Final-thesis Report

A Color vision Approach Considering Reflection Co efficient Based on Autoencoder Techniques Using Deep Neural Networks.

Submitted By

Shakib Izaz Mahmud
17101269
Sartaz Islam Shovon
17101178
Md. Abrar Hasnat
17101276
Md. Fahim Nafis
17101171

Group ID: T2030063

Supervised By:

Dr. Md. Ashraful Alam
Assistant Professor

Department of Computer Science and Engineering
Brac University

Sep 26, 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:

Shakib Izaz Mahmud
17101269

Md. Abrar Hasnat
17101276

Sartaz Islam Shovon
17101178

Md. Fahim Nafis
17101171

Approval

The thesis/project titled "A Color vision Approach Considering Reflection Co efficient Based on Autoencoder Techniques Using Deep NeuralNetworks" submitted by

1. Shakib Izaz Mahmud (17101269)
2. Md. Abrar Hasnat (17101276)
3. Sartaz Islam Shovon (17101178)
4. Md. Fahim Nafis (17101171)

Of Summer, 2015 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on August 23, 2015.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Ashraful Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Color vision approach using auto encoded technique is an effective way to detect objects. This approach considers various factors like movement detection, size and shape detection, color detection etc. Here we have considered reflection co efficient as another parameter to detect object material in different ambient lighting conditions. We are proposing to use deep learning methods to train our AI from values of light intensity of different objects in many controlled environments using digital illuminance meter also deep learning architecture on image data for detecting surface reflectance.

Keywords: Auto Encoded Techniques, Reflection Co efficient, Neural Network, Deep Learning, KNN, ANN, Random Forest, Logistic Regression, Naive Bayes, Decision Tree, Digital Illuminance meter.

Dedication

We would like to dedicate this thesis work to our loving parents. As well as, all the amazing faculties we came across and learnt from in the course of pursuing our Bachelors degree. It's been a worthwhile experience...

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our supervisor Dr. Md. Ashraful Alam sir for his kind support and advice in our work. He helped us whenever we needed help.

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

Chapter 1

Introduction

Object detection and tracking are two of the most operating research fields in computer vision consisting with a variety of applications such as object identifications, augmented reality, intelligent monitoring, interpersonal interaction, video retrieval, traffic detection, vehicle navigation.

1.1 Background Study

In the last two decades, object detection has often gone through two historical periods: "traditional object detection (before 2014)" and "deep learning-based detection (after 2014)." We can perceive "the wisdom of the cold weapon age" if we consider today's object detection to be a technical aesthetic powered by deep learning. The majority of the early object detection algorithms were built using handcrafted attributes. Due to the lack of adequate picture representation at the time, people had no choice but to construct intricate feature representations and a variety of speed-up abilities to exhaust the use of limited computing resources. Despite the fact that today's object detectors beat DPM in terms of detection accuracy, many of them, such as mixture models, hard negative mining, bounding box regression, and others, are still greatly influenced by its beneficial insights.[1]

For a complete understanding of an image, we should not only focus on classifying different images rather we should also focus on the concepts and location contained in individual image. Object detection as one of the most fundamental computer vision issues, it may provide vital information for semantic comprehension of pictures and videos. It is also relevant to a wide range of applications. Meanwhile, success in these domains will create neural network algorithms and will have a large impact on object identification approaches that can be termed as learning systems, thanks to neural networks and related learning systems. However, because of the wide range of views, positions, occlusions and lighting conditions, accurate object detection with an additional object localization duty is tough. Therefor much work is going on in this field.[2]

In this paper, we describe an algorithm which will help us to estimate the reflection co efficient of objects to determine the surface elements in a controlled environment. We assume that the parameter reflection co efficient will increase more accuracy

of detecting object material if we consider it with other object detection factors. Our main goal is to find out the reflection co efficient with both intensity and image data.We will detect the object material with both sensor data and image data separately and will add our parameter as a layer in different models to justify rather it is increasing the object detection efficiency effectively or not. Here we will use federated learning techniques along with the synthetic data we will simulate.

1.1.1 Artificial Intelligence

The study of mental qualities using computational models is known as artificial intelligence.The main preliminary goal of the AI is to complete a task more efficiently which humans can also do but not in efficient way or it will be more time consuming. [3]

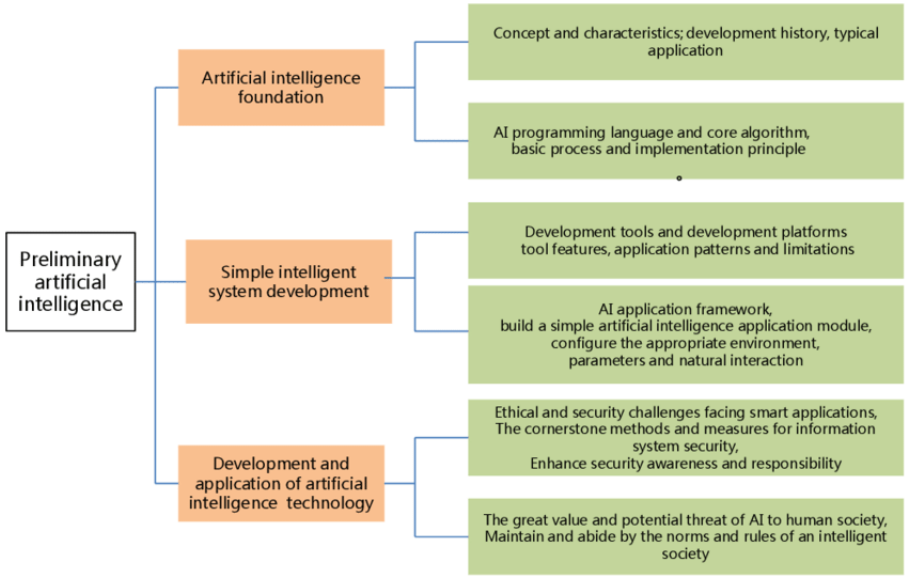

Figure 1.1: Standard Structure Diagram of Artificial Intelligence [4]

1.1.2 Deep Neural Network

Deep neural networks (DNNs) use low-level raw features to produce high-level intelligence. Deciphering the notions on which they base their decisions as human-level reasoning can help us understand this high-level intelligence. Deep neural networks (DNNs) have shown great success in numerous tasks (Goodfellow et al., 2016), from understanding images (Zoph et al., 2017) to answering questions (Devlin et al., 2018).[5]

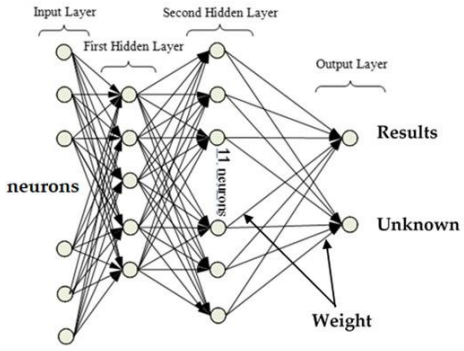

Figure 1.2: Basic Structure of the Deep Neural Network

1.1.3 Convolutional Neural Network (CNN)

Convolutional neural networks are multi-layer neural networks with a convolution layer (Conv), a down-sampling layer (or pooling layer), and a full-connection layer that are not fully linked (FC). To begin, the convolution layer applies several filters to the raw image, resulting in multiple feature maps. The feature is subsequently blurred by the down-sampling layer. Finally, to obtain a set of eigenvectors, a full connection layer is used. A convolutional neural network's architecture is shown in following Figure.

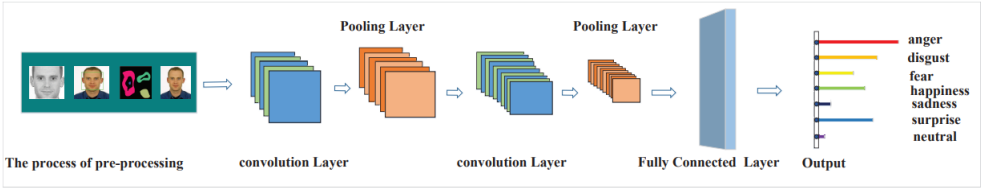

Figure 1.3: General structure of a Convolutional Neural Network

1.1.4 Machine Learning

Machine learning frequently simplifies the process of analyzing data and establishing relationships between different elements. To represent every occurrence in each dataset, machine learning algorithms use the same collection of features. Continuous, categorical, or binary attributes are possible. Unsupervised learning happens when instances are given with unknown labels (the corresponding correct outputs), whereas supervised learning occurs when instances are supplied with known labels (the corresponding correct outputs) (the corresponding correct outputs). Using these unsupervised (clustering) techniques, researchers hope to discover previously unknown but useful classes of things.[6] Another sort of machine learning is reinforcement learning. The learning system receives training information from the environment (external trainer) in the form of a scalar reinforcement signal, which is a measurement of how well the system operates. The learner is not told which actions to take; instead, by attempting each activity one at a time, he or she must find out which activities result in the best reward.[7]

1.1.5 Image Processing

The term "image processing" mostly relates to the manipulation of digital images. Three major issues have prompted the development of Image Processing:

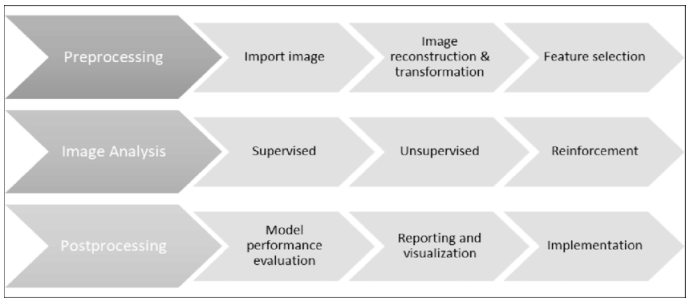

Figure 1.4: The basic structure of image processing

- Digitization and coding of photographs in order to improve transmission, printing, and storage of photographs;
- Picture enhancement and restoration, for example, to make images of the surface of other planets taken by various probes easier to interpret;
- Segmentation and description of images as a precursor to Machine Vision.

1.1.6 Performance Measuring Parameters

Accuracy

Random 80:20 split was used in the experiment. This work was evaluated using accuracy, F-measure, recall, precision. Accuracy was measured over correct and incorrectly classified instances.

$$Accuracy = \frac{Number\,of\,correct\,predictions}{Total\,Number\,of\,predictions} \tag{1.1}$$

To calculate accuracy in terms of positives and negatives, the following formula was used:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive\ +\ True\ Negative\ +\ False\ Positive\ +\ False\ Negative} \tag{1.2}$$

Precision

Precision refers to how precise/accurate the model is in terms of how many of the anticipated positives are actually positive. When the costs of False Positive are high, precision is a good statistic to use. (Reference)

$$Precision\ = \frac{True\ Positive}{True\ Positive\ +\ False\ Positive} \tag{1.3}$$

Recall

Recall calculates the number of Actual Positives in the model capture by labelling it as Positive (Reference)

$$Recall \ = \frac{True \ Positive}{True \ Positive \ + \ False \ Positive} \tag{1.4}$$

Epoch

The number of passes the machine learning algorithm has completed across the entire training dataset is referred to as an epoch in machine learning. It is calculated as d*e = i*b, where dataset size is d, number of epochs is e, number of iterations is I and batch size is b. if we try to learn epoch that would be

$$e = \frac{i \ * b}{d} \ or \ epoch \ = \frac{Iteration \ * batch \ size}{database \ size} \tag{1.5}$$

While there have been attempts to transform this process into an algorithm, determining how many epochs a model should run to train is reliant on numerous criteria linked to both the data and the model's aim. Epoch is essential in machine learning modeling because it is used to identify the model that best reflects the sample with the least amount of error. Before training the neural network, the epoch and batch size must be selected. Researchers who are looking for nearly perfect results on non-training data that requires more than one pass through the training data. Then we might need multiple epochs.

Validation

Model validation is the process of evaluating a trained model against a testing data set in machine learning. Model validation is performed after model training. It is kind of an auto check by computer to ensure that the data is reliable and sensible. The validation set is used to test a model; however, it is only utilized on a regular basis. After testing the data, we can set the hyper-parameters. But the model never learns from this validation it only shows us if we need to change the hyper-parameters or not. The validation set affects a model indirectly. Validation is an essential tool in the Data Science. It helps us to use the data profoundly. Also, it gives us much knowledge about the algorithm that we used and also about its performance. It's easy to lose track of sophisticated machine learning models and utilize the same data in multiple stages of the pipeline. Validation helps to mitigate that situation.

Testing

The goal of testing is to compare the neural network's outputs against targets from a separate collection (the testing instances). Testing findings are very dependent on the situation at hand, and certain figures may be correct for one application but incorrect for another. (Reference) The data set is divided into 3 portion training, validating and testing. After validating the dataset, the test data set was unused

from the very beginning so it can be compared and reevaluate the model if it has least error or best output. Test is important because Individual factors' effects on a neural network's prediction may be determined by using these tests. The test statistic can be used to rank variables in order of importance. The simulation results show the test's computing efficiency and performance. By which we can ensure the dependencies in the dataset and also predict the output which leads us to the ideal result. Testing is important as without testing there is no way to be sure about the methods, models and architecture that we are using or we might need to use. [8]

Training

Learning or determining suitable values for all the specified weights and biases from the labeled examples or dataset is what training a data set or model in an artificial neural network or deep neural network entails. Unsupervised and supervised training methods are the two types of training procedures. Supervised training comprises either manually "grading" the network's performance or providing the network with the required outputs with the inputs. When the network is left to its own devices to make sense of the inputs, this is known as unsupervised training. (Reference)The basic idea behind training is to randomly remove units from the network as it is being trained, resulting in a smaller neural network with each iteration. When a unit is dropped, it is the same as ignoring it during forward or backward propagation. This, in some ways, prevents the network from reacting to a certain set of characteristics. The iterative training process of neural networks addresses an optimization problem by obtaining parameters (model weights) that result in the least amount of error or loss while analyzing examples in the training dataset. Training is the only way to improve production or results. [9]

1.2 Research Objective

In this paper, we describe an algorithm which will help us to estimate the reflection co efficient of objects to determine the surface elements in a controlled environment. We assume that the parameter reflection co efficient will increase more accuracy of detecting object material if we consider it with other object detection factors. Our main goal is to find out the reflection co efficient with both intensity and image data. We will detect the object material with both sensor data and image data separately and will train our data-set in different models to justify rather it is increasing the material detection efficiency effectively or not. Here we will use federated learning techniques along with the synthetic data we will simulate.

Input feature consist of both numerical values and image in the data set. According to the literature we reviewed most of them were conducted on parameters like color, size, shape, movement, infrared vision and the object detecting models utilize one or more of these parameters to detect an object. That is why in our research we are considering another parameter called surface reflectance/reflection co efficient which is capable of determining material of the object. We expect that this approach may facilitate the process of object detection. Therefore, our system will be trained on multiple machine learning classifiers as well as utilize ensemble methods, to increase the accuracy of the models. The models will then be evaluated using multiple

classifier evaluation techniques. In addition to using these evaluation results, we will develop a prediction system based on our machine learning model to detect an object.

1.3 Research Gap

Various types of parameters are considered for object detection in ambient lighting condition. For example: Shape, Size, Color, Movement, Infrared vision. None of the para-metres can determine what a material really is. Right now they are only using this factors and predict what can be the object.

1.4 Problem Statement

Object detection is a crucial part in Computer Graphics section. As we know Autonomous Vehicle is the next big revolution in the current world. In current scenario there are many models which considers various object detection parameters: Shape, Size, Color, Movement, Infrared Vision to detect what an object is. But right now no models can not identify the material of any object in ambient lighting condition. That is why often autonomous vehicle and other technologies are facing problems to differentiate between two similar like objects.

Our vision is to consider surface reflectance as another parameter in object detection technology. With the rise in machine learning approaches, the initial identifying process is improving remarkably. Big data analytic may be used to examine large datasets and uncover hidden information and patterns, allowing user to gain knowledge from the data and anticipate outcomes accordingly. categorization and prediction accuracy of the present approach is not very good because of not considering surface reflectance. Due to the fact of COVID-19, it was quite impossible to fabricate an environment and collect the necessary data accurately . Though we have tried to make a controlled environment and collected data by the help of minimum resources.

In this paper, Our motive is to develop a model and detection system which can differentiate between two object material.Therefore, we have used machine learning algorithms namely Logistic Regression, for numeric data and for the image data in the study to detect an object material. Precision, Accuracy, F1-Score, Recall have been used to evaluate the performance of all these methods. Furthermore, we validated our data.In the beginning we have used Adam optimizer to optimize our data.

1.5 Challenges Faced

It is worth noting that, to conduct the whole thesis work was quite challenging for us. As the world is now facing the most challenging time due to the pandemic outbreak of the COVID-19 disease, it was quite difficult to analyze research work smoothly. As we were preparing our own data set, our main obstacle was finding and making a space's environment fully controlled as per our requirement but during this time

because of the limited resources we have faced many problems during the time of originating the required data-set of mainly "Numerical Data". Also, when we were searching for various material's "Shininess co efficient, Alpha" data, we could find limited data of it in a book to conduct our research work. That is why we needed a "Dark room" to measure various grade material's "Shininess co efficient, Alpha" but because of the unfortunate circumstances' world is facing right now we could not find any suitable place to collect "Shininess co efficient, Alpha" and make a data library for various grade materials. However, we have used two material "Brass" and "Copper" as we have found it in a book. Also, it was only affordable material for us then. Besides, we, the team members had to prepare our work while facing the obstacles in between creating our own data set.

1.6 Thesis Outline

To begin with, in the first chapter (Chapter 1), we have provided the overview of evolution object detection period and explained the related term which are discussed in whole paper , the Problem Statement to address the recent issues related to object detection and the Research Objective to clarify the goals and the proposed works of the authors. In the second chapter (Chapter 2), the work that has been done in the area of object detection using machine learning is discussed. Researchers' major findings are summarized. The inadequacies of previous approaches are also highlighted.

In the third chapter (Chapter 3) Mathematical Model is being explained which s used during the research.

In the fourth chapter (Chapter 4), the fundamental assessment of several supervised algorithms, as well as their implementation specifics are explored. Later on, these related algorithms will be used in the research pipeline. In the fifth chapter (Chapter 5), the Research Methodology and Workflow are introduced. The research is described in detail, including data collection, processing, and feature selection.

In the sixth chapter (Chapter 6), the various algorithms were subjected to a comparative analysis and in the last chapter (Chapter 7), we have concluded our paper and talked about our future research on this subject.

Chapter 2

Literature Review

Although physical models might be used to study light–material interactions, we chose to employ a model that allows for efficient computations, especially when combined with our pipeline rendering model. Phong proposed the current reflection model, which was later improved by Blinn. It has shown to be efficient and accurate enough to provide decent renderings under a variety of lighting circumstances and material qualities. [10]

For text categorization, this study recommends using a basic non-weighted features KNN algorithm. We propose employing a feature selection strategy that uses feature interaction to locate the relevant features for the learning job at hand (based on word interdependencies). This allows us to significantly lower the number of selected characteristics from which to learn, allowing our KNN algorithm to be used in situations where both the number of documents and the breadth of the vocabulary are large, such as the World Wide Web. [11]

The authors [12] built a logistic regression equation in python. They divided the class in two sections. Among them ten data are transaction ID and another ten data are Actual fraud. Then they computed the probability of fraud. So, by studying this book we understand that logistic regression is used in python when we can divide data in to two class.
The research paper [13] mentioned that by combining expectations based on prior experience (prior probabilities) with information from accessible data, Bayes theorem seeks to calculate the conditional probability of parameter values given the data. The Bayes theorem is utilized in their research to assess the probability of a certain facies' occurrence given well-log readings and assign the facies with the highest posterior probability to that observation depth. By studying their study, we can say it is good for small dataset and dimensions.

In the research paper [14] The authors only have a little amount of data to work with. They've trimmed it to fill it.Every negative image yielded 12 (60-by-90) pictures. As a result Any component of the image can be used as a negative image.Another glaring flaw. As a result, 180 genuine negative photos have been created.2160 negative photos have been added to the database.With k=8 and no normalization 1002 metal objects were accurately detected as metal by kNN, 45 metal objects were incorrectly labeled as Plastic, and three metals were incorrectly labeled as nothing. And, while

10116 of the photos (1002+84+9030) were correctly classified, 258 were incorrectly classified. That means, 97.5 percent of the objects in the dataset were successfully recognized.Out of 1302 samples, (1002+84) 1086 of the positive photos were properly recognized, yielding an accuracy of 83.4 percent for True Positive (TP) object classification.

In the research paper [15] Following the blob extraction process, two classifications are performed: by hand and using the KNN algorithm. Users are expected to manually determine the relevance of each blob Once all of the blobs have been placed,After each class has been selected, a training set is produced for them.Once the ground truth image and training set have been created,After that, a KNN algorithm based on the data will be processed for training.

The research paper [16] mentioned Small neighborhoods will arise and spread throughout the territory if K is equal to 1. Small neighborhoods, on the other hand, will be ignored if K is equal to 20. Then, determining the value of K has a significant impact on object classification accuracy. The distance measuring tool used by KNN is the Euclidean distance.

In the research paper [17] Numerous alternatives can be produced by adjusting the number of hidden layers in the neural network. The study examined five alternative models, each with a different number of hidden layers. During the training and validation phases, it achieves almost flawless categorization. However, it performs well in the testing and final phases, with areas under the curve of 0.9984 and 0.9995, respectively, with a minor inclination.
In the research paper [18] The experimental findings are evaluated using two publicly available benchmark datasets (I MSRC-21 [45] and (ii) Instance Saliency image database [46]. In the MSRC-21 dataset, which contains 591 RGB photos, all classes are categorized, including cow, automobile, sheep, building, plane, face, horse, tree, bicycle, and grass. In the experiments, ten classes are used. Realities on the ground are available for all photos. The pictures have a resolution of 213 $\times$ 320 pixels. Each image stands out from the rest, with its own set of colors and backdrops. Experiments were done on the MATLAB platform using a PC with a 2.5 GHz Intel Core-i3 processor and 6 GB of RAM. After attempting decision tree models for several objects, the accuracy was 71.3, 78.1, 75.3, 69.5, 70.3, and 77.2.
In this paper [19] for detecting an object shape-based method was used. For this task sensitive data like traffic light, pedestrian and moving car was used. When working with non-rigid objects like pedestrians, models or other parametrizations do not need to be developed explicitly. Instead, the approach is capable of off-line generation of an efficient representation using sample shapes, with on-line matching utilizing a new type of Distance Transform (DT) based matching.

The essential concepts of CNN are discussed in this review article, as well as how they are used to various radiological jobs, moreover the challenges and future directions in the field of radiology. Short datasets and over fitting are two barriers to applying CNN to radiological jobs and this study will look at how to overcome them. [20]

Yufeng Zhenga proposed using a Convolutional neural network (CNN) and follow-up scans to develop a computer-aided detection (CAD) technique for screening breast cancer. A digitized mammographic database is being used in the proposed strategy. Mammographic pictures are evaluated to detect potential malignant regions, as well as all regional images are transmitted to a trained CNN based on the pre-trained VGG-19 model to filter false positives. [21]

Glaucoma damages optic nerve and results in partial or blindness over time. To prevent blindness raised intravascular pressure might be modified. Periodic screening and early stage of detection also prevent blindness. Computer aided diagnosis (CAD) technique can be useful by applying digital fundus images to detect glaucoma in its early stages. [22]

To create a general-purpose automatic leaf recognition for categorization, Stephen Gang Wu integrated Probabilistic Neural Network (PNN) with image and data processing approaches. They developed a leaf recognition method that used simple features and seemed to be extremely efficient. Their algorithm is the most accurate artificial intelligence approach in comparison to other approaches for developing simply and quickly. To be effective, this system requires a varied database. Glaucoma is accurately detected using a CAD tool that employs deep learning techniques. A 18-layer convolutional neural network (CNN) is trained to extract robust characteristics from digital fundus images. U Raghavendraa scored 98.13 percent on 1426 fundus pictures (589 normal and 837 glaucoma). Their experiment demonstrates the system's robustness. It can be used as a complement to help professionals validate their decisions.[23]

Yingying Wang devised a facial expression identification approach based on the convolutional neural network (CNN) model, which has been widely employed in the image recognition sector due to its high performance. The deep neural network's potential nonlinear activation function gave it genuine artificial intelligence. Among typical activation functions, the ReLu function is one of the best. When the input value is negative, however, the derivative of the ReLu function is always zero, indicating the phenomena of neuronal necrosis. The activation function's impact on the CNN model is investigated. Sigmoid, tanh, ReLu, leaky ReLus, and softplus–ReLus, as well as the new activation function, have been analyzed and evaluated in facial expression recognition tasks using the Keras framework. According to experimental results on two publicly available facial expression databases, the convolutional neural network enhanced activation function (i.e., JAFFE and FER2013). [24]

To recognize leaves, T L I Sugata applied a deep convolutional neural network. They extended the dataset via data augmentation, which included horizontal reflection, contrast enhancement, and rotations, to ensure the accuracy level. Their findings show that by combining deep convolutional neural networks with data augmentation, their system can achieve accuracy near to that of state-of-the-art systems.[25]

In order to test state-of-the-art deep learning technologies, multispectral RapidEye optical images are used to classify difficult wetland classifications. Wetland mapping in Canada was investigated using the capabilities of seven well-known deep convnets:

DenseNet121, InceptionV3, VGG16, VGG19, and Xception. The classification results obtained from deep CNNs were compared to those obtained from traditional machine learning methods such as Random Forest and Support Vector Machine in order to assess the former's efficacy in classifying wetlands. The results of the study reveal that for all convnets, full training utilizing five spectral bands outperforms the other options. Support Vector Machine (SVM) and Random Forest, respectively, achieved 74.89% and 76.08% classification accuracy (RF). The integration of Inception and ResNet modules yielded state-of-the-art classification accuracies of 96.17%, 94.81%, and 93.57%, respectively, indicating that the integration of Inception and ResNet modules is an efficient architecture for classifying complex remote sensing scenes like wetlands. [26]

Convolutional neural networks are widely used in the field of computer vision, as well as in academia. Because it is a unique deep learning architecture, it is capable of accurately extracting picture features. Andrew G. Howard investigated a range of measures for improving the present state-of-the-art deep convolutional neural network using an image classification pipeline. They appear to have employed multiple strategies, including adding more image modifications to the training data, adding more transformations to generate more predictions at test time, and employing complementary models on higher resolution photos. [27]

M Baccouche suggested an automated deep learning algorithm that learns to characterize human activities without prior information. In light of the advantages and uses of CNN in image identification, this research developed a facial expression recognition approach based on the CNN model. Experiments on the KTH dataset showed that their proposed strategy outperforms existing deep models and produces results that are equivalent to the best relevant research[28]

Convolutional neural networks are made out of feed-forward artificial neural networks that were effectively used for visual vision. CNN is known as a shift invariant or space invariant NN since their architecture is based on common weights. It is also based on translation invariance characteristics. Because it can activate the properties of neurons to solve nonlinear issues, the activation function is the core of the CNN model. A correct activation function improves the ability to map data in dimensions. To investigate the influence on the MNIST dataset, Arun Kumar Dubey employed a rectified linear unit (Relu) and Leaky-Relu activation for the inner CNN layer and softmax activation function for the output layer. [29]

Chapter 3

Mathematical Model

I= $I_{amb} + I_{spec} + I_{diff}$

$\Rightarrow I = I_{amb} + k_s I_s cos^{\alpha}\phi + k_d I_s cos\theta$

Where,

I= *Total Intensity of reflected light from object*

I_{amb} = *Light Intensity of ambient*

I_{spec} = *Specular Intensity*

I_s = *Intensity of light source*

I_{diff} = *Diffused Intensity*

k_s = *Specular Reflection Co − efficient of the object material*

α = *Shininess Co − efficient of the object material*

k_d = *Diffused Reflection Co − efficient of the object material*

ϕ = *Angle bet^n object normal and viewing vector*

θ = *Angle bet^n object normal and light source*

(3.1)

Here, k_s, k_d, are material dependent parameters and I_{amb}, I_s, θ, ϕ are material independent parameters. It can be written as:

$$k_d, k_s, \alpha = f(I, I_{amb}, I_s, \phi, \theta) \tag{3.2}$$

Chapter 4

Methodology

4.1 Working Process

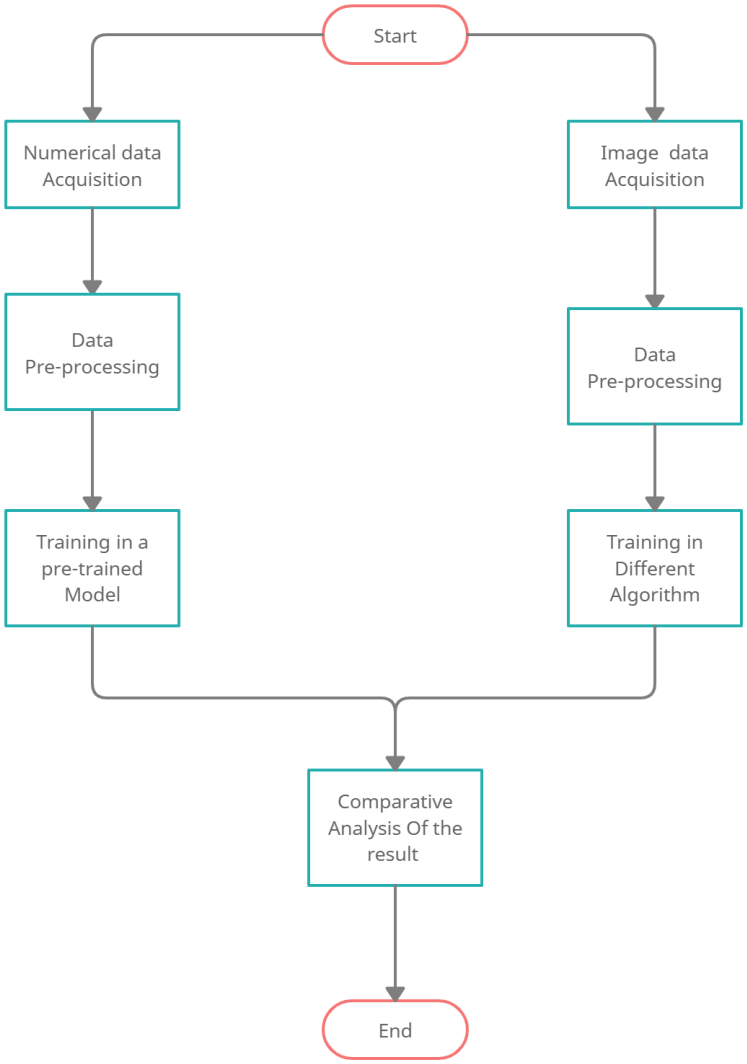

Figure 4.1: Workflow Diagram

4.2 Used Architectures

4.2.1 VGG16

The Keras library comes with a profusion of models namely, Restnet, RestNetV2, RestNetXt, InceptionV3 or a mix of InceptionRes- NetV2, VGG16,VGG19, DenseNet etc. These are all well established and well acknowledged models trained over the ImageNet containing an abundance of image data, identifying as a pre-trained model.[21] In our thesis we are using the VGG16 model. As we are using image data it is often a good practice to use a network which is trained on large data sets.[22] VGG16 is constructed of mainly 2 parts, a convolutional base built from a series of layers which is MaxPooling2d and Conv2D.It is also named VGG16 because it has 16 layers where we can put weight. VGG16 is notable in that, rather than having a large number of hyper parameters, it uses a much simpler network. VGG16 is an improvement over AlexNet. VGG16 replaces bigger kernel sized filters with 3x3 kernel sized filters for Conv2d with stride 1 and 2x2 for MaxPoolnig2D with stride 2 where in the end both are fully connected and softmax as output. (Reference 3) In our thesis, after importing our dataset we imported the VGG16 library. As VGG16 is pre trained we did not change any weight here. We also did not change any layer as well. Then we attended the data. After that the Training test phase starts. Finally, softmax pulled all the data together and gave visual representation.

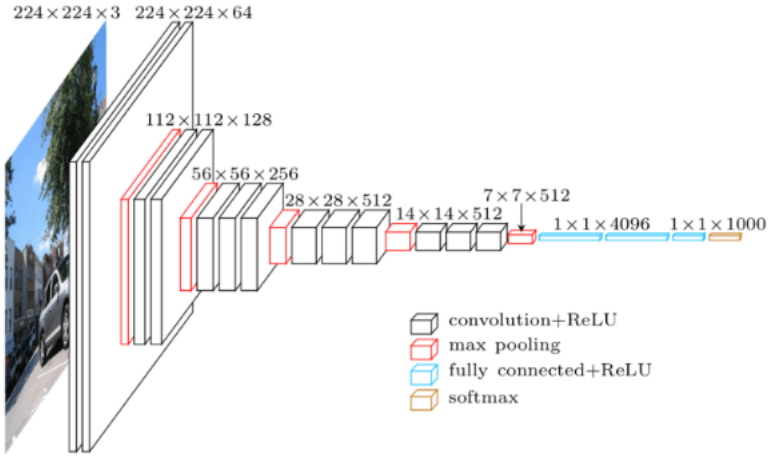

Figure 4.2: VGG16

4.2.2 VGG19

VGG-19 model consists of 19-layer which includes 16 convolutional and 3 fully-connected layers CNN devised by Simonyan and Zisserman of the University of Oxford that uses filter count of 33 along with stride and pad count of 1, as well as 22 layers of max-pooling along with stride 2.28,29 The VGG-19 (see Fig. 8)is a much deeper CNN with increased layers than AlexNet. It uses a modest 33 filters in all convolutional layers in order to minimize the quantity of parameters in deep networks, and performs best with its 7.3 percent error rate count. The VGG-19 model could not achieve the winning position in the ILSVRC30 2014 competition, but the VGG Net was one of the most prominent papers since it rearmed the idea

that CNNs need a deep network of layers to perform properly with hierarchical visual data representation. Keep the depth and the approach simple.

In ILSVRC 2014, the VGG-19 model, which has 138 million parameters, was ranked second in classification and first in localization. The ImageNet27 database was used in the ImageNet Large-Scale Visual Recognition Challenge, which was used to train this model (ILSVRC). [30]Over a million photos have been trained by the VGG-19 and is able to classify them into 1000 different object categories which includes keyboards, pencils, mouse and a large selection of other animals. As a result, the model has taught itself a collection of rich feature representations for a huge number of images.

R-CNN 31 is a region-based CNN that creates a region proposal and utilizes CNN to detect and classify objects. Later on, the Fast RCNN32 and Fasted R-CNN33 algorithms are engineered to enhance speed and accuracy. The RCNN approach seems incongruous for detecting breast cancer because mammograms in texture and size of lesion (cancer) vary significantly for different cases.

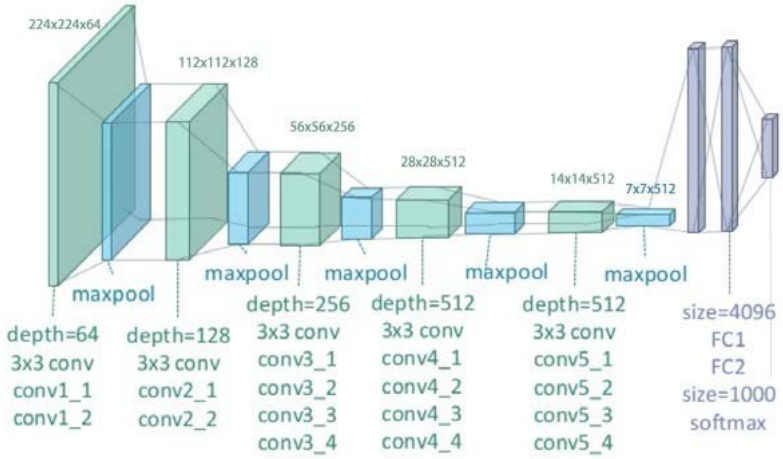

Figure 4.3: VGG19

4.2.3 MobileNetV2

Under the Convolutional Neural Network architecture, MobileNetV2 is mainly utilized for mobile vision applications. MobileNet is a very utilitarian feature extractor. As a source of non-linearity, the layer in between filters the features using convolutions by lightweight depth. It can classify images, detect objects and semantic segmentation. TensorFlow Slim Image Classification Library includes MobileNetV2 [11]. MobileNetV2 is the successor of MobileNetV1 which introduces 2 additional features which are Linear Bottlenecks between the layers and Shortcut connection between bottlenecks. MobileNetV2 consists of 17 layers of 3x3 convolutional layers in a row with 1x1 convolution [12]. Additionally, there is also an average layer of MaxPooling and a layer of classification. In our work, we imported MobileNetV2 library and did not change any weight nor any layer. Afterwards we attended the data before the test train phase.

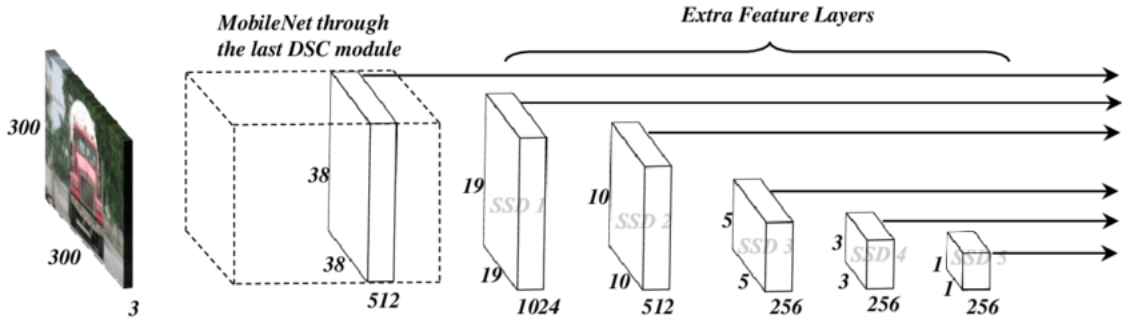

Figure 4.4: MobileNet SSD network architecture

4.2.4 Decision tree

The decision-tree method is a supervised learning approach that works with both continuous and categorical output variables (regression and classification). It is built from the top down, segmenting the data into subsets that contain samples with similar values (homogenous). The standard deviation is used to determine the homogeneity of a sample instance. When a sample's standard deviation is zero, it is termed completely homogeneous.

$$(Standard\ Deviation).sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)}, where(Mean)\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$Co\ efficient\ Of\ Variation\ (CV) = \frac{\delta}{\mu} \times 100$$

The standard deviation for two variables (Target, Feature) is:

$$S(T,\ X) = \sum_{c \epsilon X} P(c)S(c) \tag{4.1}$$

The property with the largest standard deviation reduction is chosen for the decision node. After a data set has been split by an attribute, the standard deviation reduction technique entails lowering the standard deviation. Identifying the characteristic that decreases standard deviation the greatest (SDR) is required to design a decision tree. The feature with the biggest standard deviation reduction (SDR) is picked for the decision node.

$$SDR(T, X) = S(T) - S(T, X), \tag{4.2}$$

Where T= Target, X= Feature and S= Standard Deviation.

Based on the values of the specified feature, the data set is separated into two sections. The approach is repeated for non-leaf branches until all of the samples in the dataset

have been processed. The coefficient of variation (CV) is commonly employed as a criterion for recursion termination. If the CV for a certain branch falls below a certain threshold, such as 10%, the splitting process is terminated, and the average value for that subset is assigned to the leaf node.

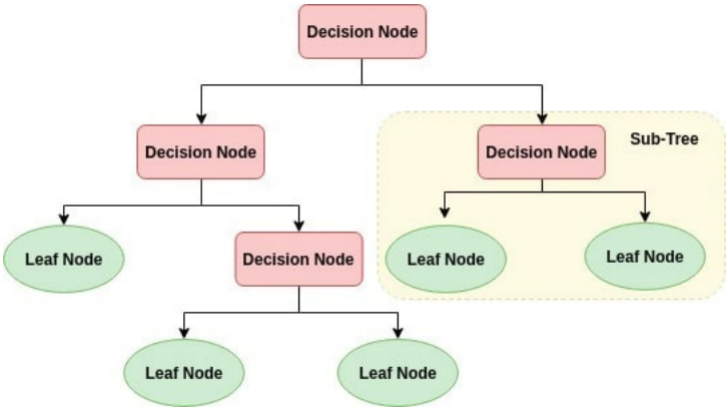

Figure 4.5: Decision Tree

4.2.5 Artificial Neural Networks (ANN):

It's a well-known classification and regression Neural Network model. The Artificial Neural Network (ANN) is a data processing method based on how information is processed by the biological nervous system, such as the brain. It consists of a large number of tightly connected processing elements (neurons) that cooperate to solve a problem.
The goal of an artificial neural network is to combine multiple combinations of artificial neurons to produce more powerful outputs. As a result, a typical artificial neural network's conceptual structure looks somewhat like this:

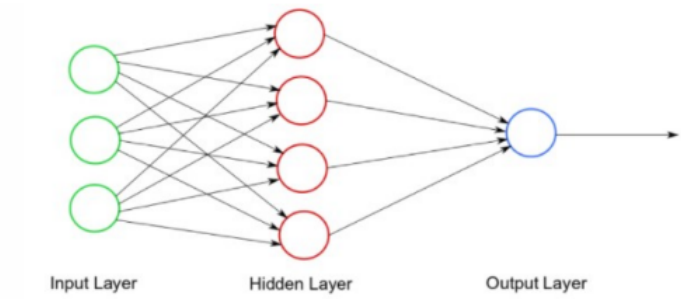

Figure 4.6: Artificial Neural Network

The net input for the above general model of artificial neural network can be calculated as follows:

$$y_i n = x_1.w_1 + x_2.w_2 + x_3 w_3 .....x_m.w_m \tag{4.3}$$

The activation function can be applied to the net input to calculate the output.

$$Y = F(y_i n) \tag{4.4}$$

4.2.6 Naïve Bayes (gaussian)

The Naive Bayes classification technique is built on the Bayes theorem and the concept of predictor independence. A Naive Bayes classifier, in simple terms, assumes that the existence of one feature in a class has no influence on the presence of any other feature in that class. The Naive Bayesian model is simple to construct and is particularly useful when working with huge data sets. Because of its simplicity, Naive Bayes is known to outperform advanced classification algorithms.

Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4.5}$$

4.2.7 Logistic Regression

The logistic function, also known as the sigmoid function, was created by statisticians to reflect the features of population increase in ecology, such as how it expands exponentially and finally approaches the ecosystem's carrying capacity.

It's an S-shaped curve with the ability to change the value of any real-valued integer between 0 and 1, but never exactly between those two points.

$$\frac{1}{(1 + e^{valve)}} \tag{4.6}$$

Where e is the base of natural logarithms, Euler's number, or the spreadsheet's EXP () function, and value is the numerical value to be changed. The logistic function was used to shift the values between -5 and 5 into the range 0 and 1, as illustrated below.

4.2.8 Random Forest

The random forest algorithm is a supervised learning model that "learns" how to categorize unlabeled data using labeled data. This is the polar opposite of the K-means Cluster algorithm, which we discovered was an unsupervised learning model in a previous post. The Random Forest Algorithm is a versatile model that is commonly used by engineers. It can tackle both regression and classification issues.

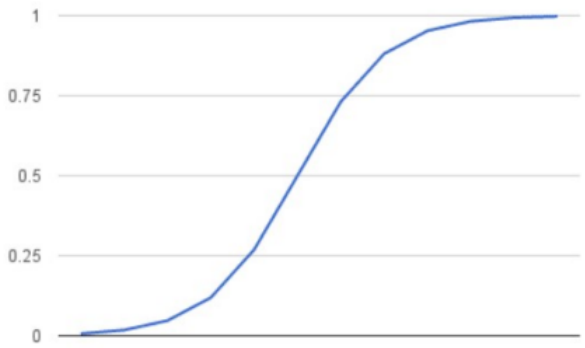

Figure 4.7: Logistic Function

During the process of solving regression problems Random Forest model is needed. Mean squared error (MSE) is needed to show the data branches from each node.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(fi - yi)^2 \tag{4.7}$$

Where N is the number of data points, fi is the value returned by the model and yi is the actual value for data point i.

Gini index formula is used in the period of performing Random Forest based on classification data. The formula is given below:

$$Gini = 1 - \sum_{i=1}^{c}(pi)^2 \tag{4.8}$$

This formula calculates the Gini of each branch on a node based on the class and probability, deciding which branch is more likely to occur.

$$Entropy = \sum_{i=1}^{c} -p_i * log_2(p_i) \tag{4.9}$$

4.2.9 K Nearest Neighbors (kNN)

K nearest neighbors is a straightforward method that maintains all available examples and categorizes new ones using a similarity metric (e.g., distance functions). KNN has been utilized as a non-parametric technique in statistical estimates and

pattern recognition since the early 1970s.

A case is classified by a majority vote of its neighbors, with the case being allocated to the class with the most members among its K closest neighbors as determined by a distance function. If K = 1, the case is simply allocated to the nearest neighbor's class.

$$
\begin{aligned}
&\textit{Euclidean } \sqrt{\sum_{i=1} k(x_i - y_i)^2} \\
&\textit{Manhattan } \sum_{i=1}^{k} |x_i - y_i| \\
&\textit{Minkowski } (\sum_{i=1}^{k} (|x_i - y_i|)^q)^{\frac{1}{q}}
\end{aligned}
\tag{4.10}
$$

It's worth noting that all three distance measurements are only applicable to continuous variables. The Hamming distance must be employed when categorical variables are present. It also raises the issue of numerical variable normalization between 0 and 1 when the dataset contains both numerical and category variables.

Chapter 5

Research Methodology

5.1 Data Acquisition

5.1.1 Data Acquisition Environment

The experimental setup of our study shown in figure X. There were six planes of the controlled environmental room. The room size is (Size). Letting one of the walls referred as reference wall, two side adjacent to the reference wall and top adjacent to the reference wall are white plastered. Bottom adjacent to the reference wall is white tiled. The opposite wall of the reference wall is covered in three replaceable backgrounds (Black, Green, White). Necessary steps are taken to make the room fully light proofed. Two 100-watt 6500K source LED flood light attached to two light stands with heights of (size) are placed at two different location side adjacent to the reference wall. A light holder is attached to the reference wall, located in the upper middle part of the wall at the height of 78inch. One Intensity Measuring device (Lutron LX-101A), Five types of power source LED lights (3-watt, 7-watt, 15-watt, 30-watt, 50-watt), a color changeable platform with height of (Size) and the surface of (Size) are used in the experimental setup. Two types of cameras (SONY IMX682, SONYIMX519) attached to a camera stand with height of (Size) are used in capturing image data. There are two types of materials (Brass sheet, Copper Sheet) in three different shapes (Cubic, Cylindrical, Conic). Two types of materials of three shapes are converted into three identical sizes. The sizes are given below:

Small Conic- Diameter: 4inch, Height: 3.7inch
Medium Conic- Diameter: 4.5inch, Height: 4.1inch
Large Conic- Diameter: 5inch, Height: 4.4inch
Small Cylindrical- Diameter: 1.8inch, Height: 1.8inch
Medium Cylindrical- Diameter: 2inch, Height: 2.2inch
Large Cylindrical- Diameter: 2.5inch, Height: 2.8inch
Small Cubic- Side: 3.1inch
Medium Cubic- Side: 3.8inch
Large Cubic- Side: 4.2inch

These two types materials of three shapes and three identical sizes are included in our experimental setup. The platform is placed in five different positions. Which are (62.3-degree, 28.95-degree, 30.1 degree); (53.7-degree, 26.21-degree, 23.55 degree);

(47.5-degree, 19.2 degree, null); (41.9-degree, null, null); (37.66-degree, 9.99-degree, 11.73 degree). The angels are taken with respect to source light, platform midpoint and plane.

5.1.2 Image Data Acquisition

In different positions materials are placed in the midpoint of the platform. Then a set of images were taken in different parameters (Background, Four different Lighting Condition, Position). The camera position is always fixed. The aperture of the camera is respectively F2.2 and F1.7. The shutter speed, ISO, Automated white balance is automatically adjusted depending on the different types of images.

5.1.3 Numerical Data Acquisition

The materials are placed in the midpoint of the platform and the intensity measuring device is adjacent to the platform. Intensity measuring device is placed in a distance of null meter from the midpoint of the platform diagonally. After that intensity data are taken in different parameters (Background, Three different Lighting Condition, Position). Different types of readings are fabricated during the experiment in respect to different parameters and material.

	Base	Variable Source, W	Background	I(s)	Distance (in)	Theta (d)	Angle,Radiance	Cos Theta	I(amb)	Material	n	Ks	Shape	Size	I	Kd
0	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	27.8974	0.913725	Cubic	S	4.0	-1.030000e-09
1	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	27.8974	0.913725	Cubic	M	4.0	-1.030000e-09
2	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	27.8974	0.913725	Cubic	L	4.0	-1.030000e-09
3	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	27.8974	0.913725	Cylindrical	S	5.0	8.605074e-02
4	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	27.8974	0.913725	Cylindrical	M	4.0	-1.030000e-09
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4045	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	12.8000	0.160138	Cylindrical	M	298.0	-1.245942e-02
4046	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	12.8000	0.160138	Cylindrical	L	296.0	-1.704448e-02
4047	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	12.8000	0.160138	Conic	S	295.0	-1.933701e-02
4048	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	12.8000	0.160138	Conic	M	296.0	-1.704448e-02
4049	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	12.8000	0.160138	Conic	L	295.0	-1.933701e-02

4050 rows × 16 columns

Figure 5.1: raw Data Set

5.1.4 Data Preprocessing

Image Data Preprocessing

*As a first step of image data preprocessing, images are converted from (462*347) pixel to (224*224) pixel because for VGG19, VGG16, MobileNEt, this quality is sufficient. Converted images are used for all Test- Train – Split.*

```
x_train=[]
for folder in os.listdir(train_path):
    sub_path=train_path+"/"+folder
    for img in os.listdir(sub_path):
        image_path=sub_path+"/"+img
        img_arr=cv2.imread(image_path)
        img_arr=cv2.resize(img_arr,(224,224))
        x_train.append(img_arr)
x_test=[]
for folder in os.listdir(test_path):
    sub_path=test_path+"/"+folder
    for img in os.listdir(sub_path):
        image_path=sub_path+"/"+img
        img_arr=cv2.imread(image_path)
        img_arr=cv2.resize(img_arr,(224,224))
        x_test.append(img_arr)
x_val=[]
for folder in os.listdir(val_path):
    sub_path=val_path+"/"+folder
    for img in os.listdir(sub_path):
        image_path=sub_path+"/"+img
        img_arr=cv2.imread(image_path)
        img_arr=cv2.resize(img_arr,(224,224))
        x_val.append(img_arr)
```

Figure 5.2: Resizing the Image

These images are nothing but pixels and the value of each pixel are in between 0-255. After this procedure images are converted in to an array.

```
train_x=np.array(x_train)
test_x=np.array(x_test)
val_x=np.array(x_val)
```

Figure 5.3: Code for Converting Image into Array

Models face difficulties to handle the 0-255 range data. That's why these data are divided by 255 to normalize the value from 0-1.

```
train_x=train_x/255.0
test_x=test_x/255.0
val_x=val_x/255.0
```

Figure 5.4: Code for Normalization

5.2 Feature Selection Engineering

5.2.1 Feature Selection

This criteria is used to determine which features of the dataset are the most important and which features are dependent on a number of factors. Highly relevant features are those which have the highest correlation or fluctuation with the aim. As a result, in order for the model to generalize and interpret the new data, less relevant features are removed.

5.2.2 Feature Engineering

Feature engineering is a machine learning technique that involves picking and modifying variables from raw data to create a prediction model. It aids in the preparation of appropriate dataset inputs and the creation of features that are suitable with the machine learning algorithm's criteria.
To investigate the impact of each feature on target value, we used the Correlation Finding technique to uncover relationships between each feature in both datasets. Features with little effect on class prediction were observed so that they might be given less weight in the final model. The.corr() function was used to extract the correlation matrix.

corrmat = dfn . c o r r ( )

```
corrmat = df.corr()
corrmat
```

Figure 5.5: Code for Correlation table

Later a correlation table has been generated. Which is following:

	Variable Source, W	I(s)	Distance (in)	Theta (d)	Angle,Radiance	Cos Theta	I(amb)	n	Ks	I	Kd
Variable Source, W	1.000000e+00	0.185477	-2.190155e-16	2.193080e-16	-4.111283e-17	-5.898623e-17	1.379062e-01	-2.919877e-17	6.188220e-17	0.139504	0.011578
I(s)	1.854772e-01	1.000000	-1.753038e-01	1.514508e-01	1.514508e-01	-1.438606e-01	7.393510e-01	1.714280e-04	1.714280e-04	0.733888	0.158422
Distance (in)	-2.190155e-16	-0.175304	1.000000e+00	-9.773387e-01	-9.773387e-01	9.651001e-01	-2.104332e-01	8.762839e-17	-3.667559e-17	-0.208723	-0.094185
Theta (d)	2.193080e-16	0.151451	-9.773387e-01	1.000000e+00	1.000000e+00	-9.985563e-01	2.286871e-01	-6.403832e-19	0.000000e+00	0.225741	0.087103
Angle,Radiance	-4.111283e-17	0.151451	-9.773387e-01	1.000000e+00	1.000000e+00	-9.985563e-01	2.286871e-01	-3.401585e-17	8.626118e-17	0.225741	0.087103
Cos Theta	-5.898623e-17	-0.143861	9.651001e-01	-9.985563e-01	-9.985563e-01	1.000000e+00	-2.320947e-01	0.000000e+00	2.353835e-19	-0.228702	-0.083746
I(amb)	1.379062e-01	0.739351	-2.104332e-01	2.286871e-01	2.286871e-01	-2.320947e-01	1.000000e+00	4.369903e-18	-1.367920e-18	0.973086	0.113160
n	-2.919877e-17	0.000171	8.762839e-17	-6.403832e-19	-3.401585e-17	0.000000e+00	4.369903e-18	1.000000e+00	1.000000e+00	0.007337	0.083069
Ks	6.188220e-17	0.000171	-3.667559e-17	0.000000e+00	8.626118e-17	2.353835e-19	-1.367920e-18	1.000000e+00	1.000000e+00	0.007337	0.083069
I	1.395038e-01	0.733888	-2.087229e-01	2.257408e-01	2.257408e-01	-2.287024e-01	9.730858e-01	7.336509e-03	7.336509e-03	1.000000	0.328482
Kd	1.157802e-02	0.158422	-9.418499e-02	8.710295e-02	8.710295e-02	-8.374635e-02	1.131599e-01	8.306862e-02	8.306862e-02	0.328482	1.000000

Figure 5.6: Correlation Table

After that for better understanding a heat map has been created.

Plt.figure(figsize= (10,11)) Sns.heatmap(df.corr(),annot=true) Plt.plot

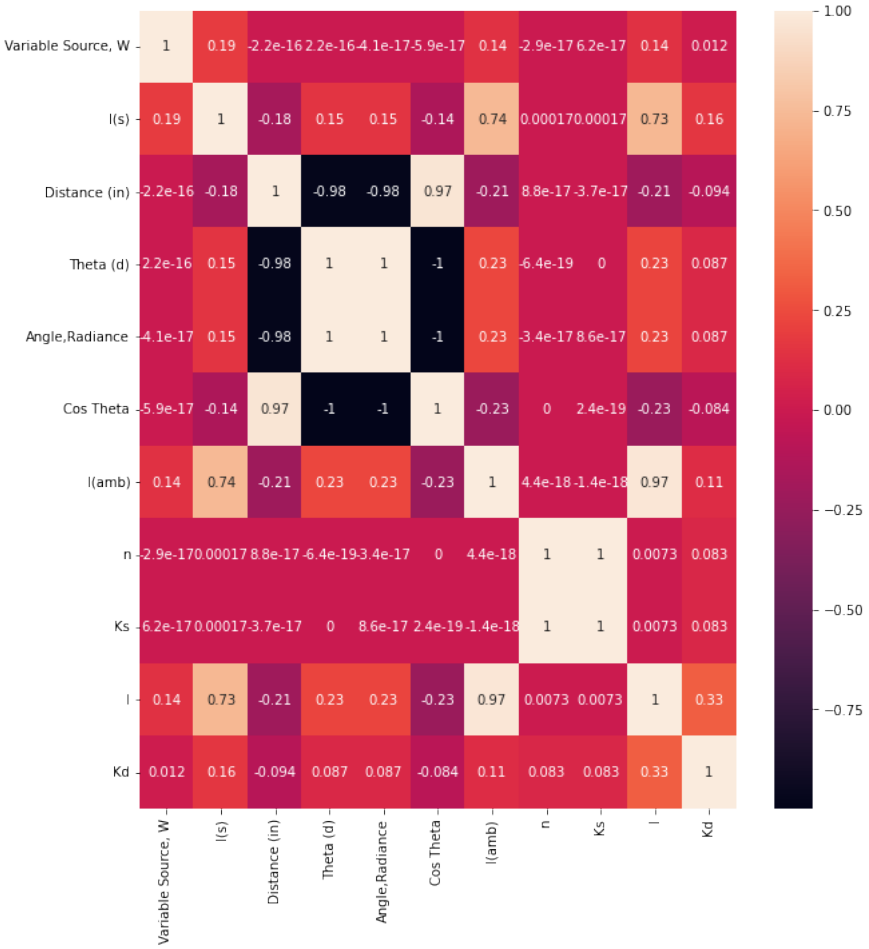

Figure 5.7: Correlation Heatmap

Later, After using .info() function following data can be extracted.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4050 entries, 0 to 4049
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Base                4050 non-null   object
 1   Variable Source, W  4050 non-null   int64
 2   Background          4050 non-null   object
 3   I(s)                4050 non-null   int64
 4   Distance (in)       4050 non-null   float64
 5   Theta (d)           4050 non-null   float64
 6   Angle,Radiance      4050 non-null   float64
 7   Cos Theta           4050 non-null   float64
 8   I(amb)              3996 non-null   float64
 9   Material            4050 non-null   object
 10  n                   4050 non-null   float64
 11  Ks                  4050 non-null   float64
 12  Shape               4050 non-null   object
 13  Size                4050 non-null   object
 14  I                   3996 non-null   float64
 15  Kd                  4050 non-null   float64
dtypes: float64(9), int64(2), object(5)
memory usage: 506.4+ KB
```

Figure 5.8: Info Number of Data

As models can determine which is brass and which is copper just by seeing the value of ks, kd and n from the column. That is why the mentioned elements of the column will be removed from our data set with the help of del function.
These elements will be extracted from the Image data to complete the object detection. The del function is given below:

```
del df['Ks']
del df['Kd']
del df['n']
```

Figure 5.9: Code for k_s k_d n Column

Then there will be a new dataset after clearing some column. The new dataset is given below:

	Base	Variable Source, W	Background	I(s)	Distance (in)	Theta (d)	Angle,Radiance	Cos Theta	I(amb)	Material	Shape	Size	I
0	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	Cubic	S	4.0
1	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	Cubic	M	4.0
2	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	Cubic	L	4.0
3	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	Cylindrical	S	5.0
4	Both Off	3	Black	25	64.1	62.30	1.087340	0.464842	4.0	Brass	Cylindrical	M	4.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4045	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	Cylindrical	M	298.0
4046	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	Cylindrical	L	296.0
4047	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	Conic	S	295.0
4048	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	Conic	M	296.0
4049	Right On,Left Off	50	White	551	96.5	37.66	0.657291	0.791650	299.0	Copper	Conic	L	295.0

4050 rows × 13 columns

Figure 5.10: New dataset after deleting k_s k_d n Column

There are some string or object which can not be handled by algorithms. So, string values need to be replaced by integer value. Base, Background, Shape, Size column's elements are being replaced by:

Base		Background		Shape		Size	
Both Off	0	Black	0	Cubic	0	Small	0
Right On Left Off	1	Green	1	Cylindrical	1	Medium	1
Both On	11	White	11	Conic	11	Large	11

Figure 5.11

For delet .replace() of Pandas are being used.

df['base '].replace(['Both Off' , 'Right On, Left Off' , 'Both On' ,] , ['00' , '01' , '11'], inplace= True)

df['Background'].replace(['Black' , 'Green' , 'White'], ['00' , '01' , '11'], inplace=True)

df['Shape'].replace(['Cubic' , 'Cylindrical' , 'Conic'] , ['00' , '01' , '11'] , inplace=True)

df['Size'].replace(['S' , 'M' , 'L'] , ['00' , '01' , '11'], inplace=True)

After delet the new dataset is given below:

```
df['Base '].replace(['Both Off ','Right On,Left Off','Both On '],['00','01','11'],inplace=True)
df['Background'].replace(['Black','Green','White'],['00','01','11'],inplace=True)
df['Shape'].replace(['Cubic','Cylindrical','Conic'],['00','01','11'],inplace=True)
df['Size'].replace(['S','M','L'],['00','01','11'],inplace=True)
```

Figure 5.12: Code for Feature Extraction

Now With the help of "convert$_d$ict = "
function the string which has been converted to the integer value need it's data type change.

Convert$_d$ict = 'Base' : int,
'Background' : int,

'Shape' : int,

'Size' : int

df= df.astype(convert$_d$ict)

Finally, our data is being ready. Final data set is given below:

	Base	Variable Source, W	Background	I(s)	Distance (in)	Theta (d)	Angle,Radiance	Cos Theta	I(amb)	Material	Shape	Size	I
0	00	3	00	25	64.1	62.30	1.087340	0.464842	4.0	Brass	00	00	4.0
1	00	3	00	25	64.1	62.30	1.087340	0.464842	4.0	Brass	00	01	4.0
2	00	3	00	25	64.1	62.30	1.087340	0.464842	4.0	Brass	00	11	4.0
3	00	3	00	25	64.1	62.30	1.087340	0.464842	4.0	Brass	01	00	5.0
4	00	3	00	25	64.1	62.30	1.087340	0.464842	4.0	Brass	01	01	4.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4045	01	50	11	551	96.5	37.66	0.657291	0.791650	299.0	Copper	01	01	298.0
4046	01	50	11	551	96.5	37.66	0.657291	0.791650	299.0	Copper	01	11	296.0
4047	01	50	11	551	96.5	37.66	0.657291	0.791650	299.0	Copper	11	00	295.0
4048	01	50	11	551	96.5	37.66	0.657291	0.791650	299.0	Copper	11	01	296.0
4049	01	50	11	551	96.5	37.66	0.657291	0.791650	299.0	Copper	11	11	295.0

4050 rows × 13 columns

Figure 5.13: Dataset after Feature Extraction

Some important relations are shown below:

I VS I_{amb}

With the help of sns.faceGrid( ) function of seaborn

```
sns.set_style("whitegrid")
sns.FacetGrid(df, hue="Material", size=16) \
   .map(plt.scatter, "I(amb)", "I") \
   .add_legend()
plt.show()
```

Figure 5.14: Code for i VS i_{amb}

Just like previous I VS I_s graph is given below:

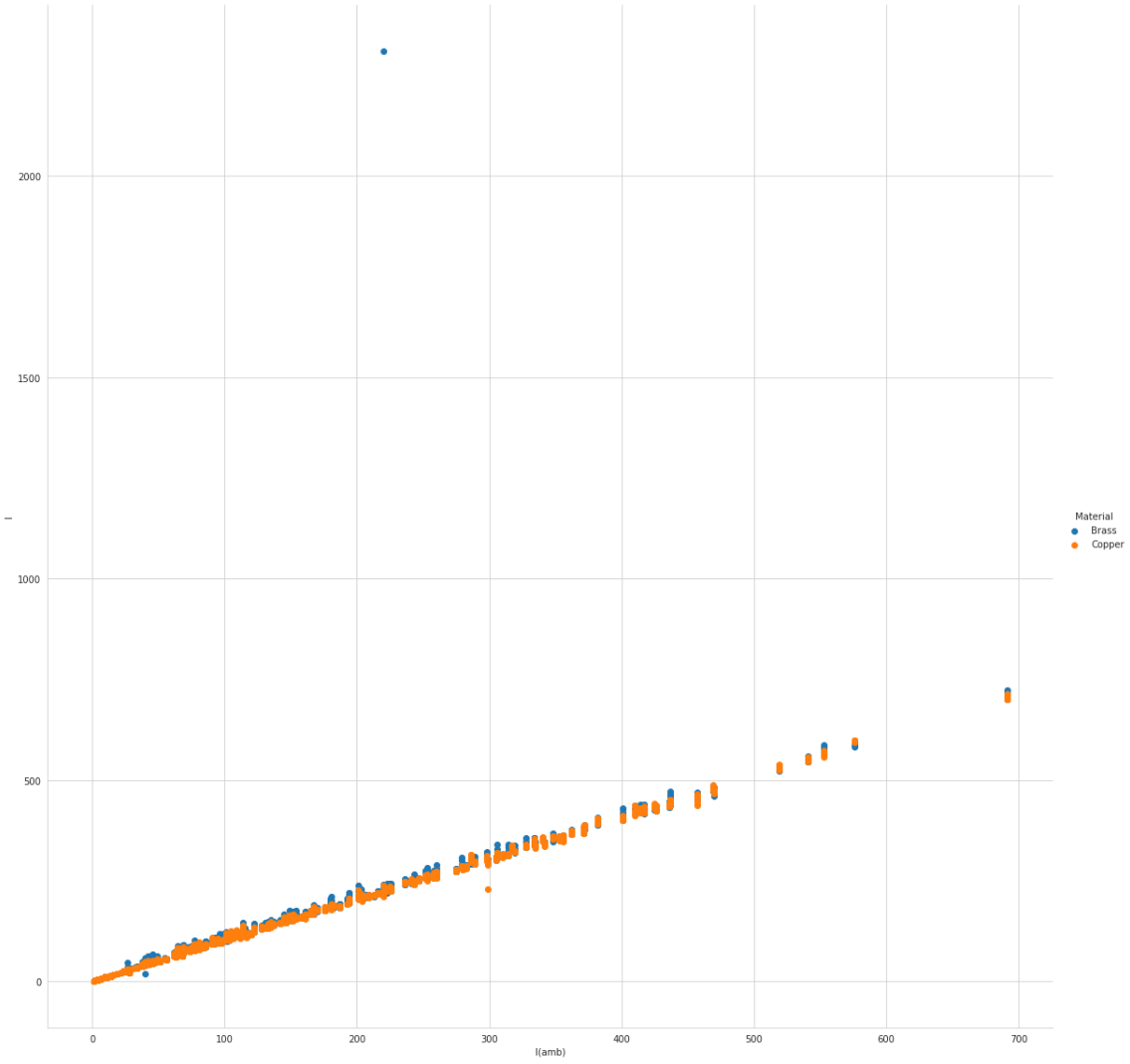

Figure 5.15: Scattered Plot of i VS i_{amb}

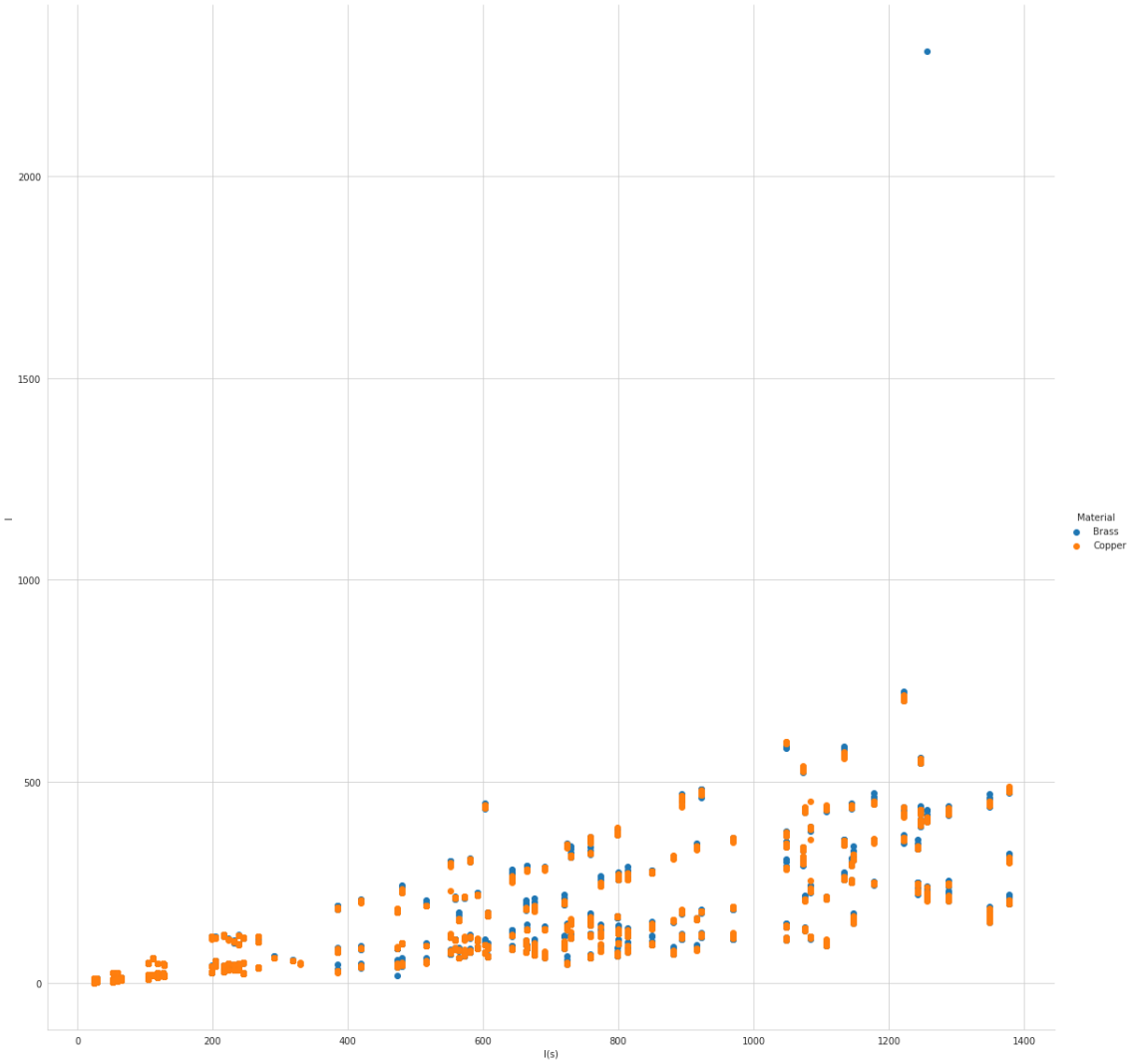

Figure 5.16: Scattered Plot of i VS i_s

For understanding relation between every feature sns.pairplot( ) function is being used.

sns.set$_s$tyle(\whitegrid");

sns.pairplot(df), hue="Material", size=3);

plt.show()

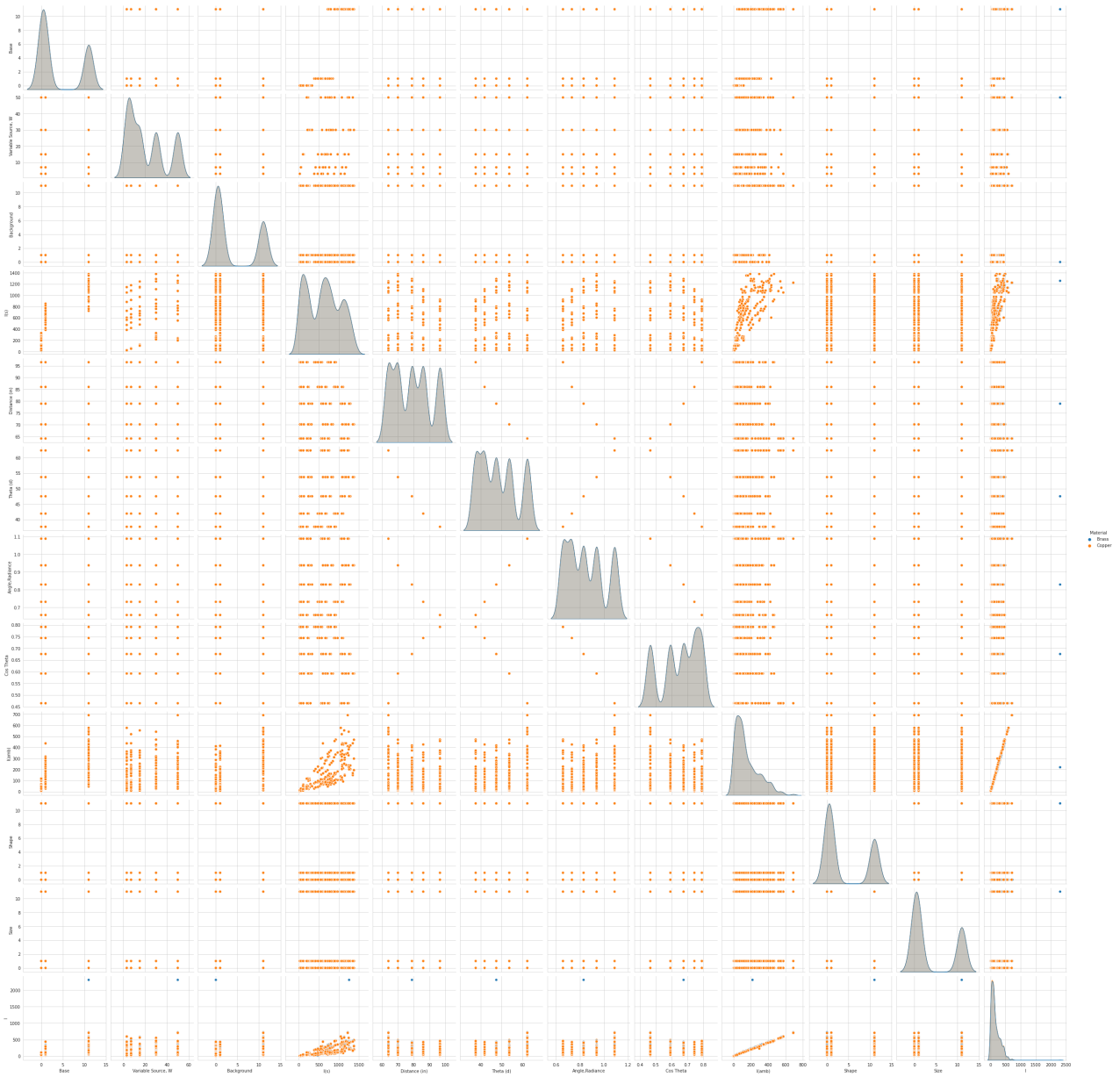

Figure 5.17: Relation Between All Features

Then data set is shown with boxplot.

In descriptive statistics, a box plot is a method of graphically depicting groups of numerical data through their quantities.

For that python's matplotlib.pyplot library's pH.figure( ) function has been used.

Plt.figure(figsize = (15, 17))

Df.boxplot( )

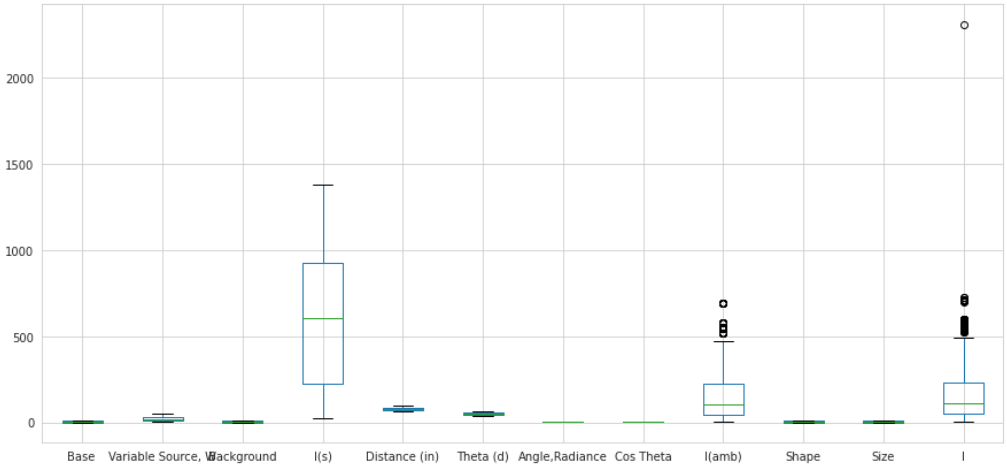

Figure 5.18: Box Plot

5.2.3 Tran-Test-Split

There are 2696 each Brass and copper in total 5392 image data available in our dataset.
At first the dataset is splitted in train-test data by 80:20. Further train set data is changed into train-validation data by 80:20 again.

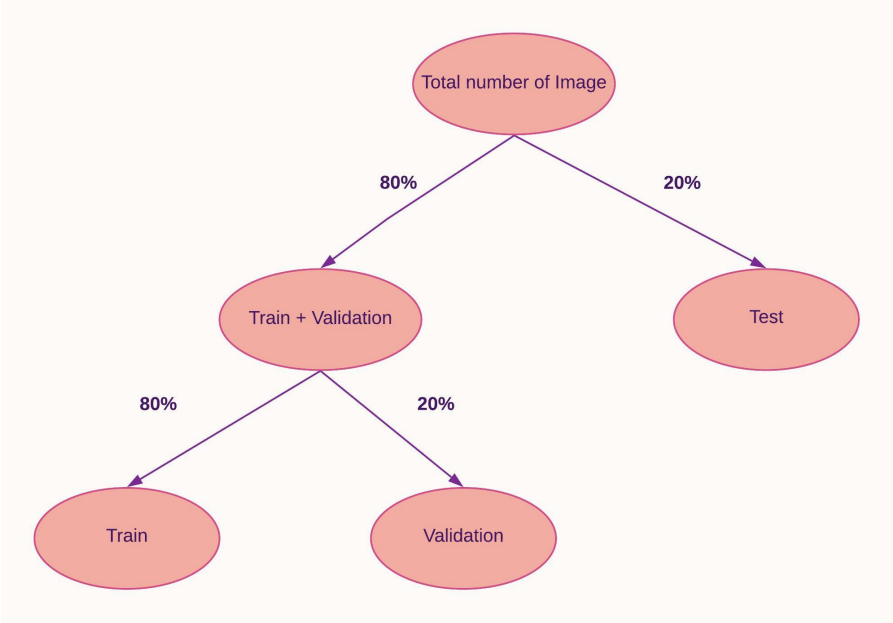

Figure 5.19

Category	Total Number of Image (T)	Number of Training Image (0.8*0.8*T)	# of Validation Image (0.2*0.8*T)	# of Test Image (0.2*T)
Brass	2696	1726	431	539
Copper	2696	1726	431	539

Figure 5.20

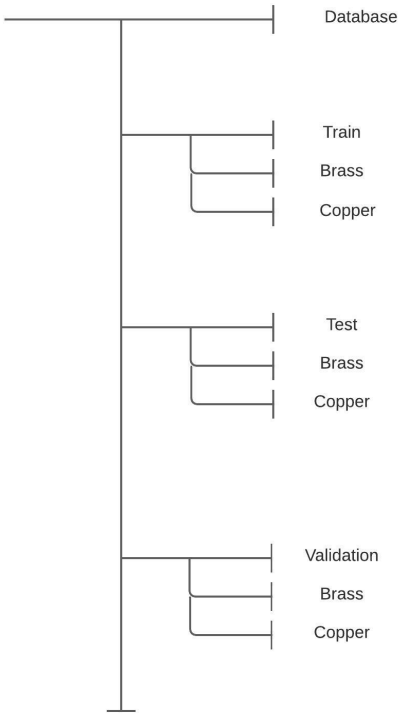

Figure 5.21

5.2.4 Numerical Data:

Numerical data is shuffled and splitted in 80:20 train-test. Then
df= df.sample(frac=1) Shuffle
from sklearn.model$_s$*electionimporttrain*$_t$*est*$_s$*plit*
XTrain, XTest, yTrain, yTest= train$_t$*est*$_s$*plit(df[[\Base", \VariableSource, W", \Background", \I(s*
$0.2, random_state = 42)$

XTrain

	Base	Variable Source, W	Background	I(s)	Distance (in)	Theta (d)	Angle,Radiance	Cos Theta	I(amb)	Shape	Size	I
3684	11	50	1	893	96.5	37.66	0.657291	0.791650	168.0	1	0	173.0
2950	11	30	11	1107	86.0	41.91	0.731467	0.744195	425.0	11	1	436.0
393	11	3	1	1144	70.1	53.70	0.937242	0.592013	289.0	11	0	293.0
444	11	3	1	724	96.5	37.66	0.657291	0.791650	133.0	1	0	136.0
2903	11	30	11	1377	70.1	53.70	0.937242	0.592013	469.0	1	11	479.0
...	...	...	...	...	...	...	...	...	...	...	...	...
2727	11	30	0	1377	70.1	53.70	0.937242	0.592013	194.0	0	0	199.0
2312	1	15	1	572	86.0	41.91	0.731467	0.744195	107.0	11	11	108.0
2289	1	15	1	676	78.9	47.50	0.829031	0.675590	92.0	1	0	93.0
2144	11	15	11	799	96.5	37.66	0.657291	0.791650	371.0	0	11	378.0
2693	0	30	11	217	96.5	37.66	0.657291	0.791650	120.0	0	11	116.0

3240 rows × 12 columns

Figure 5.22: x Train

yTrain

	Material
3684	Copper
2950	Copper
393	Copper
444	Copper
2903	Brass
...	...
2727	Copper
2312	Brass
2289	Brass
2144	Brass
2693	Copper

3240 rows × 1 columns

Figure 5.23: y Train

XTest

	Base	Variable Source, W	Background	I(s)	Distance (in)	Theta (d)	Angle,Radiance	Cos Theta	I(amb)	Shape	Size	I
2368	1	15	11	720	70.1	53.70	0.937242	0.592013	194.0	0	1	203.0
2924	11	30	11	1288	78.9	47.50	0.829031	0.675590	417.0	11	11	420.0
3747	11	50	11	1076	86.0	41.91	0.731467	0.744195	426.0	1	0	428.0
3154	1	30	11	758	64.1	62.30	1.087340	0.464842	319.0	1	1	322.0
3458	0	50	11	239	78.9	47.50	0.829031	0.675590	98.0	0	11	98.0
...	...	...	...	...	...	...	...	...	...	...	...	...
3244	0	50	0	204	64.1	62.30	1.087340	0.464842	42.0	1	1	42.0
701	1	3	1	473	86.0	41.91	0.731467	0.744195	85.0	11	11	87.0
184	0	3	11	25	64.1	62.30	1.087340	0.464842	13.0	1	1	13.0
3302	0	50	0	223	86.0	41.91	0.731467	0.744195	32.0	11	11	34.0
1682	0	15	0	119	86.0	41.91	0.731467	0.744195	15.0	11	11	15.0

810 rows × 12 columns

Figure 5.24: x Test

yTest

	Material
2368	Copper
2924	Brass
3747	Brass
3154	Brass
3458	Brass
...	...
3244	Brass
701	Copper
184	Brass
3302	Brass
1682	Brass

810 rows × 1 columns

Figure 5.25: y Test

5.3 Model Implementation Optimization

5.3.1 Model Implementation

In pre trained model VGG19 and Mobilenet model are used for for image processing.

```
##model building

vgg = VGG19(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

Figure 5.26: Code for VGG19

```
##model building

mob = Mobilenet(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

Figure 5.27: Code for Mobilenet

For both models Imagenet data weight is used.

Imagenet is 14, 197, 122 or image data set according to the heirarchy. Pre layer is not going to be trained again. That's why

layer.trainable= false.

```
#do not train the pre-trained layers of VGG-19
for layer in vgg.layers:
    layer.trainable = False
```

Figure 5.28: Non Pre trainable Parameters

After that a flatter version will be used. Flattening is the process of transforming data into a single number. Softmax layer is the last layer of VGG19 and Mobilenet and here multiclass classification is being used. As these classification is 2 class classification after customization by the help of sense function we need to take the value of 2.

```
#adding output layer.Softmax classifier is used as it is multi-class classification
#customize the layer

prediction = Dense(2, activation='softmax')(x)

model = Model(inputs=vgg.input, outputs=prediction)
```

Figure 5.29: Customizing softmax with dense function

Structure can be seeing by model.summary. 2 crosses are for 2.

block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 2)	50178

Figure 5.30: VGG19 Model Summary

conv_pw_11_relu (ReLU)	(None, 14, 14, 512)	0
conv_pad_12 (ZeroPadding2D)	(None, 15, 15, 512)	0
conv_dw_12 (DepthwiseConv2D)	(None, 7, 7, 512)	4608
conv_dw_12_bn (BatchNormaliz	(None, 7, 7, 512)	2048
conv_dw_12_relu (ReLU)	(None, 7, 7, 512)	0
conv_pw_12 (Conv2D)	(None, 7, 7, 1024)	524288
conv_pw_12_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_12_relu (ReLU)	(None, 7, 7, 1024)	0
conv_dw_13 (DepthwiseConv2D)	(None, 7, 7, 1024)	9216
conv_dw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_dw_13_relu (ReLU)	(None, 7, 7, 1024)	0
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1048576
conv_pw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_13_relu (ReLU)	(None, 7, 7, 1024)	0
flatten_1 (Flatten)	(None, 50176)	0
dense (Dense)	(None, 2)	100354

Figure 5.31: Mobilenet Model Summary

Model	Total Perameter	Trainable Perameter	Non Trainable Perameter
VGG19	20,079,562	50,178	20,024,384
Mob Net	3,329,218	100,354	3,228,864

Figure 5.32: Model Summary

Optimizer

Adaptive Movement Estimation is a gradient descent optimization technique that uses an algorithm. The method is very efficient for problems with a lot of data or parameters. Moreover it takes less memory. It's essentially a hybrid of the 'RMSP' algorithm and the 'gradient descent with momentum' algorithm.

"Sparse categorical cross entropy" is being used for cross function in the optimizer.

It compares the predicted label and true label. That's how it calculates the loss.

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

Figure 5.33: Code for Optimizer Loss Function

Metric

Accuracy is used for metric. The important reason behind it is, it signals the model loss.

$$Accuracy = \frac{CorrectPrediction}{TotalPrediction} \tag{5.1}$$

5.4 Data Analysis

Initially for image data Random Forest, Decision Tree, Artificial Neural Network, Gausian Naive Bayes, Logistic Regression, K Neighbor Model Algorithms are applied. Among them Logistic Regression showed the highest accuracy of 50.49 Percent.

```
#Import svm model
from sklearn import svm


svmModel = svm.SVC(kernel='linear') #model building and fitting
svmModel.fit(XTrain, yTrain)

svmResult = svmModel.predict(XTest)  # Prediction


from sklearn.metrics import accuracy_score   # Performance
svmAccuracy = accuracy_score(yTest,svmResult)
print(svmAccuracy)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:760: DataCon
  y = column_or_1d(y, warn=True)
0.5049382716049383
```

Figure 5.34: SVM Model Implementation and Accuracy

The reason behind the less accuracy is K_d, K_s has been dropped from the numerical data column during the numerical data pre processing.

K_d, K_s is the material's property which defines what the material is and during the process of image data in some models it shows convincing accuracy of what the material is. So it can be predicted that when the image and numerical data will run in a hybrid model simultaneously the matrial detection accuracy will be convincing as per the mathematical model.

During the time of implementation of numerical data with K_d, K_s the object detection result accuracy is 100 Percent.

```
df = df.sample(frac=1) # Shuffle


from sklearn.model_selection import train_test_split
XTrain, XTest, yTrain, yTest = train_test_split(df[["Base ", "Variable Source, W", "Background", "I(s)",
"Distance (in)", "Theta (d)", "Angle,Radiance", "Cos Theta", "I(amb)", "Shape", "Size", "I", 'Ks', 'Kd']], df[["Material"]],
test_size=0.2, random_state=42)
```

Figure 5.35: Including k_s k_d in feature column

```
#Import svm model
from sklearn import svm


svmModel = svm.SVC(kernel='linear') #model building and fitting
svmModel.fit(XTrain, yTrain)

svmResult = svmModel.predict(XTest)  # Prediction


from sklearn.metrics import accuracy_score   # Performance
svmAccuracy = accuracy_score(yTest,svmResult)
print(svmAccuracy)
```

```
1.0
```

Figure 5.36: SVM model implementation and accuracy including k_s k_d

During the implementation of image data on VGG19 and Mobilenet the accuracy is given below:

VGG 19	0.7214
Mobile Net	0.8131

Figure 5.37: Image Data Accuracy

Here 30 epoch has been run.

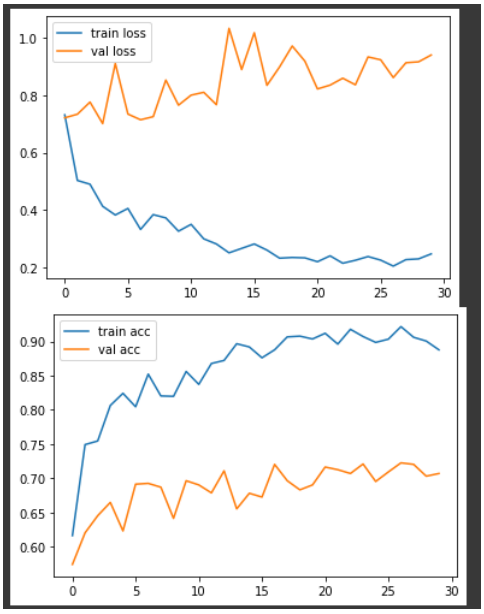

Figure 5.38: VGG 19 Loss and Accuracy Graphs

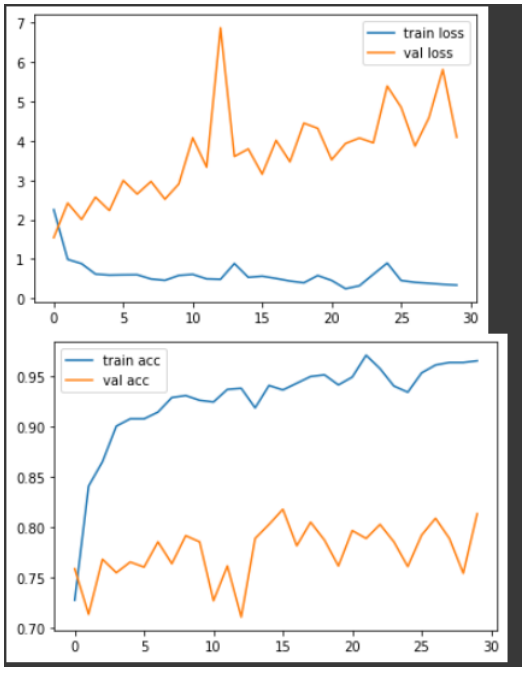

Figure 5.39: MobileNet Loss and Accuracy Graphs

Chapter 6

Result Analysis

6.1 Numerical Data Accuracy

Algorithm	Result
Random Forest	0.2
Decision Tree	0.387
Artificial Neural Network	0.47
Naive Bays	0.4691
Logistic Regression	0.5024
KNN	0.2765

Figure 6.1: Numerical Data Accuracy

6.2 Image Data Accuracy

Mob Net	81.31%
VGG19	72.14%

Figure 6.2: Image Data Accuracy

If material can be identified from image data then the value of k_d, k_s can be found and then after the implementing the numerical data with K_d , k_s in various models most of the time 100 Percent accuracy can be found.

In that case image data model's accuracy is the combined model's accuracy.

Moreover as data is collected in a unfortunate circumstances there can be measurement error in the numeric data or else more accuracy can be expected also from numeric data model.

Chapter 7

Conclusion and Future Work

7.1 Future Work

In future our main goal is to develop a hybrid model which can process both Numeric and Image data simultaneously. As Numeric data processing models give 100 Percent accuracy on detecting different materials upon knowing the value of K_d, K_s and with image data processing models we can differentiate between two materials with convincing accuracy, we can predict that our model can differentiate between two materials when the two data set will be handled together it will give a accurate result about any material.

7.2 Conclusion

An AI can recognize an object based on its posture, movement and shape among other factors. But in some cases, it might not be able to detect the object material. Therefore, analyzing the properties of the surface material like reflection coefficient could be an effective approach. As we could not find any authentic dataset related to our situation, we will simulate an environment that is similar to our surroundings. Then we will collect and analyze surface reflectance data to train our AI to determine the reflection coefficient of object material. We will let the AI predict surface data using different algorithms and choose the most efficient one. We are expecting better results after adding our chosen algorithm as another layer of a conventional object detecting AI.

Bibliography

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[2] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[3] E. Charniak, *Introduction to artificial intelligence*. Pearson Education India, 1985.

[4] Y. Yu and Y. Chen, "Design and development of high school artificial intelligence textbook based on computational thinking," *OALib*, vol. 05, pp. 1–15, Jan. 2018. DOI: 10.4236/oalib.1104898.

[5] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, P. Ravikumar, and T. Pfister, "On concept-based explanations in deep neural networks," 2019.

[6] J. Quackenbush, "Microarray data normalization and transformation," *Nature genetics*, vol. 32, no. 4, pp. 496–501, 2002.

[7] ——, "Microarray analysis and tumor classification," *New England Journal of Medicine*, vol. 354, no. 23, pp. 2463–2472, 2006.

[8] E. Horel and K. Giesecke, "Significance tests for neural networks," *Journal of Machine Learning Research*, vol. 21, no. 227, pp. 1–29, 2020.

[9] H. Wang and H. Zheng, "Model validation, machine learning," in *Encyclopedia of Systems Biology*, W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, Eds. New York, NY: Springer New York, 2013, pp. 1406–1407, ISBN: 978-1-4419-9863-7. DOI: 10.1007/978-1-4419-9863-7_233. [Online]. Available: https://doi.org/10.1007/978-1-4419-9863-7_233.

[10] E. Angel and D. Shreiner, *Interactive Computer Graphics with WebGL*. Addison-Wesley Professional, 2014.

[11] P. Soucy and G. W. Mineau, "A simple knn algorithm for text categorization," in *Proceedings 2001 IEEE international conference on data mining*, IEEE, 2001, pp. 647–648.

[12] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[13] Y. Li and R. Anderson-Sprecher, "Facies identification from well logs: A comparison of discriminant analysis and naıve bayes classifier," *Journal of Petroleum Science and Engineering*, vol. 53, no. 3-4, pp. 149–157, 2006.

[14] I. Mesecan and I. Ö. Bucak, "Searching the effects of image scaling for underground object detection using kmeans and knn," in *2014 European Modelling Symposium*, IEEE, 2014, pp. 180–184.

[15] K. Inthajak, C. Duanggate, B. Uyyanonvara, S. S. Makhanov, and S. Barman, "Medical image blob detection with feature stability and knn classification," in *2011 Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, 2011, pp. 128–131.

[16] D. I. H. Putri, C. Machbub, *et al.*, "Object detection and tracking using sift-knn classifier and yaw-pitch servo motor control on humanoid robot," in *2018 International Conference on Signals and Systems (ICSigSys)*, IEEE, 2018, pp. 47–52.

[17] M. Rashidan, Y. Mustafah, S. Hamid, N. Zainuddin, and N. Aziz, "Detection of different classes moving object in public surveillance using artificial neural network (ann)," in *2014 International Conference on Computer and Communication Engineering*, IEEE, 2014, pp. 240–242.

[18] A. Ahmed, A. Jalal, and K. Kim, "Region and decision tree-based segmentations for multi-objects detection and classification in outdoor scenes," in *2019 International Conference on Frontiers of Information Technology (FIT)*, IEEE, 2019, pp. 209–2095.

[19] D. M. Gavrila and V. Philomin, "Real-time object detection for" smart" vehicles," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE, vol. 1, 1999, pp. 87–93.

[20] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[21] Y. Zheng, C. Yang, and A. Merkulov, "Breast cancer screening using convolutional neural network and follow-up digital mammography," in *Computational Imaging III*, International Society for Optics and Photonics, vol. 10669, 2018, p. 1 066 905.

[22] U. Raghavendra, H. Fujita, S. V. Bhandary, A. Gudigar, J. H. Tan, and U. R. Acharya, "Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images," *Information Sciences*, vol. 441, pp. 41–49, 2018.

[23] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *2007 IEEE international symposium on signal processing and information technology*, IEEE, 2007, pp. 11–16.

[24] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences*, vol. 10, no. 5, p. 1897, 2020.

[25] T. Sugata and C. Yang, "Leaf app: Leaf recognition with deep convolutional neural networks," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 273, 2017, p. 012 004.

[26] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery," *Remote Sensing*, vol. 10, no. 7, p. 1119, 2018.

[27] A. G. Howard, "Some improvements on deep convolutional neural network based image classification," *arXiv preprint arXiv:1312.5402*, 2013.

[28] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *International workshop on human behavior understanding*, Springer, 2011, pp. 29–39.

[29] A. K. Dubey and V. Jain, "Comparative study of convolution neural network's relu and leaky-relu activation functions," in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, Springer, 2019, pp. 873–880.