset v4 consisted of images that were manually labeled.

CHAPTER 5: IMPLEMENTATION

Chapter 4 discussed the dataset information, preprocessing on the dataset to make it consistent and unbiased. It also discussed an application that helped in labeling sandpaper images. It also highlighted the process of selecting images for training, validation, and testing.

This chapter will discuss the hardware information used to perform the research and implementation of the two approaches of computer vision: the traditional approach and the deep learning computer vision approach.

Hardware Information

Computer vision requires an extensive amount of computational power for executing and training deep neural networks. Table 2 shows the hardware specification of the computer that was used in this study. The neural networks were trained on GPU, as CPU takes 10x more time in training than GPU.

Traditional Approach

As discussed in the earlier chapter, the Canny algorithm was used for edge detection in this thesis to detect faulty sandpaper. The Following libraries and tools were used:

1. Python was used as a programming language
2. The Jupyter notebook was used as a development IDE
3. OpenCV library was used for the implementation of the Canny algorithm.
(Concept of Canny edge detection, 2019)

In this approach, two filters were applied to the images to calculate the damage that is called score. Figure 16 shows a source code for calculating a score of the sandpaper images using a canny edge detector, and the following steps were performed for the calculation of the score:

Table 2 Hardware specification

Name of Hardware	Hardware Specifications
CPU	Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz
CPU Cores	6
GPU	Quadro RTX 4000, 8 GB GDDR6 memory
Memory	48 GB DDR4 2666 MT/s

```
img = cv2.imread(filePath)
text_edges = cv2.Canny(img,100,150)
all_edges = cv2.Canny(img,30,110)
edgeImages = all_edges - text_edges
score = cv2.countNonZero(edgeImages)
```

Figure 16 Score calculation



Figure 17 Text edges detection



Figure 18 All edges

1. The first filter finds all text edges in the image. Figure 17 shows the result of this filter on the sandpaper image.



Figure 19 Image subtraction operation response

2. The second filter finds all the edges in the image, including the text edges. Figure 18 shows the result of this filter on the sandpaper image.
3. Later, a pixel subtraction operation was performed on the responses of both filters to remove the text from the images. Figure 19 shows the result of the image subtraction operation.

Pixel subtracting operation removes all the text and brand information from the image; after that, for checking the damage, non-zero numbers are counted in the image response to determine the score of the image. This score indicates the level of damage or faults, and by setting the threshold values, it is possible to determine which of the sandpaper is with or without an error.

The threshold setting on the score gave insights about the sandpaper, but it was not enough because sandpapers have different types of defects. Some markings are minor, and it is difficult to decide the quality with only the score parameter. It was required to calculate other statistical parameters that can help determine sandpaper quality along with a score. Two parameters were calculated:

1. The maximum difference from average, in which an image resulting from the all edges filter was divided into six identical tiles. The non-zero number was calculated for each tile. Afterward, two statistical values were calculated: the average score and the maximum score. Lastly, the difference between the maximum and the average was calculated.

2. The difference between maximum and minimum score, in which an image resulting from the all edges filter was divided into six identical tiles. The non-zero number was calculated for each tile. Afterward, two values were determined: the minimum and maximum values, and lastly, the difference between maximum and minimum values was calculated.

The score was not enough to determine the minor faults in the sandpaper images. These two parameters were calculated. Later, setting a threshold value on these parameters and a score value helped identify the faulty sandpapers.

Deep Learning

In deep learning, two models were used to determine the quality of sandpaper, CNN and VGG16. The following tools and libraries were used in the deep learning approach:

1. Python programming language
2. Jupyter notebook as a development IDE
3. Tensorflow library is used with Keras in the backend. Keras is a library that provides a python interface for an artificial neural network.

Convolutional neural network

For the prediction of sandpaper quality, a simple convolutional neural network was developed that consisted of three convolutional layers, two fully connected, and one output layer. Later, different alterations were performed to obtain high accuracy. These alterations include adding convolutional layers and tuning the hyperparameters of the model.

Lastly, a model was developed that was producing the expected accuracy. It consisted of five convolutional layers, three fully connected layers, and one output layer. Figure

20 shows the source code of the CNN model, and Table 3 describes the hyperparameter used for the CNN model.

Table 3 Convolutional neural network hyperparameters

Parameters	Value
Convolutional layers	5
Fully Connected Layers	3
Output layer	1
Batch Size	2
Optimizer	Rmsprop
Image Size	1900 X 550

```

model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(3,3), padding='same', strides=2, activation='relu', input_shape=(IMG_HEIGHT, IMG_WIDTH, 1)))
model.add(MaxPooling2D(pool_size=2))

model.add(Conv2D(filters=64, kernel_size=(3,3), padding='same', activation='relu', strides=2))
model.add(MaxPooling2D(pool_size=2))

model.add(Conv2D(filters=128, kernel_size=(3,3), padding='same', activation='relu', strides=2))

model.add(Conv2D(filters=256, kernel_size=(3,3), padding='same', activation='relu', strides=2))

model.add(Conv2D(filters=512, kernel_size=(3,3), padding='same', activation='relu', strides=2))

model.add(Dropout(0.3))
model.add(Flatten())

model.add(Dense(2048, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(512, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(3, activation='softmax'))

model.compile(optimizer='RMSprop', loss='categorical_crossentropy', metrics=['accuracy'])

```

Figure 20 Convolutional neural network source code

VGG16

The Keras built-in VGG16 (Keras VGG, 2021) implementation was used to determine sandpaper quality. Keras VGG16 neural network consists of thirteen convolutional

layers, two fully connected layers, and one output layer. Table 4 describes the hyperparameter used for the Keras VGG16 model.

Table 4 VGG16 hyperparameters

Parameters	Value
Convolutional layers	13
Fully Connected Layers	2
Output layer	1
Batch Size	1
Optimizer	Rmsprop
Image Size	1900 X 550

There were different types of errors in the sandpaper. Cropping them to reduce size was not an option; because faulty sandpaper might have a portion that has no faults. Thus, cutting an image into multiple tiles was not a choice.

For training the VGG16 model, the images in the dataset were reduced to 475 x 140 pixels, which is one-fourth the size of the original images. Initially, the VGG16 model was trained on dataset v2. It did not produce high accuracy. Upon investigation, two issues were found that caused low precision of the model:

1. This model could not detect an image having minor markings or errors, as shrinking the size of the images made it more difficult to detect those faults.
2. The image classification of the line scan camera was not accurate, as it misclassified the images that made the model struggle to detect the faults.

Later tuning and optimization of hyperparameters of VGG16 have made it possible for a model to run with the original image size:

1. Reduce the batch size of the model
2. Reduce the number of dense layers
3. Reduce the number of neurons in fully connected dense layers

Figure 21 contains a code for the Keras VGG16 model, which was trained to predict the sandpaper quality. After running the code on the original resolution, it still did not achieve significant accuracy.

```
base_model = VGG16(include_top=False, weights='imagenet', input_shape=(1900, 550,
3))
inputs = tf.keras.Input(shape=(1900, 550, 3))
bm = base_model(inputs, training=False)
flatten_layer = Flatten()(bm)
dense_1 = Dense(64, activation="relu",
kernel_initializer='he_uniform')(flatten_layer)
dropout_1 = Dropout(0.3)(dense_1)
dense_2 = Dense(32, activation="relu",
kernel_initializer='he_uniform')(dropout_1)
dropout_2 = Dropout(0.3)(dense_2)
outputs = Dense(1, activation='sigmoid')(dropout_2)
model = tf.keras.Model(inputs, outputs)
```

Figure 21 Keras VGG16

```
filters = 32
model = Sequential()
model.add(Conv2D(input_shape=(1900,550,1),filters=filters,kernel_size=(3,3),padding="same", activation="relu"))
model.add(Conv2D(filters=filters,kernel_size=(3,3),padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=2*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=2*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=4*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=4*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=4*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=8*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=8*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=8*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=16*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=16*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=16*filters, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Flatten())
model.add(Dense(units=256,activation="relu"))
model.add(Dense(units=128,activation="relu"))
model.add(Dense(units=3, activation="softmax"))

model.compile(optimizer="Rmsprop", loss="categorical_crossentropy", metrics=['accuracy'])
```

Figure 22 Custom VGG16 source code

After that, dataset v4 was used to train the neural network. This dataset contained the greyscale images, and VGG16 works only on RGB images. The Keras VGG16 model creates a dummy variable if it is trained on the greyscale images. Thus, a custom CNN model was developed on the architecture of VGG16, having the same convolutional, fully connected, and output layer. Figure 22 shows a source code of a custom VGG16 neural network.

The dataset v4 was used for the training of custom-made VGG16. This model still could not produce satisfactory accuracy and achieved almost the same accuracy as Keras VGG16. However, hyperparameter tuning was performed to attain higher accuracy, like increasing the filters, changing the kernel size, changing the number of dense layers and neurons. However, this network still did not achieve satisfactory accuracy.

CHAPTER 6: RESULTS AND DISCUSSION

The last chapter discussed the hardware used in the research and the implementation of the traditional computer vision and deep learning approaches.

This chapter will discuss the outcome of both approaches of computer vision used in this thesis. Besides this, it will also discuss all the experiments done to achieve these results.

Traditional Approach

In the traditional approach, the image score was calculated to determine the faultiness in the sandpaper. The threshold value was set to determine the quality. It was determined that images with a score of more than 2500 and less than 5000 have no faults, and images having scored between 5000 and 12,000 have minor faults in them, and scores beyond 12,000 have significant defects.

Besides this, two statistical parameters were also calculated to use them with a score to determine sandpaper quality. The first one is the difference between the maximum and minimum scores of the tiles. The second is the difference between the maximum and average scores of the tiles.

The difference between the maximum and minimum tile score was calculated for acceptable and non-acceptable sandpapers in the first statistical parameter. Later, an average and standard deviation of results were calculated for setting the threshold value. Table 4 shows the average and standard deviation of the first statistical parameter for both categories of images.

The difference between the maximum and average tile score was calculated for both acceptable and non-acceptable sandpaper in the second statistical parameter. Later, an

average and standard deviation of results were calculated for setting the threshold value. Table 5 shows the average and standard deviation of the second statistical parameter for both categories of images.

Table 5 Maximum and minimum tile score difference

Category	Average	Standard Deviation
Acceptable Sandpaper	2126	536
Faulty Sandpaper	4834	5576

Table 6 Maximum and average tile score difference

Category	Average	Standard Deviation
Acceptable Sandpaper	1203	374
Faulty Sandpaper	2623	3221

After analysis, it was determined the first statistical parameter is more efficient than the second statistical parameter. Thus, the first parameter was used along with the image score to determine the sandpaper quality. Steps that were used to assess the quality of sandpaper via edge detector are as follow:

1. Firstly, the score of the images was calculated. If the score was between 2500 and 5000, it was an acceptable image, and if the score was above 12,000, it was faulty sandpaper.
2. If the score was between 5000 and 12000, then the first statistical parameter was calculated, which is the difference between the maximum and minimum tile score. If the difference was in the average range of acceptable sandpaper, it was marked as acceptable. Otherwise, it was marked as faulty sandpaper.

The accuracy achieved from this method was 75%. There were two issues found with the edge detection approach:

1. Firstly, it struggled with images having minor faults and a score between 5000 and 12,000 even though an extra statistical parameter was used along with a score.

2. Edge detection was used to classify the first batch of images, as the light conditions were different in the second batch. Thus, the score of the second batch was different from the first batch. It was found that the score was dependent on the exposure and contrast of the images. Thus, this method was not suitable for detecting sandpaper quality when having different light conditions.

Deep Learning

In the deep learning approach, CNN and VGG16 neural networks were used to determine faulty sandpaper. Obtained accuracy of these models for the datasets is as follows:

Convolutional neural network

The convolutional neural network was trained for four datasets in this research study, and it produced different accuracy for every dataset.

Dataset v1

Initially, the company provided a hundred images for testing before the actual batches of datasets; twenty images were faulty sandpaper, and eighty were without faults. Due to the restrictions of Covid-19, it was not possible to access the factory at that time.

The CNN model trained on these images predicted 80% of the images as acceptable, because it overfits due to insufficient training data and could not distinguish between acceptable and unacceptable sandpaper. Although hyperparameters such as an optimizer, number of convolution layers, batch size, output function, number of the kernels, and kernel size were tuned and altered to obtain better outcomes, it did not work out.

Figure 23 shows the CNN model loss graph on dataset v1. Figure 24 shows the obtained accuracy for the model for this dataset. It achieved almost 80% accuracy, but

as the data were insufficient for training and the ratio of acceptable and unacceptable images was not equal, it failed to determine the faulty sandpaper.

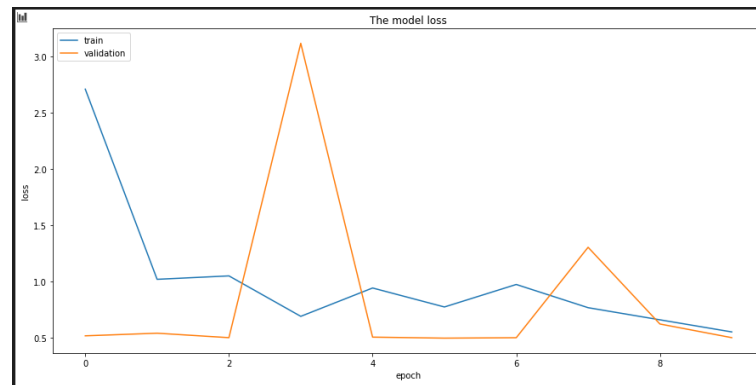


Figure 23 CNN model loss for dataset v1

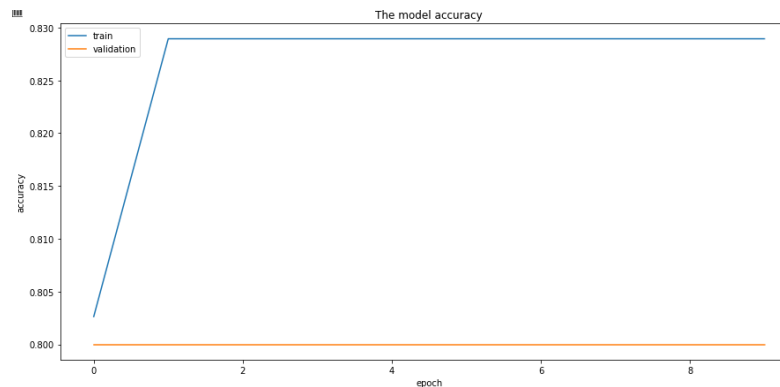


Figure 24 CNN accuracy for dataset v1

Dataset v2

A line-scan camera classified the first batch images. It showed better results in determining sandpaper quality. The alteration and tuning of hyperparameters were done on this model, and it was concluded that the Adams optimizer (Keras Adam Optimizer Implementation, 2021) does not produce good results. Thus, the Rmsprop optimizer (Keras Rmsprop Optimizer Implementation) was used to predict the sandpaper quality. The model successfully predicted the quality of the sandpaper by changing hyperparameter values. It was able to distinguish sandpaper which has significant faults, and sandpaper having no flaws. However, it struggled to identify minor errors in the sandpaper. The model struggled for two reasons:

1. The first batch images were not classified accurately by a line scan camera.
2. The size of the dataset was insufficient to train the model, although it is bigger than the image dataset v1.

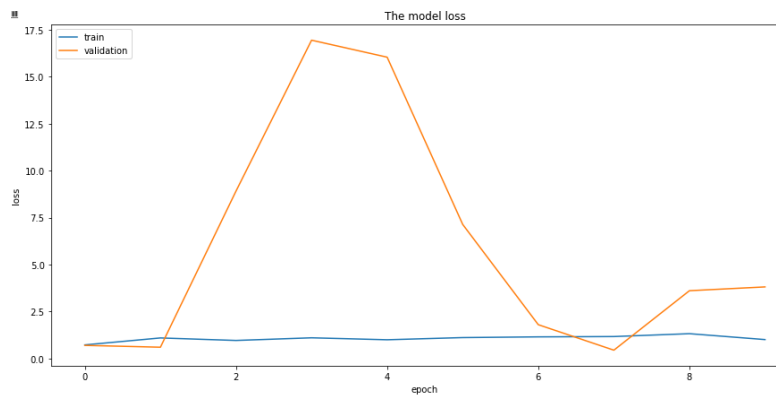


Figure 25 CNN model loss for dataset v2

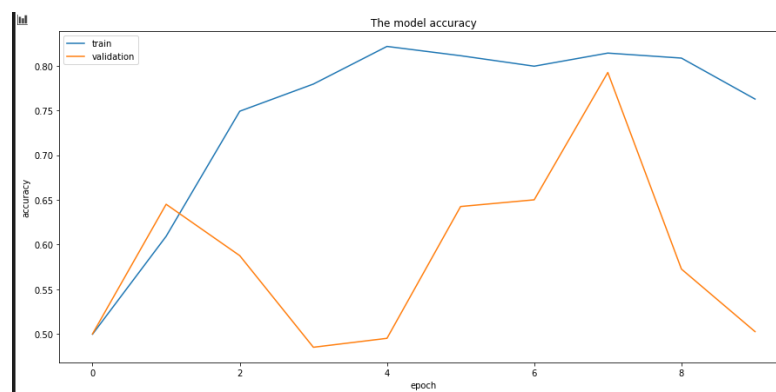


Figure 26 CNN accuracy for dataset v2

Figure 25 shows the model loss graph on the v2 images dataset. The diagram shows that the model was learning throughout the training, and the loss function decreased in every iteration.

Figure 26 describes the obtained accuracy of the model for the v2 dataset of images. The accuracy achieved by this dataset was 76%. However due to insufficient dataset size and misclassified images, it could not produce a higher accuracy.

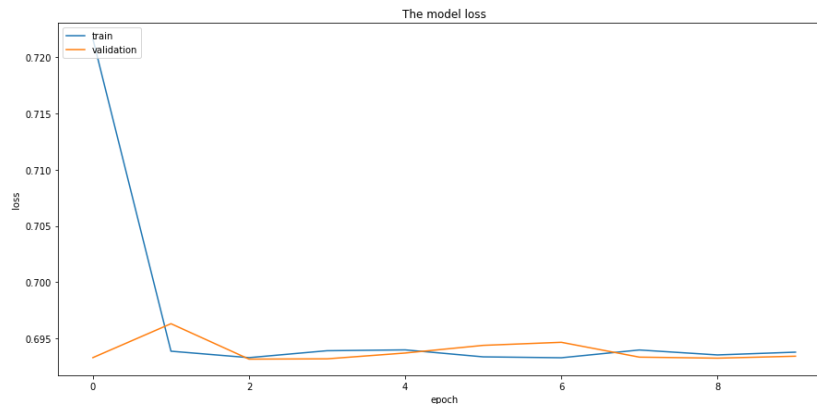


Figure 27 CNN model loss for dataset v3

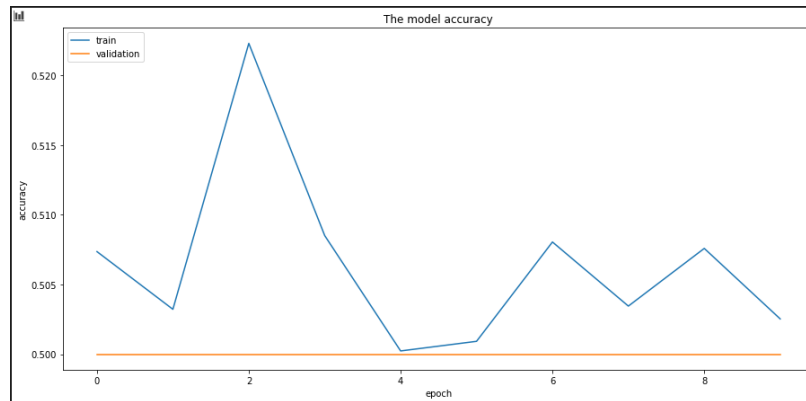


Figure 28 CNN accuracy for dataset v3

Dataset v3

The image dataset v3 was prepared using a classification done by canny edge detection. This dataset made by a traditional approach was not accurate. Thus it did not help to gain higher accuracy. It dropped the model accuracy as compared to image dataset v2. Although tuning of hyperparameters had been done to achieve better results, it did not work out, and the model still struggled to identify the quality of sandpaper.

Figure 27 shows the model loss graph of the CNN model on the dataset v3. This graph shows that models failed to reduce loss function throughout the training and did not learn any valuable features.

Figure 28 shows the obtained accuracy for the model for the dataset v3. The training accuracy was 52%, and validation accuracy was 50%. As the model could not reduce loss function, it could not improve the precision as well, and throughout the training, it remained constant.

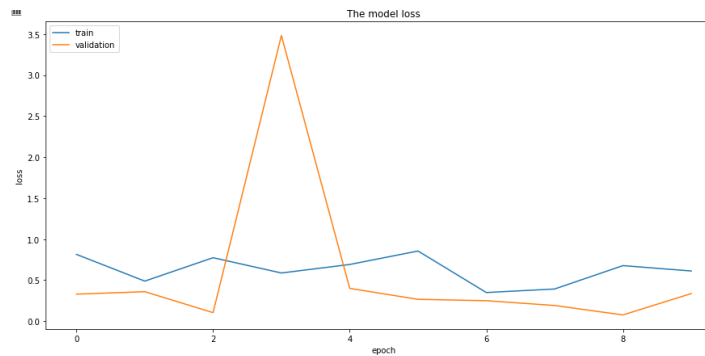


Figure 29 CNN model loss for dataset v4

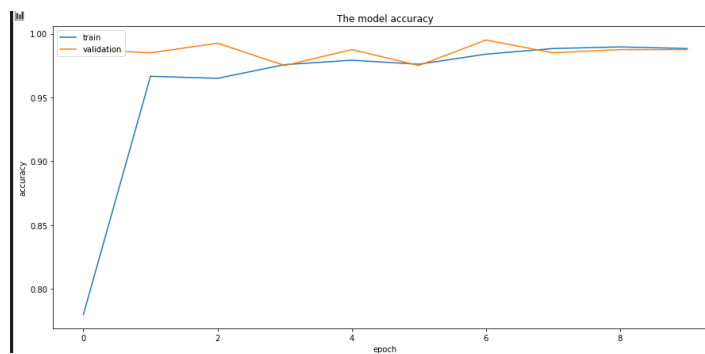


Figure 30 CNN accuracy for dataset V4

Dataset v4

The images in dataset v4 were labeled manually. The CNN model produced high accuracy in predicting sandpaper quality on a manually classified dataset.

Figure 29 shows the model loss graph of the CNN model on the dataset v4. The diagram shows that the model was learning throughout the training process, and the loss function continuously decreased while training the model.

Figure 30 indicates the obtained accuracy for the model for the dataset v4. This dataset significantly increased the model accuracy. It achieved more than 95% accuracy in

both training and validation. It means the model was able to recognize and learn the features of the sandpaper. The model was able to identify both acceptable and unacceptable sandpaper images.

Besides this, it was a requirement to check whether one model can be implemented on every type of sandpaper produced by the manufacture. Thus, for testing, the model trained on image dataset v4 was used to classify other types of sandpaper; However, it did not predict those with high accuracy as it struggled with faulty sandpaper images. Still, it gave a better result and was able to predict acceptable photos of other types of sandpaper.

VGG16

In this thesis, Keras built-in VGG16 and a custom CNN model developed on the architecture of VGG16 were used to identify faulty sandpaper.

Keras VGG16 for dataset v2

Initially, dataset v2, classified with a line scan camera, was used to train the Keras VGG16 model. As discussed earlier, for the VGG16, the image size was reduced. The model was not able to achieve significant accuracy and struggled to determine the sandpaper quality. It happened for two reasons:

1. The image size was reduced; hence that the model could not identify minor faults in the sandpaper images.
2. Images that were classified by a line scan camera were not accurate.

Figure 31 indicates the loss of the function throughout the training, and the graph suggests that the model was not able to learn throughout the training.

Figure 32 shows the achieved accuracy of the model. The model achieved 49% accuracy in training and 50% accuracy in the validation of the model. The accuracy remained constant throughout the model training, which tells that it did not learn anything while training.

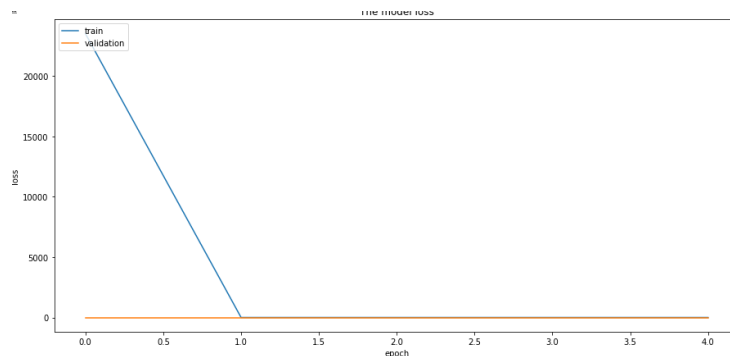


Figure 31 Keras VGG16 loss for dataset v2

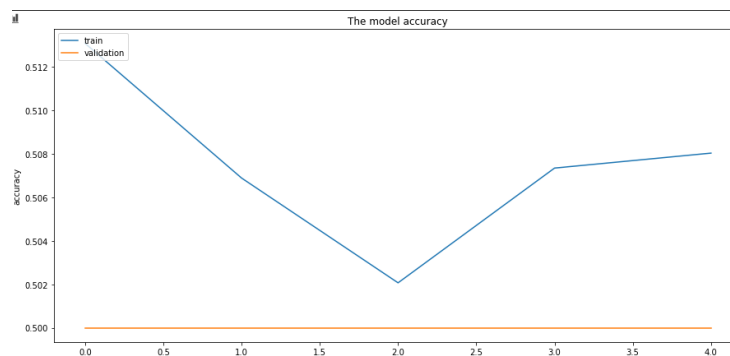


Figure 32 Keras VGG16 accuracy for dataset v2

Keras VGG16 for dataset v4

The dataset v4 was manually labeled. As the dataset was in the greyscale and VGG16 supports only RGB images, it creates a dummy variable if trained on a greyscale image. Thus, it did not achieve any significant results and could not learn the features in the images of sandpaper.

Figure 33 shows the graph of the model loss function. The charts indicate that the model could not learn any significant features while training and struggled to reduce the loss function. After some iterations, the loss function became constant.

Figure 34 determines the accuracy achieved by the model on dataset v4. The graph tells that the model struggled to obtain higher accuracy and did not achieve significant

accuracy. It achieved 50% accuracy, because it could not learn the features and could not reduce the loss.

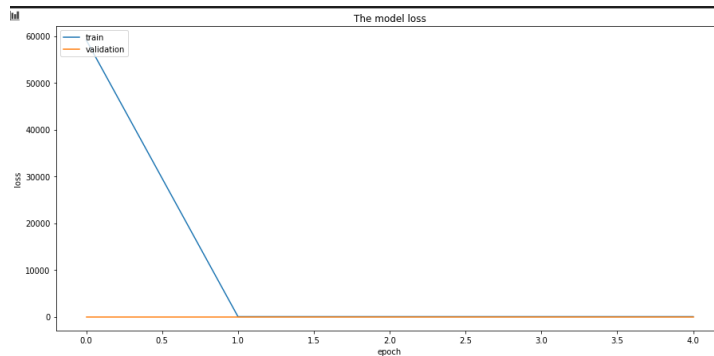


Figure 33 Keras VGG16 loss for dataset v4

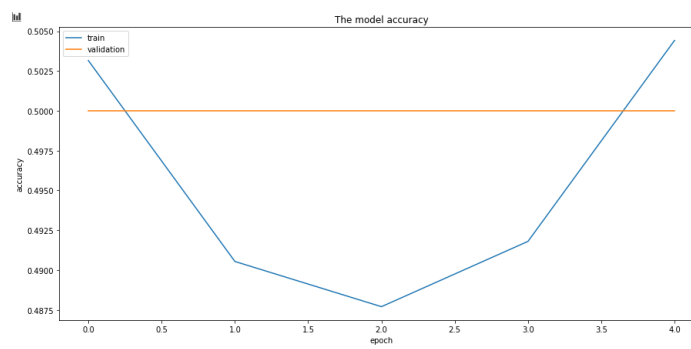


Figure 34 Keras VGG16 accuracy for dataset v4

Custom VGG16 for dataset v4

The custom neural network was developed on the architecture of VGG16 that can work on greyscale images. The dataset v4 was used to train this model. It produced better results than Keras VGG16 but still could not achieve any significant results.

Figure 35 shows the graph of the model loss function. The charts indicate that the model could not learn any significant features while training and struggled to reduce the loss function. It resembled Keras VGG16 and showed that the loss function became constant after a couple of iterations.

Figure 36 shows the accuracy achieved by the model on dataset v4. The graph tells that this model also struggled to obtain higher accuracy but still did not achieve

significant results. It reached 52% accuracy, which is better than Keras VGG16, but the model could not learn the features and could not reduce the loss.

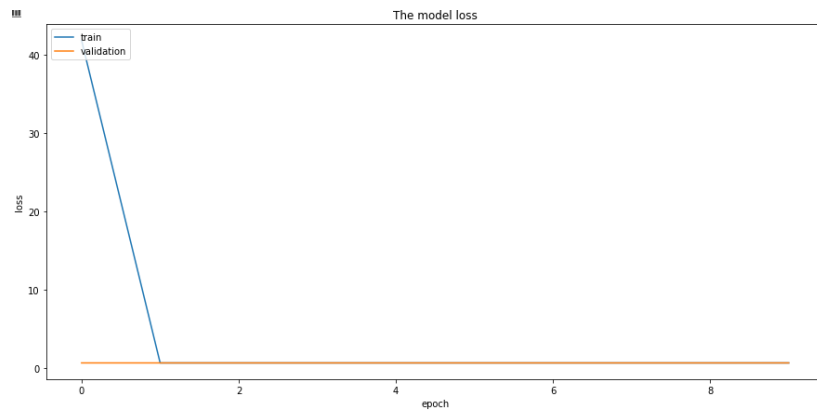


Figure 35 Custom VGG16 loss for dataset v4

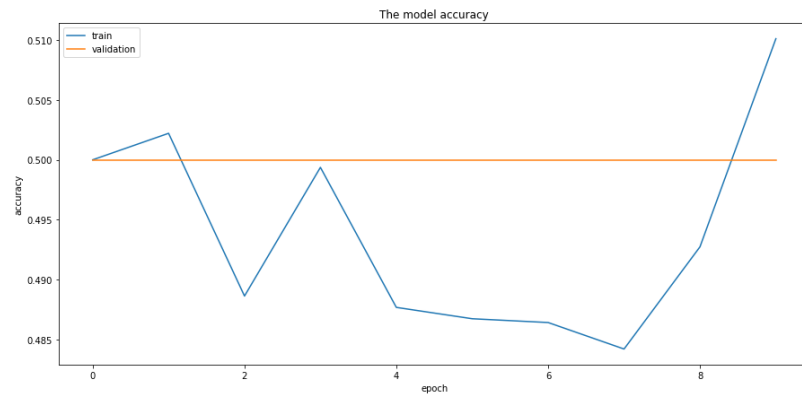


Figure 36 Custom VGG16 accuracy for dataset v4

CHAPTER 7: CONCLUSION

The last chapter discussed the outcome of both approaches of computer vision used in the thesis and the experiments conducted throughout the research study.

This chapter will discuss the conclusion of the thesis, accuracy comparison of the deep learning computer vision algorithms, edge detection, and line scan camera, and future work that needs to be done.

In this thesis, two approaches were used to classify faulty sandpaper. Firstly, the traditional approach was used in which the edge detection method was used to classify the sandpaper. It achieved 75% accuracy in determining sandpaper quality, but it has three problems:

1. It struggled to identify the images with minor faults, and it only classified the images that had significant flaws or images that did not have any defects.
2. This approach was not satisfactory because it did not classify the images that had different light conditions.
3. By using this approach, the accuracy of the model could not be modified or enhanced in the future. The accuracy would remain constant.

The VGG16 neural network was not able to produce any significant results. It was observed that both models: Keras built-in VGG16 and the custom CNN made upon the architecture of VGG16, did not learn the features from the sandpaper images and could not achieve higher accuracy. Both achieved nearly 50% accuracy, which is not significant.

Lastly, the convolutional neural network produced significant results in the determination of faulty sandpaper. However, this model struggled with a dataset classified by a line vision camera and an edge detection technique. However, when

trained on a manually labeled dataset, it produced 95% accuracy. It was able to learn the features in the sandpaper images and successfully identify faulty sandpaper.

Besides this, it was required to decide whether one neural network can assess different types of sandpaper. It was concluded that one neural network was not enough to determine the quality of different types of sandpaper. The neural network must be trained for each type of sandpaper separately to identify faulty sandpaper.

Analysis of the dataset and this research study suggests that sandpaper can be classified into three categories:

1. The images having no faults
2. The images having minor flaws or markings on them
3. The images having significant markings

The images with minor imperfections can be classified as sellable sandpaper if it does not affect the usage. Thus, future work includes separating the dataset into three categories, acceptable, images with minor faults, and images with significant faults. After separating the dataset, the algorithm should be trained on this dataset to find defective sandpaper images.

REFERENCES

Richmond Alake (2020). A Beginner's Guide To Computer Vision. Available at: <https://towardsdatascience.com/a-beginners-guide-to-computer-vision-dca81b0e94b4> [Accessed May. 2021].

Danit Brown (2020). The Secrets Of Sandpaper. Available at: <https://indianapublicmedia.org/amomentofscience/the-secrets-of-sandpaper.php> [Accessed May. 2021].

Concept of Canny edge detection. 2021. Available at: https://docs.opencv.org/master/da/d22/tutorial_py_canny.html [Accessed May. 2021].

CS231n Convolutional Neural Networks for Visual Recognition. (2021). Available at: <https://cs231n.github.io/convolutional-networks/>[Accessed May. 2021].

Teledyne DALSA (2015). Understanding Line Scan Applications. Available at: https://www.visiononline.org/vision-resources-details.cfm/vision-resources/Understanding-Line-Scan-Applications/content_id/5338 [Accessed May. 2021].

Mohamed Eelgendy (2020). Deep Learning for Vision Systems. Manning.

In-Sight 9902L Series Features (2021). Available at: <https://www.cognex.com/en-fi/products/machine-vision/2d-machine-vision-systems/in-sight-9902-line-scan-vision-system>. [Accessed May. 2021].

Keras Adam Optimizer Implementation (2021). Available at: <https://keras.io/api/optimizers/adam/> [Accessed May. 2021].

Keras Rmsprop Optimizer Implementation. 2021. Available at: <https://keras.io/api/optimizers/rmsprop/> [Accessed May. 2021].

Keras VGG. 2021. Available at: <https://keras.io/api/applications/vgg/> [Accessed May. 2021].

Niall O'Mahony (2019). Deep Learning vs. Traditional Computer Vision.

Rohit Thakur (2019). VGG16 implementation in Keras for beginners. Available at: <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c> [Accessed May. 2021].

Types of Sandpaper and Abrasives (2021). Available at: <https://www.homedepot.com/c/ab/types-of-sandpaper-and-abrasives/9ba683603be9fa5395fab901899be695>.

Karen Simonyan, Andrew Zisserman (2015), Very Deep Convolutional Networks for Large-Scale Image Recognition. Available at: <https://arxiv.org/abs/1409.1556>

J. Canny, "A Computational Approach to Edge Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.