7-2021

Material Detection with Thermal Imaging and Computer Vision: Potentials and Limitations

Jared Poe
University of Arkansas, Fayetteville

Material Detection with Thermal Imaging and Computer Vision:
Potentials and Limitations

A thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

by

Jared Poe
University of Arkansas
Bachelor of Science in Mechanical Engineering, 2019

July 2021
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

_____
Zhenghui Sha, Ph.D.
Thesis Director

_____
David Jensen, Ph.D.
Committee member

_____
Yue Chen, Ph.D.
Committee member

ABSTRACT

The goal of my masters thesis research is to develop an affordable and mobile infrared based environmental sensoring system for the control of a servo motor based on material identification. While this sensing could be oriented towards different applications, my thesis is particularly interested in material detection due to the wide range of possible applications in mechanical engineering. Material detection using a thermal mobile camera could be used in manufacturing, recycling or autonomous robotics. For my research, the application that will be focused on is using this material detection to control a servo motor by identifying and sending control inputs based on the material in an image. My thesis is driven by the following research question: how does infrared imaging compare to visible light in terms of prediction accuracy both in ideal and non-ideal scenarios? This question is motivated by the fact that there is a lack of knowledge on the distinction between the qualities of thermal imaging and RGB imaging for computer vision, especially with the use of an affordable mobile camera. To address this gap and answer the research question, this thesis aims to achieve three objectives: 1) to create a dataset and train a thermal imaging convolutional neural network (CNN) for material detection, 2) to create a testbed that will utilize the material detection for the control of an actuator, and 3) to compare the performance of thermal imaging vs. RGB imaging in terms of detection accuracy for both ideal and non-ideal scenarios. To achieve these objectives, a large number of infrared and RGB images must be collected and pre-processed to create a dataset for the training of CNN models and the prediction of material types. A protocol must also be developed to establish the real-time communication between the mobile thermal device and the actuator to relay this material information. An in-depth understanding is gained of the benefits and drawbacks in terms

of accuracy's in ideal and non-ideal scenarios while using an affordable thermal mobile camera as opposed to traditional RGB cameras for material detection. These methods were tested on a small-scale prototype device consisting of a Raspberry Pi and a SG90 servo motor. The way each data type is pre-processed is different, e.g., using dynamic range quantization vs. standardization, in order to obtain the best model performances. Our results show that the thermal imaging model performed better than RGB model in non-ideal scenarios where is was dark (52% average accuracy vs. 46%), but was not able to outperform RGB imaging in ideal scenarios (74% average accuracy vs. 95%). While this conclusion is not surprising and falls in our expectation, the quantification of the differences between RGB imaging and thermal imaging for material detection and the systematic approach developed are the new knowledge generated. It reveals the potentials and limitations of infrared image-based computer vision and therefore sets the foundation for future work with thermal imaging as it relates to environmental sensing, autonomous applications, and under what conditions this application can be made.

ACKNOWLEDGEMENTS

I would like to thank Dr. Sha for his support and guidance during this study. Without whom I would never have made it to the writing of this thesis. The amount of professional growth I have experienced under his guidance is extraordinary. I would also like to thank the Mechanical Engineering department at the University of Arkansas and the Office of Vice Chancellor for Research and Innovation for financial aid and funding for this research. Thank you to Dr. David Jensen, associate professor at the University of Arkansas, and Dr. Yue Chen, assistant professor at the University of Arkansas, for their willingness to serve on my thesis committee and for their much needed feedback on my work and how it can be improved. All of the System Integration Design Informatics Laboratory members (Laxmi Poudel, Xingang Li, Yinshuang Xiao, John Clay, Molla Rahman, and Sumaiya Tanu) deserve a huge thank you for the feedback and answered questions over the past year and a half that I have received from them.

I want to thank Dr. Youngjun Cho from the department of computer science at University College London, Dr. Charles Xie from the Institute for Future Intelligence, and Chenglu Li, a Ph.D. student from the University of Florida, for their added support and assistance during this research. For the many questions that they answered and the helpful feedback they have provided.

Last, but certainly not least, I want to thank my wife for her never-ending support through this process. She has never hesitated to encourage me every step of the way. I would not be where I am today if she were not by my side.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Background and Motivation

In recent years, there has been a dramatic increase in the exploration of computer vision. Computer vision was introduced around the 1960's and since 2010 this topic has been growing exponentially [5]. When this topic was first being explored, the majority, if not all, of the effort was being poured into visible light images which consist of Red, Green and Blue color channels (RGB). However, in the early 2000's there started to become some interest in extending this knowledge about computer vision to the infrared spectrum [6]. Since that time, there has been work done for object detection, velocity calculation, and trajectory prediction using infrared (or thermal) image-based computer vision techniques [7] [1]. In this master thesis research, I propose to give quantitative proof on the merits of thermal imaging as it is compared to RGB imaging for material detection and to determine the best approach to obtaining this proof. Thermal imaging has the obvious benefit of being able to detect temperature values in a particular scene/image which visible light imaging cannot. This allows flexibility on the amount of environmental sensing that can be done with just one device. Using computer vision techniques in the infrared spectrum will allow this temperature data to be leveraged for different applications than can be achieved with the visible light spectrum using RGB images. Although the overarching goal of this research is to have a comprehensive infrared-based environmental sensing system for the closed-loop control of unmanned ground vehicles, the immediate objective of this thesis is focused on material detection and to use this information for the control of a simple prototype device composed of a Raspberry pi and a servo motor. In future work, this material detection can be utilized for controlling a robot's operating conditions. For example, the

1

robot can adapt the speed and torque of the motors automatically based on the pathway material that is present, e.g., concrete vs. grass. In addition, this material detection could prevent the robot from coming into contact with materials that are unwanted. If the robot is tasked to follow a side walk, anytime the robot started to encounter grass or dirt, proper adjustments could be made to keep on the sidewalk. To achieve this objective, we must first prototype this control system with a simple device and in the process answer these research question: how does the performance of thermal imaging compare to that of RGB imaging both in ideal (in daytime) and non-ideal scenarios (in darkness)?

While there are many different approaches for material and object detection using sensors such as lidar, laser scanners, etc. that can already be used to obtain autonomous robot functionality with environmental sensing, the cost of such devices is a major barrier that can impede its application in some circumstances. For example, applications like personal use such as in the case of disabled persons or private projects. In the case of disabled persons, whether they are in a wheelchair or walking without sight, a cheap alternative for environmental sensing is important. Another important area where an inexpensive device would be helpful is in the education system. This device could be used easily to aid in student learning objectives in robotics and control. With respect to manufacturing, there may be a need to use an inexpensive sensing method for material detection and/or sorting for a particular temporary production process. With the proposed mobile thermal device, the cost will be significantly less than alternative sensing methods while allowing for high flexibility in the modes of sensing that can be achieved. Some work has been done to use computer vision techniques with RGB imaging for robot control, such as the work done by Christian Bodenstein et al. [8] using a mobile phone and as done in Robotic Weed Control System for Precision Agriculture [9] where they simply used a

standalone camera. In our literature review, there are no published research studies in the area of closed-loop control using thermal imaging or for comparing the performance of RGB and Thermal imaging for such an application.

Using thermal imaging has an important benefit over RGB: thermal imaging is not dependent on having sufficient lighting. This will allow for materials to be identified even in low lighting scenarios or even no lighting at all. In the next section, I present a review of the relevant literature which helps identify the research gaps and questions and how to approach them.

1.2   Research Questions and Objectives

As before mentioned, the question that this thesis seeks to answer is how does the performance of using thermal imaging for material detection compare to using RGB imaging in ideal and non-ideal scenarios. This is a question that has not been answered in previous literature and by answering this question, a foundation for future application and development is established. In answering the questions, the best methods for image processing and network development are discovered and applied. There are some important objectives that have to be achieved in order to answer this important question. First, a dataset has to be collected, both for thermal data and RGB data. This data collection was accomplished by recording videos of the appropriate material and extracting the data from those videos in order to create the image datasets for thermal and RGB images. The data extracted for thermal imaging consisted of temperature values, while the data extracted for RGB imaging consists of pixel intensities of the three different color channels. This difference is important when looking at training models for each data type, which leads to the second objective. The second objective that must be achieved is to determine the optimal image processing techniques for each data type. On the one hand, the

thermal data consists of temperature matrices while the RGB data consists of images containing three channels of pixel data. Therefore, the way that each of these data types are pre-processed must be accounted for in the models. The third objective is to develop the CNN architectures that will be trained using these collected RGB and thermal datasets. How must the thermal CNN differ from the RGB CNN architecture and how must the hyperparameters be tuned in order to obtain high accuracy's. The fourth objective is to use the trained thermal model to control a servo motor based on the material present in an image. By completing this motor control, it is demonstrated how this new knowledge about thermal imaging can be applied in future work.

1.3   Outline and Road Map

    The general outline and road map can be seen in Figure 1.1. In the road map, it can be seen that Stage 1 is broken up into parts 1.1 and 1.2. The first part of stage one is used to collect thermal data and train the thermal imaging CNN model on that data. It also includes the data pre-processing on the thermal data. This processing is done using the Dynamic Range Quantization and cropping the center portion of each thermal matrix. The second part of stage one covers the data collection and training of the RGB imaging CNN model with the collected data. The pre-processing of this data is completed by using featurewise standardization and resizing the images before feed-forwarding to the CNN. Stage 2 is the implementation of the thermal model for the control of a servo motor. This stage is used as a way to demonstrate the future capabilities of this research. The trained thermal model is used to identify the material in a given image and that data is the communicated to the Raspberry Pi via TCP/IP communication. Then the Raspberry Pi sets the servo to a predetermined angle based on the material detected and the motor then updates the Raspberry Pi when the angle is changed. The third and final stage is the quantitative

comparison between the accuracy's of the RGB model and thermal model with the validation dataset. In this stage, each model is trained 10 different times on the thermal model and 5 times on the RGB model and the accuracy of each model is calculated for each trained model. The average, maximum, and minimum of the accuracy's are then compared to give hard evidence of the performance that each model can obtain. The accuracy of the thermal model and the RGB model will be compared in both ideal scenarios (daytime) and non-ideal scenarios (nighttime) to determine how well the thermal model compares to the RGB model in a range of circumstances. By using the validation dataset, a real application is replicated because this validation data was collected at a different time and place than the training dataset and is used to show the real life accuracy's that the models can produce.
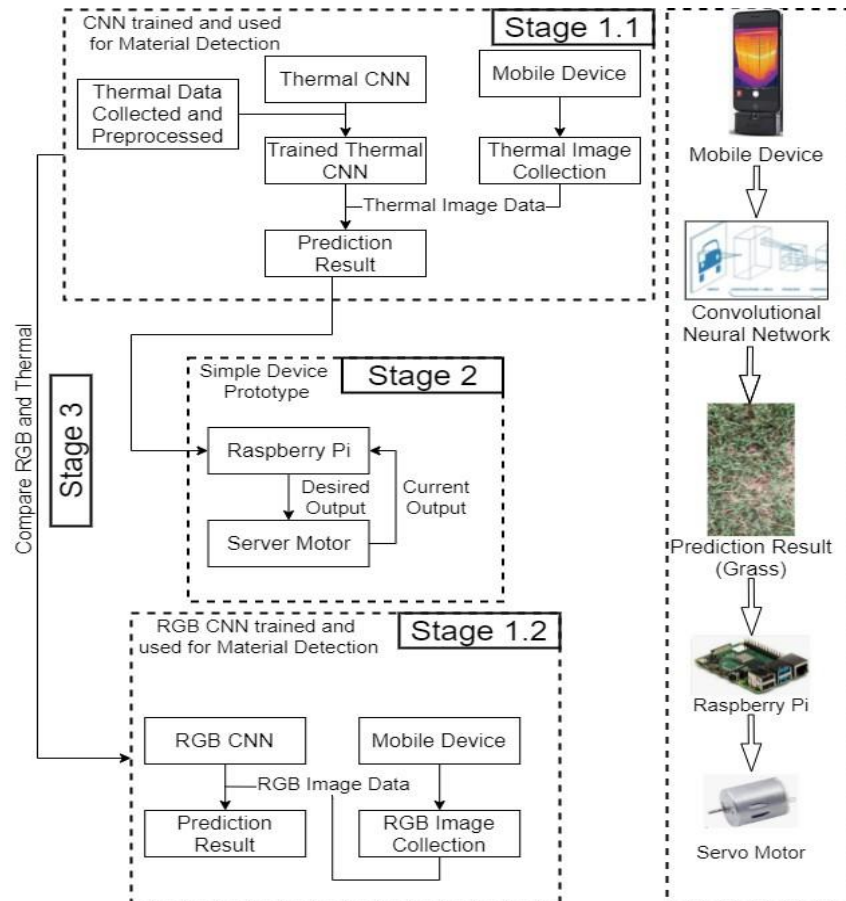
Figure 1.1: Outline and Road Map

2   LITERATURE REVIEW

2.1   Relevant Literature

Computer vision seeks to create computational systems that can analyze and interpret images and videos as humans are able. This concept has proven to be very effective and useful in areas such as autonomous vehicles, face recognition, object detection, and so much more [1]. Among many computer vision techniques, one particular deep learning-based method, called Convolutional Neural Network (CNN) [2], has been widely used recently. For example, Jangblad [1] used thermal imaging for object detection to aid in the landing of airplanes by detecting important landmarks such as the runway, approaching lights and PAPI lights. In this study, he found that the prediction time was longer in higher resolution images, but the accuracies were better with higher resolution images. This detection information, however, was meant to be used by the pilots flying the planes when they are landing in poor weather conditions. It was not used for autonomous control of these aircraft and was not compared in performance to RGB imaging.

Other studies involving thermal imaging have been done for object detection. One study, performed by Zingoni and his co authors [7], used a flexible algorithm for detecting moving objects. The algorithm that they developed involved evaluating each pixel of a video, updated frame-by-frame, and rejecting pixels that had no significant change between frames. The results of this study showed that moving objects could be detected with a detection rate of 96% and that there would only be one false alarm for every 14 video frames. However, the use of this algorithm for the control of an autonomous robotic platform was neglected and the comparison was never made between RGB and thermal imaging.

There have been medical studies conducted using thermal imaging as well. One example of this is a study conducted by Cho and his co-authors [10] that deals with the monitoring of the human respiratory rate. Thermal imaging can also be used to detect inflammation areas and even be used to monitor and help in treating arthritis [11]. These studies used temperature data to identify these conditions. These studies are an example of some of the inherent benefits that thermal imaging has over RGB in that they use temperature data and do not rely on visible light.

Within the application of material recognition, Dr. Youngjun Cho and his coauthors have developed a deep-learning approach using thermal imaging [12]. They accomplished this using a CNN in MATLAB's "MatConvNet" framework. In their study, they were able to achieve a prediction accuracy of 98% on indoor materials and an accuracy of 89% on outdoor materials. However, when the outdoor materials were wet (i.e. during rainfall) the accuracy of the trained network dropped to below 5%. This is most likely due to the substantial change in the emissivity of the materials when the materials are exposed to moisture. Also, the accuracy in real application scenarios, the accuracy's dropped to approximately 68% for outdoor materials. The CNN structure that they used to obtain this amount of accuracy was drawn from the study performed by Jaderberg et al. [13] which provides a robust CNN architecture that can handle a wide range of variations in data. This was helpful in the work done by Cho et al. [12] because of the wide range of materials and the variances in the data. Again, it should be noted that this study never compared the performance of the thermal imaging to RGB imaging in this application of identifying material types.

It is worth noting that with regard to applying their work to real world applications, Cho et al. left it for future work. They discuss the possibilities for integrating this mobile thermal

camera for use with automatic cleaning robots, such as vacuum, sweeping, and mopping robots for floor type detection. Another real world application was to use this technology as a third eye for impaired people who use wheelchairs or for care takers who have limited visibility of the footpath they are walking. In this thesis, Dr. Cho's work is leveraged to set a foundation to explore the benefits of using thermal imaging for material detection compared to RGB imaging. There are some key differences in the methods used in this thesis, but several of the methods from Dr. Cho's work are strongly utilized. These differences and similarities will be discussed in later sections. This thesis also seeks to apply the resulting thermal model for servo motor control as a way of demonstrating the future possibilities of autonomous robotic control.

In summary, there has been an increase in the amount of work done in computer vision on thermal imaging in recent years. However, research gaps exist in the lack of application of thermal imaging in the control of a system and in comparison between the use of thermal imaging with that of RGB imaging to determine the performance of each method. To fill these gaps, a thermal and RGB image dataset must be collected for detecting materials that are of interest and the best methods must be developed for processing these datasets. Although this type of application for real-time control has be utilized with RGB imaging [9], the use of thermal imaging has been widely neglected. Because thermal imaging is being used, the possibilities for analysis and the range of applications is extended. It can be used in the dark (or low lighting) and the images collected can be used for thermal analysis of buildings in addition to being used for material recognition. Before discussing the details of this thesis work, there is some technical background that should be covered.

2.2   Technical Background

This section focuses on background information for the methods used in this thesis. The topics that are covered include CNN terminology and structure, the differences between RGB and thermal image data, and different processing techniques for RGB and thermal data.

2.2.1   CNN Terminology and Structure

There are some key concepts and definitions that should be discussed regarding CNNs. The first concept is Kernel (K): this refers to a set matrix that is used to scan, or stride over the input matrix (I) and perform multiplication and addition on each stride (see Figure 2.1).
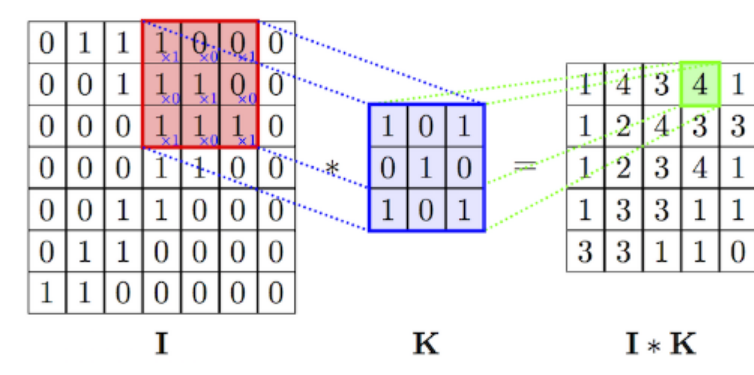
Figure 2.1: Convolution Operation by a 3x3 Kernel [1]

A stride can be (1,1), which means the kernel moves one pixel on each stride horizontally and one pixel when it scans vertically. The same goes for (2,2), (3,3), etc. The larger the stride is, the smaller the output convolved image will be. Another way of manipulating the size of the output convolved image is called padding. Padding is when there is an extra perimeter of pixels placed around the input image. These extra pixels are typically assigned a value of zero, which is called zero padding (see Figure 2.2a). As shown in the figure, because of the padding, the convolved image (green) is of the same dimensions of the input image (blue). Now that the initial

9

convolution operation has been performed, the next step is called *pooling*. There are two types of

pooling operations that can be done: max pooling and average pooling. Max pooling simply

takes the largest value that is contained in the kernel of the input data at a certain stride and

places it in the corresponding location in the output matrix (see Figure 2.2b). Average pooling is

the same concept as max pooling, except it takes the average value of all the elements of a kernel

[2]. These convolution and pooling operations are used to extract important features from the

input images, such as edges, corners, etc.

(a) Padding of a 5x5 Input Image
to Produce a 5x5 Convolved Image

(b) Pooling Operations by a 2x2 Kernel
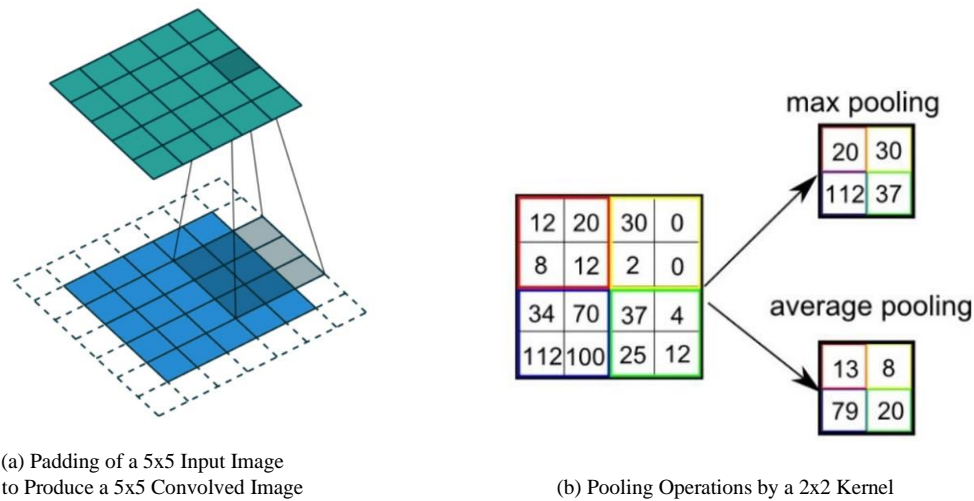
Figure 2.2: Pooling and Padding [2]

Activation functions are used in a CNN when a linear model is not able to capture all the

variations in the data while training the network. These activation functions are able to more

adaptively train the network even with large variations in data, thus allowing it to learn more

complex patterns. Common activation functions consist of sigmoid activation and Rectified

Linear Unit (ReLU) activation functions. The ReLU (Figure 2.3) is the more commonly used

activation function as the sigmoid activation function saturates and is no longer useful in

training. The equation for the sigmoid activation function is given as: $\text{sig}^{(t)} = \frac{1}{1+e^{-t}}$. The

sigmoid activation function is typically only discussed for historical purposes as it is not readily used in neural networks at present [3]. The ReLU does come with one downfall, that is the "dying ReLU". This is caused due to the zero value for any negative inputs, which in turn can cause some nodes to remain untrained and essentially "die". One other activation function to note is the Softmax. The softmax is generally used as the final layer in a CNN for multi-class classification.
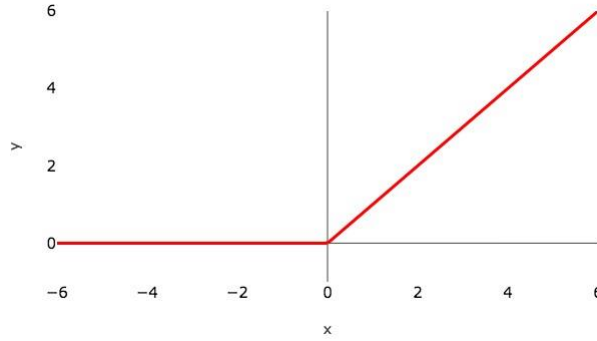
Figure 2.3: Rectified Linear Unit Activation Function [3]

A loss function compares the predicted value of an image during network training and compares that to the actual value given by the dataset. The most commonly used, and the method used in this thesis, is the Cross-Entropy Loss function (Equation 2.1). This loss function is used when a softmax classifier is present in a model.

$$L_{CE} = -\sum_{i=1}^{n} t_i \log p_i$$

(2.1)

Where t_i is the truth label and p_i is the softmax probability value for the ith class.

In a CNN, batch size is the hyper-parameter that refers to the number of samples that are utilized before the model parameters are updated. A sample is any single row of data, in our case, one sample would be a single image. The common values that are used for batch size are 32, 64, 128, and 256. Another key hyper-parameter is called the number of epochs. This refers to the number of times that the given training dataset will be iterated over until a sufficiently small

11

error is obtained in the model. The final hyperparameter that we will discuss is the learning rate of the network. The purpose of this hyperparameter is intuitive. The faster the learning rate is, the faster the network approaches an optimal value in terms of the number of epochs needed. On the other hand, the slower the learning rate, the more epochs the network will need to reach an optimal value. Furthermore, the faster the learning rate, the more rapid the changes in the model and this can cause poor results. However, if the learning rate is too slow, this can cause the network to get stuck and lead to poor results. This is why the learning rate is often considered the most important hyperparameter.

Backpropagation is an essential step in the CNN training process. After each batch of data has been fed through the network and the loss values have been calculated, the parameters and weights of the neurons are updated by the backpropagation step. Backpropagation is much like it sounds, after the weights and parameters have been determined in the forward direction of the network, the calculated loss is propagated backwards through the network to update weights and parameters of the neurons. In this way, the optimal weights are determined for the neurons. This step is repeated for every batch of data until all of the dataset has been used, which constitutes one epoch. When Training a neural network, overfitting needs to be avoided. Overfitting occurs when any single neuron is relied on too heavily for the correct classification of the input. This might mean that the network performs extremely well on the training data, but when unseen data is introduced, it will perform poorly. This overfitting problem can be solved by adding some dropout regularization layers in the network. By doing this, the network is forced to not rely so much on any one neuron to classify the input correctly and the network consequently performs better on unseen data.
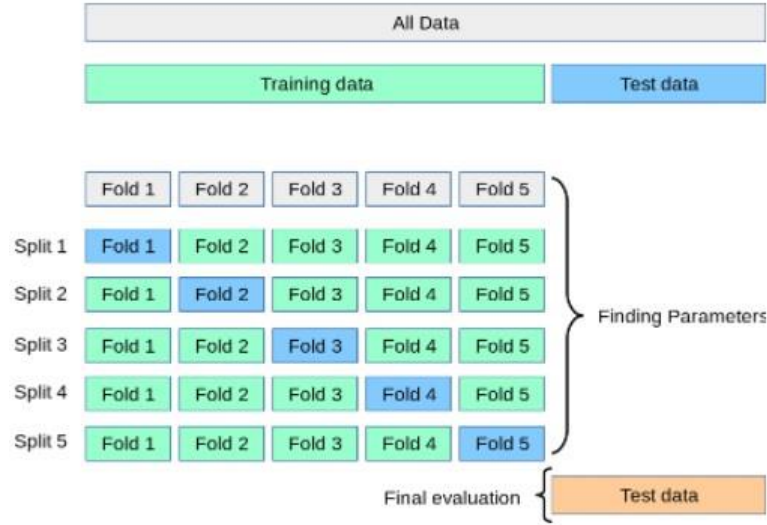
Figure 2.4: K-Fold Cross Validation [4]

K-Fold Cross Validation is the process of splitting a given dataset into K different partitions called folds. The network is then trained on K-1 of these folds, while one fold is used as a test set. This process continues until every fold has an opportunity to be set as the test set. In this way, the network is able to better fine tune the hyperparameters without loosing any data to a strictly dedicated test set (see figure 2.4).

In this thesis, K-fold validation was not used. The model was tuned by manually tuning the hyperparameters, the performance was optimized based on human judgement, then using those hyperparameters, the thermal network was trained 10 different times (5 times for RGB model) using an 80/20 split of training and test data. This train and test data were selected randomly, thus, every time the model is trained, the achieved accuracy is different. Therefore, by training the model on the dataset multiple times, the overall accuracy's can be obtained and a conclusion of the robustness of the model can be made. Before any of these techniques could be implemented, the image dataset needed to be preprocessed to obtain the best performance possible.

2.2.2   Image Processing Techniques

Image processing is a highly important step to any image driven CNN. There are many techniques ranging from changing the dimensions of the image to cropping the image to only include specified pixels to completely altering the pixel values across the entire image. The most important pre-processing methods that we use for our thermal images and RGB images in this thesis are Dynamic Range Quantization and Image Standardization, respectively. Each of these concepts will be discussed in detail later. First, the difference between the thermal data and RGB data must be established.

The Flir One mobile thermal camera uses the temperature values of the scene and then color maps that temperature data into a colorful image that one can observe on the phone screen. Thus, when the data was collected, the images that were captured consisted of this heat map. These heat map images are not adequate for network training. When using these heat map images, there are no defining patterns between one material or another that the network can learn, thus creating a very poor network with little to no accuracy in classifying unseen data. However, using the SmartIR application for the Flir One camera, the raw temperature values are saved to a special file which can then be accessed later to extract this raw thermal data. By using these raw temperature matrices and the DRQ processing method discussed in the next paragraph, the CNN model accuracy increases dramatically.

Dynamic Range Quantization (DRQ) is considered for thermal data. The DRQ method involves scanning the entire raw thermal matrix obtained from the thermal camera, identifying the maximum and minimum temperature values and then uses these values to "quantize" the remaining pixels in the thermal matrix. The equation that results from this process is shown in Equation (2.2):

14

$$A_{new} = \frac{(A(x, y) - min)}{max - min} \qquad (2.2)$$

This equation allows us to reduce the environmental effects captured in the image. Therefore, regardless of the absolute temperature due to the time of day or what time of year, by using the DRQ method, these effects are taken out of the image and the temperature values are only compared to neighboring pixels. From Equation (2.2), A(x,y) is the value of each pixel being processed, this value is then scanned over each pixel of the image. The min value is the minimum temperature value of the entire image and the max value is the maximum temperature value. Thus, when these processed images are feed-forward into the CNN, each pixel is not being learned absolutely, but rather it is being learned relative to neighboring pixels. Therefore, for varying materials with varying porosity and texture, these changes in pixel values are specific to that certain material and not for what the absolute temperature that material may be experiencing.

The image data that was collected with the normal RGB camera are comprised of three channels: Red, Green, and Blue. The CNN then has to train on these three channels of red, green, and blue pixel values. While this makes the training time for the RGB network longer than the thermal network (where there is only one channel), the amount of data is inherently greater, which produces a better accuracy in scenarios with good lighting. However, this good accuracy only happens in the ideal lighting scenario; this will be discussed in depth later. When considering what type of processing technique to use for RGB images, it is clear that the DRQ method will not help because in these three channels (over the entire image), there would be values that are zero and others that are 255. Thus, making the DRQ equation reduce to simply dividing each pixel by 255. This method is often used in some applications, but this leads to poor accuracy for material detection.

Image Standardization was used for the RGB images. This method is similar to the DRQ method, but with some important differences. The equation used for this method is shown in Equation 2.3 [14]:

$$A_{new} = \frac{(A(x, y) - \mu)}{\sigma} \tag{2.3}$$

The μ value is the mean of the image pixel values and the σ value is the standard deviation from the mean. This allows the image data to have properties as a Gaussian distribution where the mean is zero and the standard deviation is 1, in other words, the mean is removed from the image, which in turn aids in the CNN learning and classifying process by centralizing the data. There are two ways that this standardization can be applied in a CNN. The first way it can be applied is samplewise, where each image is standardized by its own standard deviation. The second is called featurewise standardization and this is where each input image is standardized by the standard deviation of the entire dataset.

3   CONVOLUTIONAL NEURAL NETWORK: SETUP AND EXPERIMENTATION

3.1   Dataset Collection

The data for the training and validation contained in this thesis were collected periodically over the course of a year using an affordable mobile camera. Using this affordable thermal camera has the benefit of being obtainable by almost anyone, but it also has some drawbacks with respect to quality and performance of the data acquired. These drawbacks cause some issues with processing the thermal data and will be discussed later. The first group of data was collected in July of 2020, however, this data was discarded as we purchased newer equipment and recollected data. The second group was collected with a new equipment in September of 2020, this was the largest collection. Then there was more data collected to extend our data-set further in April and May of 2021. This data was collected by recording a video of the materials from a distance of approximately 30 inches from the surface, the overall flow of this process can be seen in Figure 3.1. The mobile thermal camera plugs straight into the charging port of the phone and the SmartIR app is used to capture the video during data collection.

After collecting this data, a .mp4 file is saved as well as a .vir file. The raw thermal data is stored in this .vir file and it is possible to extract the data in a particular way which will be discussed in detail later. This raw thermal matrix was used for training and testing
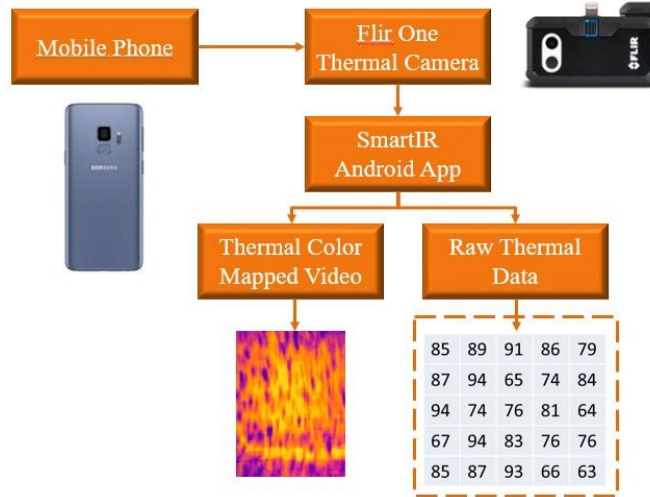
Figure 3.1: Thermal Dataset Collection

after preprocessing with the DRQ method. In Figure 3.2, an example of the color mapped

thermal image can be seen and the final DRQ image and size can be seen as well. The original

thermal resolution is 160×120, however, there was a study that shows that when using a cheap

thermal camera, the temperatures that are at the edges of the frame are sometimes inaccurate.

Therefore, by using the 60×60 cropped portion from the center of the frame, the possibly

skewed values from the edges are removed and the most accurate data is retained [12]. The

cropped DRQ image is what is used to train and validate the thermal model. The raw thermal

data was used for training because, although the color mapped image looks more pleasing to the

naked eye, the raw temperature values allow the CNN to more readily identify differences

between the materials by learning the distinct thermal patterns. When using the color mapped

images, the network has a very poor validation accuracy and, in some cases, yielded a zero

accuracy. The image processing challenges are discussed at length later.

As for the RGB data, the data was collected much the same way as the thermal data, by

recording a video of the material approx. 30 inches away from the surface. The recorded
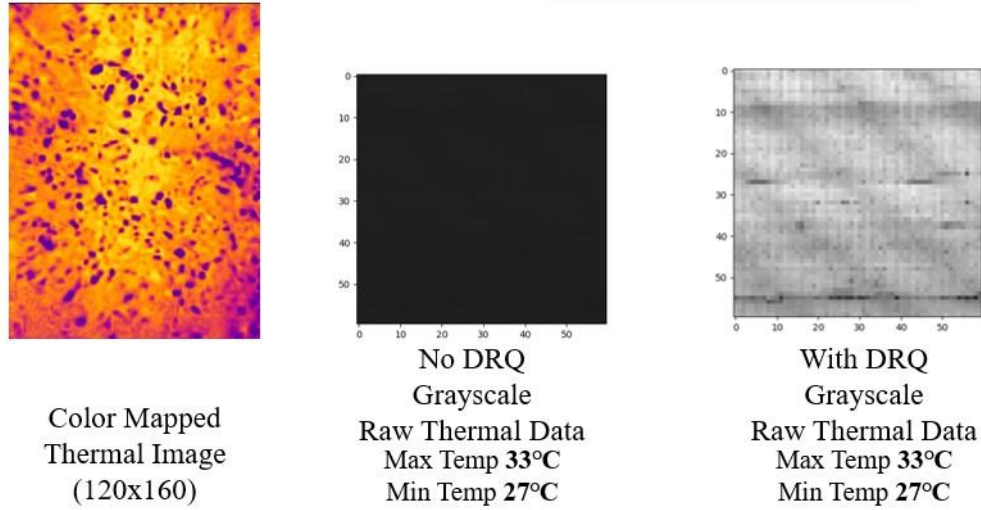
Figure 3.2: Thermal Dataset Examples

videos were then broken down frame-by-frame to obtain the images which were then fed into the network after the featurewise standardization had been performed. Figure 3.3 shows an example image of asphalt that was resized to 96x96. This resizing is performed to save computational time. The 96×96 is used with RGB images as opposed to 60×60 as in the thermal data case, because the standard dimensions that were observed in the literature review was 96×96 and that is what was adopted here. The resize option was also commonly used with RGB images, so that is what was used with this RGB CNN model as opposed to cropping. An example image is not provided after the standardization is applied, because this processing is performed inside the model training structure. Each RGB image was resized to 96x96 in order to save computational time because the full resolution image is not needed to produce good results.

This collected thermal and RGB data was then split in several different ways to be used in the thermal and the RGB network model. The dataset was split into 80% training and 20%
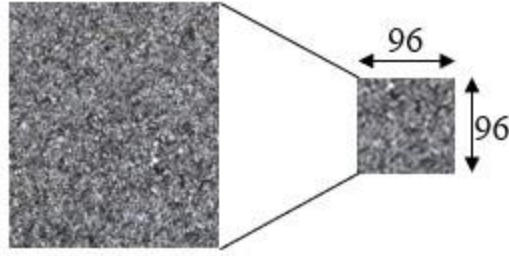
Figure 3.3: RGB Dataset Example

testing groups. Thus, 80% of the data was used to train the network and the other 20% was used

to test the network during training. It should be noted that this testing dataset is not of high

importance in this thesis because the main focus here is to obtain the highest prediction

validation accuracy. In other words, the accuracy that is of highest importance is the accuracy on

validation data that is collected at a different time and place than the training and testing sets.

Thus, by collecting this validation data it can be used to mimic real application circumstances to

evaluate the trained network models as it would perform in a real life scenario. The different

types of material data, data collection times and location, and how the dataset was split is

tabulated in Tables 3.1 and 3.2.

Table 3.1: Thermal Dataset Collection

Dataset Collection and Use (Thermal)					
Material	Collection Time	Location	Data Splits	Dataset Size	Real Data for Validation
Asphalt	September 2020, April 2021, May 2021	ENRC, UARK Campus	80% Training 20% Testing	1,507	470
Concrete	September 2020, April 2021, May 2022	ENRC, UARK Campus	80% Training 20% Testing	1,603	461
Grass	September 2020, April 2021, May 2023	ENRC, UARK Campus	80% Training 20% Testing	1,449	445

Table 3.2: RGB Dataset Collection

Dataset Collection and Splits (RGB)					
Material	Collection Time	Location	Data Splits	Dataset Size	Real Data for Validation
Asphalt	September 2020, April 2021, May 2021	ENRC, UARK Campus	80% Training 20% Testing	1,844	30
Concrete	September 2020, April 2021, May 2022	ENRC, UARK Campus	80% Training 20% Testing	3,693	47
Grass	September 2020, April 2021, May 2023	ENRC, UARK Campus	80% Training 20% Testing	1,936	48

3.2 CNN Configuration Testing and Prediction Validation

The tests and experiments that were carried out on this data are listed below:

1. Experimented with many different hyperparameter configurations with the thermal CNN

2. Used the best performing hyperparameters and completed the pseudo K-Fold cross validation

3. Collected nighttime thermal images to experiment with the versatility of the CNN performance

4. Repeated items 1,2,3 for RGB images

3.2.1 Thermal Hyperparameter Configuration

The purpose of the first experiment was to find the best combination of hyperparameters based on human heuristic that resulted in the highest accuracy on the validation dataset.

The first hyperparameters that should be considered are those contained in the CNN layers themselves. The CNN structure that was finally determined to be the best performing is shown in Figure 3.4:
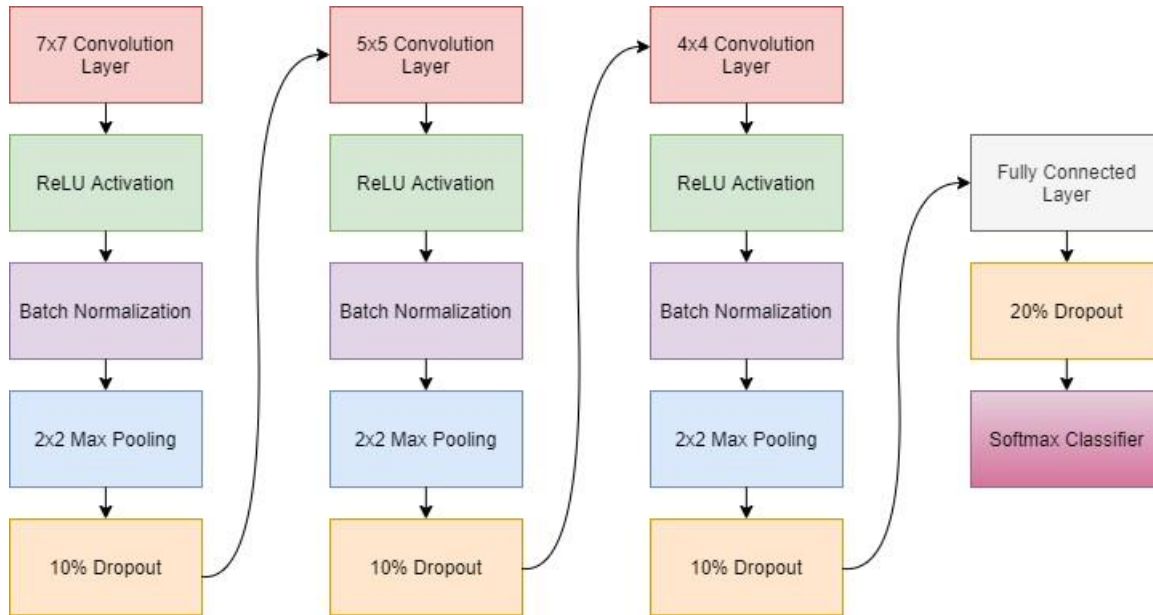
Figure 3.4: The Developed Convolutional Neural Network Structure

This network structure starts with a 7×7 convolution layer. This convolution layer is followed by a ReLU activation layer which is used to catch all variations in the inputs and more adaptively train the model. After the ReLU, a batch normalization layer is added. This batch normalization is the same concept as the featurewise standardization image processing, except instead of standardizing over the entire dataset, it standardizes over each batch. By adding this layer to the CNN, it allows for more robust learning accuracy and robustness. Because in addition to applying the DRQ for each input image, the images are now normalized over the inputs for the entire batch. Without this layer, the final prediction validation accuracy of the model is lower and more inconsistent. Another important layer to note are the dropout layers. These dropout layers are used after each pooling layer, and then again, after the fully connected

layer (FCL). These layers aid in preventing the network from relying too much on any one node in the neural network, which in turn reduces the possibility of overfitting. The amount of dropout was iterated many times to get the best performing model. These iterations included values ranging from 5% to 30% and by using different values for after each layer. For example, when using a single dropout layer of 30% after the FCL, the resulting prediction validation was approximately 50%. After many iterations, the best performance was obtained by using 10% dropout after each pooling layer and 20% dropout after the FCL. The purpose of the fully connected layer at the end is to flatten the output of the last pooling layer and is used to learn the differences between materials. This FCL then feeds into the softmax classifier for class identification. This structure is based heavily on the work done in by Cho et al. [12], but with some differences in the number of layers and with the addition of the batch normalization which was found to increase the accuracy by approximately 9%. Now that the core network structure has been established, the final hyperparameters must be optimized.

The final hyperparameters that were tuned consist of the number of epochs, the learning rate, and batch size. By tuning these hyperparameters, the best accuracy is achieved and the resulting hyperparameter values are used for the prediction validation tests. The learning rate was iterated a few times, but this had little affect of the performance of the final model, thus it was left at the most typical value of 1^{-3}. The number of epochs was iterated, along with the batch size, in order to find the combination that provided the best validation accuracy. Some of the highlights from this hyperparameter testing are shown in Table 3.3:

The combination of hyperparameters that resulted in the best validation accuracy was

Table 3.3: Thermal Hyperparameter Testing

Hyperparameter Testing				
Test Number	Number of Epochs	Learning Rate	Batch Size	Validation Accuracy
1	350	1E-03	256	53%
2	350	1E-03	192	55%
3	350	1E-03	120	69%
4	300	1E-03	120	61%
5	200	1E-03	120	74%
6	250	1E-03	120	75%

250 epochs with a learning rate of 1^{-3} and a batch size of 120. The smaller batch size allowed for

more iterations in each epoch, which leads to more opportunities that the model has to update the

neuron weights. This process of iteration and discovery of what hyperparameters and layers were

important for our model took several months to complete. The largest challenge with training the

network was identifying of which layers should be included in the CNN and where they should

be added. After it was identified that a dropout layer should be added after each max pooling

layer and that the batch normalization should also be included, the rest of the hyperparameter

iterations followed quickly.

As can be seen from Figure 3.5, the training accuracy was at nearly 100% after the first

couple epochs.

The loss function (blue line) periodically spikes and at these spikes it can be seen that the

testing accuracy (gray line) on the this 20% testing split decreases, but then as this loss is

backpropigated through the network, the accuracy returns to 100%. For the majority of the

epochs, this model accuracy on the testing data (gray line) tends to be 100%. This high accuracy

on the testing split occurs because the testing split of the data was collected at the same time,

place and conditions as the training split (the testing dataset is simply a random
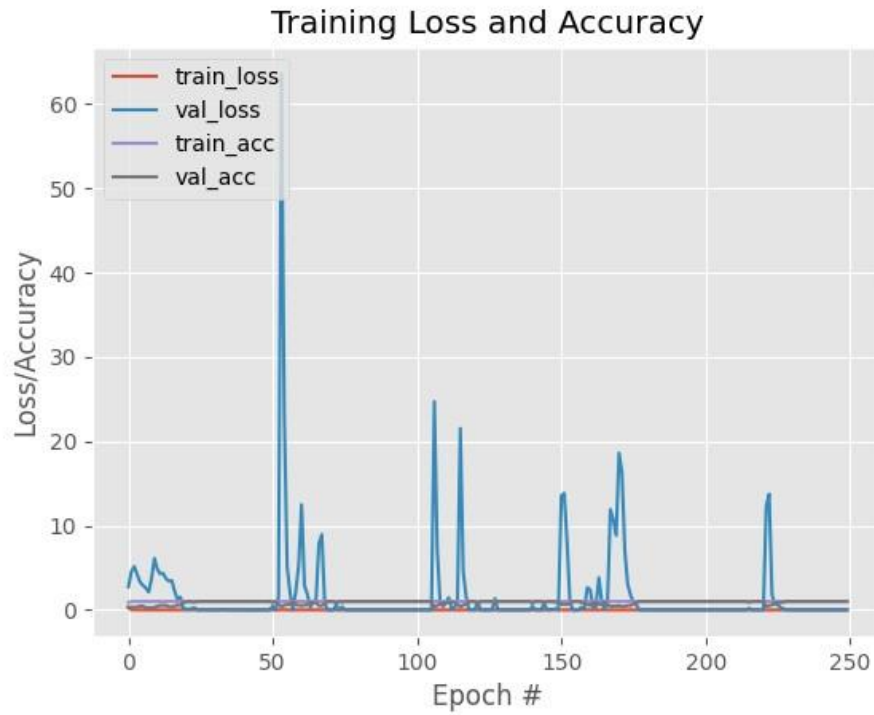
Figure 3.5: Thermal Training and Testing Plot

20% split of the collected dataset). This makes it easier for the model to identify and predict the images contained in this testing dataset. This is one of the reasons it is important to create the validation dataset which was collected at a different place and time; and under different weather conditions. This way, the actual prediction capability of the model could be confirmed. Thus, the most important part is the model accuracy on the validation dataset and the testing accuracy's are ignored (this will be discussed in more depth later).

3.2.2   Thermal Model Prediction Validation

Now that the best performing hyperparameters have been identified, it is time to implement the prediction validation as discussed previously. This will provide the average prediction validation accuracy that can be obtained for thermal images. This testing is shown in

Table 3.4.

Table 3.4: Thermal Prediction Validation

Prediction Validation Testing				
Test Number	Number of Epochs	Learning Rate	Batch Size	Validation Accuracy
1	250	1E-03	120	75%
2	250	1E-03	120	67%
3	250	1E-03	120	74%
4	250	1E-03	120	82%
5	250	1E-03	120	69%
6	250	1E-03	120	85%
7	250	1E-03	120	66%
8	250	1E-03	120	77%
9	250	1E-03	120	76%
10	250	1E-03	120	71%
			Average	74%

This testing is completed by training the neural network on the training dataset and then calculating the prediction accuracy on the validation dataset with the final model obtained for each training iteration. This process of training and then calculating the prediction accuracy on the dataset is repeated 10 times for the thermal model. Because the 80% split of data for training the model is chosen randomly every time the model is trained, the prediction accuracy on the validation dataset is different for each test. This is done to find the average prediction accuracy that can be obtained by the given model hyperparameters. This provides insight on how robust the thermal model is and how this robustness compares to that of the RGB model.

3.2.3   Thermal Model Prediction Validation on Nighttime Images

The final step for the thermal model is to evaluate the prediction accuracy on unseen nighttime images. These nighttime images represent the non-ideal scenario. The CNN has no nighttime data introduced for training the model, thus the prediction accuracy is expected to decrease. The purpose of this test was to determine the robustness of the thermal model and the

advantage of thermal imaging over RGB imaging for a non-ideal scenario. The results of this test

are shown in Table 3.5

Table 3.5: Thermal Nighttime Prediction Validation

Night Image Testing				
Test Number	Number of Epochs	Learning Rate	Batch Size	Validation Accuracy
1	250	1E-03	120	54%
2	250	1E-03	120	55%
3	250	1E-03	120	53%
4	250	1E-03	120	55%
5	250	1E-03	120	45%
6	250	1E-03	120	52%
			Average	52%

3.2.4 RGB Model Configuration and Prediction Validation

The RGB CNN structure was the same as the thermal CNN structure (see Figure 3.4). As

such, the same dropout and batch normalization layers are used. However, the number of epochs

and batch size are iterated to determine the best performing combination for the RGB model.

This testing can be seen in Table 3.6. After the best hyperparameters are identified, the validation

prediction is completed by the same method discussed in the thermal model prediction

validation. These results are tabulated in Table 3.7 for the ideal scenario (daytime) and Table 3.8

for the non-ideal scenario (nighttime):

Table 3.6: RGB Hyperparameter Testing

Hyperparameter Testing					
Test Number	Number of Epochs	Learning Rate	Batch Size	Test Accuracy	Validation Accuracy
1	150	1E-03	256	100%	98%
2	100	1E-03	256	100%	98%

Table 3.7: RGB Prediction Validation

Prediction Validation Testing					
Test Number	Number of Epochs	Learning Rate	Batch Size	Test Accuracy	Validation Accuracy
1	100	1E-03	256	100%	98%
2	100	1E-03	256	100%	94%
3	100	1E-03	256	100%	91%
4	100	1E-03	256	100%	92%
5	100	1E-03	256	100%	99%
6	100	1E-03	256	100%	98%
				Average	95%

Table 3.8: RGB Nighttime Prediction Validation

Night Image Testing					
Test Number	Number of Epochs	Learning Rate	Batch Size	Test Accuracy	GB Validation Accurac
1	100	1E-03	256	100%	48%
2	100	1E-03	256	100%	43%
3	100	1E-03	256	100%	48%
4	100	1E-03	256	100%	43%
5	100	1E-03	256	100%	48%
				Average	46%

3.3   Discussion of Results

When the thermal dataset was being collected with the Flir One mobile camera, there was an issue that arose while recording the videos. If the video was not long enough, the thermal matrix would result in skewed and even sometimes a corrupted raw temperature matrix so that the model could not recognize the images. The color mapped image was not affected, it was only the raw thermal data that was stored in the .vir file that was corrupted. It was discovered that the length of the video needed to be 60 seconds long to avoid this error. One possible cause for this phenomenon is that the raw thermal matrix takes this long to fully calibrate and save an uncorrupted raw temperature matrix file. This caused many problems when originally testing the thermal model, because this issue was not identified immediately and the results were very poor because of it. This error took a week to identify and correct. However, once this problem was identified, the validation data was collected again and used for the prediction validation testing.

This was only the beginning of the issues that were encountered while collecting and processing the thermal data using the Flir One camera.

One of the most challenging parts of this research was the method by which the raw temperature data is obtained and how to process this raw data. The Flir One camera is operated by the SmartIR app [15]. In this SmartIR app the color mapped video is recorded and saved to the smartphone gallery as a .mp4 file. In the beginning stages of this research, each frame from this mp4 file was then extracted and used to train the thermal model. However, when using these images, the thermal model was not able to accurately identify any material type. This was a puzzling result, until it was discovered that the color mapped video was not comprised of the raw temperature data. It was using the raw temperature matrix to map it into the colorful video as a way of visualizing this temperature data. In an effort to collect this raw thermal data, the Flir One SDK was used to create a simple android application that could be used to collect this data. However, after weeks of attempting this development, it was discovered that the SmartIR app saved a separate .vir file to the phone files in which the raw thermal data was stored. After this .vir file was found, the extraction of the raw thermal data could be accomplished and this raw data was used to train the thermal model. This raw thermal data was saved as a UInt8 1D array in the .vir file. It was manipulated in order to create a $160 \times 120 \times L$ matrix (L is the number of video frames). This manipulation is completed by taking the final length of the 1D array (N), then subtracting 8 and dividing by 4 which provides the number of pixels $(N_p = \frac{N-8}{4})$ in the entire video. By going one step further and dividing by 160 and again by 120 $(L = \frac{N_p}{160 \times 120})$, the total number of frames of the video can be obtained. This allows the

final form of $160 \times 120 \times L$ to be obtained. Each frame of this final raw thermal matrix form was then processed using the DRQ function. Once this challenge was overcome, the thermal model could be tested and validated.

It can be seen from the hyperparameter testing for the thermal model and the RGB model that the number of epochs and the batch size used for each model are different from each other. This is has to do with the larger dataset for the RGB model. With this larger dataset, the number of epochs could be less and the batch size could be larger than the thermal model without sacrificing performance. Other than these two changes, the thermal model is identical to the RGB model. The reason the two models were made to be so similar was to create an even playing field for the comparison of each image type. Because if one model was vastly different from the other, then that would not be a very good representation of the performance solely based on the type of data used. Computer vision using RGB images has been in development longer than thermal computer vision, as such, there are much more sophisticated models that have been developed for RGB image data than the thermal model developed in this thesis. The question that this thesis seeks to answer is how does thermal imaging compare to RGB imaging for material detection. Thus, it is important that the same model structure is used for each method so that the comparison is being made purely on the difference in data type and not on the CNN model development.

The model prediction validation is the way in which the thermal and RGB model performance is quantified. This prediction validation is conducted using the validation dataset that was collected at a different time and place from the training dataset. This validation dataset is used as a way to mimic real world applications of these models. In other words, how will these trained models perform on real world data to detect material types. This prediction validation

accuracy is determined after the network model is fully trained. This final model, which is obtained after being trained through every epoch on the training dataset, is then used to predict the materials present in each image of the validation dataset. The prediction accuracy is then calculated from the number of correct label predictions out of the entire validation dataset. This process is repeated multiple times for both the thermal and RGB models to obtain an average accuracy over multiple tests. Therefore, the model is trained multiple times and the predication validation accuracy is calculated each time the model is retrained. The accuracy's vary with each test, because the model randomly selects 80% of the dataset to train with and the other 20% is used to test the model. However, the performance of this testing dataset split is not of importance in this thesis because the model accuracy on the validation dataset is the main focus. This process of training the network and calculating the accuracy on the validation dataset is repeated 10 times for the thermal model and 5 times for the RGB model. By doing this, the robustness of the thermal model and the RGB model are determined regardless of what training data is used. The comparison of these prediction accuracy's are discussed next.

Comparing Tables 3.4 and 3.7, and as shown in Figure 3.6, it can be seen that the RGB model outperforms the thermal model on in an ideal scenario where the data was collected in daylight. The average prediction validation accuracy for the thermal model was 74% and the average accuracy of the RGB model is 95%.
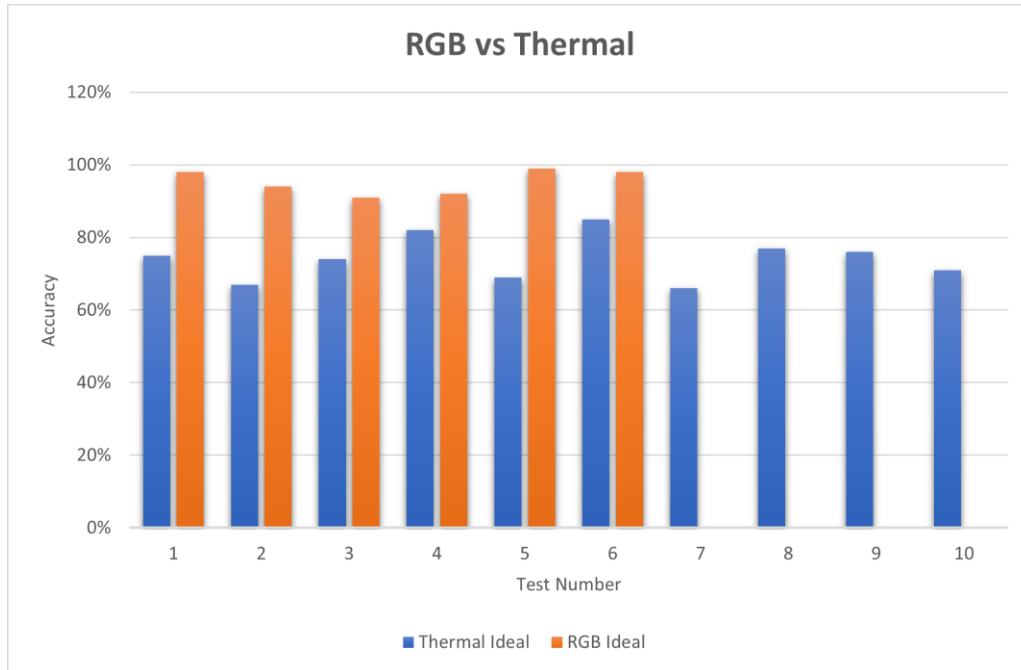
Figure 3.6: RGB vs Thermal Validation in Ideal Scenario

This is a significant difference and there are a few reasons that this is the case. First, the

RGB dataset was larger because of the higher camera frame rate, which allowed more images to

be captured in a shorter time period. The data available in the RGB image is inherently greater

because the RGB image is comprised of three different channel (red, green, and blue) and the

model uses all of these channels for training. This would allow the RGB model to have more data

and give a wider of variety of data to train on. Second, the RGB images rely solely on the visible

light that is in the scene. Thus, in an ideal scenario the RGB has three channels of data which can

give the model a high performance largely based on the color of the material. Which in the case

of this thesis, the materials considered are asphalt, concrete, and grass, so the color differences

are drastic between each material. Because thermal images do not have these properties and are

relying simply on the temperature distribution across the material, it is at a disadvantage.

However, looking at Tables 3.5 and 3.8, and as in Figure 3.7, it can be seen that the thermal model significantly outperforms the RGB model in the non-ideal scenario.
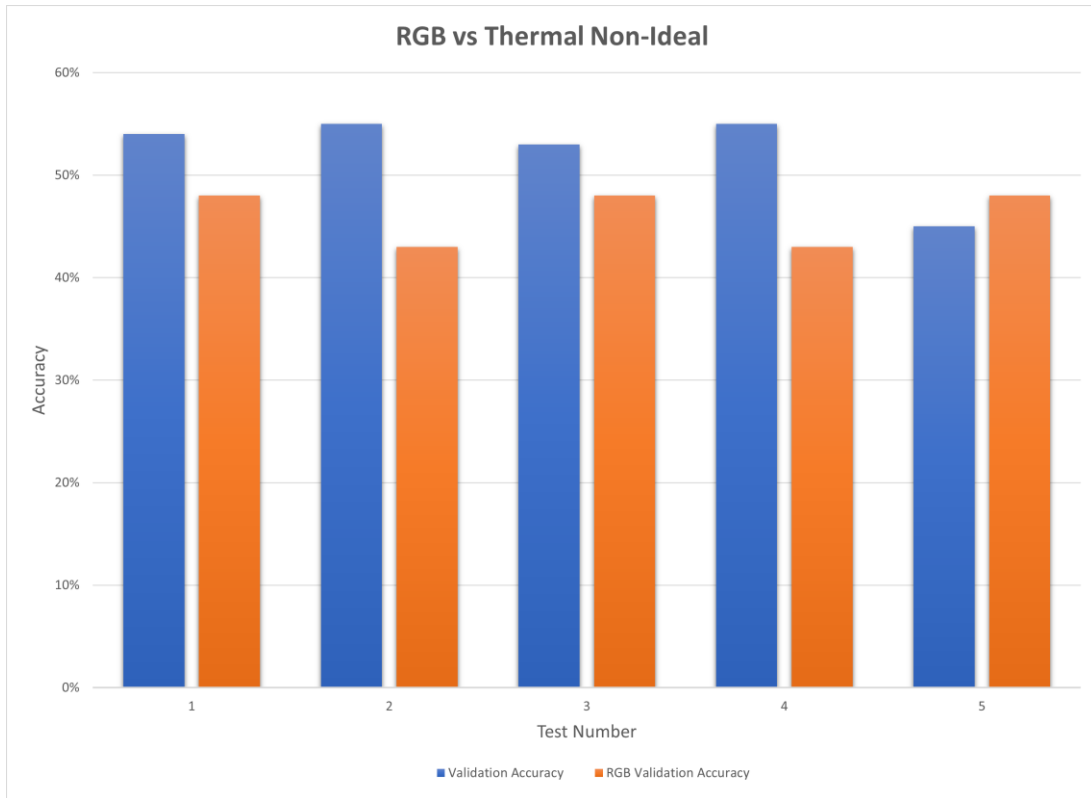

Figure 3.7: RGB vs Thermal Validation in Non-Ideal Scenario

The reasons for this are the flip side of the ideal scenario, now that there is very little visible light in the scene, the RGB model is not able to identify the material as well because now the material color is obscure and there are not real patterns for the model to detect, thus the performance drops drastically. The percentage decrease of the RGB model was $95\% - 46\% = 49\%$ from the ideal to non-ideal scenario, while the percentage decrease of the thermal model was only $74\% - 52\% = 22\%$. Thus, the thermal model is much more robust to changes in different scenarios. The thermal model detects material types based on how the temperature changes across the material due to porosity, cavities, and emissivity. Thus, even though the materials absolute temperature changed from daytime to nighttime, the temperature changes across the

material mostly stay the same. This allows the thermal model to be more robust. Furthermore, the more that visible light is absent from the scene, the larger the gap will become between the thermal and RGB model accuracy's.
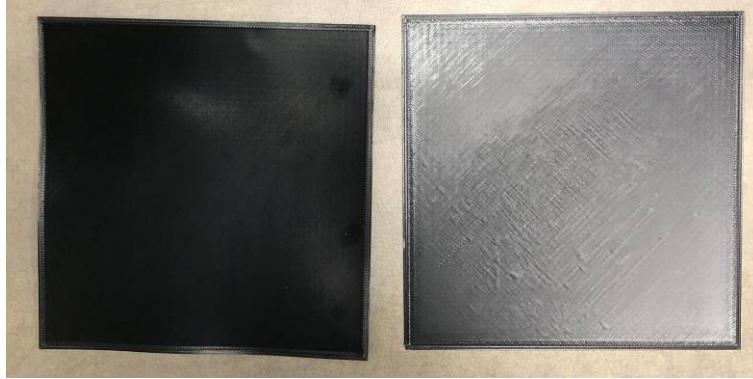

Figure 3.8: PLA (left) and ABS (right) 3D Printing Materials

The ability to detect 3D materials would be very advantageous in manufacturing and in cooperative 3D printing applications. If a 3D printer could identify other 3D materials in its work area and react in real time to this information, it could be helpful in improving the speed and accuracy of the 3D printers. Because of this, thermal data was collected for PLA and ABS 3D printing materials. These materials were used to train the thermal model to see if they could be learned and identified. However, these two materials were very similar in surface texture and the model was unable to learn any recognizable patterns. It can be seen from Figure 3.8 that the difference between PLA and ABS is almost indiscernible. The lack of available data was also a contributing factor to the inability to get any positive results.

The prediction accuracy's of each model on the individual material classes was not calculated because for the purposes of this research that was not vital information. However, it should be noted that this model works best on materials that have noticeably different surface textures. As it was seen when testing the 3D printing material detection, the model was not able to

differentiate between the two different printing materials, because they were very similar in

surface texture. This is a case where combining the RGB data and the thermal data could be very

advantageous so that the RGB data could better detect color differences in the material and the

thermal model could detect slight differences in thermal patterns. Then by combining the two

different data types, similar materials could still be identified (i.e. steel vs aluminum). This work

on combining RGB and thermal data for model training is left for future work and is discussed

briefly in the conclusion.

4 PHYSICAL EXPERIMENT: SETUP AND TESTING

In this chapter, the physical experimentation of servo motor control using the trained thermal model is presented. The purpose of this experiment is to demonstrate the future work and applications that are possible on this topic. These tests are carried out to demonstrate how the trained thermal model can be used to send control signals to a servo motor that will then respond with the correct adjustment based on what material is present. The servo motor in this demonstration is a simplified representation of a robotic platform.

4.0.1 Setup and Communication Protocol

A Raspberry Pi (RPi), SG90 Servo, and required hardware was purchased for this testing. First, it was attempted to run the thermal model on the Raspberry Pi itself. However, the RPi did not have the appropriate software to run the CNN model. Therefore, a TCP/IP communication was establish between the RPi and a laptop via an Ethernet cable. The RPi is established as the server which will receive data from the client, the laptop. A diagram of this setup is shown in Figure 4.1.

WiFi could also be used for this communication, which will be much more practical in future testing and implementation so that the RPi can be used to control a mobile platform (i.e. robot). By setting up this connection, the predicted material from the laptop can be sent directly to the RPi and the RPi can then send this as a control input to the servo motor to update the position. In a real application, This can be thought of as the RPi sending wheel turning updates on an autonomous robot based on the material present in order to prevent collision or detouring from a planned path.
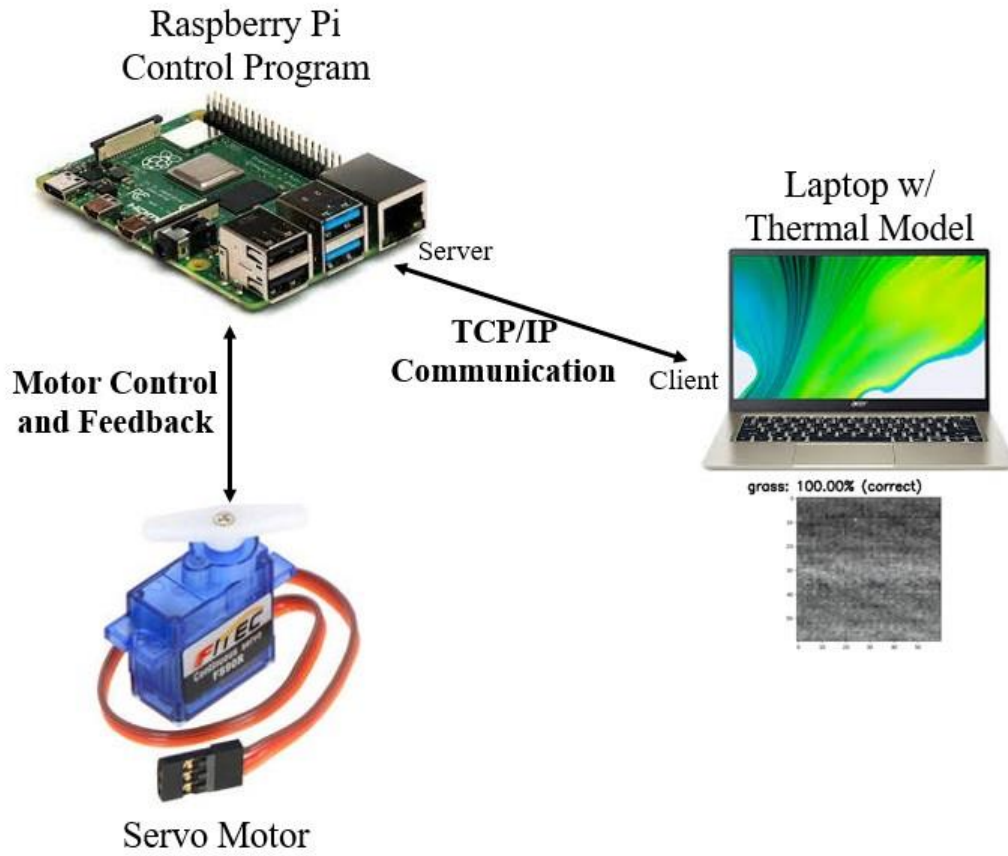
Figure 4.1: Physical Experiment Setup

4.0.2   Discussion of Results

This testing successfully proved that this thermal model can be deployed in real-life

application for autonomous robotics or even for manufacturing application where some machine

operation may need to adapt to the materials that it is coming into contact with. In this

demonstration, the RPi was programmed to update the position of the servo based on the criteria

in Table 4.1. The material that was correctly detected by the model on the laptop was grass with

a 100% certainty (Figure 4.2). By using the criteria in Table 4.1, the servo was then updated to

position itself at 180 degrees.

Table 4.1: Testbed Motor Control Criteria

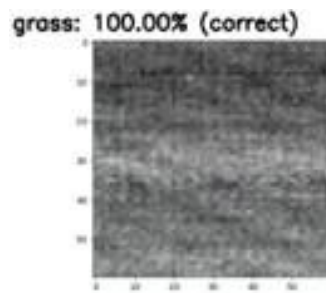| Material Detected	Position Angle
Asphalt	0
Concrete	90
Grass	180

grass: 100.00% (correct)

Figure 4.2: Grass Correctly Detected

5   CONCLUSION

There are some key results from this research that identify methods and quantitative evidence in applying thermal imaging and RGB imaging for material detection. These methods include how to pre-process thermal data using the Dynamic Range Quantization and RGB data using image standardization for the best performance in the CNN models. The quantitative evidence is based on how well the thermal imaging and RGB imaging both perform in ideal scenarios where the lighting and environmental conditions are good; and for non-ideal scenarios where it may be dark or environmental conditions are poor (e.g. excessive fog). The conclusions drawn from these items are discussed below.

The best performance for the thermal model was obtained by utilizing the Dynamic Range Quantization (DRQ) method for processing the thermal data. This method works well for the thermal data because when dealing with absolute temperature in matrices (as is the case with the mobile thermal camera), it is crucial that the temperature values be scaled in relation to all other values in the matrix. This allows for the highest degree of variance between neighboring matrix values and thus results in a more robust training of the CNN model. It was also found that when using the SmartIR app, the thermal data is stored in a separate file called a ".vir" file. Within this ".vir" file, the thermal data is stored in a 1D array, which must be extracted in a particular way in order to get out a $160x120xL$ thermal matrix. The third value ("length") is the length of the video, or in other words the number of frames in the video that was recorded during the dataset collection. The best performance for the RGB model was obtained by utilizing the standardization method for processing RGB images. This method works by transforming the input data to have a mean value of zero and a standard deviation of 1. By doing this, the model

was able to identify the difference between each material much more accurately. There are two different ways in which this standardization can be applied to the data set. The first is called samplewise standardization and this works by applying the standardization on each input image. The second is called featurewise standardization and it works by applying the standardization based on the input values over the entire input dataset. The featurewise application yielded the best performance in the final RGB model. When developing the two different models, the hyperparameters that were focused on for tuning were the number of epochs, the batch size, and the amount of dropout used in the CNN. The dropout was critical because it allowed the model to not rely too heavily on any particular neuron in the network, which would then cause overfitting. If overfitting occurs, the model performance decreases dramatically.

After the image pre-processing was completed and the models were tuned and trained, the thermal model was found to have inferior accuracy as compared to RGB imaging for material detection in ideal lighting scenarios. The average accuracy of the thermal model on validation data after 10 folds was found to be 74%. The average accuracy of the RGB model, on the other hand, had an average accuracy of 95% after 6 folds. Thus, the RGB model was able to outperform the thermal in the ideal case. However, for the non-ideal case (after dark), the thermal model has a noticeably better performance as compared to the RGB model. The average accuracy of the thermal model on validation data after 5 folds was 52%, while the RGB model accuracy was only 46%. These results were based on images that were taken after dark, but the images were not completed dark, there were still small sources of light. Thus, if there were to be no lighting present in the image, it can be expected that the gap between the accuracy thermal model and the RGB model will only become more drastic. This will be very advantageous for autonomous operations in dark areas and for disabled persons to use after dark.

This thermal model for material detection was deployed for the control of a servo motor as a proof of concept and as a demonstration of what can be done in future work. This control demonstration was accomplished by establishing a TCP/IP communication between a laptop and a Raspberry Pi, which then relayed the control input from the laptop to the control of the servo motor. The thermal model was employed on the laptop to identify the material that was present in a given image, this information was then sent to the Raspberry Pi, processed and then sent to the servo as input. Once the servo motor is set to the predetermined position (which is based on the material type detected), the Raspberry Pi then displays a message that states that the servo motor position has been updated. This demonstration successfully showed the possibilities of using the thermal model for autonomous functions.

The work presented in this thesis does have some limitations. Data was only collected for three materials; asphalt, concrete, and grass. This was due to the amount of time it takes to collect the data and the amount of data that is necessary to properly train the models. Thus, the number of materials was reduced to just these three. The models created in this thesis are deployed for use in the control of a single servo motor as a demonstration of the possible uses. These models should be deployed in more complex systems such as a robot for autonomous control. Also, the process in which this motor control demonstration was conducted required the manual process of sending an image to the model on the laptop and having it processed and detected before the laptop could send this information to the Raspberry Pi for motor control. This process should be implemented using a more automatic control algorithm and also be employed for real-time detection and control. Regardless of these limitations, the findings in this thesis have laid a foundation for future work in deploying this affordable mobile thermal camera for use in autonomous robotics, manufacturing, and personal use. Future work should first include

expanding the already created dataset both in the amount of data and the variety of data. Secondly, it should also involve more tuning of the thermal model in terms of hyperparameter values to obtain an optimal prediction accuracy on the validation data. Thirdly, future work should involve deploying this thermal model in more advanced robotic platforms for autonomous control. Deploying these findings for the control of a robotic platform will provide an affordable alternative to conventional sensing methods and can also serve multiple sensing capabilities from one device (i.e., material detection, object detection, temperature monitoring, etc.).

Another area that should be explored in future work is that of combining the thermal data and the RGB data so that both methods can be simultaneously leveraged in the applications discussed above. As can be seen in Figure 5.1, the thermal camera itself can combine the RGB image and the thermal data to create a mixed version. In this way, the objects in a scene can be easily detected and the temperature information can also be seen. By using this same concept, the raw thermal data that is collected and processed could be combined with the RGB data before training so that The benefits of both data types can be leveraged for any scenario. This would have to be done in a separate step before training that model. The mixed image shown above could not simply be taken, processed, and used to train the model, because of the same issue observed when using the pure thermal image for training the thermal model. Therefore, the raw temperature data would have to be extracted and then later combined with the RGB data in a way such that both data types could be effectively used for training and validation.
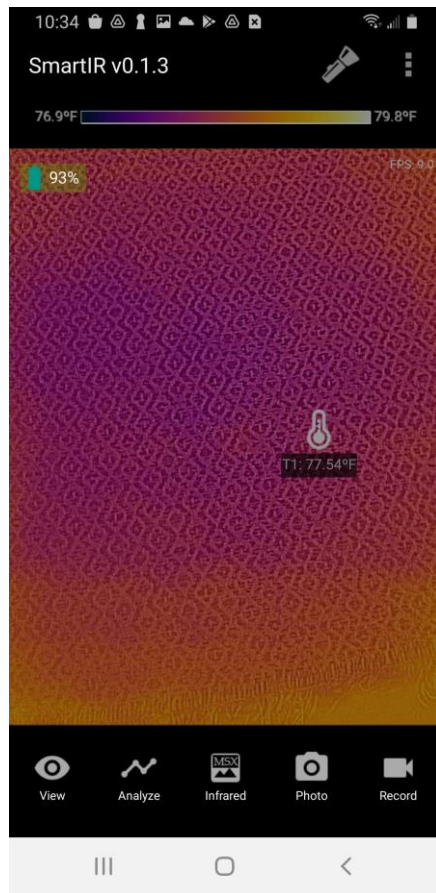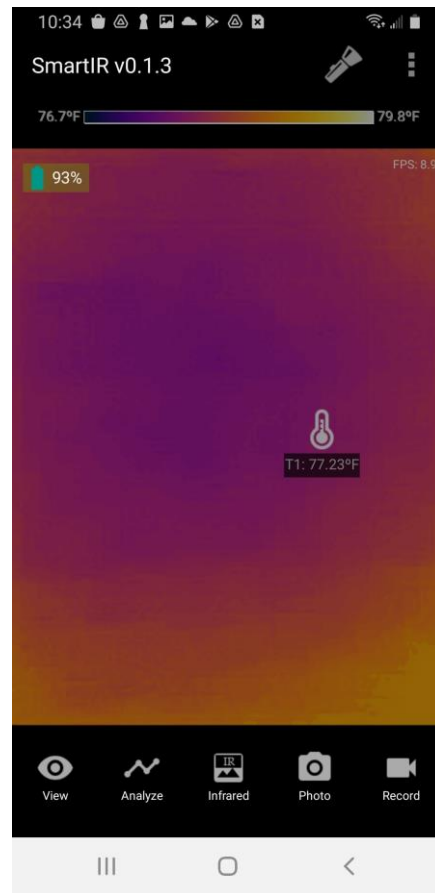
(a) Mixed View   (b) Pure Thermal View

Figure 5.1: Different View Types with SmartIR

Bibliography

[1] M. Jangblad, "Object detection in infrared images using deep convolutional neural networks," 2018.

[2] S. Saha. (2018) A comprehensive guide to convolutional neural networks. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guideto-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[3] V. Jain. (2019) Everything you need to know about "activation functions" in deep learning models. [Online]. Available: https://towardsdatascience.com/everything-youneed-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253

[4] scikit-learn developers. (2020) Cross-validation: evaluating estimator performance. [Online]. Available: https://scikit-learn.org/stable/modules/cross\ validation.html

[5] P. Editor. (2018) A brief history of computer vision and ai image recognition. [Online]. Available: https://www.pulsarplatform.com/blog/2018/brief-historycomputer-vision-vertical-ai-image-recognition

[6] S.-S. Lin, "Extending visible band computer vision techniques to infrared band images," 2001.

[7] A. Zingoni, M. Diani, and G. Corsini, "A flexible algorithm for detecting challenging moving objects in real-time within ir video sequences," *Remote Sensing*, vol. 9, no. 11, p. 1128, 2017.

[8] C. Bodenstein, M. Tremer, J. Overhoff, and R. P. Wurtz, "A smartphone-controlled au-¨ tonomous robot," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2015, pp. 2314–2321.

[9] M. P. Arakeri, B. V. Kumar, S. Barsaiya, and H. Sairam, "Computer vision based robotic weed control system for precision agriculture," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017, pp. 1201–1205.

[10] Y. Cho, S. J. Julier, N. Marquardt, and N. Bianchi-Berthouze, "Robust tracking of respiratory rate in high-dynamic range scenes using mobile thermal imaging," *Biomedical optics express*, vol. 8, no. 10, pp. 4480–4503, 2017.

[11] W. Hardin. (2018) Thermal imaging to diagnose disease. [Online]. Available: https://www.visiononline.org

[12] Y. Cho, N. Bianchi-Berthouze, N. Marquardt, and S. J. Julier, "Deep thermal imaging: Proximate material type recognition in the wild through deep learning of spatial surface

temperature patterns," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–13.

[13] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, pp. 2017–2025, 2015.

[14] J. Brownlee. (2019) How to normalize, center, and standardize image pixels in keras. [Online]. Available: https://machinelearningmastery.com/how-to-normalize-center-andstandardize-images-with-the-imagedatagenerator-in-keras/

[15] C. Xie. Infrared street view. [Online]. Available: https://charxie.github.io/