To the Graduate Council:

I am submitting herewith a dissertation written by Elizabeth Piersall entitled "A Comparison of Machine Learning Based Arc Detection For a Custom Low-Cost Multispectral Camera for UAV Applications." I have examined the final paper copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Data Science and Engineering.

Dr. Peter Fuhr, Major Professor

We have read this dissertation
and recommend its acceptance:

Dr. Yilu Liu

Dr. Fangxing Li

Dr. Ali Ekti

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

To the Graduate Council:
I am submitting herewith a dissertation written by Elizabeth Piersall entitled "A Comparison of Machine Learning Based Arc Detection For a Custom Low-Cost Multispectral Camera for UAV Applications." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Data Science and Engineering.

Dr. Peter Fuhr, Major Professor

We have read this dissertation
and recommend its acceptance:

Dr. Yilu Liu

Dr. Fangxing Li

Dr. Ali Ekti

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

A Comparison of Machine Learning Based Arc Detection For a Custom Low-Cost Multispectral Camera for UAV Applications

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Elizabeth Piersall

May 2025

Abstract

Unmanned Aerial Vehicles (UAVs), also known as drones, have seen increasingly widespread use in recent years with advancing technology in a variety of fields, and present an option to improve the capability and accessibility of remote data collection. One area that may benefit from this capacity is the maintenance of electrical equipment, particularly in circumstances in which the equipment owned by electrical utilities covers wide areas, and more conventional sensor options may be expensive to implement at scale. However, the cost and accessibility of UAVs and their sensor payloads can also limit the use of this technology, and options for specific sensor capabilities may be limited to certain UAV platforms by weight, power usage, and compatibility. This dissertation will investigate an approach to implementing a custom, relatively low-cost camera system with accessible components, and compare the performance of machine learning based image classification to integrate the separate components for the example application of the detection of electrical arcing. Specifically, this research will examine a low-cost camera system composed of three industrial cameras, which collect light in different parts of the electromagnetic spectrum: visual, ultraviolet, and infrared. Each of these wavelength bands contains information that could assist in the detection of an electrical arc, with light emitted from the arc in the visual and ultraviolet spectra, and infrared radiation from the arc increasing the radiant heat of nearby metal components which in turn emit infrared radiation. This dissertation will use this prototype camera system to generate a custom dataset of electrical arcing examples in varying ambient outdoor conditions,

including bright sunlight and low light, and use this data to train separate and ensemble machine learning based image classification models. A comparison of different model parameters and ensemble approaches will be used to address the strengths and limitations of the cameras and the approach to arc detection. The goal of this research is to provide insights into the use of machine learning for custom low-cost sensor systems, what strengths it can demonstrate, and what potential weaknesses a developer or user of such a system may benefit from addressing.

Table of Contents

1 Introduction **1**

1.1 Introduction 1

1.2 Purpose 3

1.3 Cameras 4

1.3.1 Existing Cameras 4

1.3.2 Custom Cameras 6

1.4 Research Questions 6

1.4.1 Question 1: Can low-cost visual, ultraviolet, and thermal cameras collect enough information to be used for detection of an electrical arc with machine learning? 7

1.4.2 Question 2: How does an ensemble camera system as a means of classifying data for arc detection compare to models for each individual camera? 7

1.4.3 Question 3: How do other conditions impact the ability of a machine learning model to detect electrical arcing from the data captured by each of the three cameras individually? As an ensemble? 8

1.4.4 Question 4: What are the implications of the accuracy results of the individual camera data and ensemble models, and how can they be used to design a more effective, low-cost camera sensor? 8

2 Literature Review **9**

2.1 Sensor Integration and Fusion . . . 9

2.2 UAV Based Sensing . . . 11

2.3 Multispectral Analysis . . . 13

2.4 Camera-based Arc Detection . . . 17

2.5 Analysis and Machine Learning . . . 19

2.5.1 Sources of Data Noise . . . 25

2.5.2 Summary . . . 26

3 Hardware and Data Collection **28**

3.1 AIMS Project and Drones . . . 29

3.2 Hardware . . . 31

3.2.1 Camera Prototype . . . 31

3.2.2 Spectrometer . . . 33

3.2.3 Arc Source . . . 34

3.3 Data Collection . . . 36

3.4 Summary . . . 39

4 Experiments **41**

4.1 Dataset . . . 42

4.1.1 Ambient Conditions . . . 43

4.1.2 Cleaning and Organization . . . 49

4.2 Machine Learning . . . 50

4.2.1 Training . . . 50

4.2.2 Models . . . 52

4.3 Individual Cameras . . . 57

4.3.1 Detailed Results . . . 60

4.4 Ensemble . . . 61

4.4.1 Majority Vote . . . 61

4.4.2 Classification Layer . . . 65

4.5 Classification Layer + Majority Vote . . . 71
4.6 Summary . . . 72

5 Discussion of Experiments **73**
5.1 Separate Cameras . . . 73
5.1.1 Visual . . . 74
5.1.2 Ultraviolet . . . 80
5.1.3 Thermal . . . 86
5.2 Ensemble . . . 89
5.2.1 Majority Vote . . . 90
5.2.2 Classification Layer . . . 93
5.2.3 Classification Layer + Majority Vote . . . 99
5.3 Summary . . . 99

6 Conclusions **101**
6.1 Research Questions . . . 101
6.1.1 Question 1: Can low-cost visual, ultraviolet, and thermal cameras collect enough information to be used for detection of an electrical arc detection with machine learning? . . . 102
6.1.2 Question 2: How does an ensemble camera system as a means of classifying data for arc detection compare to models for each individual camera? . . . 102
6.1.3 Question 3: How do other conditions impact the ability of a machine learning model to detect electrical arcing with the data from each of the three cameras separately? As an ensemble? . . . 104
6.1.4 Question 4: What are the implications of the accuracy results of the individual camera data and ensemble models, and how can it be used to design a more effective, low-cost camera sensor? 105
6.2 Future Work . . . 106

Bibliography **109**

A Data Collection Code **121**

A.1 Spectrometer 121

A.2 Images 122

B Detailed Result Metrics **128**

B.1 Individual Cameras 128

B.2 Classification Layer 133

Vita **136**

List of Tables

4.1 Validation data weather conditions, with conditions taken from NOAA data for location of data collection, and weather conditions visually verified. Times shown are from timestamps of the spectrometer data, which was taken immediately before collection of each validation dataset. 48

4.2 The composition of each of the custom datasets collected. 48

4.3 A binary confusion matrix. 55

4.4 Summary of accuracy results for one camera input, averaged over other parameters. 59

4.5 Summary of MCC results for one camera input, averaged over other parameters. 59

4.6 Accuracy results for majority vote ensemble models. 63

4.7 MCC results for majority vote ensemble models. 64

4.8 Accuracy results for ensemble models, Classification Layer Model 1, transfer learning. 69

4.9 Accuracy results for ensemble models, Classification Layer Model 2, transfer learning. 69

4.10 MCC results for ensemble models, Classification Layer Model 1, transfer learning. 70

4.11 MCC results for ensemble models, Classification Layer Model 2, transfer learning. 70

4.12 Accuracy results for majority voting of classification layer models. . . 71

4.13 MCC results for majority voting of classification layer models. 72

5.1 Best performing models for the visual data. 75
5.2 Worst performing models for the visual data. 75
5.3 Results of the summarized data for all the validation sets for the visual data. 75
5.4 Best performing models for the ultraviolet data. 82
5.5 Worst performing models for the ultraviolet data. 82
5.6 Results of the summarized data for all the validation sets for the ultraviolet data. 82
5.7 Best performing models for the thermal data. 87
5.8 Worst performing models for the thermal data. 87
5.9 Results of the summarized data for all the validation sets for the thermal data. 87
5.10 Best models for classification layer models. 95
5.11 Best models for classification layer models, summary of all validation datasets. 96

B.1 Accuracy results for individual cameras without transfer learning. . . 129
B.2 Accuracy results for individual cameras with transfer learning. 129
B.3 Other metric results for individual cameras without transfer learning 130
B.4 Other metric results for individual cameras with transfer learning . . 131
B.5 MCC results for individual cameras without transfer learning. 132
B.6 MCC results for individual cameras with transfer learning. 132
B.7 Accuracy results for ensemble models, Classification Layer Model 1, no transfer learning. 133
B.8 Accuracy results for ensemble models, Classification Layer Model 2, no transfer learning. 134
B.9 MCC results for ensemble models, Classification Layer Model 1, no transfer learning. 134

B.10 MCC results for ensemble models, Classification Layer Model 2, no transfer learning. 135

List of Figures

2.1 An example of a quadcopter drone. 11
2.2 Electromagnetic spectrum of sunlight, the light to most Earth-looking satellite or aircraft sensors, including the light above the Earth's atmosphere. 15
2.3 Neural network diagrams. 21
2.4 Diagram of a CNN used for image processing. 23
2.5 The process of convolution in CNN layers. 24
2.6 Literature examples with relevance to dissertation research. 27

3.1 A version of the triband camera prototype in a 3D printed case. . . . 33
3.2 A diagram of data collection from the three cameras. 34
3.3 Spectrum taken with validation data. 35
3.4 Example images from each camera, of each class. 36
3.5 An AIMS drone with the triband prototype camera system attached. 37
3.6 Images taken from each camera at the first AIMS demo. 37
3.7 A diagram of data collection from the three cameras. 38
3.8 Examples of data collection. 39

4.1 Spectrum of electrical arc. 46
4.2 Spectrum taken with validation data. 47
4.3 Diagram of the AlexNet network. 54
4.4 Diagram of VGGNet network. 54

4.5 Diagram of Resnet residual block diagram. 54
4.6 A diagram of the majority vote ensemble. 61
4.7 Ensemble network diagrams. 66

5.1 Examples of visual camera images from Valid Data 2. 77
5.2 Examples of visual camera images from Valid Data 4. 77
5.3 Examples ultraviolet camera images from Valid Data 2. 83
5.4 Examples ultraviolet camera images from Valid Data 1. 85
5.5 Examples ultraviolet camera images from Valid Data 3. 85
5.6 Examples ultraviolet camera images from Valid Data 1. 89

Chapter 1

Introduction

1.1 Introduction

The ability to collect data at a distance, in many different conditions and environments, is greatly valuable in a variety of applications and circumstances. Historically, the platform for data collection over large areas, or in remote or dangerous areas, was often either manned aircraft or satellite [1]; however, rapidly advancing technology for Unmanned Aerial Vehicles (UAVs), widely known as drones, has presented an alternative with its own unique advantages. UAVs can offer lower costs and more adaptable remote data collection options than those that have existed before, making remote sensing capabilities more accessible for users in a wider variety of contexts and applications [2]. However, sensors for various types of remote data collection are often very expensive, and often designed for specialized applications such as aircraft or satellites. While such sensors may be very capable of high-quality data collection, they may not be compatible to either the physical design or data inputs of many available UAVs, may not be well suited to the specific needs of a user of UAV-based sensing, or may be economically impractical for an application with a lower budget or higher risk of crash.

Commercially purchased UAVs often do come with sensors integrated into their systems, both for data collection as well as navigation and safety reasons [3]. Common sensors include antennas to connect to a Global Navigation Satellite System (GNSS), often specifically Global Positioning System (GPS) for the United States based system, for measurements of the location and altitude of the UAV. This external data is often combined with internally measured accelerometer and gyroscope data to monitor the flight and behavior of the UAV, making sure it is stable and within a safe area. Many commercially available UAVs also have some level of automation to correct for interference such as wind to maintain stable flight, as well as monitoring of airspace to prohibit flight in areas with legal restrictions. Additionally, sensors are included as part of collision avoidance systems that are found on many drones, which can include cameras, Light Detection and Ranging (LiDAR), and other sensors to help detect obstacles and prevent crashes. In addition to other flight controls, UAVs are usually controlled, or at least monitored, by a human pilot, who can interact with the UAV through radio telemetry or other means of correspondence. The requirements for the pilot, including the range at which they can operate the UAV, airspace restrictions, and other factors, are regulated by various organizations around the world. In the United States, this management is done by the Federal Aviation Administration (FAA) [4], and all research and prototypes flown must comply with their requirements.

It is after the requirements for monitoring and controlling the UAV, as well as flight requirements including power capacity for flight time, that the sensor payload and onboard data analysis, or data broadcasting, may be considered in its design. All of the sensors described above are primarily concerned with the UAV itself, maintaining its flight, safety, and the safety of the people and objects around it. The sensors used for data collection will havewill be have to be secondary to these in terms of the UAV's capacity to support them, including for power consumption and lift capacity. The relatively small size of these UAVs is in part an advantage that allows for lower cost

and accessibility, but can also place limitations on their sensor payloads, especially for sensors that are not necessary to maintain safe flight.

While purpose-made sensors do also exist for UAVs, those available may be intended for certain types of UAV platforms that may be restricted for other reasons, or may offset the affordability of the UAV system enough to make it a less practical option than other means of data collection. In order to meet the various limitations of a UAV-mounted system, as well as specific requirements for the sensing application required, a custom sensor system may be an option to consider for various data collection applications.

1.2 Purpose

A common type of sensor for remote sensing applications is a multispectral camera, a type of camera which collects data across multiple subsections, often called bands, of the electromagnetic spectrum. Multispectral sensing allows more information to be extracted from remote sensing data than would be possible from a single camera, to measure light from reflected sunlight or light emitted from other sources, and collect data based on how the surface material interacts with that light. This often includes visual imagery, as well as near, middle, and far infrared imagery, [5] and can be used to extract information about the surface, whether covered with vegetation, soil, snow, cloud cover, buildings, or asphalt, and to make assessments of that material.

For an application like the inspection of electrical utility equipment, a multispectral sensor may be useful in the detection of electrical anomalies, as electrical arcing may heat equipment in patterns measurable by a thermal camera, or may produce an arc flash, which includes an ultraviolet component [6] and may be more measurable to an ultraviolet camera than a broad spectrum visual camera. However, options for UAV sensor packages with those capabilities preexisting are limited. While standard visual cameras are widely available, thermal cameras can have more limited accessibility, and are often designed for specific applications.

Electrical utility equipment monitoring is an application that would potentially benefit greatly from an additional source of data collection, both because equipment often covers large, potentially rural areas, making inspection difficult, and because failures of equipment can have serious consequences, including extended power outages and fires. Advanced sensing capability within the grid can significantly increase awareness of the state of electrical equipment [7]; however, there will likely still be physical limitations to such systems, such as identifying physical obstructions, damage to poles, and precisely identifying the location of a failure. Additionally, such advanced systems may not be readily available to all utilities, including those that are rural and cover wide areas. Relatively low-cost, UAV-based systems could assist in risk mitigation, anomaly detection, and event identification. However, the sensors used by such a system could both need to be able to identify events outdoors, in a variety of conditions including direct sunlight, which could overwhelm a camera attempting to detect another light source such as an electrical arc, as well as not be cost prohibitive for a utility to be able to meet the needs of inspection for a potentially large network.

1.3 Cameras

Many types of thermal and ultraviolet cameras exist for purchase, including those intended for UAV applications. A summary of some of the common options that exist for purchase is discussed in the following section, both in capability and price, to add additional context to the custom camera system that will be examined in the rest of this dissertation.

1.3.1 Existing Cameras

Teledyne FLIR, a widely-used manufacturer of infrared cameras for thermal detection and night vision, offers a variety of handheld and smartphone cameras, as well as UAV

mounted options. The options on their webpage at the time of writing range from $199.99 (USD) for a handheld smartphone camera add-on, to $3,000-$4,000 handheld options. Their thermal payload camera, VUE® TV128+ Payload, is sold stand-alone for around $4,000, or $9,760 for their fully-equipped SILAS drone. Other drone options for pre-installed thermal capability include the Autel EVO II Dual, which is currently priced for $4,799, the DJI Mavic 3, which is currently priced for $5,899, and the Parrot ANAFI, which currently sells for around $7,000. Regulatory concerns may also see limitations in the use of certain drones in certain markets, further limiting options available.

Ultraviolet cameras have fewer built-in UAV options than thermal cameras, and can be difficult to find with listed prices. One camera from Aerial Solutions, an enterprise partner of DJI, sells a camera for the applications of detecting electrical corona in the ultraviolet spectrum for a listed price of €37,900. LDP LLC, a manufacturer of ultraviolet cameras and ultraviolet lenses, sells DJI-compatible cameras with ultraviolet, visual and infrared capability ranging from $862.50 to $3,737.50. LDP also sells ultraviolet modifications of Canon, Nikon, and Sony cameras as well as industrial ultraviolet cameras, though most of these are handheld digital cameras that would be incompatible with UAV use. The UAV industrial cameras sold by LDP range in cost from $265.00 to $2,150.69. A quartz lens assembly for their small industrial cameras, which is needed to capture light through a lens from wavelengths shorter than around 365 nanometers, is priced ranging from $1,085 to $1,587.

Multispectral cameras collect with both a spatial and spectral component, collecting data for the same spatial data across multiple spectral bands often ranging between 5-12 different bands across the visual, near infrared, and far infrared ranges [8]. Examples of multispectral cameras are often more expensive than standalone cameras, such as the Parrot Sequoia+ for $3,850, the DJI-compatible Slantrange 3PX for $5,250, or the Sentera 6X Multispectral Sensor for $10,000.

1.3.2 Custom Cameras

Combining separate cameras as a form of multispectral data collection loses some of the advantages of an integrated multispectral sensor, and introduces potential complications such as variations in camera hardware and firmware, lenses, and differences between the cameras. However, it does allow for the capabilities of the camera system to be chosen specifically for the use case, which can help optimize both cost and other constraints of a UAV payload, such as weight and power usage. Some custom implementations of multispectral sensors, such as the one described in [9], overcome some of these difficulties by having different filters move in front of a stable camera, removing the variability in camera position. Another approach is to use processing, including machine learning, to extract information from data collected by cameras with physical separation, that may not fully overlap [10]. This type of analysis could allow less specialized, lower-cost cameras to be used in a more modular way, improving the accessibility of multispectral equipment for a wider variety of applications.

Machine learning has gained significant prominence as a data analysis approach in recent years, and has an advantage of being able to train the results with training data, allowing different sensors to be used as long as training data can be collected with them. For this research, differences between the hardware and firmware of the cameras, as well as limitations of each of the cameras, presented challenges to more traditional analysis approaches, and so a more easily generalized machine learning based approach was used.

1.4 Research Questions

The research in this dissertation will focus on analyzing strengths and limitations of a low-cost, custom camera system with machine learning based classification for

the application of electrical arc detection, through the use of a custom dataset. The specific research questions addressed in this dissertation are as follows:

1.4.1 Question 1: Can low-cost visual, ultraviolet, and thermal cameras collect enough information to be used for detection of an electrical arc with machine learning?

Before attempting to combine the outputs of each of the cameras, I will first compare image classification accuracies for each of the camera datasets separately. This will allow me to compare the quality of the data in each of the models, as well as the effect of different model parameters on each of the cameras. This will also provide a baseline to compare the ensemble results.

1.4.2 Question 2: How does an ensemble camera system as a means of classifying data for arc detection compare to models for each individual camera?

This question will compare the accuracy of two different ensemble models, with variations of parameters, to the accuracy of models for the individual cameras also across different parameters. This comparison will also be done across the same variety of light conditions as the individual cameras to evaluate how the effect of different ambient light affects the ensemble models.

The visual and ultraviolet cameras capture similar types of data, reflected sunlight, and as a result there may be some overlap in the information they can measure, as well as how ambient light may impact that information. The thermal camera, by contrast, measures infrared light from re-emitted thermal radiation, and may not be affected in the same way. How well an ensemble model can take advantage of data from similar or different sources, and which models may be better suited to do so, will be investigated.

1.4.3 Question 3: How do other conditions impact the ability of a machine learning model to detect electrical arcing from the data captured by each of the three cameras individually? As an ensemble?

This question covers both ambient conditions and parameters of the model. For an outdoor camera-based sensor to detect arcs, or another light source, it will need to perform well in a variety of light conditions. If different cameras perform better or worse in certain conditions, that could emphasize the value of a multispectral camera system, rather than a camera with only one spectral band. Additionally, the parameters of models can significantly affect the output in many machine learning applications. How different parameters can affect not just the overall accuracy of classification models, but also the same models under different light conditions, is also important when designing the sensor system.

1.4.4 Question 4: What are the implications of the accuracy results of the individual camera data and ensemble models, and how can they be used to design a more effective, low-cost camera sensor?

The results of the experiments done in this dissertation are on simplified cases, and cannot represent the full complexity of field applications of arc detection, or UAV-based sensing more broadly. However, the goal of this research is to identify strengths and weaknesses in this process, from the hardware, data collection for training, machine learning approach, and ensemble approach, that can help inform field applications for custom sensor systems. This question will address those lessons learned, and how they could be expanded on in future work.

Chapter 2

Literature Review

This research, and the larger AIMS project, is based on the combination of technologies and therefore the integration of multiple areas of research. This literature review intends to provide a general overview of these areas, as is relevant to the later sections of this paper. The areas of interest reviewed here include the following: The integration and fusion of data from related sensors. The use of drone-based sensors, including implementations and applications. The use of multispectral data, specifically camera data in different types of monitoring and analysis. Finally, machine learning for detection and classification, specifically for image data, and some discussion of approaches to analysis and sources of noise that may be considered for image data.

2.1 Sensor Integration and Fusion

Early research in the field of data fusion, or the combination of data from multiple sensors or related datasets, was done as in part by the US Department of Defense for military application, including remote sensing applications and as guidance for autonomous vehicles. Applications expanded into civilian technologies, including robotics, maintenance of complex machinery, and medical uses [11]. These

applications included object or target detection and tracking, and identification of targets or events.

A distinction between multi-sensor integration and multi-sensor fusion is defined by [12], where multi-sensor integration is described as the use of multiple sensors to accomplish a task, while multi-sensor fusion is the combination of information from multiple sources into one representational format. The combination of information from different sensors offers potential advantages in the accuracy of the information, through the reduction of uncertainty through redundant data and the statistical benefit of multiple observations. Measurements of separate features by different types of sensors allows the information from those different features to be used by the system. Additionally, the combination of lower power sensors may provide lower cost options over individual sensors [13]. Potential challenges of a multi-sensor system can be the result of error of individual sensors such as damage or calibration issues, or issues in the combination, due to inconsistent measurements between sensors, errors in control systems, or challenges with feature identification from potentially large amounts of trivial data.

Error can be partially managed by using measurements from the individual sensors to characterize the uncertainty and error of the specific device, which can be used as part of the integration process. Additionally, input from the sensors can be used to build and update a model of the environment, to assist in characterizing measurements taken, and the various states of the environment that are being measured [14].

The level of processing at which data is fused varies with the system and application. Information can be combined at the signal level, or pixel level for image processing data, with minimal processing, or real-time noise reduction, redundancy, or similar applications. Combination at the feature level combines data that has already been processed enough for features to be identified. Symbol or decision level fusion combines information that has already been processed to represent an aspect of a task [15]. The specific approaches depend on the sensors, processing capabilities, and objectives of the system designed.

2.2 UAV Based Sensing

UAVs, or airborne drones, can take many forms, and be suited to a wide variety of applications, including many contexts of data collection and monitoring. Commonly used classifications of drones include VTOL (Vertical Take-Off and Landing), LALE (Low Altitude, Long Endurance), and HALE (High Altitude, Long Endurance), [16], variously suited to different response times, sensor payloads, and flight times, which also often correlate to different costs and flight restrictions. An example of a quadcopter drone, a type of VTOL drone, is shown in Figure 2.1, image reference from [17].

At relatively high altitudes, drones can cover large areas for data collection that can be comparable to data that can be gathered by satellites, but with more control



Figure 2.1: An example of a quadcopter drone.

of the flight path and, depending on the sensor payload, higher resolution due to less atmosphere between the sensor and the ground, and potentially less restriction on payloads. As a result, they can be a good option for collecting high quality data for training and validating classifiers.

A specific example, from [18], is used in mapping agriculture, specifically to augment satellite data for the identification of rice plants to map growing areas and monitor growth. This type of data collection can include sensors other than visual cameras, such as in [19], which used cameras with multiple visual wavebands in the range from 400-900 nanometers (nm) in addition to thermal and video data. This paper also investigated applications of classification on agricultural applications, including the identification of leaf chlorophyll content and canopy nitrogen content, and how the use of multiple wavelength bands can augment the process.

UAV collected thermal data was utilized by [20], for the measurement of roosting fruit bat populations, using a Zenmuse XT 19 mm radiometric thermal camera. This approach gave researchers access to environments that would otherwise have been difficult to access for human researchers. Accessibility to difficult terrain or environments are a strength of drones, also described in [21], which combined RGB (Red, Green, Blue channel) camera data and multispectral imagery in the visual and near-infrared spectrum, which were collected with drones to measure shallow water bathymetry. Drone collected camera data used to monitor soil erosion is described in [22], with the application of collecting reliable datasets to assess vegetation and soil cover. Similarly, [23] used camera data to estimate vegetation cover and biomass.

In addition to data collection for classification, drone collected data can also be used for real-time or near real-time parameter detection and identification. Drones were used for forestry applications in [24] to improve response time and data quality to assess pest outbreaks, wildfire spread, wildlife and flora. Data for these applications was variously collected in visible light, near-infrared, shortwave infrared, thermal infrared, Radar, and Lidar. The use of a combination of ground-based cameras and drone-based sensors to monitor wildfires is also described in [25], utilizing visual and

infrared cameras. This paper describes the combination of camera data to estimate variables such as flame height and fire direction.

Drones can also be equipped with other sensors beyond cameras, such as chemical sensors for gas detection as in [26], presented as a low-cost option that can operate in hazardous conditions, with the collected data transferred over a wireless network to an external controller. Drones can also be used to deploy equipment that will remain in place, such as in [27], which describes a system to deploy sensors to measure vibration to assess health of civil structures, with potential applications in remote or difficult to access areas.

The use of drones to monitor remote electrical substations in hard weather conditions is described in [28], facing challenges including performance and safety in difficult operating conditions, challenges to navigation in mountainous terrain, and interference from weather and electromagnetic equipment.

The use of drones to inspect the quality of ceramic insulators is described in [29], where a user-operated quadcopter drone is described carrying a visual RGB camera, an infrared camera to measure temperature, and a video transmitter. A Raspberry Pi is used for onboard computer vision for the results for the inspection. Other camera data was also processed by a ground-based computer.

2.3 Multispectral Analysis

The first US manufactured Earth-looking satellite was launched in 1960, and the first use of satellite data for remote sensing was established shortly after [30]. Satellite data, and other overhead data collected from aircraft, has been used in the study of wide areas of land for many applications ever since, including assessment of renewable and non-renewable resources.

The use of data collected by Earth-looking space borne or airborne sensors for studying wide areas of land has continued to be used and developed since. These systems generally rely on illumination from the sun to collect information from

optical wavelength bands, taking advantage of the variable absorption and reflection of different materials to assist in data analysis and the detection of objects of interest. Exceptions could be artificial light sources, or living organisms with infrared emission, as could be found in several sources listed in the prior section.

Multispectral data is electromagnetic data collected across multiple, often many, wavelengths, often in the visible or near visible range, generally collected by RGB (red, green, blue) cameras. Additional spectral data can provide information of use in determining physical and chemical properties of the area or object being observed that may be invisible in the visual range, while RGB information may also provide spatial context. An early example of a source of multispectral data from a space-based source was Landsat-1, a collection of satellites launched between 1972 and 1975, which collected data in the visual, near-visual, and thermal bands [31], as well as later Landsat constellations. Much of the data measured with these cameras is reflected sunlight, especially in the visual and near-visual bands, while much of thermal bands is energy from sunlight that has been absorbed and reemitted as lower energy infrared radiation. An image of the electromagnetic spectrum of sunlight is shown in Figure 2.2, image sourced from [32].

As would be expected in a field with many users and applications, a variety of techniques have been studied and used to combine data taken from different spectral bands to optimize the resolution and utility of the data for whatever application is relevant. A summary of many techniques used, particularly for data collected by space borne systems, is described by [33]. This review describes analysis techniques used both for multispectral data, or data combined from multiple distinct spectral bands, and hyperspectral data, or data collected with a larger span of continuous spectral bands. Examples of datasets evaluated in this comparison include data from AVIRIS: Airborne Visible/Infrared Imaging Spectrometer, which measured solar reflected light from 400 nm to 2500 nm developed by the Jet Propulsion Laboratory [34], and HYDICE: HS Digital Imagery Collection Experiment [35], an aircraft mounted hyperspectral sensor, sponsored by the U.S. Navy Space and Warfare

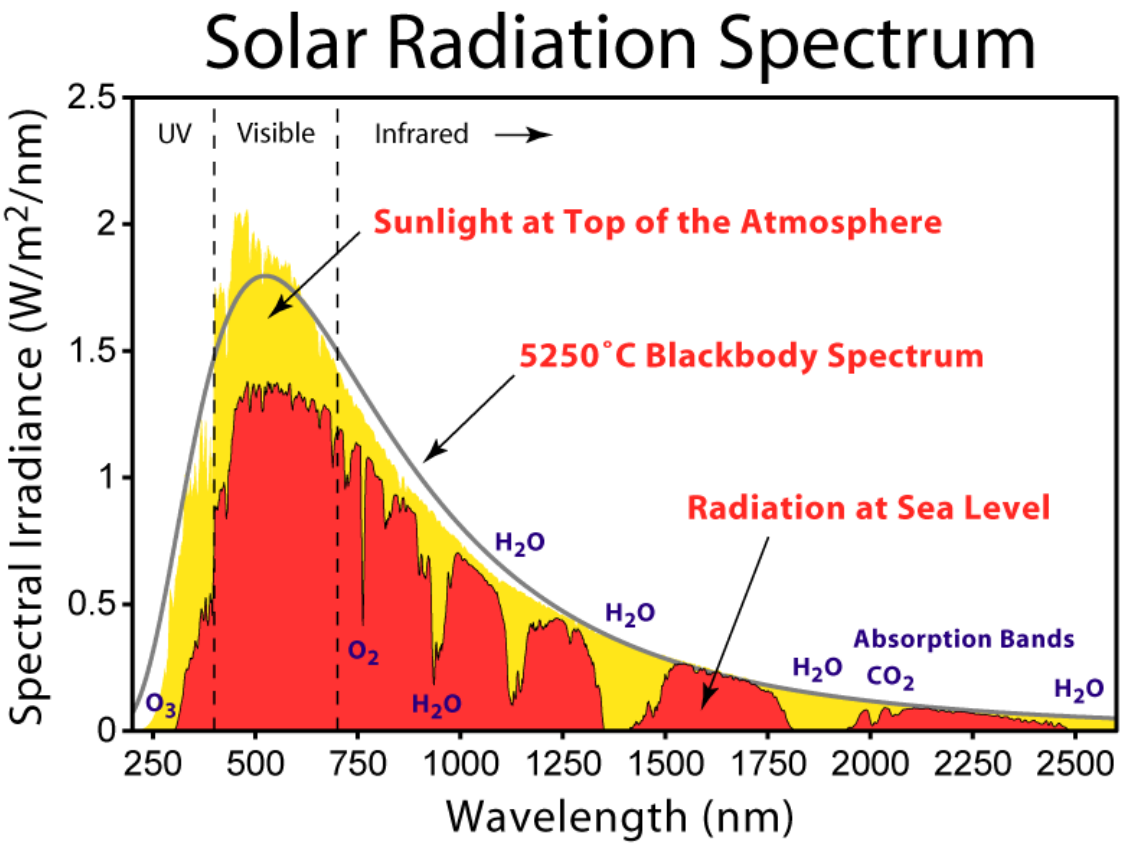


Figure 2.2: Electromagnetic spectrum of sunlight, the light to most Earth-looking satellite or aircraft sensors, including the light above the Earth's atmosphere.

Systems Command. These datasets generally consist of data covering very large areas, but with image resolution that maybe limited by noise, limitations of the sensor, or limitations of information in the waveband being measured.

The datasets in the cited review are evaluated with techniques such as Stochastic Mixture Models [36] and wavelet based approaches [37], which take advantage of the fact that those datasets generally contain overlapping data, where each of the spectral bands are taken with the same sensor equipment. The information in the different bands therefore has a redundancy to assist in the combination of the spectral bands.

However, this level of redundancy is not found in every application of multispectral data, as smaller collection systems, such as drones, may have more strict limitations of the weight or power of the sensor payloads than the equipment used to collect the

larger datasets, or may have other limitations on the quality or design of the sensors. As a result, different approaches to data processing and analysis are sometimes warranted. A quadcopter drone mounted sensor system is described in [38], which takes data with an RGB camera and eight monochrome cameras with filters ranging from 390 to 950 nm from ultraviolet to infrared. These drones are used to collect data for agriculture, using sunlight reflected from different areas of land, but from lower altitudes and with more direct control over the movement of the platform than higher altitude sensor approaches. The location of these points is recorded with Global Navigation Satellite System (GNSS) data, for use in future analysis. Other UAV based applications of multispectral data for precision agriculture include the use of normalized difference vegetation index (NDVI) to measure yield of maize in [39]. NDVI utilizing bands of near-infrared (NIR) and visual to measure vegetation, and collected data with two cameras, one RGB and one NIR. UAVs with thermal imaging were used by [40] to assist with irrigation management of sugar beets.

A review of hyperspectral imaging for agriculture and forestry was done in [8], This paper includes of description of commercially available hyperspectral sensors for UAV use, noting the limitations that the development of such sensors require coordination between the sensor manufacturer, the UAV manufacturer, and an additional party to provide integration between those system. These requirements can limit both the access utility of commercially available sensors for UAV applications, through higher price points, limited supply, and compatibility issues with other UAV platforms.

Another example of a dataset with multiple, smaller cameras is the KAIST dataset [41], a multispectral color and thermal dataset for pedestrian detection. Based on this dataset, and a similar CVC-14 Dataset [42], an approach to combining data that isn't fully overlapping and redundant is described in [10]. Lack of overlapping data is more likely to be collected in cases with more specialized, often smaller specific application-based sensors such as drones. Such data can lack several of the advantages in more fully overlapped datasets, and present different challenges. The KAIST dataset is fully overlapping, but this paper uses geometric transformations to modify the dataset to

comparatively test an approach for unpaired images, based on an SSD-like neural network, or Single Shot MultiBox Detector as defined in [43], a convolutional neural network that assigns bounding boxes to detected object classes.

Another approach to handling unpaired data for pedestrian detection is described in [44] which uses a generative adversarial network, or in [45], which uses a cyclical CNN model that loops through a fusing and refinement of thermal and RGB data, which performed well on well-aligned data and had challenges with misaligned data.

Autonomous vehicles are an example application area for the processing of multispectral data that may not be perfectly overlapped. For this application, the data will also need to be processed very near the time of collection. Object detection for autonomous vehicle applications is investigated in [46], using a custom dataset generated by side-by-side cameras collecting RGB, near-infrared, mid-infrared, and far-infrared image data. The study then performed object detection with an ensemble detector, using the YOLOv1 model to detect objects for each individual camera, then combining the outputs for a final decision, generally finding improvement, if not always significant improvement, for the ensemble result.

2.4 Camera-based Arc Detection

Electrical arcs emit light in the ultraviolet as well as the visual spectrum, and the use of detectors for ultraviolet light has been investigated in several contexts before.

The detection of electrical arcs is a field of interest to many applications, including those suited to camera-based monitoring. The use of camera-based detection is used as a step in a hierarchical algorithm in [47]. A microphone array is used to identify a characteristic sound by an electrical arc, and is used to signal the system that one has occurred. The system then estimates the distance to the fault, and verifies the fault location by checking for high temperatures on the insulator with a thermal camera. [48] instead uses a charged coupled device (CCD) camera with a high frame rate to

capture an image of a low voltage electrical arc, and uses an edge detection algorithm to identify it.

Detection of arcs in an pantograph-catenary system on a train is discussed in [49]. Arcs occur when the contact wire does not stay attached to the pantograph, or due to wear or damage to components. A vision based detection method described in [50] detects arcs from a video data recording the pantograph. The use of a photomultiplier tube to detect ultraviolet light for arc detection of a pantograph is described in [51]. In [52], data from an infrared camera is stored and analyzed to detect overheating or arcing to assist with preventative maintenance. A computer vision based approach for pantograph-catenary arc detection based on Otsu's method is described in [53], using a collected time series of grayscale images.

The value of ultraviolet detectors in arc detection is discussed in [6], which characterizes spectra measured from electrical arcs used in urban rail, measuring from 200-1100 nm. Parts of these spectra were located in the UV range in the solar-blind region, indicating the UV light could be measured without too much interference from sunlight, and could therefore be useful data in arc detection. Another application, described in [54], uses an ultraviolet sensor, which measures the presence of ultraviolet light as a signal, to detect partial discharges on ceramic insulators. The resulting spectrograms were classified with k-nearest neighbour (kNN), Naive Bayes, Decision Tree and SVM classifiers. Another study, [55], used images of a porcelain insulator taken with a FILIN6 UV imager with flashover, and used AlexNet, a convolutional neural network, to classify the results and evaluate their use as a flashover detector. The use of a combination of ultraviolet and visual cameras were also described as a low-cost detector for coronas caused by partial electrical discharges in aeronautic environments in [56]. Long exposure images of corona were measured with a Sony IMX586 backlit stacked CMOS sensor in [57] to study the relationship between the energy of the images and the energy dispersed by the partial discharges, suggested for application in aircraft environments.

Spectrum data of electrical arcs for the use of optical arc detection for aircraft power systems was measured by [58]. They compared anode and cathodes made of copper and aluminum with DC power, and found wavelengths λ throughout the UV-A and UV-B spectrum, with peaks around 307, 324, and 392 nm, as well as others in the visual spectrum.

Looking at different sensor implementation, [59] compares the effectiveness of three different sensors, ultrasonic, and RF antenna, and an infrared camera to detect different types of arcs, dry band arcing, corona, and surface discharge. Of these, this paper only had success in detecting dry band arcing with the infrared camera.

2.5 Analysis and Machine Learning

Many of the image based classifiers and detectors described in the above sections utilized some kind of neural network, as do the classification model assessed within this dissertation. Machine learning has been implemented in many data analysis applications including identification of features, or an identifiable section of a system [60], and patterns that may under other conditions require an expert to identify. Usually the data for classification is collected, formatted and preprocessed prior to classification or identification by the model. Models are either usually regression based, or define a relationship between data and future values, or classification based, or identify and separate data into distinct class labels [61].

The quality of the dataset used for training can also impact the quality of the trained model. An imbalanced dataset is one where there are different numbers of samples of one class than another, such as for a rare event that is difficult to collect data for. When an imbalanced dataset is used to train a model, the model will likely prioritize the samples with the more data [62].

Neural networks utilize artificial neurons, data structures with value based activations and weights that can be set with training, connected into layered network. During training, theses structures extract features from input data. A many layered

neural network is called a deep neural network, and has shown advantages over other machine learning approaches when identifying patterns in complex data [63].

An artificial neural is based on a simplified model of a biological neural, with an output determined by the weighted sum of some n inputs, modeled by

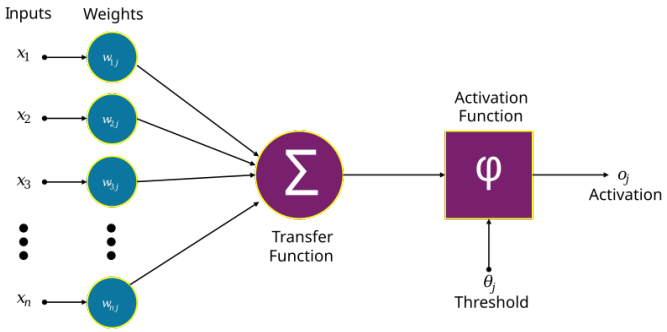
$$O = f(\text{net}) = f\left(\sum_{j=1}^{n} w_j x_j\right)$$

where w represents weights and x represents input values [64]. For the simplest case, the value of the output is defined as

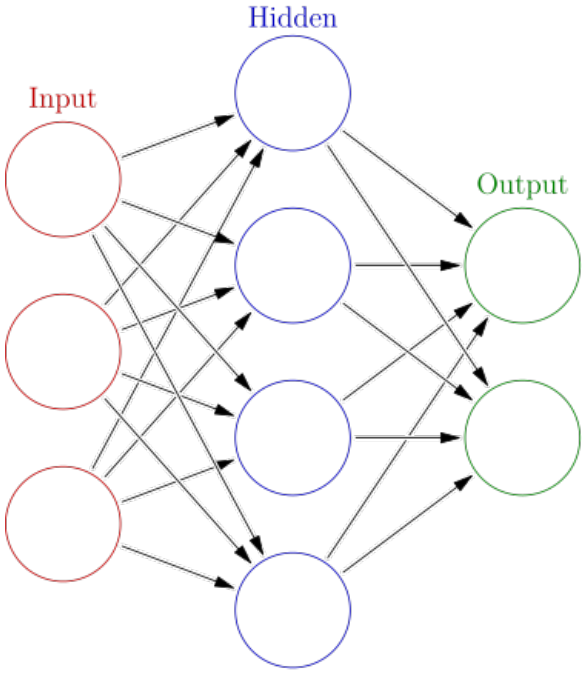
$$O = f(\text{net}) = \begin{cases} 1 & x \text{ if } w^T x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

though more complex threshold functions are used in many applications. A single artificial neural, also called a perceptron, is only capable of solving linearly separable problems, and more complicated problems are addressed by forming a larger network. A simple, linear neural network is the combination of layers of artificial neurons combined [64]. Some examples diagrams are included in Figure 2.3. An example of the structure of a simple artificial neuron is shown in Figure 2.3a from source [65]. The structure of a simple artificial neural network is shown in Figure 2.3b, from source [66].

The process of training a neural network involves passing data through the network first forward, and calculating the error, or loss, of the network with the current parameters compared to the ground truth values, or target. The weights are then modified according to stochastic gradient descent, which uses the partial derivatives of the loss with respect to each weight to calculate the magnitude and direction of the error, and therefore the gradient to follow when modifying the parameters to move them towards ground truth according to the following equation, in which where z represents the loss and w^i represents the weights within the network.



(a) Structure of artificial neural.



(b) Artificial neural network with linear layers.

Figure 2.3: Neural network diagrams.

$$(w^i)^{t+1} = (w^i)^t = \eta \frac{\nabla z}{\nabla (w^i)^t} \tag{2.1}$$

The process of passing this error value back through the network to train it is called backpropagation, calculating the error of the network, then propagating backward through the network and making corrections on the weight values. This overall algorithm is represented by

$$\Delta w_{ij}(n) = -\eta^* \frac{\nabla E}{\nabla w_{ij}} + \alpha^* \Delta w_{ij}(n-1) \tag{2.2}$$

where η and α respectively represent the learning rate, or size of changes applied for each partial derivative, and momentum, or the effect of past weight changes that will be applied to the current change [64].

Convolutional neural networks, or CNNs, are neural networks which contain convolutional layers for feature extraction, and have several advantages over other types of the previously described fully connected neural networks. Each neuron in a CNN is only connected to a small number of neurons in the previous layer, reducing parameters compared to linear layers, which are connected to all previous neurons. CNNs also frequently contain pooling layers, which combine inputs of local weights and can take advantage of local image correlation to reduce the amount of data while retaining useful information [67]. As an artificial neuron is based on a simplified biological neuron, the kernels that make up a CNN are inspired by a simplified version of biological visual perception. A diagram of a CNN used for image processing is shown in Figure 2.4, from source [68].

A common input structure for a CNN used for color image data is a three order tensor, with H rows for height, W columns for width, and three channels for color input (R, G, B). Similar to fully connected networks, a CNN is made up of layers, which include convolutional layers but also often include fully connected layers, normalizing layers, pooling layers, and other architectures that depend on the structure of the overall network being used.

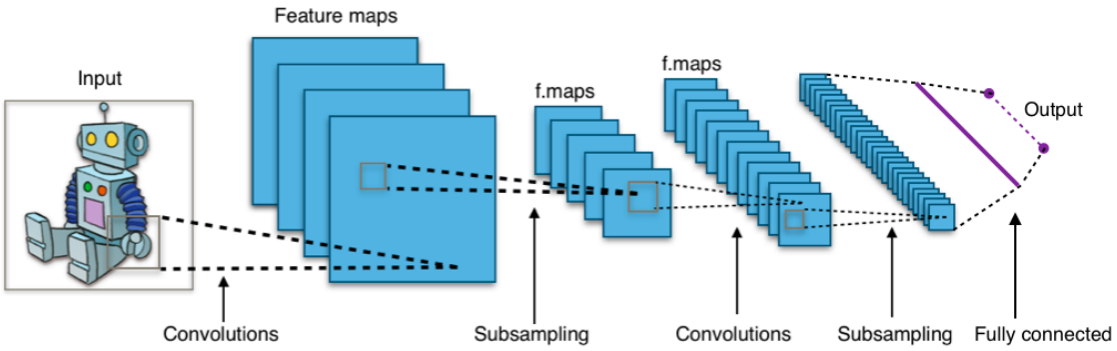


Figure 2.4: Diagram of a CNN used for image processing.

The process of training a CNN is generally similar to training other types of neural networks, defined by minimizing the error, or loss, of the calculated value relative to ground truth values [69]. A diagram showing the processing of convolution used to calculate the inputs of successive layers in a CNN model is shown in Figure 2.5, image source [70].

While many applications of machine learning are trained on pre-established datasets, the use of custom sensors as a data source both for collecting datasets and for real-time applications is also widely established, discussed in [71]. However, any data collected will need to be cleaned, to have errors and redundancy removed and parameters identified, and preprocessed in a way that is consistent between the training data and the data used in an application.

Ensemble learning describes the use of more than one model to process data, where the results are combined into a single output with the intention of improving the quality of the prediction. A common approach is to train different model architectures on the same data and combine the results. This approach was used in [72], which classifies medical images according to the results from four different medical image datasets based on the combined results from different models. The processing of combining different models is referred to as stacking. This was used in combination with other techniques such as augmenting, or performing rotations or

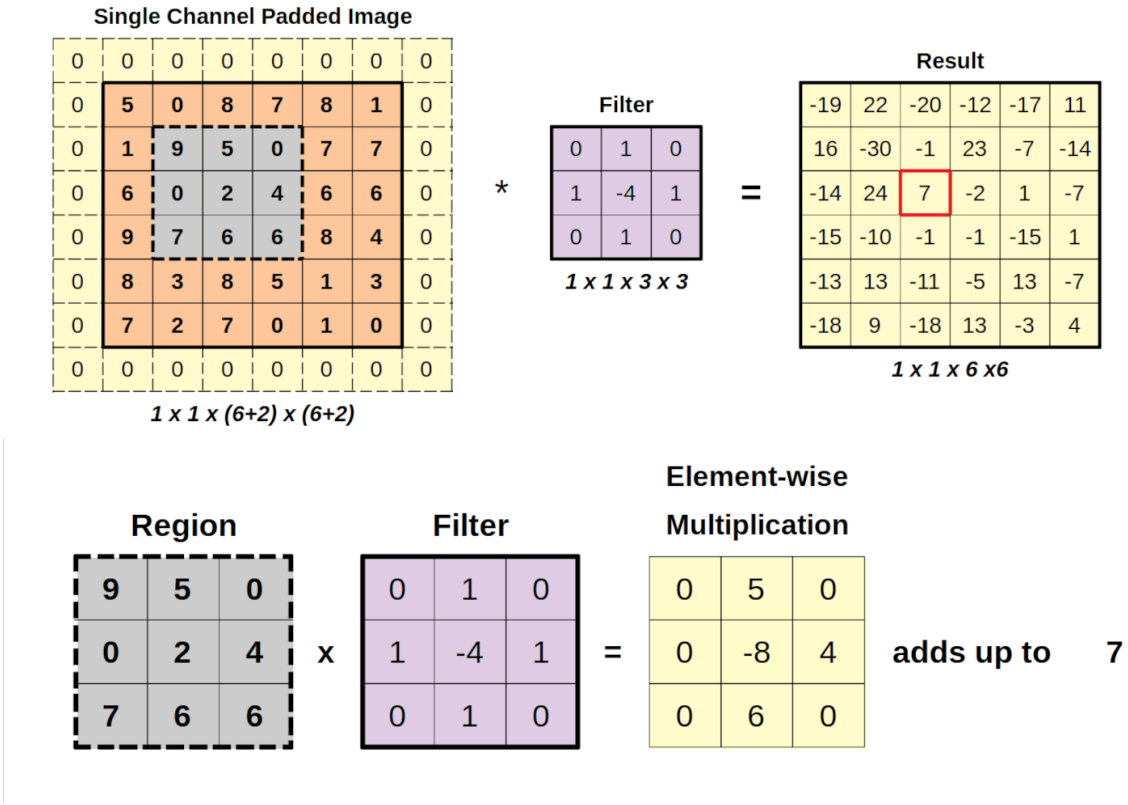


Figure 2.5: The process of convolution in CNN layers.

other modifications to training data to reduce overtraining, and bagging, or training the multiple models from randomly chosen selections from the training dataset.

Ensemble methods are discussed in the survey [73] addition to other classification approaches for hyperspectral data. The ensemble methods described include the use of SVM and random feature selection, the integration of different CNN based classifiers, and the use of separate CNN and deep residual classifiers, as well as different approaches to the integration of spacial and spectral data provided by hyperspectral sensors. In [74] a custom CNN architecture was used for the base classifiers, concatenating the results to models augmented with Gaussian noise before fusing the results. A CNN ensemble used to address small sample size in hyperspectral data is described in [75], using a pixel-pair feature (PPF) algorithm to integrate with an ensemble CNN classifier followed by majority voting.

An example of the use of an ensemble model to address uncertainty from lighting conditions is discussed in [76], which describes the use of image classification in a robotic grasping system for an industrial application. A random cropping ensemble neural network (RCE-NN), which uses randomly cropped subsections of an image of the material to be classified, used an ensemble of CNN based feature extractors made of of AlexNet_bn, Inception, VGGNet, and ResNet, and combined with a weighted majority voting algorithm.

2.5.1 Sources of Data Noise

Data collection, for images and videos, can introduce different types of noise and present different types of challenges. When collecting data in non-ideal conditions, which will be required for a system that will need to operate outside mounted on small drones, the data classification system will need to function with images that contain noise and distortions. There are multiple types of noise that can be introduced into an image, either when the image is captured or in saving and transmitting [77]. Impulse noise, or salt and pepper noise, black and white dots that appear on the image, is

often the result of dust particles or faulty components in the camera. Gaussian noise, or amplifier noise, is when the value of each pixel is added to a random noise value, as determined by a Gaussian distribution. Poisson noise, or photon noise, is caused when not enough photons are measured to provide detectable information. Speckle noise is caused by random multiplications of pixel values and generally occurs in radar measurements [78].

2.5.2 Summary

This literature review aimed to provide a summary of the areas of research most relevant to providing context and background to the research described in the following sections. This included prior applications of drone-based sensors, and how the data from them was collected, used and applied. This also included different kinds of multispectral sensors, approaches used to analyze them, and why some of those techniques may or may not be relevant to the types of data collected in this work, and the related use of camera and other optical based electrical arc detection. Finally, a short summary of machine learning approaches and sources of data noise was included to provide context to the approaches used in analysis described in Chapter 4, as well as those considered for future work. A summary of how some of these examples relate to the research described in later chapters is described in Figure 2.6, including, respectively, [9], [10], [59], and [57].

Literature Examples	Dissertation research
A Multispectral Camera Development: From the Prototype Assembly until Its Use in a UAV System • Custom multispectral camera with filter wheel, visual to near infrared	A light weight, low-cost camera, but can only take one image at once, does not include an ultraviolet option, and requires more specific build.
MLPD: Multi-Label Pedestrian Detector in Multispectral Domain • Pedestrian detection with side-by-side cameras, not fully overlapping data, as is more common	Similar side by side camera approach, different application, with a focus on collecting an effective training dataset, rather than uses of an existing dataset.
A Study on the Ability of Different Non-Intrusive Sensors for Diagnosing Outdoor Insulator Defects • Investigate detection of corona, surface discharge, and dry-band arcs with infrared, RF, and ultrasonic sensors	Measurements were taken in outdoor conditions instead of a lab in order to compare interference of sunlight and variable movement and position.
Insulation Failure Quantification Based on the Energy of Digital Images Using Low-Cost Imaging Sensors • Lab measurement of UV and visual data from arc corona.	Long exposure images would not be well suited to UAV application, application is outdoor equipment rather than low pressure aircraft systems.

Figure 2.6: Literature examples with relevance to dissertation research.

Chapter 3

Hardware and Data Collection

The research in this dissertation began as an aspect of sensor development for the Autonomous Intelligence Measurement Sensors and Systems (AIMS) project, a prototype autonomous drone response system intended to assist electrical utilities with maintenance, and disaster prevention and response. The AIMS system is a network of UAVs with access to information from utility sensors and ground-based processing systems. When directed by the overall system, whether in response to some detected event or anomaly or according to planned maintenance inspections, a UAV with a sensor payload can be sent to a given area for data collection. Based on that first UAV's findings, additional UAVs with different sensor payloads may be requested for assistance in collecting additional data. This data is then passed to ground-based systems for processing, decision making, or to be assessed by human analysts or technicians.

One of the goals of AIMS was to include low-cost options, both for the UAVs themselves and the sensor payloads. This was to present a more accessible option than other prebuilt UAV sensor options that may be available, and also to account for the risk of the field operation of UAVs used in partially or fully autonomous operations. Lightweight and lower power usage options were also examined in order

to optimize flight time, which can be a significant limiting factor for small quadcopter or hexcopter UAVs running off battery power.

One sensor type of interest is a multispectral camera system, with capacity for thermal detection to identify overheating components and potential sources of fire, as well as an ultraviolet camera to assist in the detection of electrical anomalies including electrical arcing. The options to purchase such a combined system were limited, especially when considering the cost of the entire system of the UAV, sensors and other needed components. To address this challenge, an investigation into a custom camera system was developed using separately purchased industrial cameras, which extended into this dissertation work.

The objective of this research is intended to test different aspects of the camera system, including the capability of the cameras separately as well as with ensemble implementations. The conditions used for the collection of the datasets described in this work are simplified cases compared to real world conditions that may be expected in field applications, however, the challenges of outdoor measurement were a focus of the comparison, as the equipment measured by the AIMS system would all be outdoors, and in a variety of weather conditions.

Simplified cases of real world challenges are included in the form of variable light conditions such as sunlight, which can be one of the primary challenges for camera based data collection, especially when measuring a light source such as an electrical arc. The results from the experiments done in this research can help inform how to approach more complex challenges as this system and others like it mature.

3.1 AIMS Project and Drones

A summary of the Autonomous Intelligence Measurement Sensors and Systems (AIMS) project is provided here to give context to the camera based sensor system used in this research.

AIMS is a prototype autonomous drone response system, consisting in its current form of different options of UAV platforms and sensor payloads, including low-cost options for accessibility for a variety of electrical utilities across a variety of use cases. The AIMS system's planned implementation is to be connected to a larger alert system, such as a utility's outage management or Supervisory Control and Data Acquisition (SCADA) system, which can then provide information on the time and general location of some kind of anomalous event, or any other unknown that the utility may want to gather more information on without immediately sending workers. The AIMS UAVs with their varying sensor payloads can be sent to the location of interest to determine more definitive information, such as precise location or prediction of the nature of the event, with the goal of aiding the response for the utility's human crews.

AIMS UAVs, or drones, are separated into two primary types: scout drones and specialists. The scout drones are intended to be general purpose use, sent to the general location of a known or suspected event to identify a more accurate location of the event, which may be visually obvious, such as a fallen power pole, or some more difficult to identify equipment failure. A scout drone will carry a standard sensor payload, including a camera as well as other options that are still being evaluated, to make a preliminary assessment of the event's location and potential cause. This information can then be used to identify the more specialized drones to send to the identified location if the cause of the event cannot be readily identified. Specialist drones with more specific sensing capability can then be sent to more definitively identify the cause of the event, or other information that may benefit utility workers.

One such set of sensors, which has been used for initial lab and field testing, is a triband multispectral camera. The intention of this camera system was to combine three types of spectral data: visual, ultraviolet, and infrared, each of which can provide additional information for the identification of certain types of events. The images or video taken from the three cameras is time synchronized and with a consistent physical offset, intending to aid in the use of this camera as a combined

sensor in order to improve capacity for event detection over a visual camera alone. The goal of this camera is not only to measure an anomaly detectable in each of these spectral ranges separately, but to also combine the results for improved capacity for event detection.

An example of an event of interest is electrical arcing, which may occur with damaged electrical equipment. As described in the literature review, electrical arcs emit light in both the visual and ultraviolet components of the electromagnetic spectrum. The presence of arcing also results in heat, giving potential information to all three camera components of the triband system.

Building on the initial research done for AIMS, in this research I will evaluate the approach to data collection with these cameras, as well as compare how well the collected data from each individual camera can be utilized for the detection of an electrical arc source under variable outdoor light conditions, as well as how well the prediction performs as an ensemble.

3.2 Hardware

3.2.1 Camera Prototype

A goal for some of the example specialist drones was to have a multispectral camera, for the detection of electrical arcs or other light or heat sources that may present fire or safety risks, that was in the range of $1,000-$2,000 for the drone, including the full payload. More typical uses of drone based camera detection, such as those described in the literature review, are often much more expensive for the cameras alone, so the use of industrial and other lower-cost cameras was considered as an option. However, these cameras have several disadvantages over prebuilt multispectral systems.

The prototype multi-camera implementation used in this research is based on one our research group developed for UAV based demonstrations, with modifications to streamline data collection of the custom dataset. The cameras used were an ELP 48

Megapixel USB visual camera, a LDP LLC XNiteUSB2S-MUV: USB 2.0 Megapixel HD Monochrome ultraviolet camera, and a USB FLIR Lepton 3.5 thermal camera. These three cameras were set in a 3D printed mount like the one shown in Figure 3.1, which maintained the order and spacing of the three throughout the dataset. The prices for these cameras, at the time of writing, are $135.99 for the ELP visual camera. The LDP ultraviolet camera is $245 with the stock glass lens. A quartz lens would improve the spectral range from 365-380 nm to 300-380 nm, with the glass blocking wavelengths shorter than 365 nm and a shortpass filter blocking visual light above 380 nm, but the cheapest quartz lens sold for this camera is $1,085, and so was not included. The FLIR Lepton is $164 for the camera itself, with an additional $119.99 for the PureThermal Mini Pro JST-SR webcam, which processes in input from the FLIR Lepton, turning the inputs into a functional thermal camera, developed by GroupGet's GetLab. The FLIR Lepton thermal camera is a long wave infrared (LWIR) camera, with a range of 8,000 to 14,000 nm. The PureThermal Mini Pro JST-SR processes the output data from the FLIR Lepton with the open source firmware to display the output.

The data collection approach for the AIMS prototypes had the three USB cameras connected to a Raspberry Pi 4 micro-controller when mounted on a UAV. To receive data from the drone while in flight, the camera stream is sent over an RSTP [79] stream over a local Wi-Fi network. This stream was then captured by a local device and saved, either using VLC, or FFmpeg [80], a command line tool for recording and converting digital and audio data to save video output in set time increments script, within a bash script.

For ease of use for the data collection for this research, a python script using the OpenCV library was used instead to capture individual images, which were captured and stored on a laptop before cleaning and processing for machine learning. The details of this process are included in Chapter 4. A diagram of the process of data collection is shown in Figure 3.2.



Figure 3.1: A version of the triband camera prototype in a 3D printed case.

3.2.2 Spectrometer

In addition to camera data, spectral data was collected during the data collection, both of the source and ambient conditions for the validation data to provide additional context to the results. This was done using an Ocean Optics USB4000 Spectrometer. This spectrometer is a small, portable USB connected fiber optic spectrometer[81]. Data can be collected from this spectrometer either with Ocean Optics' own software, OceanView, which I used initially to understand the general behavior of the spectrometer under different light conditions, as well as with the third party API Seabreeze [82]. For most of the measurements I used Seabreeze to capture the data with python scripts, so that the data could be more easily collected and displayed, and so that the same Ubuntu laptop could be used to collect data from the spectrometer as the USB cameras, since OceanView does not have a Linux version of the application.

The code used for collecting the spectrometer data is included in Appendix A.

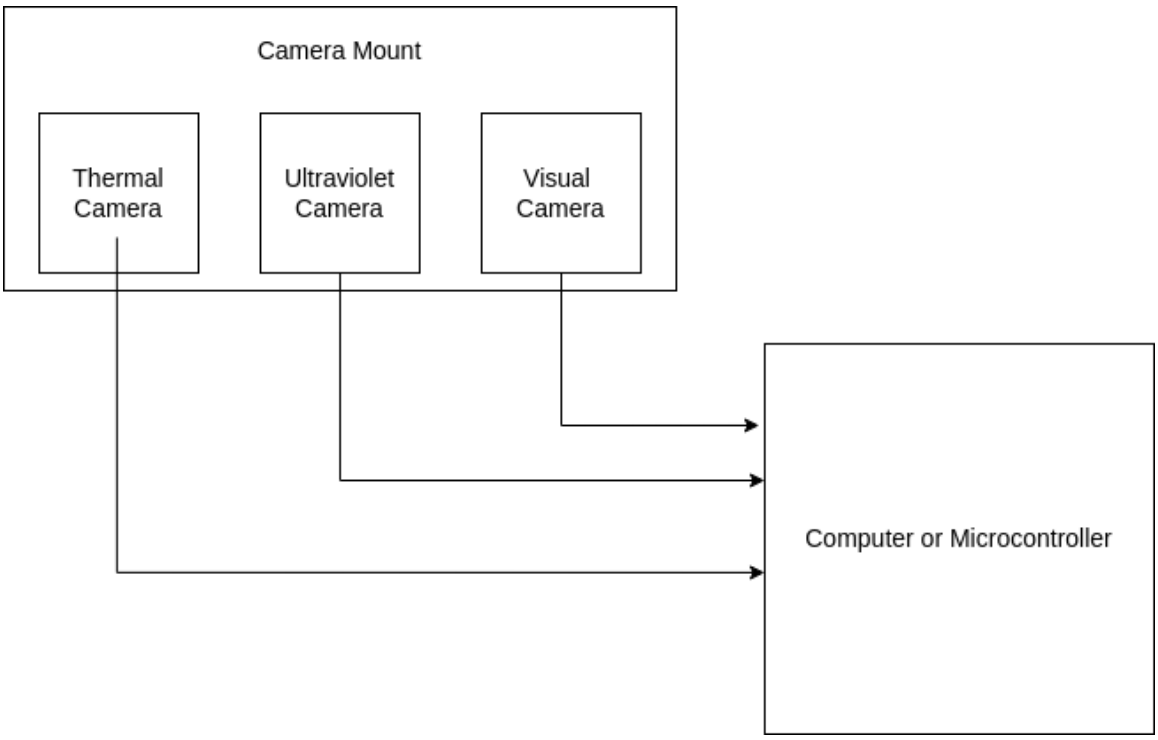


Figure 3.2: A diagram of data collection from the three cameras.

The spectrometer was used both to collect data of the arc source to provide context for how much the source overlapped with expected ranges of the ultraviolet and visual cameras, as well as the ambient conditions used for the validation datasets to evaluate how much the ambient light overlapped with the source to provide context for the performance of the classification results.

3.2.3 Arc Source

The arc source used was a Zippo Rechargeable Candle Lighter, which emits a small electrical arc intended for use as a household lighter, and was therefore a small, relatively safe option for testing the behavior of the cameras, separately and combined, for detection of a source with visual and ultraviolet light emission, which also heats up when used in a way that is measurable to the thermal camera.

The electric lighter in this project acts as a stand-in for electrical arcing that could occur on electrical equipment. It is not intended to perfectly represent a large arc

that may occur on equipment used by an electrical utility, as realistic arcs on real equipment was out of scope for this stage of sensor development for both access and safety reasons. With the spectrometer data of this lighter, I intend to show that this small arc source is adequate to generate measurable images in the ultraviolet part of the spectrum, which will be adequate as a proof of concept for the use of cameras for other data that contained an ultraviolet component, including larger electrical arcs. Additionally, this arc source can be used to examine the difference in classification accuracy by the three different cameras, and how that result can be improved.

A spectrometer measurement of the arc lighter alone taken with the spectrometer is shown in Figure 3.3. Example images captured by each camera of the lighter for each of the two classes, 'on' or 'off', is shown in Figure 3.4. This data was taken indoors in a relatively dark room to limit noise. A measurement of the background was taken just before this one was collected, using the Seabreeze API to put a time delay between the readings to allow the two readings, with the lighter off and on, to be taken close together for more accurate removal of background noise. Multiple measurements were taken, and the least noisy included, with the most common source of noisy measurement due to alignment issues with the spectrometer and the lighter.

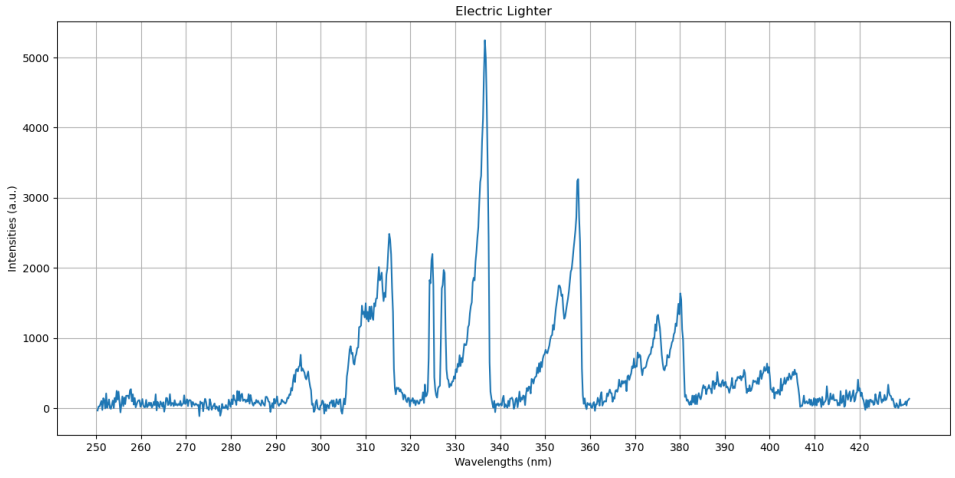


Figure 3.3: Spectrum taken with validation data.

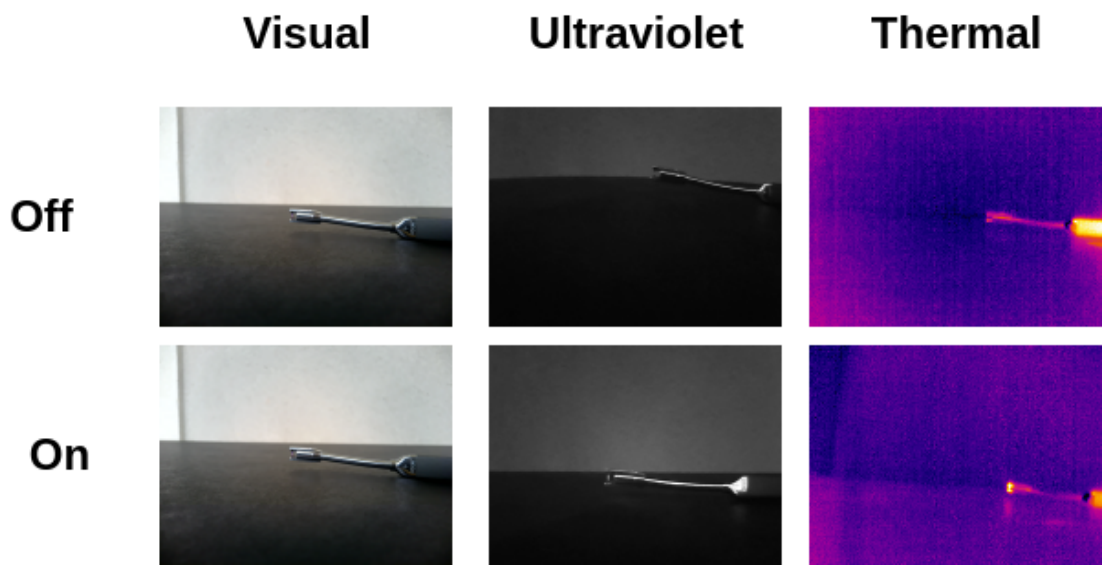


Figure 3.4: Example images from each camera, of each class.

Spectrometer data was taken for each of the validation datasets to help contextualize the results, in addition to descriptions of weather conditions. This is described in more detail in Chapter 4.

3.3 Data Collection

This section will describe the process of collecting data from the cameras.

The original AIMS prototype camera was connected to a Raspberry Pi, which was connected to the drone's power supply to provide power both to the cameras and the limited onboard processing on the UAV. An example of an AIMS scout drone with the prototype triband camera attached is shown in Figure 3.5. Example images from this version of the prototype, taken during an AIMS sponsor demonstration, are shown in Figure 3.6. A diagram of the process of data collection used in this research is shown in Figure 3.7.



Figure 3.5: An AIMS drone with the triband prototype camera system attached.



(a) Visual camera **(b)** Ultraviolet Camera **(c)** Thermal Camera

Figure 3.6: Images taken from each camera at the first AIMS demo.

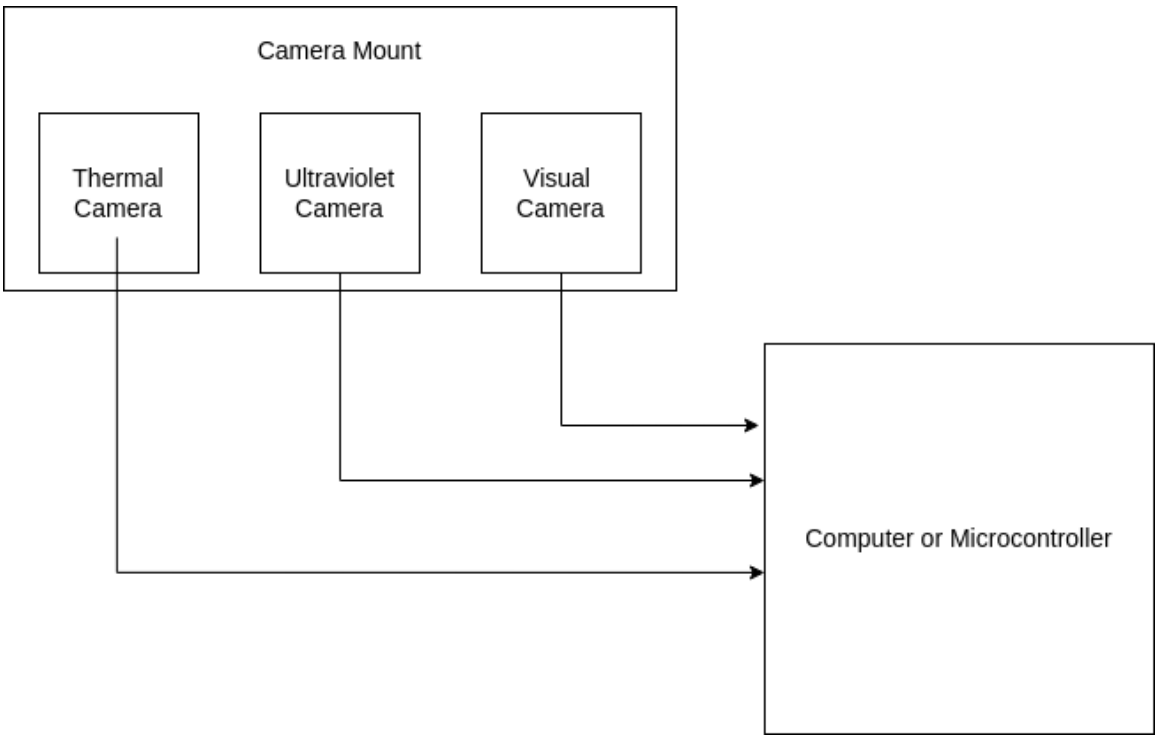


Figure 3.7: A diagram of data collection from the three cameras.

In order to build a dataset for experiments for machine learning, individual images were instead captured to increase control over the data, to improve the accuracy of the training data and make sure the synchronization of timing between the images of the cameras was consistent.

For ease of use for the data collection for this research, a python script using the OpenCV library was used instead to take individual images which were captured and stored on a laptop before cleaning and processing for machine learning. Some examples of the data collection setup with different levels of sunlight is shown in Figure 3.8.

Data collection was done on a laptop running the Ubuntu operating system, and so the cameras were controlled by the video4linux drivers, which could be access with the command line tool `v4l2-ctl`, which controls input and output for cameras, radio, and other applications. The command `v4l2-ctl --list-devices` was needed to access the device IDs to pass to OpenCV, as these values could change based



(a) Data collection sunlight



(b) Data collection partial shade

Figure 3.8: Examples of data collection.

on the order the cameras were connected, or if other processing were being used by the laptop to modify the numbers. The tool `v4l2-ctl` could also be used to check and modify other parameters related to the cameras, as could OpenCV directly, but attempts to manually adjust the gain or exposure time were largely unsuccessful with these cameras, and limited fine control over the camera conditions, as did largely unsuccessful additional attempts to calibrate the cameras in order to manually overlay the images. This challenge motivated the choice to compare the performance of the cameras with minimal preprocessing, since the hardware and firmware of low-cost cameras chosen for a given application may not reliably allow for specific preprocessing to be done. I decided to instead compare how a generalized process focused only on the training data could perform.

The full code used for data capture is included in Appendix A.

3.4 Summary

This section provided a discussion of the hardware used for data collection, and the motivations from the AIMS project that informed the design choices. The challenges from these hardware choices, which could realistically be encountered in other low-cost implementations of non-specialized hardware for a given approach, motivated the

more general machine learning based approach to the image classification described in the Chapter 4.

Chapter 4

Experiments

This chapter will build on the descriptions of the motivation, hardware, and data collection approach described in Chapter 3, describing the datasets in more detail, including providing detail on the conditions of the data collection, and how that affected the data used in the experiments. These conditions include the approach to collecting data under different conditions of ambient light to create variation in both the training data and the four different validation datasets, and how those differences impact the results of image classification.

The processing of the image data is described in detail, including the approach to cleaning and organizing the data to form the training dataset. The approach to machine learning based image classification is described, including the process of training with supervised learning and the specific model architectures that were compared in this research. The models chosen also are well known and well studied models options for the field of image classification, relatively small versions of which were chosen for this relatively straightforward example dataset. The goal of these experiments was to compare the models were able to classify data from the same source captured with the three different cameras, how they performed separately compared to in an ensemble, and how those results might inform other camera or other combined sensor implementations that may use similar machine learning based

approaches. The numerical results to the individual experiment follow this section, to be elaborated on in Chapter 5.

4.1 Dataset

As described in the previous chapter, the dataset used for these experiments was collected with the prototype multispectral camera system of three low-cost cameras. Each of the cameras took images simultaneously, with the same number of total images collected from all three cameras, so that the number of images from the separate cameras could be consistent, and the input and output labels for the ensemble models could be correctly synchronized.

For the image collection the lighter, which was the source of the electrical arcs, was held in front of the cameras at a distance varying between 6-18 inches. For each session of data collection, the images were save in directories for their camera type, labeled "visual", "uv", and "thermal", and further divided into two directories which represent the two classes, 'on' or 'off', with the arc from the lighter on or off in those images respectively. Both the top level label for session of data collection and the individual image files contained the data, with the individual images containing time stamps to keep the data organized, as well as a numerical count within the session to synchronize the images for each camera and class.

All the data was taken in the same general outdoor location, outside of a building, against a wall facing approximately south, so that the area was in sunlight throughout much of the day, especially mid-morning to mid-afternoon. The training dataset was taken in this area in several different positions, different angles, and during different times and day and weather conditions, intending to be generalized enough to be able to predict the varied conditions of the validation data. The four validation sets were also set at different times, but all in the same positions.

4.1.1 Ambient Conditions

One of my priorities when building the training dataset was for it to contain a well balanced variety of images in different ambient light conditions. As discussed in Chapter 2, a balanced dataset is important for accurate classification, but earlier attempts at image classification with this research made it clear that a balanced variety of other conditions can also be important for good results. A neural network learns the variety of parameters from available data in the dataset used to train it, so the training dataset needs to be reflective of the variety of conditions it may encounter.

The validation datasets were in the same general location, with the lighter moved around within the frame of the three cameras. None of the training data was taken in the exact location of the validation data, instead the training data data was collected in front of several different locations to make sure the models were not dependent on a specific background. Additionally the lighter was held in different positions throughout the training data so that the presence of the arc, rather than some other feature, would be identified as the source of the classification result. For example, if every image in the lighter "on" class has the lighter on the right of the image, and every image in the lighter "off" class is on the left of the image, the model may identify those as the parameters of interest in addition to or even instead of the presence of the arc itself, since, with this approach to machine learning, the only information the models has to determine correctness is the class value.

Sunlight could interfere with data from all three cameras, but most predominately the visual and the ultraviolet, in which spectral ranges the sun emits a significant amount electromagnetic radiation that is not blocked by the Earth's atmosphere, and so could potentially overwhelm another light source. The four different validation datasets used for these experiments includes two taken in the middle of the day in sunny and partially sunny conditions. These are intended to account for this potential

interference and compare how well the models are able to classify images with arcs under those conditions.

The most commonly used definition of the visible spectrum is based in human biology, and as such has range based on the response of the retina, with the lower range between 360-400 nm and the upper range of the visible spectrum generally considered to be around 760-830 nm [83]. Spectral ranges visible to digital cameras, such those used here for data collection, also have some variation in response. RGB (red, green, blue) color cameras, such as the visual camera used, have three different ranges of response, one for each channel, with the best response being around the center of each channel, or 'color'. The analysis of responses for digital cameras done by [84] focused on the range of 400-720 nm, their results showing how the blue channel measures the shortest wavelengths and the red the longest, each with a range of response that is strongest around the center of its range and tapers off to either end.

Based on the Ocean View spectrometer measurements from the arc, I expected the majority of the detection of the arc by the RGB camera would be in the blue channels, as most of the light emitted from the arc was under 450 nm, the section of the visual spectrum with the shortest wavelengths. However, a comparison of histograms of the channels of the images with and without the arc present did not show significantly more response in the blue channel than the green or red, possibility because the arc was being recorded as mostly white in the images, and the camera was not passing the raw data. This may represent another limitation of the camera, either in the hardware, firmware, or something else in the processing used for this low-cost prototype. Another type of camera could have made this information available, but would also have placed another limitation on camera selection for such a sensor, and so I decided to proceed with this camera.

I also attempted to calibrate the cameras to each other in order to attempt to manually overlay the images, which was unsuccessful. As with the histograms, this seemed to be a limitation of the low-cost cameras. The visual camera had an auto-focus that was not responsive to the tools I had access to, and the different,

though relatively low resolutions of the other cameras, as well as the difference in light conditions they performed in well, also seemed to hinder the attempt.

After these two attempted approaches to more directly process the data for the cameras, I decided to forgo modifying the data directly and compare how well a minimally processed machine learned based approach could perform. While some amount of additional processing was likely possible, it was becoming clear that one of the challenges of using low-cost sensors could be a reduced predictability or control of the behavior of those sensors. Instead, I decided to instead rely on data taken with those sensors, and see how well a neural network could learn the general behaviors of the camera. If successful, this more automated approach could assist in other applications of lower cost, or otherwise less specialized, equipment.

The use of more specialized equipment, with more predicable results, being included with these as a means of helping validate the predictions may be worth future investigation.

While the visual camera may have some sensitivity at or below 400 nm, its much larger range means a greater potential for another light source to interfere with the detection of the arc, especially bright sunlight, compared to the ultraviolet camera's narrower 365-380 nm range. Considering again the spectrum of the electrical arc taking with the Ocean Optic spectrometer in Figure 4.1, all of the information visible to these cameras is above 365 nm, limited by the glass lenses, which blocks shorter wavelengths. The largest peak in this range is between 370-380 nm. This would most likely not be detectable to the visual camera, which may give the ultraviolet camera an advantage over the visual for detecting this arc source. The additional advantage an ultraviolet camera not limited by a glass lens can be seen here, as it could capture more of the light emitted from the arc, and other approaches that could possibility take advantage of this information while navigating the increased cost of a quartz lens may also be worth future investigation.

Considering the thermal camera, it may have an advantage over the other two cameras in very bright ambient conditions, as it does not measure solar radiation

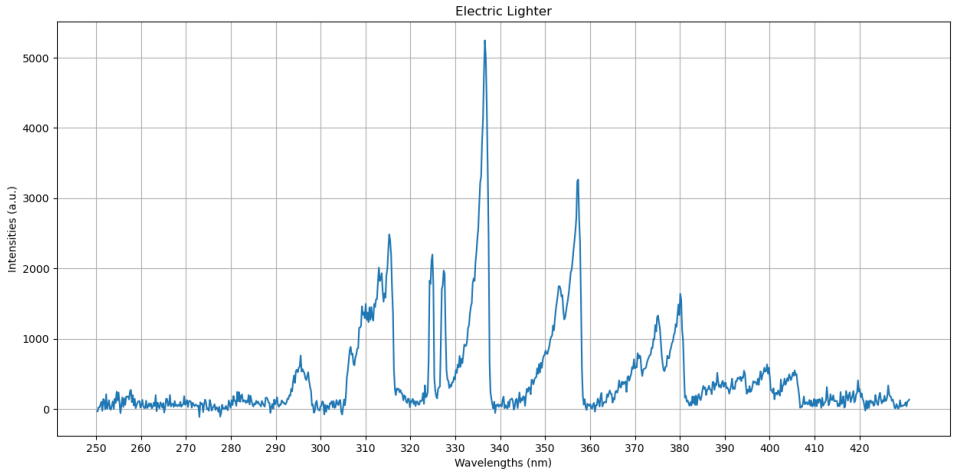


Figure 4.1: Spectrum of electrical arc.

directly, but instead infrared radiation, most commonly present from radiant heat. This source may also be subject to distortion from sunlight in the form of ambient heat, especially on objects that are in direct sunlight in warm conditions. There is the potential that the way an object heats from an arc may be a more consistent pattern that is not as easily hidden by sunlight. However, this may be very dependent on regional conditions, material of components, and other factors. The data examined here was collected in early winter in a temperate climate, and while differences in temperature in surrounding objects can be seen, it is likely less intense than might be measured in other conditions. For a field application, the thermal behavior of the objects measured should be compared in the more extreme conditions that could be expected in that region, and if machine learning is used, those examples should be included in the training data.

In addition to collecting data for training at varying times of day and with varying backgrounds, the validation datasets were collected specifically to compare the accuracy of predictions relative to varying light conditions. Four separate validation datasets were captured, each at a different time, and across several days to capture some various in weather conditions as well, but in a consistent location and consistent

angle, so the variation in validation data results would be limited to those light conditions. The location was in the same general area as the training data, but not at the exact location or angle as any of the training data, to make sure the results were separate. The spectrometer data from these sets is compared the the arc measurement in Figure 4.2.

The ambient conditions for each of the validation datasets are summarized in Table 4.1. At the time of year recorded, sunrise was at approximately 7:38 AM and sunset was at approximately 5:23 PM, with the location against a building to the north and open to the south, providing the most sun mid-afternoon.

The datasets taken, including the training dataset as well as the four validation datasets, were intended to be well balanced between the 'on' and 'off' classes, with variations mostly due to different number of images removed during cleaning. The sizes and class composition of each of the datasets is shown in Table 4.2

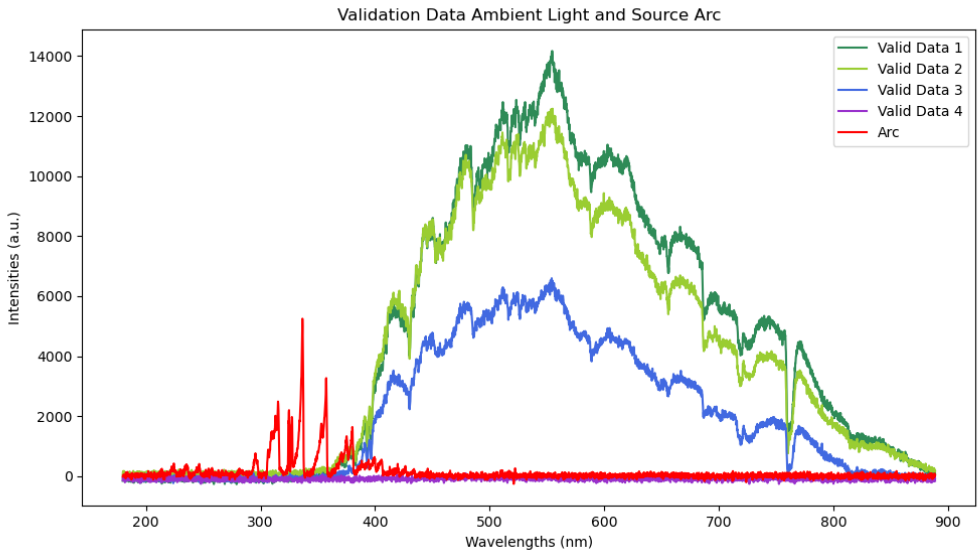


Figure 4.2: Spectrum taken with validation data.

Table 4.1: Validation data weather conditions, with conditions taken from NOAA data for location of data collection, and weather conditions visually verified. Times shown are from timestamps of the spectrometer data, which was taken immediately before collection of each validation dataset.

Validation Dataset	Temperature	Humidity	Weather Condition	Time (EST)
Valid Data 1	41° F	76%	Clear	12:18 PM
Valid Data 2	50° F	58%	Partially Cloud to Sunny	2:41 PM
Valid Data 3	50° F	58%	Partially Cloudy to Sunny	12:06 PM
Valid Data 4	38° F	41%	Partially Cloudy to Sunny	5:09 PM

Table 4.2: The composition of each of the custom datasets collected.

Dataset	On	Off	Total
Train	2,418	2,406	4,824
Valid Data 1	115	124	239
Valid Data 2	127	122	249
Valid Data 3	135	133	268
Valid Data 4	122	115	237

4.1.2 Cleaning and Organization

After being collected, the data had to be manually cleaned to make sure the class labels were correct, and that there were no other issues that could have interfered with training, to the extent that it could be controlled at this step. The primary source of mislabelled images that I found was images were the lighter had either ended up out of frame for one of the cameras, or were the lighter had turned off, which was particularly prone to happening when the lighter was low on battery. If a problem was noticeable in an image from one of the cameras, the corresponding images taken with the other cameras would need to be removed as well, since the because the images and labels in each dataset needed to be kept consistent, and because they problem may be present in the other images as well. This was done with a python script, in which the directory for each collection of images could be set, and the number assigned when the image was taken could be deleted form each of the three datasets at once. This insured the data remained synchronized across the three cameras, with the same number of images from each set, which each directly corresponded to the image from the other two cameras.

Once the datasets were cleaned, the training data needed to be organized into one large dataset. Another python script was used to sort through each subdirectory of images, and add them to a new directory, with a unique numerical identifier to each set of images (visual, ultraviolet, thermal), to allow the ensemble models described later in this chapter to handle the data correctly.

In the new directory, the images were saved in files labeled "on" or "off", so that the file structure looks like

`train/visual/on/<image_file.png>`

or

`train/uv/off/<image_file.png>`.

The validation data was further separated into each of the separate validation sets, such as

valid1/visual/on/<image_file.png>

or

valid2/visual/on/<image_file.png>.

4.2 Machine Learning

The approach used for the detection of the arcs in these images is machine learned based image classification, which was used to sort the images into those that do or do not contain an image of the arc, the 'on' and 'off' classes, respectively. As discussed in the previous section of this chapter as well as Chapter 3, this approach was chosen due to challenges presented by these cameras in using more direct image detection approaches, as well as to assess it as a more generalized approach to combining inputs from low-cost, non-specialized sensors. This approach also allows the same software to be used for each of the three cameras.

The first task of the machine learning was to classify the data from each of the cameras individually in order to establish a baseline of how well the cameras could identify the arc alone. The separate camera predication models were also used to test several different parameters, including model architectures. The four different validation sets were used to help compare the results.

Following these results, ensemble approaches to combining the models were compared.

4.2.1 Training

Three different model architectures were used for image classification, all of which are convolutional neural networks, the theory of which is described in general terms in Chapter 2, which were trained with the training data described in the previous section. Training is the process of fitting a neural network or other machine learning

algorithm to data. For a neural network, this means using data to set the values of its weights, a process which can be either supervised or unsupervised.

The weights in a model are modified through the training process according to the model's output through the training process, according to the metric being used. A example of weights within a model can be seen in the simple case of a multi-layer perceptron, where the output can be modeled as

$$o(\boldsymbol{x}) = f\left(w_0 + \sum_{i=1}^{n} w_i x_i\right) \tag{4.1}$$

where x represents the input values and w represents the model weights [85]. On untrained models, weights are often randomly set when the model is initialized, though they can also be loaded from a previous process of training a model with the same architecture on a related but different problem, through a process known as transfer learning.

Transfer learning is the process of transferring knowledge of models trained on a source domain to a task in a target domain, where the task of the source domain is related to the task of the target domain [86]. In the case of the following experiments, both the source and task domains are intended for the task of image classification. The pretrained weights of the source domain were trained on the ImageNet dataset, as made available by the PyTorch library.

The process for machine learning used in this research was supervised learning, as opposed to unsupervised learning. This approach to training the classification models allows the expected outputs of the model to be known to the training algorithm, and compares the output of the model with the predicted output using an error function. The error function calculates the error, also called the loss, of the model's output compared to the expected output and uses that value to modify the weights for one pass through the training data, called an epoch. The error function used here was the cross entropy loss, where the error, or the loss, is calculated with

$$l(x, y) = Ll_1...l_N{}^T, l_n = -w_{y_n} log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})} \cdot 1y_n \neq \text{ignore index} \tag{4.2}$$

where x is the input, or result form the model, y is the target, or expected value, C is the number of classes, 2 in binary classification, and N is the minibatch dimension, or a subset of the training data processed together [87].

When training, the dataset loads the file paths of each image in the directory, and creates a list of the name of the next file to form the class labels. These labels are sorted alphabetically and mapped to integers, so that this binary classification is defined as `"off" = 0` and `"on" = 1`.

Some preprocessing was done to all images, as needed for convergence of the model and to meet the requirements of the PyTorch imported models. This preprocessing included image normalization and resizing, as well as converting the data to the correct formats. Additionally, images used for training were given a randomly assigned vertical or horizontal flip to add some additional variety to the samples without changing the data itself.

The number of epochs used was determined by an early stopping algorithm, which tracks the loss, and stops training if the model stops improving over epochs and starts to degrade over several epochs. An improving loss value will decrease from epoch to epoch, and degrading one will start to go back up.

4.2.2 Models

The models compared with this data are described in this section. Several other, models were compared in earlier stages of research, included LeNet, however models more simple than AlexNet generally did not have consistently good results with datasets similar to this one. While more options may be possible, especially with additional preprocessing or additional parameter extraction, these models were chosen

as the basis for comparison with this dataset, with this minimally preprocessed approach.

AlexNet was published in 2012, and was developed to classify images in the ImageNet dataset, consisting of 1000 different classes. AlexNet was designed with the intention of creating a network with a large learning capacity, as needed to successfully classify ImageNet. AlexNet is made up of eight layers, five convolutional and max pooling layers and three fully connected layers [88]. The output of the neurons in AlexNet use a nonlinear activation function known as Rectified Linear Units (ReLUs) characterized by the function $f(x) = \texttt{max}(0, x)$ as opposed to alternative functions such as $f(x) = tanh(x)$, which allows faster training times. The architecture includes max pooling layers, which summarize adjacent neurons and can help reduce overfitting, and dropout layers, which set a neurons to zero with a probability of 0.5, and force the trained network to become more robust. A diagram of the AlexNet network is shown in Figure 4.3, image source [89].

VGGNet was published in 2015, and was developed to classify the ImageNet dataset, with the goal of improving accuracy over other convolutional neural networks for image classification. VGGNet uses small convolutional filters in layers, separated by max pooling layers with eleven to eighteen weight layers, and ending with three fully connected layers [90]. A diagram of the VGGNet network is shown in Figure 4.4, image source [91].

ResNet was published in 2015, and represents an approach to addressing the problem of vanishing and exploding gradients that occurs with deeper networks, and can prevent very large convolutional networks from training effectively. ResNet approaches this problem by using residual layers as an attempt to optimize feature mapping within the network. This network was originally assessed on the PASCAL and COCO datasets [92]. A diagram of a residual block, used to make up the layers of the ResNet network, is shown in Figure 4.5, image source [93]. A table showing a confusion matrix is shown in Table 4.3.

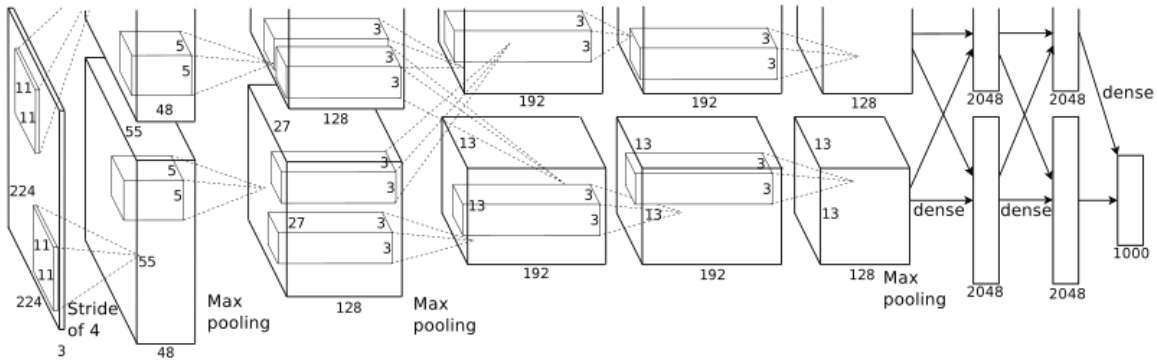


Figure 4.3: Diagram of the AlexNet network.

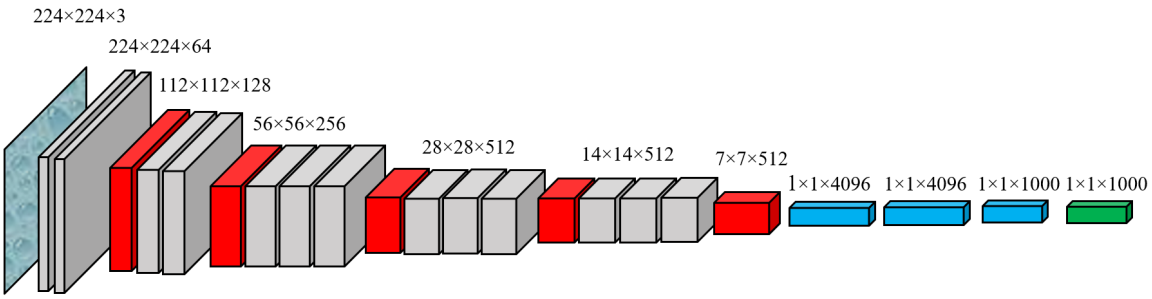


Figure 4.4: Diagram of VGGNet network.

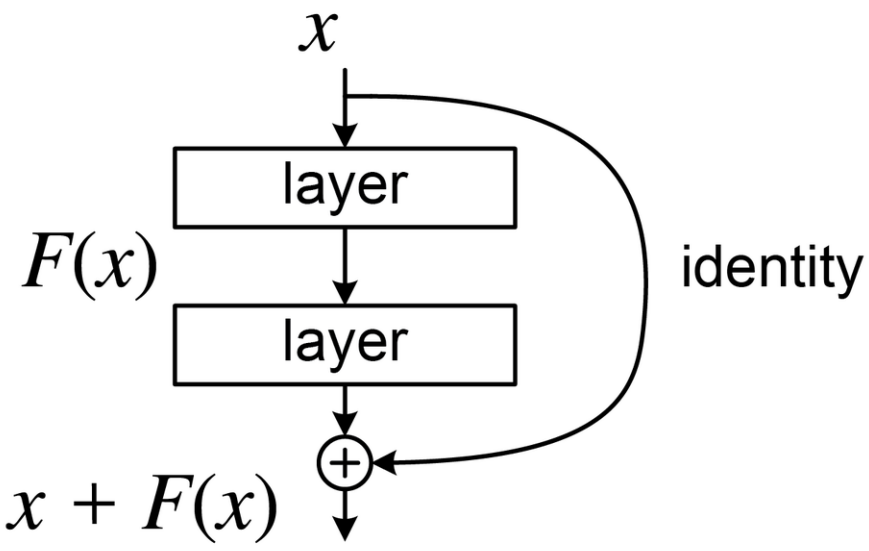


Figure 4.5: Diagram of Resnet residual block diagram.

Table 4.3: A binary confusion matrix.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

To compare the effectiveness of models, as well as other parameters that may affect the success of classification in including ambient conditions of the data, the outcomes will need to be compared. An common approach to displaying the outcome of a model is a confusion matrix, where the outcome of a classification an be described with four categories for binary classification: True Positive (TP) and True Negative (TN) for correct classification, and False Positive (FP) and False Negative (FN) for incorrect classification [94].

Other metrics that describe the behavior or models include the accuracy [95], which describes the ratio of correct predictions to total instances:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.3}$$

Error rate, which gives the misclassification ratio:

$$ErrorRate = \frac{FP + FN}{TP + FP + TN + FN} \tag{4.4}$$

Precision, which describes the ratio of the correct prediction with the positive case:

$$Precision = \frac{TP}{TP + FP} \tag{4.5}$$

Recall, which describes the fraction of positive cases that are correctly classified:

$$Recall = \frac{TP}{TP + FN} \tag{4.6}$$

Specificity, which describes the fraction of negative cases that are correctly classified:

$$Specificity : \frac{TN}{TN + FP} \tag{4.7}$$

Additional metrics build on these definitions to provide more information about the behavior of the models . For example, accuracy can be an effective metric if the main concern is how often an individual class is identified correctly, but can hide classification issues for a class that has far fewer examples compared to other classes [96]. A well known alternative metric is the Balance Accuracy, which averages the values of the recalls for each class, or the Balanced Accuracy Weighted, which multiplies the recall by the weight of each class, or its frequency in the dataset, where row_k represents a row in the confusion matrix:

$$BalancedAccuracyWeighted = \frac{\sum_{k=1}^{K} \frac{TP_k}{Total_{row_k} \cdot w_k}}{K \cdot W} \tag{4.8}$$

The F1 score is the harmonic mean of the precision and the recall. The metric is intended to favor models with similar Precision and Recall scores, and provide more weight to the effects of smaller classes in the dataset.

$$\text{F1 Score} = 2 \cdot \left(\frac{precision \cdot recall}{precision + recall} \right) = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.9}$$

The F1 score is calculated from the Macro Average Precision

$$MacroAveragePrecision = \frac{\sum_{k=1} KPrecision_k}{K} \tag{4.10}$$

and the Macro Average Recall, defined as

$$MacroAverageRecall = \frac{\sum_{k=1} Recall_k}{K} \tag{4.11}$$

with the Macro F1-Score defined as

$$MacroF1 - Score = 2 \cdot \left(\frac{MacroAveragePrecision \cdot MacroAverageRecall}{MacroAveragePrecision^{-1} + MacroAverageRecall^{-1}} \right) \tag{4.12}$$

A discussion of the strengths and limitations of different metrics used is described in [97]. This paper defines 'macro' metrics as those that balance the results to class size, while 'micro' only gives information about right or wrong classification, ignoring the class distribution of a given dataset.

4.3 Individual Cameras

Before examining approaches to combine the camera outputs, I started with a comparison of how well this approach of using machine learning based image classification was able to identify images containing the arcs for each camera separately. To evaluate this, I wanted to compare the performance of the collected datasets from each camera over several different conditions, both to determine the best conditions for the models, but also to see how much those conditions changed the results. The literature review showed a variety of different analysis approaches, from machine learning based methods similar to those used in this dissertation, to a variety of other analytical and data based methods. While this dataset is relatively simple compared to something suited to a more general application, the effectiveness of some models over others, as well as how the data from each camera performs relative to the other cameras, could be useful in designing a more complex system, or a similar one for more general application.

The three model architectures used were those described above, AlexNet, ResNet18, and VGGNet11, which in the following section will be shorted to ResNet and VGGNet. Different starting conditions for the weights were used, with a version of the models started with randomly assigned weights prior to training, while another version used transfer learning, and was loaded with weights for the same model

architecture originally trained on the ImageNet dataset, as loaded with the PyTorch library. Each of the above conditions was trained with data collected with the three cameras, visual, ultraviolet, and thermal. As described above, each of the datasets from each camera, both for training and validation, contains the same number of images, and were all taken concurrently, so that the only differences are the type of camera, and a slight difference in the angle relative to the target, the lighter, due to the small physical offset of each camera. This resulted in a total of eighteen different trained model parameters, or trained weights that could be reloaded with the architectures used.

Those eighteen trained models were then used to classify each of the four validation dataset described above, with only basic preprocessing was done to make sure the images were of a consistent size and format. The output predictions could then be compared to the expected outputs, determined by the labels of the validation set, to determine the accuracy and other metrics regarding the performance of the models.

A high level summary of the results is shown in Table 4.4, with the accuracy of the outputs summarized across all other variables for the values in the rows. Add additional table with Mathews correlation coefficient (MCC) is included in Table 4.5, which ranges from a coefficient of 1 for a perfect prediction, 0 for a random prediction, and -1 for an inverse prediction. The columns represent each of the validation datasets, with the final "All" column showing the accuracy over all the validation data. A more detailed breakdown of these results is included in the following section.

Table 4.4: Summary of accuracy results for one camera input, averaged over other parameters.

Single Camera Input					
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Camera Summary					
Ultraviolet	0.623	0.27	0.79	0.954	0.799
Visual	0.508	0.064	0.566	0.668	0.6
Infrared	0.942	1	0.881	0.928	0.89
Model Summary					
AlexNet	0.636	0.415	0.669	0.85	0.694
ResNet	0.773	0.41	0.836	0.837	0.845
VGGNet	0.664	0.509	0.732	0.864	0.753
Transfer Learning Used Summary					
Yes	0.687	0.461	0.849	0.88	0.857
No	0.695	0.429	0.643	0.82	0.658

Table 4.5: Summary of MCC results for one camera input, averaged over other parameters.

Single Camera Input					
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Camera Summary					
Ultraviolet	0.248	-0.519	0.591	0.911	0.321
Visual	0.015	-0.879	0.132	0.363	0.11
Infrared	0.89	1	0.773	0.855	0.366
Model Summary					
AlexNet	0.27	-0.174	0.338	0.701	0.64
ResNet	0.545	-0.184	0.682	0.687	0.677
VGGNet	0.33	0.012	0.479	0.728	0.682
Transfer Learning Used Summary					
Yes	0.388	-0.148	0.298	0.646	0.644
No	0.376	-0.085	0.698	0.762	0.682

4.3.1 Detailed Results

This section includes the full result metrics for each of the model parameters compared, which is included to make sure no results of interest were hidden by the summarized data. Additional metrics in addition to the accuracy will also be included. A more in depth discussion into these results will be included in Chapter 5.

The accuracy metric compares how well the results match the expected labels, which were a combination of both the 'on' and 'off' labels. So a results were the results for a set expected to be 'on' with results of [1, 0, 1, 1, 1], for an expected label of [1], and values expected to be 'off' with results of [0, 0, 1, 1, 0] for an expected label of, would be summarized as:

Predictions: [1, 0, 1, 1, 1, 0, 0, 1, 1, 0]

Labels: [1, 1, 1, 1, 1, 0, 0, 0, 0, 0]

For an accuracy measurement of: 0.7.

The precision, recall, and F1 score used in the following section were based on binary targets, which return that metric for the positive label, here 1, or 'on'. For the test data here, the additional metrics would be (rounded after the third decimal):

Precision: 0.667

Recall: 0.8

F1: 0.727

The dataset taken was intended to be well balanced between classes, and a comparison of classification metrics can show how effective this was. For a well balanced dataset, accuracy is a reasonably descriptive metric, but if there are imbalances, or other variations, then the information provided by other metrics can be useful. Detailed tables of these metrics across the metrics compared are included in Appendix B.

4.4 Ensemble

Building upon these results for the individual cameras, I compared a few different approaches to combining the outputs of these models into one prediction. As described in the literature review, ensemble models, which combine multiple models of the same or similar inputs, are an approach to improving the accuracy of the prediction over the use of one model. This can take different forms, often utilizing models of different architecture taking inputs of the same data and combining the results.

This research will compare a few different ensemble model approaches, all relatively straightforward and not specific to the types of data. The goal of this approach is to investigate approaches to combining the camera data that might generalize to other camera data, and potentially other sensor data.

4.4.1 Majority Vote

The first, and simplest, ensemble approach compared is a majority vote, a basic diagram of which is shown in Figure 4.6. The majority vote is described by the

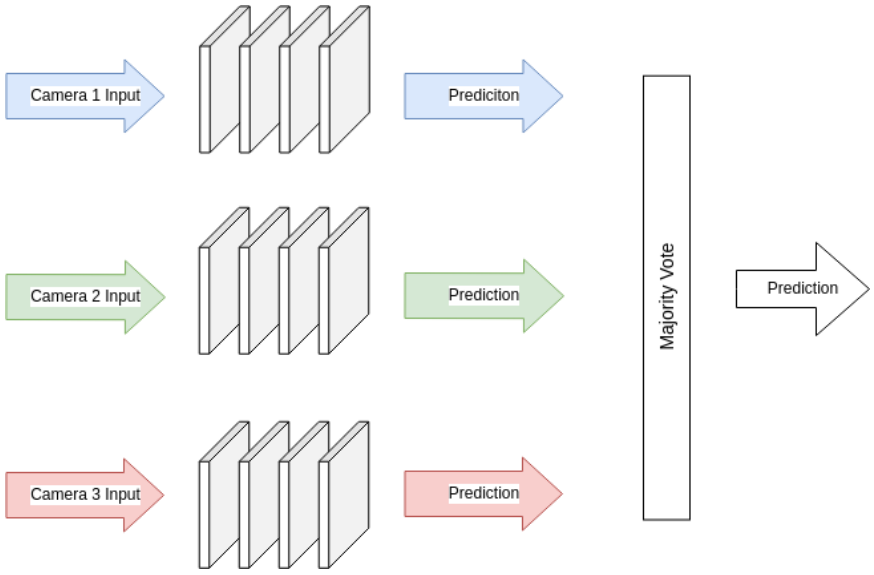


Figure 4.6: A diagram of the majority vote ensemble.

following equation, which determines the output based on the inputs of some number of models

$$V = \text{argmax}_j \sum_{i=1}^{M} c_i^j \tag{4.13}$$

Considering there are two different sets of three variables in this data, the three cameras and the three models, I compared different combinations of this simple majority vote, all of which used models that were already evaluated separately in the previous section. The first two approaches were relatively simple, and only used models with transfer learning, based on those having slightly better results in the separate models. The first of these, labeled Majority Voting 1, was to combine the camera outputs, visual, ultraviolet, and thermal. The prediction from this approach is included for each model. The second, labelled Majority Voting 2, is to combine the outputs of each model for each of the camera predictions, so combining the visual outputs for AlexNet, ResNet, and VGGNet into one visual prediction. This was used to generate an ensemble prediction for each camera. Finally, I combined these two approaches in the category labelled Majority Voting 3, and took the ensemble prediction for each camera, and used that to create an ensemble prediction overall for each of the parameters. Each of these are shown in Table 4.6. The results with the MCC metric are included in Table 4.7.

Additionally, the last few sections of the results table combine all the models, including those without transfer learning. An additional case was added removing the model that was generally the worst performing, AlexNet without transfer learning. These results start with those labelled Majority Voting 4, which is all six models per each camera, with the three models with and without transfer learning, combined with majority vote into one prediction per camera.

Table 4.6: Accuracy results for majority vote ensemble models.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Majority Voting 1: Ensemble Cameras					
AlexNet	0.657	0.285	0.604	0.895	0.606
ResNet	0.883	0.390	0.996	0.992	0.816
VGGNet	0.669	0.434	0.974	0.987	0.768
Majority Voting 2: Ensemble Models					
Ultraviolet	0.603	0.241	0.925	0.966	0.686
Visual	0.343	0.088	0.821	0.751	0.506
Thermal	0.946	1	1	0.97	0.98
Majority Voting 3: Models, then Cameras					
	0.715	0.285	0.993	0.996	0.749
Majority Voting 4: All Models Per Camera					
Ultraviolet	0.552	0.181	0.903	0.962	0.652
Visual	0.326	0	0.407	0.654	0.344
Thermal	0.937	1	0.933	0.966	0.959
Majority Voting 5:, All Models Per Camera, Dropped AlexNet no TL					
Ultraviolet	0.657	0.257	0.925	0.962	0.702
Visual	0.393	0.024	0.657	0.688	0.442
Thermal	0.941	1	0.963	0.97	0.969
Majority Voting 6:, All Models					
	0.891	0.289	0.974	0.992	0.787
Majority Voting 7:, All Models, Dropped AlexNet no TL					
	0.929	0.337	1	0.996	0.816

Table 4.7: MCC results for majority vote ensemble models.

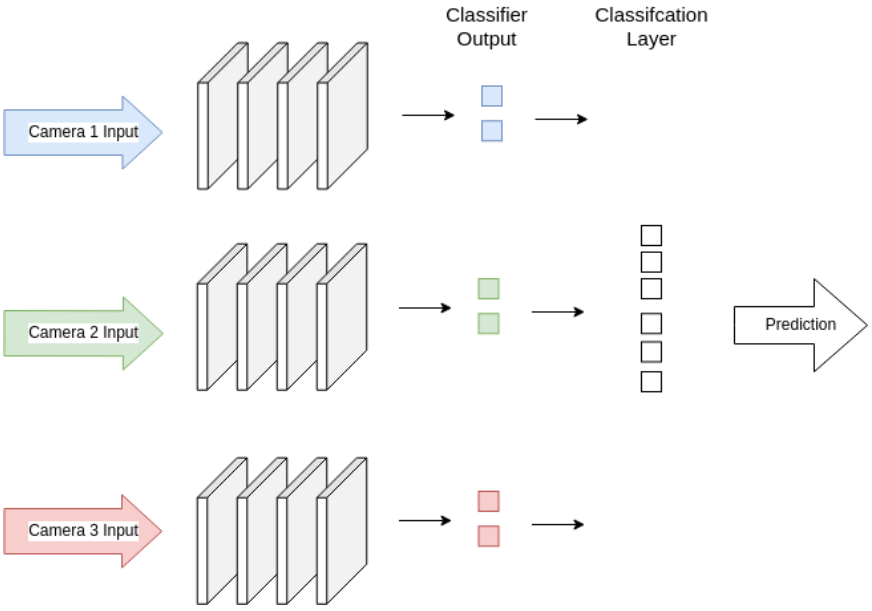
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Majority Voting 1: Ensemble Cameras					
AlexNet	0.327	-0.498	0.337	0.796	0.22
ResNet	0.786	-0.363	0.993	0.983	0.636
VGGNet	0.396	-0.158	0.949	0.975	0.551
Majority Voting 2: Ensemble Models					
Ultraviolet	0.239	-0.578	0.852	0.935	0.377
Visual	-0.313	-0.837	0.687	0.561	0.195
Thermal	0.896	1	1	0.943	0.451
Majority Voting 3: Models, then Cameras					
	0.447	-0.509	0.985	0.992	0.506
Majority Voting 4: All Models Per Camera					
Ultraviolet	0.276	-0.521	0.82	0.927	0.372
Visual	-0.2	-0.954	0.21	0.425	0.119
Thermal	0.881	1	0.878	0.934	0.371
Majority Voting 5:, All Models Per Camera, Dropped AlexNet no TL					
Ultraviolet	0.359	-0.57	0.852	0.927	0.411
Visual	-0.217	-0.953	0.428	0.471	0.148
Thermal	0.888	1	0.926	0.943	0.413
Majority Voting 6:, All Models					
	0.801	-0.387	0.951	0.983	0.6
Majority Voting 7:, All Models, Dropped AlexNet no TL					
	0.861	-0.448	1	0.992	0.637

The following section, labelled Majority Voting 5, which dropped the model Alexnet with transfer learning for a total of five models. Finally, the second to last row is a case using all trained models, labelled Majority Voting 6, for a total of eighteen models. The final row, labelled Majority Voting 7, is also all models, but as before drops the AlexNet without transfer learning, for a total of fifteen models.

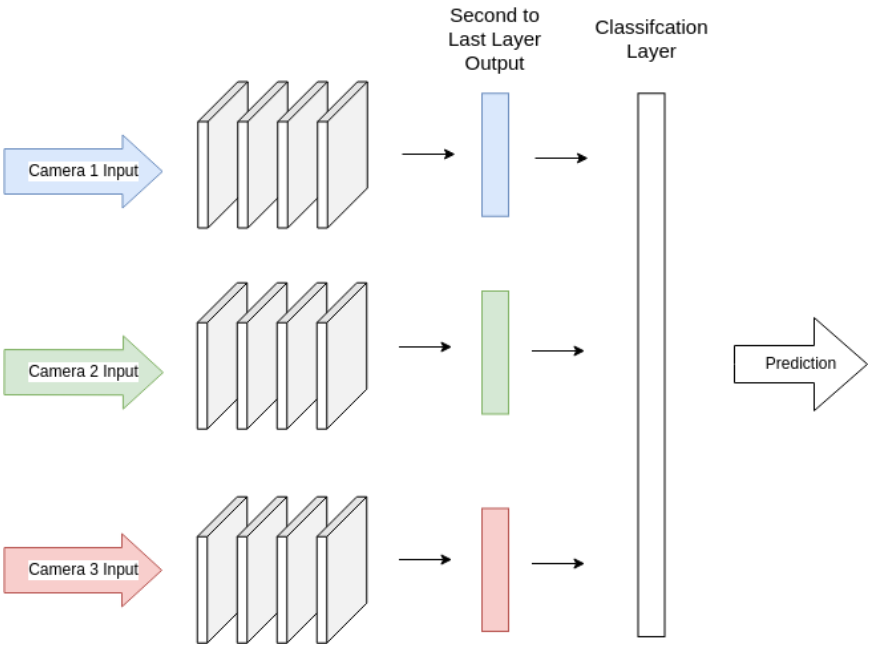
4.4.2 Classification Layer

The other approach I compared for the ensemble models was to combine the original camera models with a classification layer. I tried two different simple architectures to compare how combining the information for the models at separate points would change the result. For these models, because a majority vote was not needed, combinations of each set of cameras are also included, as well as all three. Diagrams of both of these ensemble approaches are included in Figure 4.7.

For the first, labeled Model 1, makes no change to the original model other than taking the raw prediction outputs, or logits, rather than the predicted class from the original classification layer. In a separate model shown previously, an argmax function is used to predict the class from these outputs, but without the function the outputs are a numerical value for each class. These outputs are then combined with a final classification layer to the ensemble classification value. As each model in this case will have two outputs, one for each class, this layer takes six inputs for the three cameras, or four inputs for the cases to two cameras. The output of this layer is then the final prediction of the two classes. This model is shown in Figure 4.7a, while the second architecture is shown in Figure 4.7b. The code for this ensemble model, implemented with ResNet18 built with the PyTorch library, is show below:



(a) Ensemble model 1 diagram



(b) Ensemble model 2 diagram

Figure 4.7: Ensemble network diagrams.

```
class Resnet18ThreeCamera1(nn.Module):

def __init__(self,pretrained=True):
    super(Resnet18ThreeCamera1, self).__init__()

    self.model1 = new_resnet18_model(pretrained=pretrained)
    self.model2 = new_resnet18_model(pretrained=pretrained)
    self.model3 = new_resnet18_model(pretrained=pretrained)
    self.classifier = nn.Linear(6, 2)

def forward(self, x1, x2, x3):
    x1 = self.model1(x1)
    x2 = self.model2(x2)
    x3 = self.model3(x3)
    x = torch.cat((x1, x2, x3), dim=1)
    x = self.classifier(x)
    return x
```

The other model compared was similar, but instead of taking the output from the original classification layers, it took the output from the layer just before the classification layer for both models. These outputs were then used as inputs for a new classification layer with twice as many inputs, and the same two outputs. For example, for the ResNet18 model, the second to last layer of the model has 512 outputs, which are used as the inputs into the final classification layer, which for binary classification would have two outputs. Instead, for three cameras, I combined this into a new classification layer with $3\dot{5}12 = 1536$ inputs, to the 2 output classes. This model architecture is shown in Figure 4.7b. The code for this combined model, implemented with ResNet18 built with the PyTorch library, is shown below.

```
class Resnet18ThreeCamera2(nn.Module):
```

```
def __init__(self,pretrained=True):
    super(Resnet18ThreeCamera2_no_relu, self).__init__()

    self.model1 = new_resnet18_model_weights(pretrained=pretrained)
    self.model2 = new_resnet18_model_weights(pretrained=pretrained)
    self.model3 = new_resnet18_model_weights(pretrained=pretrained)
    self.classifier = nn.Linear(1536, 3)

def forward(self, x1, x2, x3):
    x1 = self.model1(x1)
    x2 = self.model2(x2)
    x3 = self.model3(x3)
    x = torch.cat((x1, x2, x3), dim=1)
    x = self.classifier(x)
    return x
```

All of the tables shown in this section are from the models utilizing transfer learning. The results from the models without transfer learning are included in Appendix B. The accuracy results for Model 1 are shown in Table 4.8. The accuracy results for Model 2 are shown in Table 4.9.

The MCC results for Model 1 are shown in Table 4.10. The MCC results for Model 2 are shown in Table 4.11.

Table 4.8: Accuracy results for ensemble models, Classification Layer Model 1, transfer learning.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
AlexNet					
UV & Vis	0.603	0.129	0.765	0.81	0.749
UV & Therm	0.946	0.45	0.94	0.975	0.967
Therm & Vis	0.958	0.747	0.948	0.924	0.9
All Three	0.937	0.719	0.97	0.962	0.953
ResNet					
UV & Vis	0.51	0.329	0.966	0.966	0.884
UV & Therm	0.962	0.839	0.899	0.992	0.947
Therm & Vis	0.967	0.88	0.933	0.996	0.959
All Three	0.971	1	0.94	0.996	0.976
VGGNet					
UV & Vis	0.757	0.478	0.929	0.958	0.944
UV & Therm	0.908	0.635	0.959	0.992	0.989
Therm & Vis	0.695	0.028	0.851	0.886	0.85
All Three	0.937	0.896	0.966	0.992	0.989

Table 4.9: Accuracy results for ensemble models, Classification Layer Model 2, transfer learning.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
AlexNet					
UV & Vis	0.736	0.064	0.765	0.759	0.682
UV & Therm	0.921	0.273	0.94	0.949	0.958
Therm & Vis	0.946	0.739	0.914	0.895	0.88
All Three	0.95	0.751	0.978	0.975	0.967
ResNet					
UV & Vis	0.682	0.209	0.929	0.966	0.848
UV & Therm	0.711	0.522	0.951	0.992	0.898
Therm & Vis	0.979	0.964	0.91	0.992	0.95
All Three	0.9	0.892	0.978	0.987	0.94
VGGNet					
UV & Vis	0.519	0.502	0.862	0.932	0.911
UV & Therm	0.87	0.803	0.948	0.996	0.994
Therm & Vis	0.958	0.888	0.873	0.979	0.975
All Three	0.946	0.867	0.963	0.975	0.967

Table 4.10: MCC results for ensemble models, Classification Layer Model 1, transfer learning.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
AlexNet					
UV & Vis	0.302	-0.743	0.57	0.674	0.169
UV & Therm	0.892	-0.119	0.887	0.951	0.657
Therm & Vis	0.919	0.494	0.901	0.859	0.79
All Three	0.881	0.534	0.942	0.926	0.806
ResNet					
UV & Vis	0.033	-0.358	0.933	0.935	0.397
UV & Therm	0.926	0.715	0.815	0.983	0.849
Therm & Vis	0.933	0.781	0.869	0.992	0.888
All Three	0.943	1	0.882	0.992	0.952
VGGNet					
UV & Vis	0.517	-0.052	0.859	0.919	0.563
UV & Therm	0.816	0.269	0.921	0.983	0.747
Therm & Vis	0.398	-0.944	0.722	0.794	0.232
All Three	0.881	0.791	0.935	0.983	0.895

Table 4.11: MCC results for ensemble models, Classification Layer Model 2, transfer learning.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
AlexNet					
UV & Vis	0.548	-0.871	0.596	0.597	0.176
UV & Therm	0.841	-0.48	0.882	0.9	0.543
Therm & Vis	0.896	0.478	0.832	0.793	0.748
All Three	0.904	0.583	0.956	0.951	0.837
ResNet					
UV & Vis	0.377	-0.638	0.859	0.935	0.399
UV & Therm	0.527	0.113	0.907	0.983	0.645
Therm & Vis	0.959	0.929	0.834	0.983	0.921
All Three	0.8	0.801	0.956	0.975	0.881
VGGNet					
UV & Vis	0.024	0.032	0.729	0.873	0.442
UV & Therm	0.745	0.623	0.9	0.992	0.81
Therm & Vis	0.916	0.795	0.76	0.958	0.849
All Three	0.896	0.763	0.926	0.951	0.879

4.5 Classification Layer + Majority Vote

This final section will show the combination of these approaches, using the combined models as in a majority vote ensemble model. As before, several different subsets of the data were compared, which are displayed in Table 4.12. The MCC metric results are included in Table 4.13 Based on the final model overall used in the majority voting section, these use all three types of model, AlexNet, ResNet, and VGGNet, with transfer learning for most, but dropping the least accurate AlexNet without transfer learning for each camera type. The result was a very computationally heavy model in order to compare the largest number of possible models for this case. These are broken up into the two model architectures used for the classification layers, with models just using Model 1, Model 2, and using the total of models for both. For each I tested the using only the combination of two models, as shown above (ultraviolet + visual, ultraviolet + thermal, and thermal + visual), as well as combining the classification layer result as well, which are respectively labeled Pairs, and All.

Table 4.12: Accuracy results for majority voting of classification layer models.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Model 1					
Pairs	0.925	0.463	0.932	0.965	0.821
All	0.91	0.419	0.901	0.932	0.79
Model 2					
Pairs	0.915	0.574	0.96	0.976	0.856
All	0.891	0.543	0.943	0.963	0.835
Model 1 + Model 2					
Pairs	0.92	0.519	0.946	0.97	0.838
All	0.901	0.481	0.922	0.948	0.812

Table 4.13: MCC results for majority voting of classification layer models.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Model 1					
Pairs	0.853	-0.082	0.865	0.93	0.641
All	0.84	0.065	0.819	0.875	0.63
Model 2					
Pairs	0.829	0.16	0.92	0.951	0.717
All	0.807	0.316	0.893	0.93	0.715
Model 1 + Model 2					
Pairs	0.84	0.033	0.892	0.892	0.678
All	0.823	0.189	0.855	0.902	0.673

4.6 Summary

This chapter described the use of the datasets collected for machine learning based image classification to distinguish between the arc from the electric lighter being on or off, from the processing of cleaning and organizing the data to the architectures used for the image processing and the results. The comparison of both the machine learning results of the individual models was described, as well as three different architectures of ensemble models. The results for the individual models include a variety of parameters: the results for the different cameras, three different based models, as well as the use of transfer learning, all shown over the four different validation sets. The ensemble models all used transfer learning, but still show the results for different cameras and combinations of cameras, different models, also over the different validation datasets.

Chapter 5

Discussion of Experiments

This chapter will discuss the results of the experiments described in Chapter 4, including what those results imply about the information collected by each of the cameras, and how the various models tested were able to utilize that information for the image classification of the arcing dataset. Due to both the relative simplicity of both the dataset and the analytical approaches, their results are intended to provide a groundwork of strengths and limitations of these specific sensors and analytical approaches. These results can be used to develop the analytical approach to this sensor system, as well as the use of low-cost cameras and machine learning-based sensor ensembles in other potential implementations.

5.1 Separate Cameras

The summary of the individual camera results in Table 4.4, included at the beginning of the results section of Chapter 4 before the full results, shows some overall trends. For the comparison of the model architectures, it shows a somewhat better accuracy for ResNet over the other models, and better accuracy from VGGNet than AlexNet. The summary also shows better accuracy for models using transfer learning than not. However, these results are summarized over all other parameters, and a quick look

over the full results shows that they are not consistent in every case. This chapter will examine the individual camera results in more detail.

To more easily discuss the relative performances of these models across the different validation datasets, I have aggregated the best and worst models for each camera and validation set in the following section. For cases where there were multiple models with the same accuracy value, all the values were included in that column. For the few cases where all the models were the same value, I labelled the column "All".

5.1.1 Visual

As can be seen in the tables, the most accurate results were from the ResNet and VGGNet models in almost all cases. In the summary table, VGGNet had the best accuracy overall, though not by very large margins, with the accuracy results for all validation data for VGGNet being 0.553, compared to ResNet's 0.533. The model with the least accurate results for this dataset was AlexNet in over half of the cases. The use of weights from transfer learning had mixed results with the best models, but all but two of the worst models did not have weights from transfer learning.

For the best performing models, the same models was the best across all metrics for each dataset. For the worst, there was some variation, with Accuracy and Precision having the same worst model for Valid Data 3 and Valid Data 4, but different models for the Recall and F1. However, these were not significant differences. For example the lowest F1 score for the visual data for Valid Data 4 was VGGNet without transfer learning with 0.691, while the next lowest was ResNet without transfer learning with 0.703. Table 5.1 shows the models with the best metrics for the visual dataset results, Table 5.2 show the models with the worst metrics for all the visual dataset results, and Table 5.3 shows the results over all of the visual dataset validation data.

Table 5.1: Best performing models for the visual data.

Best Model - Visual				
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4
Accuracy	ResNet, TL	VGGNet, TL	ResNet, TL	VGGNet, TL
Precision	ResNet, TL	VGGNet, TL	ResNet, TL	VGGNet, TL
Recall	ResNet, no TL	VGGNet, TL	ResNet, TL	VGGNet, TL
F1	ResNet, no TL	VGGNet, TL	RestNet, TL	VGGNet, TL

Table 5.2: Worst performing models for the visual data.

Worst Model - Visual				
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4
Accuracy	AlexNet, TL	AlexNet, no TL	VGGNet, no TL	ResNet, no TL
Precision	AlexNet, TL	AlexNet, no TL	VGGNet, no TL	ResNet, no TL
Recall	VGGNet, no TL	AlexNet, no TL	AlexNet, no TL	AlexNet, no TL
F1	AlexNet, no TL	AlexNet, no TL	AlexNet, no TL	VGGNet, no TL

Table 5.3: Results of the summarized data for all the validation sets for the visual data.

Summary - Visual		
	Best	Worst
Accuracy	VGGNet, TL	VGGNet, no TL
Precision	VGGNet, TL	VGGNet, no TL
Recall	VGGNet, TL	AlexNet, no TL
F1	VGGNet, TL	AlexNet, no TL

Looking instead at the accuracy of the model from the different validation datasets may provide more information. Valid Data 2 consistently had the lowest accuracies for this camera's data, the potential causes of which will be discussed below, though the bright sunlight in that dataset likely contributed. By contrast, images from Valid Data 3 and Valid Data 4, seen in Figure 5.2taken with lower ambient light, have the arc more noticeably visible, which likely contributed to the higher accuracy in the results, as well as better performance in the other metrics.

The overall size and balance of classes of the dataset can provide some additional context for these results. The training data has 2,406 images in the 'off' class, and 2,418 images in the 'on' class, a 0.5% difference, and a fairly well balanced data set. However, how well the data in each class helps to train the specific validation sets is harder to measure. In earlier experiments, I noticed a drop in accuracy in low light cases like Valid Data 4, despite the arc being more visible. I determined that a likely cause of this result was that low light cases were possibly underrepresented in the training dataset. Despite the information needed to identify the difference in the arcing class in those images being more apparent to human inspection compared to other data taken in brighter ambient light, the trained models were not as effective in identifying it. In collecting the datasets used for these experiments, I attempted to collect a training dataset that was more balanced to the types of light conditions tested in the validation data. Ideally, a better balance would reduce failures from the model, and low accuracy for one of the validation datasets would reflect either a lack of information captured, or a failure of the model to identify what data was captured despite its presence in the training data.

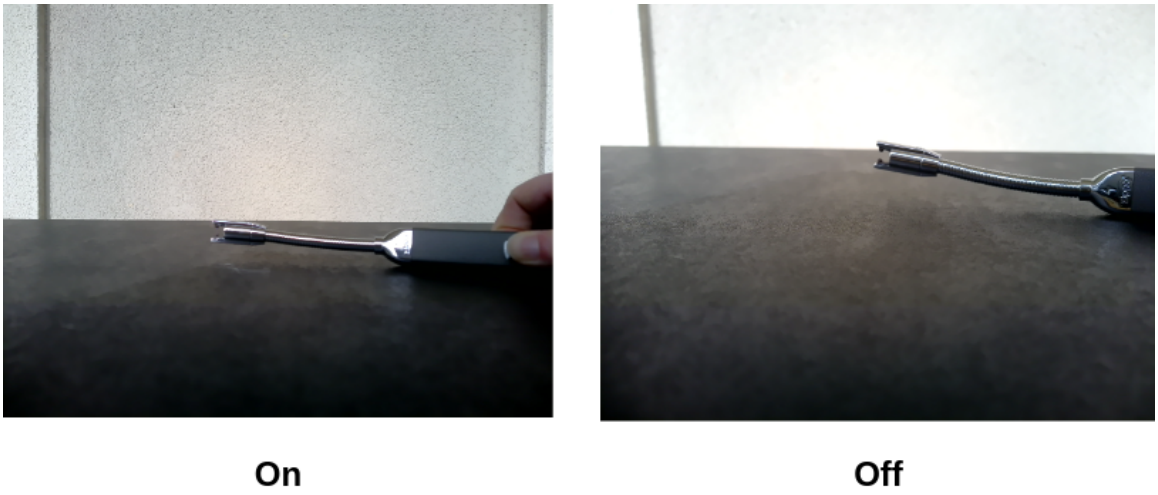


Figure 5.1: Examples of visual camera images from Valid Data 2.

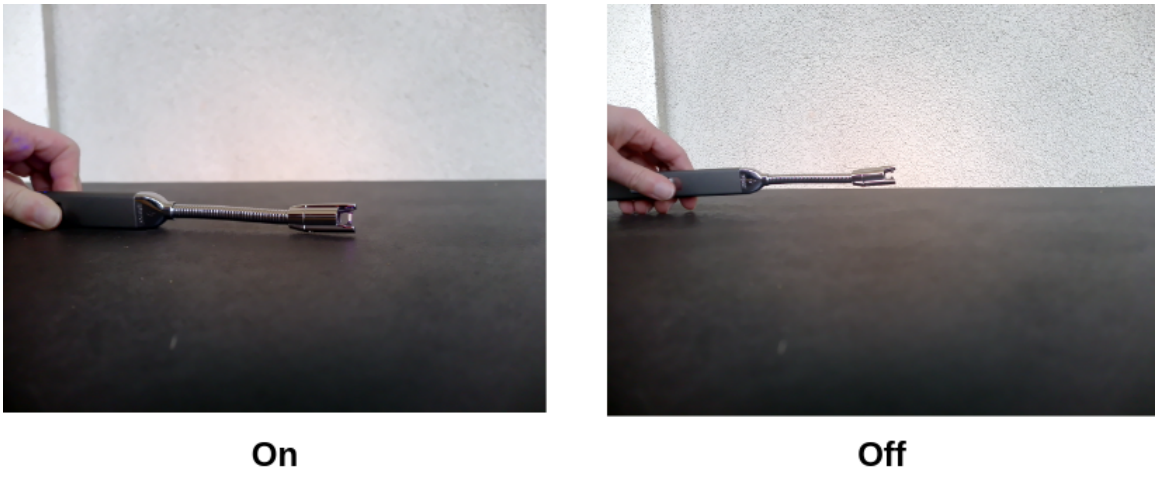


Figure 5.2: Examples of visual camera images from Valid Data 4.

Looking specifically at Valid Data 2, the validation dataset with the most ambient light, I suspect the previous poor performance with low light cases may partially inform why this dataset had such low accuracy in these experiments. After capturing the images and cleaning the data, Valid Data 2 has 124 images labelled 'off' and 115 labelled 'on'. A random guess between the two classes would give an accuracy of around 0.5. If the model always guessed the lighter was on for this dataset, it would have an accuracy of around 0.481, while if it always guessed the lighter was off it would have an accuracy of around 0.519. An accuracy lower than that, which can certainly be seen with this dataset, may imply that the model is misidentifying something as the incorrect class rather than simply guessing, which can be confirmed by looking at the raw results. For example, the raw results for AlexNet with transfer learning for Valid Data 2 were, for for the 'off' class with correct label 0:

[1, 1, 1,

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

The fact that all three models, with and without transfer learning, all had this problem to varying degrees likely indicates the source of the behavior was within the data. The ultraviolet camera also had a similar problem, to a slightly less degree, which may also indicate it was related to the ambient light. However, while the cameras were bought from separate suppliers, it it possible the original manufacturer of the ultraviolet camera was the same as the visual camera, as the case looks very similar another ELP camera tested in other prototypes, which could support the possibility that the issue was caused or enhanced by the camera's auto-focus or other adaptive features.

However, the fact that most of models classified the data as the opposite value for this dataset may imply that the models could identify a difference between the on and off images, which may otherwise be difficult for a human to identify, as shown in Figure 5.1. However, in their current state, the models compared are identifying the class associated with that difference incorrectly.

The accuracy results were higher with the other three validation datasets, with all but two cases having an accuracy greater than 0.5, comparable to or better than random guess. The visual camera datasets did have the lowest accuracy of the datasets from the three cameras, which may provide evidence that the broad visual spectrum alone may not be the most effective approach to detecting an electrical arc, a light source with a strong ultraviolet component. However, other factors such as a higher quality camera having more control over parameters such as gain and exposure, or a more complex machine learning model with well balanced data, could possibly help improve the likelihood of detection.

5.1.2 Ultraviolet

Table 5.4 shows the models with the best metrics for the ultraviolet dataset results, Table 5.5 shows the models with the worst metrics for all the ultraviolet dataset

results, and Table 5.6 shows the results for the summary of all of the ultraviolet validation data.

As with the visual dataset, there are general trends but no consistent best choice when it comes to the models compared, other than that ResNet and VGGNet tended to outperform AlexNet in most cases. AlexNet without the transfer learning weights once again performed poorly for many of the cases, and transfer learning in general often, but not always, improved the results.

Comparing the effectiveness of transfer learning for the visual and ultraviolet datasets, transfer learning performed better in each of the summary datasets. For visual summary data, there was an accuracy improvement for using transfer learning of $0.432 - 0.398 = 0.034$ for AlexNet, $0.533 - 0.437 = 0.096$ for ResNet, and $0.553 - 0.35 = 0.203$ for VGGNet. For ultraviolet, there was an improvement of $0.67 - 0.583 = 0.087$ for AlexNet, $0.706 - 0.691 = 0.018$ for ResNet, and $0.69 - 0.611 = 0.079$ for VGGNet.

The difference in accuracy for the visual datasets with and without transfer learning and the ultraviolet datasets with and without transfer learning was somewhat smaller for ultraviolet for VGGNet specifically, but overall was not significantly different. This outcome suggests that the transfer learning weights were comparably useful for the classification task for both datasets, even though the ImageNet data used to originally train those weights was more similar, at least in terms of format, to the visual dataset than the ultraviolet, as a three channel RGB dataset compared to the monochrome data from this ultraviolet camera.

Looking at the different validation datasets, Valid Data 2 seems to have the same low accuracy as it did in the visual data, though to a lesser extent. VGGNet without transfer learning had the best accuracy with a close to random guess accuracy of 0.494 as opposed to the models with the data that all favored the wrong choice for both classes.

Table 5.4: Best performing models for the ultraviolet data.

Best Model - Ultraviolet				
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4
Accuracy	VGGNet, TL	VGGNet, no TL	VGGNet, TL VGGNet, TL	ResNet, TL
Precision	ResNet, no TL	VGGNet, no TL	ResNet, no TL	ResNet, no TL VGGNet, no TL AlexNet, TL ResNet, TL VGGNet, TL
Recall	ResNet, no TL	VGGNet, no TL	AlexNet, no TL VGGNet, no TL	AlexNet, no TL
F1	ResNet, no TL	VGGNet, no TL	VGGNet, TL	ResNet, TL VGGNet, TL

Table 5.5: Worst performing models for the ultraviolet data.

Worst Model - Ultraviolet				
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4
Accuracy	VGGNet, no TL	ResNet, no TL	AlexNet, no TL	VGGNet, no TL
Precision	VGGNet, no TL	ResNet, no TL	AlexNet, no TL	AlexNet, no TL
Recall	AlexNet, no TL	ResNet, no TL	AlexNet, TL	VGGNet, no TL
F1	AlexNet, no TL	ResNet, no TL	AlexNet, no TL	VGGNet, no TL

Table 5.6: Results of the summarized data for all the validation sets for the ultraviolet data.

Summary - Ultraviolet		
	Best	Worst
Accuracy	ResNet, TL	AlexNet, no TL
Precision	ResNet, no TL	AlexNet, no TL
Recall	VGGNet, no TL	ResNet, no TL
F1	ResNet, TL	AlexNet, no TL

As before, looking at the raw results from this model, where the correct label is 0:

[1, 1, 1,

While this additional information may have helped to reduce the misclassification of this dataset for the 'on' cases, it did not fix the incorrect classification of the 'off' data. That this error happened with both datasets, with and without transfer learning, may also indicate a common parameter being used for both datasets to identify the arc, potentially the ratio of light to dark in the image, or another factor relating to the light and contrast of the images. While randomization was used in the training data, these results may indicate the value of a more deliberate step to normalize the data, or conversely, to introduce varying contrast levels in the the dataset, may improve the classification results.

Looking at the second most successful model for this data, ResNet with transfer learning, for correct label 0:

[1, 1, 1,

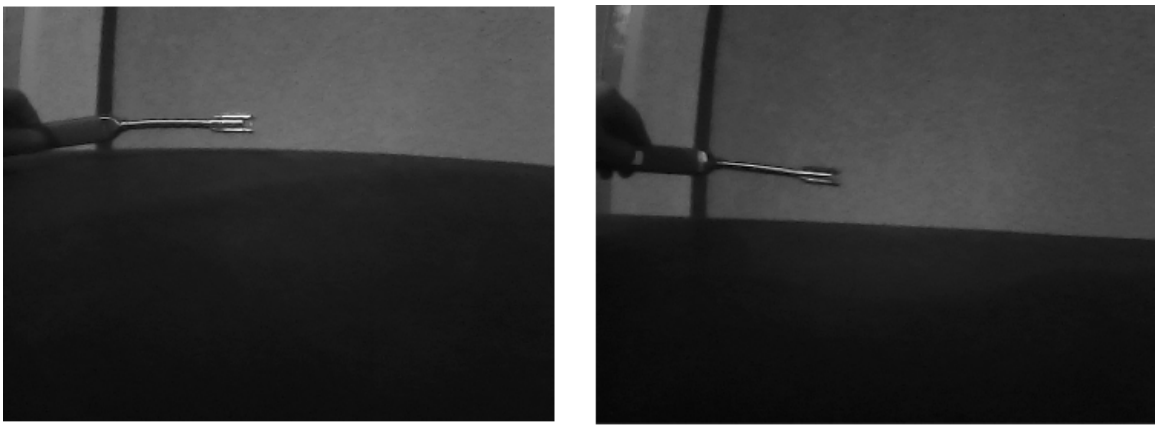


Figure 5.4: Examples ultraviolet camera images from Valid Data 1.

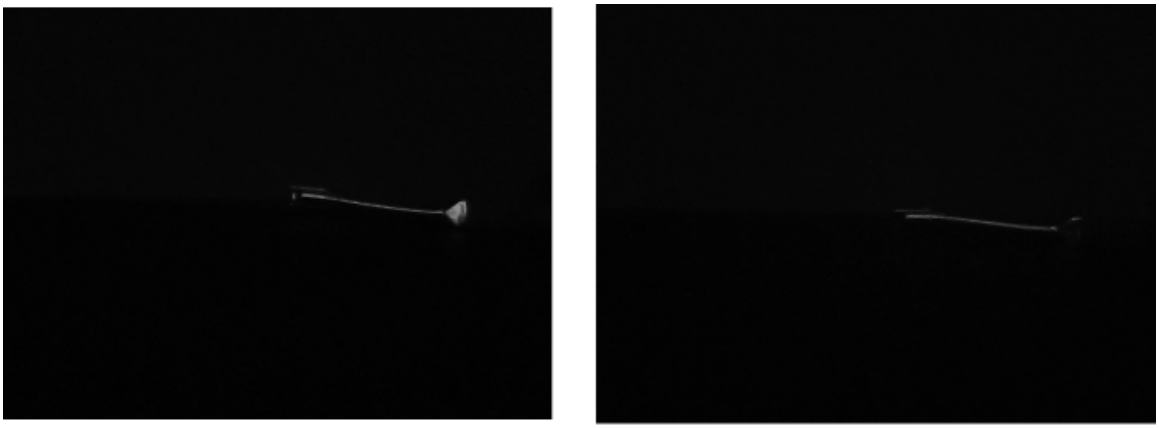


Figure 5.5: Examples ultraviolet camera images from Valid Data 3.

While examination of the training data as compared to the validation data may help illuminate why the model is behaving this way, it would be difficult to identify precisely. This lack of direct explainability of a machine learning based output is a known limitation of machine learning as an analytical approach, and something the ensemble models can hopefully help mitigate.

5.1.3 Thermal

Table 5.7 shows the best models for each metric for the thermal dataset results, Table 5.8 shows the least successful models for the thermal dataset, and Table 5.9 shows the results over a summary of all of the validation data for the thermal dataset.

The results from the thermal dataset are distinct from both the visual or ultraviolet data. This makes sense in that, while the information from the visual and ultraviolet cameras both primarily measured light from reflected sunlight, in addition to the light source from the arc, the thermal camera measured infrared data, light absorbed as heat and re-emitted as thermal radiation. It would be expected that the accuracy of the models from this camera may not follow the same patterns as the visual and ultraviolet cameras, which is confirmed by the results. Only three models had an accuracy of less than 0.9 for any of the validation datasets, and none had an average accuracy of less than 0.8.

While the best models followed the same general pattern as the other cameras, with ResNet with transfer learning having the highest accuracy and AlexNet without transfer learning the lowest, the data overall doesn't follow the same patterns. Notably, Valid Data 2, which had the lowest accuracies for the other two camera datasets, was the best performing in this case, with perfect accuracy across every model. This would indicate that the classification for this data was the easiest, or the best represented by the training data, of the validation datasets from the thermal camera.

Table 5.7: Best performing models for the thermal data.

Best Model - Thermal				
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4
Accuracy	VGGNet, no TL	All	VGGNet, TL ResNet, TL	AlexNet, TL
Precision	All	All	ResNet, TL VGGNet, TL	AlexNet, TL ResNet, TL
Recall	VGGNet, no TL	All	AlexNet, no TL AlexNet, TL VGGNet, TL	AlexNet, no TL
F1	VGGNet, no TL	All	VGGNet, TL	AlexNet, TL ResNet, TL

Table 5.8: Worst performing models for the thermal data.

Worst Model - Thermal				
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4
Accuracy	ResNet, no TL	All	AlexNet, no TL	AlexNet, no TL
Precision	All	All	AlexNet, no TL	AlexNet, no TL
Recall	AlexNet, TL	All	ResNet, no TL	ResNet, no TL
F1	ResNet, no TL	All	AlexNet, no TL	VGGNet, no TL

Table 5.9: Results of the summarized data for all the validation sets for the thermal data.

Summary - Thermal		
	Best	Worst
Accuracy	ResNet, TL	AlexNet, no TL
Precision	VGGNet, TL	AlexNet, no TL
Recall	AlexNet, no TL	ResNet, no TL
F1	VGGNet, TL	AlexNet, no TL

Examples of the validation data from the thermal camera from Valid Data 2 are shown in Figure 5.6, with the bright spot of where the arc had heated the metal visible for the 'on' data. This may indicate that the difference in Valid Data 2 was not due to some other factor in the images, such as background details that may have skewed the result, or at least nothing that was measurable by the thermal camera.

In addition to the difference in the way sunlight interacts with the data, some other differences in this data may also contribute to the difference in accuracy results. There is post processing done on the images from this camera by the PureThermal webcam, which was not done by the visual and ultraviolet cameras. In addition to translating the data to be readable by USB, this camera also adjusts the colors of the image based on the relative intensity of the brightest and darkest parts of the images. This can be seen in the examples in Figure 5.6, where the shadow of where the sunlight has not been heating the table is clearly defined in the 'off' image, in which the warm table is the warmest thing in the images. This effect is much less distinct in the 'on' image, where the hot metal of the lighter was the hottest object in the image. It is possible this relative brightness also contributed to the improved classification accuracy for this camera's data.

While all the data was collected with the intention of making sure it included challenging sets for the visual and ultraviolet cameras, including both dark conditions and direct sunlight as well as intermediate cases, challenging conditions specific to the thermal camera were not the focus, examples of which could be an image set with another very warm source, or data taken on a very hot day with metal objects in frame. The lack of such examples could have resulted in all the datasets for this camera being easier than the others as well.

However, even with these caveats, the results do show that the thermal camera provides data that can be complementary to the other cameras for the purpose of identifying an anomaly that may have some sort of thermal component, such as an electrical arc that may heat up the surrounding material, as was the case of the lighter.

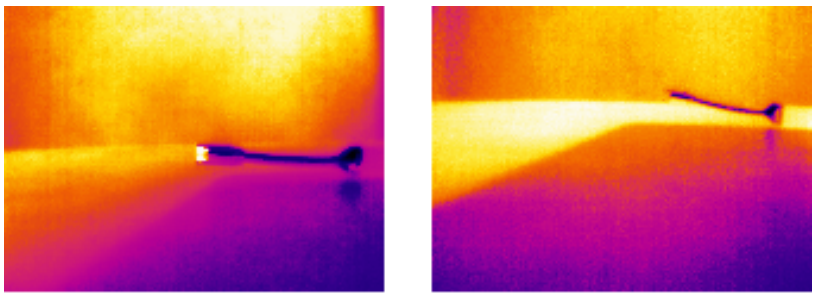


Figure 5.6: Examples ultraviolet camera images from Valid Data 1.

Another possibility that may become relevant in an application is that the heat from a thermal source may fade more slowly than the light from an arc. For this data collection I mitigated this effect by collecting the 'off' data first, and if collecting more than one dataset I allowed time for the lighter to cool between them. In other applications, this could be an advantage, since a detectable hot spot may last longer than an arc and be easier to detect, but it may also interfere with an ensemble detector, as the thermal camera could correctly identify an anomaly, while the other cameras also correctly identify that no anomaly was found. In such instances, there may be value in retaining both ensemble and separate classification results.

5.2 Ensemble

The goal of ensemble models is to combine the outputs of the data to address the limitations demonstrated by the separate camera models in the previous section, of both the cameras themselves and the models used. This section will compare the results of the different ensemble models described in Chapter 4, which combined the individual camera data inputs either with a majority vote of the model results, or with an additional classification layer. These models build upon the same data and the same model architectures as the separate camera models. The goal of the comparison

is to see how the results changed as a result of this aggregation, what improvements were made, and what limitations could or could not be improved upon.

5.2.1 Majority Vote

The majority vote ensemble models do not change the outputs of the original models, only take those outputs and combine the results for a new final classification result. One noticeable limitation of this approach with these datasets was the similarity between the visual and ultraviolet data, combined with these experiments only having three cameras in total. While the ultraviolet camera did perform better in several cases, both the visual and ultraviolet data tended to have similar strengths and weaknesses when it came to the performance of the models relative to ambient light. Whether this was due to an overlap in data between the visual and the narrow band of ultraviolet light available to the low-cost ultraviolet camera, a similarity between the camera hardware, or some other factor, would require additional investigation. Since there were only three cameras total, the majority vote results often followed the behavior of these two cameras, limiting the effectiveness of the combined result if the data from two of those cameras resulted in inaccurate predictions in the same conditions. For these datasets, especially Valid Data 1 and 2, the thermal camera often gave the best prediction accuracy, but since the one camera would be outvoted in a simple ensemble, the benefit of that accuracy was limited.

This section compares the accuracies of each of the majority voting approaches, the first of which combined each of the camera outputs for each type of model, originally trained with weights from transfer learning. The labels of the different majority voting models, which describe which models were combined to determine the prediction result, are the same as those in Table 4.6 in Chapter 4.

Majority Voting 1 combined the visual, ultraviolet, and thermal results for each of the transfer learning models, shown for each of the model architectures, AlexNet, ResNet, and VGGNet. For Valid Data 1, the highest accuracy was the ResNet with

0.883, and 0.669 for VGGNet. Comparing these to the original transfer learning values for Valid Data 1, the visual result for ResNet had an accuracy of 0.707, while ultraviolet had 0.556, and thermal had 0.95, so the ensemble result was lower than the thermal accuracy, but higher than either of the other two. For Valid Data 2, the highest accuracy was the VGGNet with 0.434. For the separate models, both visual and ultraviolet data had low accuracies of 0.233 and 0.305 respectively for VGGNet, though those were some of the best for that dataset overall. For comparison, the thermal data did very well for Valid Data 2, with accuracies of 1 for each model. The ensemble model did perform better than two of the three constituent models, but could not overcome the overall low accuracy of the lower two. For Valid Data 3 and Valid Data 4, the ensemble results were more effective when compared to the separate results, with all accuracies for both ResNet and VGGNet above 0.97, and both ResNet models above 0.99, which was above the accuracy even for the separate thermal data.

Majority Voting 2 was combined results in the opposite way as Majority Voting 1, combining each of the three models, AlexNet, ResNet, and VGGNet, for each of the three cameras, with the results shown for the ensemble results for each camera. For the visual data, this had mixed results. For Valid Data 1, the ensemble accuracy of 0.343 was lower than two of the three of the individual models. This was also true for Valid Data 2, where the ensemble accuracy was only better than one of the individual models. However, the ensemble accuracy for Valid Data 3 was better than any of the separate values. Valid Data 4 was also the second lowest, but the first and second were close, with accuracies of 0.768 and 0.759, compared to the ensemble of 0.751. For the ultraviolet data, the ensemble accuracy was between the middle and highest value for Valid Data 1, for Valid Data 2 it was between the middle and lowest, for Valid Data 3 it was better than any of the separate values, and for Valid Data 4 it was the same as the highest two. The thermal data performed well for all datasets, with the ensemble accuracy between the middle and highest for Valid Data 1, and the same or above the highest value for the rest of the datasets.

Majority Voting 3 combined the previous two majority voting results, using the ensemble model prediction to for a final prediction for the data. As before, this worked well when the average accuracy of the constituent models was high, and not as well when it was lower. The results were slightly worse than the Majority Voting 1 ensemble accuracy for ResNet for every dataset, and worse than the Valid Data 1 accuracy for VGGNet. The Majority Voting 2 accuracies for the Thermal models was the best overall, but given the better performance of the thermal data this was not surprising.

For the rest of the majority voting models compared add more model outputs, even if they were not always the best performing, to compare the effect on the results. Majority Voting 4 was the same as Majority Voting 1, but it also included the models without transfer learning. The accuracies of this model were slightly to significantly worse in every case.

Majority Voting 5 followed the same approach as Majority Learning 4, but dropped the lowest performing model, AlexNet without transfer learning. This approach worked better for some datasets than others, but was a definite improvement over the case which included the dropped AlexNet model. Compared to Majority Voting 1, which only the transfer learning models, several of the ultraviolet datasets performed the same or better, while the visual datasets gave less accurate results in all but one case.

Finally, the last two majority voting models compared, Majority Voting 6 and 7, combined all the model results together at once without any preference. The first combined all eighteen models. The second combined the same without the three lowest performing models, AlexNet without transfer learning for each camera, for a total of fifteen models. The last of these had fairly good results, with the only matching or improved accuracies being the thermal data only models, or the ResNet model from Majority Voting 1.

Overall, the majority voting models were usually a way to avoid the lowest accuracy results, but it was not a guaranteed way to find the highest accuracy result.

There were limitations of how helpful this approach could be, particularly where the weaknesses of several of the models being combined were similar, rather than different enough to be counteracted by the other cameras' data. Additionally, as discussed for the separate models, the thermal data gave much more accurate results overall than the other two cameras under the conditions compared, which resulted in the combined thermal data also demonstrating more accurate combined predictions than the data from other cameras, regardless of other parameters. For cases where similar models both gave relatively low accuracy results, such as the visual and ultraviolet datasets for Valid Data 2, the lower accuracy of the visual and ultraviolet camera data overcame much of any advantage of the ensemble, as without additional processing those inputs together made of two thirds of the total vote.

5.2.2 Classification Layer

Unlike the majority voting ensemble classification models, the classification layer models did not reuse the previously trained models. Instead, while they did use the same base models as the separate cameras, and compared the same parameters, they were fully trained with the new final classification layer.

Earlier in the process of this research, with an earlier version of the training dataset, I also tested an additional classification layer architecture, adding a few additional layers to compare if processing the data together before the classification layer would improve the result. The initial attempts had worse accuracies than either of the architectures included here, so I decided to focus on the simpler approaches for this research for a more direct comparison to the separate cameras and the majority vote ensemble.

As in the separate models, I summarized the best performing models into tables, with Table 5.10 showing the results for all the validation data, and Table 5.11 this for the summary of each dataset.

In several ways this summary shows that the classification layer approach to an ensemble model continues trends from both the separate cameras and the majority voting ensembles, in that while there may be some trends in the accuracy of the models compared, but few clear and consistent best choices. I also included the results for pairs of models for this ensemble approach. This would have also been possible for a majority vote, but this type of model will always give a result even with two inputs, while without only two choices, a majority vote often would not have had a consensus, which was also an additional source of inaccuracy for the larger majority vote models that had even numbers.

As an example, one of the classification layer models combined two AlexNet models, one with ultraviolet data, the other with thermal data. These models were combined with a new classification layer in one of two ways, as described in Chapter 4 as Model 1 and Model 2, as well as with and without transfer learning, to output one prediction for the combined output.

For many combinations of models and camera inputs compared with this type of ensemble, the use of transfer learning had a significant impact on the accuracy of the predictions. For AlexNet, Valid Data 1, Model 1 without transfer learning has an accuracy of 0.519, approximately a random guess between the two categories, while Model 1 with transfer learning had a very good accuracy of 0.946. Similarly, for Valid Data 2 for the same pair of inputs but instead with ResNet base models, Model 1 without transfer learning had an accuracy of 0.189, while Model 1 with transfer learning had an accuracy of 0.839. However, there were also categories, as can be seen in Table 5.10, where Model 1 without transfer learning, or Model 2 without transfer learning, had the best accuracy. A possible reason for the variability of the results with and without transfer learning is that the alternative I used was to randomly assign the values of weights at the start of training. Table 5.11 summarizes these results.

Table 5.10: Best models for classification layer models.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4
		AlexNet		
UV & Vis	Model 2, no TL	Model 1, TL	Model 1, TL Model 2, TL	Model 1, TL
UV & Therm	Model 2, no TL	Model 2, no TL	Model 1, no TL Model 2, TL	Model 1, no TL
Therm & Vis	Model 1, no TL	Model 1, no TL	Model 1, TL	Model 1, TL
All Three	Model 2, TL	Model 2, TL	Model 2, TL	Model 2, TL
		ResNet		
UV & Vis	Model 1, no TL	Model 1, TL	Model 2, no TL	Model 2, no TL
UV & Therm	Model 1, TL	Model 1, TL	Model 2, TL	Model 1, TL Model 2, TL
Therm & Vis	Model 2, TL	Model 1, no TL	Model 1, TL	Model 1, no TL Model 1, TL Model 2, no TL
All Three	Model 1, TL	Model 1, TL	Model 2, TL	Model 1, TL
		VGGNet		
UV & Vis	Model 2, no TL	Model 2, TL	Model 1, TL	Model 1, TL
UV & Therm	Model 1, no TL	Model 1, no TL	Model 2, no TL	Model 1, TL Model 2, TL
Therm & Vis	Model 2, no TL	Model 2, TL	Model 2, no TL	Model 2, TL
All Three	Model 1, no TL	Model 2, no TL	Model 1, TL	Model 1, TL

Table 5.11: Best models for classification layer models, summary of all validation datasets.

Summary	
	All
AlexNet	
UV & Vis	Model 1, TL
UV & Therm	Model 1, TL
Therm & Vis	Model 1, no TL
All Three	Model 2, TL
ResNet	
UV & Vis	Model 1, TL
UV & Therm	Model 1, TL
Therm & Vis	Model 1, no TL
All Three	Model 1, TL
VGGNet	
UV & Vis	Model 1, TL
UV & Therm	Model 2, TL
Therm & Vis	Model 2, TL
All Three	Model 1, TL

Some random configurations may start in a better configuration than others by chance, and lead to a faster convergence, or more effective final configuration, than others. As an alternative, weights could also be manually set to specific values, whether a median value or some other consistent values, which may improve the consistency of training results, but may not necessarily improve them. An advantage of transfer learning is that the weights are consistently set to values that are, at least to some degree, pre-configured to a task similar to the current one. Based on the summarized results, in which only two values of twelve had better accuracies without transfer learning, this seems to give at least somewhat higher accuracies in a majority of examples.

There were also more often better accuracies from models trained with Model 1 than Model 2, though not all of these results had significant differences, which can be seen comparing two of the best performing models. All three cameras' inputs combined with ResNet and with VGGNet, both with transfer learning. For the ResNet model, the accuracies across the four validation datasets were 0.971, 1, 0.94 and 0.996 for Model 1 and 0.9, 0.892, 0.978, and 0.987 for Model 2. For the VGGNet model for the same validation datasets, the accuracies were 0.937, 0.896, 0.966, and 0.992 for Model 1, and 0.946, 0.867, 0.963, and 0.975 for Model 2. Overall there would seem to be a slight preference for the slightly less complex design of Model 1, but additional tests of these kinds of architectures with more complex datasets would be worth testing.

Finally, this type of ensemble overall performed well, with the best overall results for Valid Data 2 outside of the thermal only examples. However, there are still limitations to this approach, which can be seen looking again at the results for Valid Data 2 specifically. For AlexNet with transfer learning, the accuracies for visual and ultraviolet were 0.092 and 0.112 respectively. For the ensemble with the Model 1, it was 0.129, and for Model 2 it was 0.064. For ResNet, visual and ultraviolet individual results were 0.024 and 0.378 respectively, for the ensemble Model 1 it was 0.329, and for Model 2 it was 0.209. For VGGNet, for visual and ultraviolet it was

0.02 and 0.494. For Model 1 it was 0.478, and for Model 2 it was 0.502. So, there are limitations to how much an ensemble like this can improve the results, and it did not always provide improvement. It was however at least usually closer to the more accurate result than the less accurate result, which may be useful, especially in cases where it may be unknown which model would be the more accurate for a given case.

The combined accuracy for Valid Data 2 for models that included the thermal data were somewhat more variable, and sometimes performed quite well. For examples, the accuracy for the combined ultraviolet and thermal models, for AlexNet, ResNet, and VGGNet respectively, were 0.49, 0.189, and 1 for Model 1 without transfer learning, 0.45, 0.839, and 0.635 for Model 1 with transfer learning, 0.996, 0.329, and 0.992 for Model 2 without transfer learning, and 0.273, 0.552, and 0.803 for Model 2 with transfer learning. For all three camera inputs, the accuracies for Valid Data 2 were 0.49, 0.514, and 0 for Model 1 without transfer learning, 0.719, 1, and 0.896 for Model 1 with transfer learning, 0.49, 0.51, and 0.916 for Model 2 without transfer learning, and 0.751, 0.892, and 0.867 for Model 2 with transfer learning. The results for the dataset that had lower accuracies for all the visual and ultraviolet data make a stronger case for transfer learning.

Compared to the majority vote ensemble, the improved accuracy from the thermal camera was more able to improve the overall accuracy of the prediction, and so this approach to combining models may be preferred in cases were one result is required, rather than additional comparison, even if the accuracy of the individual predictions from some of the models may not be as accurate in certain conditions.

Looking at the overall difference in accuracy between Model 1 and Model 2, I will compare the best summary cases, for all of the validation data, and for the models that combined all three cameras. Looking at the best performing models overall, ResNet and VGGNet with transfer learning, for Model 1, the summary accuracy for ResNet was 0.976 and for VGGNet was 0.989, and for Model 2, the accuracy for ResNet was 0.94, and for VGGNet, 0.967. Model 1 did give slightly more accurate

results, but the difference was small, and both were better than any of the majority voting results except for those that only used the thermal camera dataset.

5.2.3 Classification Layer + Majority Vote

Combining the other two ensemble approaches would be more computationally intensive than using either in isolation. The accuracy results for this category, while an improvement over the single models and the original majority vote, tended towards lower accuracies than the classification layer results. However, that the results were more representative of the individual camera results may be useful in some contexts, as they also may be less prone to being influenced by one camera result. As before, additional tests with poor accuracy performances from each data type may help assess additional cases where some models may have additional advantages or shortcomings.

Additionally, this approach to an ensemble model resulted in better accuracies for the combined classification layer models that only used pairs, rather than all four. This was likely at least partially due to cases where there was no majority in the vote, and the difference was not significant for most examples. The models which combined only the Model 2 outputs were also slightly more accurate overall than those that only combined Model 2, or both Model 1 and Model 2. The best accuracy, for the summary of all the validation data, was 0.856 for the three pairs of results, and 0.835 for all model outputs. Based on these results, for this dataset at least, this final approach was likely not worth the significant computation and memory usage it required.

5.3 Summary

This section reviewed the results from the experiments, comparing both the performances of the cameras separately, and the different ensemble approaches. For this dataset, the models using only the thermal camera data had some of the highest

accuracies overall, though the weaknesses of some of the visual and ultraviolet datasets provided some insights into how the ensemble models addressed varying accuracies of their constituent models. The ensemble models combined with an extra classification layer often had the most accurate outputs overall, while the majority vote models had smaller variation from the overall result of the individual models.

Chapter 6

Conclusions

The goal of this dissertation research is to evaluate an approach to building a camera-based multispectral sensor system that may be constrained both by the requirements of a small UAV platform, and by a need for lower-cost, widely compatible components. To account for possible hardware, firmware, and software limitations these constraints may apply, a machine learning based image classification was used as the approach to arc detection. The experiments compared the effectiveness of this approach for a custom dataset of electrical arcing, comparing the classification accuracies for each of the individual cameras, as well as the ensemble system through different parameter options, model architectures, and ensemble architectures.

6.1 Research Questions

This section will return to the question posed in Chapter 1 to evaluate how the results of the experiment performed addressed the goals of the research.

6.1.1 Question 1: Can low-cost visual, ultraviolet, and thermal cameras collect enough information to be used for detection of an electrical arc detection with machine learning?

Within the context of these experiments, the answer to this question was yes, though the accuracy of the data from each camera was variable. Overall, there were models that were able to correctly classify images with the electrical arcs, with at least better than a random guess, a majority of the time, for at least three of the four validation datasets. The data from the ultraviolet camera could be classified with an accuracy of over 90% for two of the four validation datasets, and the thermal camera performed very well overall. Each of the cameras were, at least in some circumstances, capturing data that could contribute to the identification of the electrical arc, even if not every camera was able to do so in every circumstance. Therefore, the cameras all collect data that is usable for arc detection from this source in certain conditions, though the ultraviolet and especially the visible camera could not be used alone to detect arcs in all conditions. As predicted, ambient light levels, in particular bright sunlight, was a limiting factor in the use of these cameras for this application.

6.1.2 Question 2: How does an ensemble camera system as a means of classifying data for arc detection compare to models for each individual camera?

The two approaches to creating the ensemble results, combining the separate outputs by majority vote or by classification layer, had somewhat different results compared to the separate cameras, though both tended to at least improve the consistency of the results, and depending on the model parameters could also often improve accuracy.

Combining the outputs by a majority vote of the results often corresponded to a better accuracy than the worst of the constituent models, but often not as good as the

best, at least for the training and validation data used in these experiments. While the outcomes of this approach were not always impressive for all datasets, a simple majority vote may be an effective way to average the predictions from several sensors without giving a strong preference to any one. This may be useful if, for example, a sensor was more reliable in training data than in application, and giving it a strong preference may not improve the results in all cases. Additionally, a result aggregated with a simple vote could be a useful summary step in analysis where the results are going to be further analyzed, such as by a human in the loop, who could then compare the separate predictions to decide if further inspection may be worthwhile. Additionally, a vote may be more useful in an implementation with more sensors, with strengths and weaknesses that vary more from each other than the visual or ultraviolet cameras here, where the majority vote may be less prone to compounding a specific trend in the predictions.

Combining the outputs with a classification layer often resulted in a model that was more likely to give an accurate prediction, or at least not a significantly worse prediction, than the models that made up the ensemble. This approach can combine two models as well as three, though the predictions were never at best much more accurate than the most accurate of the constituent models. Still, this approach to ensemble models had the best overall accuracy outside of the thermal-only data, particularly for the most successful parameters tested, using transfer learning for models based either on ResNet or VGGNet. Even the combined ultraviolet and visual data for Valid Data 2, which had the lowest accuracies overall, were above many of those for the separate models. While somewhat more computationally intensive, these combined models did provide an improved classification compared to any approach that was consistent across the four different validation datasets.

A potential disadvantage to using a classification layer could come from the lack of explainability, in which, as with Valid Data 2, some quirk of the data confuses the model, giving much lower accuracies overall for a reason that may be difficult

to identify. This sort of variability is something that a majority vote, or separate models, may be better able to identify.

A final combination of approaches, a majority vote of the classification layer models, was included for completeness. While this was less accurate than the classification layer models for the best cases, the accuracies were more consistent than most of the other approaches, and may have use in avoiding the variability that was seen throughout the experiments across other parameters. This approach, however, was the most computationally intensive of those compared, and therefore may not be practical in all applications, including those running at the edge, or near the edge, as may be done in a UAV-based application.

6.1.3 Question 3: How do other conditions impact the ability of a machine learning model to detect electrical arcing with the data from each of the three cameras separately? As an ensemble?

As predicted, ambient sunlight was a factor in the ability to accurately classify the arcs in both the visual and ultraviolet data. Also as predicted, this was a more significant factor for the visual camera, and at least for these experiments did not affect the thermal camera results.

The quality and relevance of the training data was also an important component of model accuracy. Earlier experiments had poor accuracy in low light compared to the generally quite accurate performance of Valid Data 4, the low light dataset, in these experiments, possibly due an imbalance of low light data in the dataset. This type of data was more balanced in this dataset, which may have contributed to the high accuracy of Valid Data 4. The low accuracy of Valid Data 2 for the ultraviolet datasets and especially the visual datasets may have been in part due to an inadequate amount of training data in bright, direct sunlight, though, as indicated

by the somewhat better accuracy by models using the ultraviolet data, the sunlight limited the amount of data that could be captured was likely also a factor.

Model choice also had an impact, with the smallest models tested, AlexNet, often, though not always, making less accurate predictions than ResNet and VGGNet. For more complex datasets, more complex models would likely be worthwhile to compare, though there would be a trade-off in training time and computational requirements. Transfer learning also usually, but not always, improved the classification accuracy, even for cameras that did not output three channel RGB images. Similar patterns continued with the ensemble models, with ResNet and VGGNet often performing better than AlexNet, and models that were trained with weights from transfer learning performing better than models that did not.

For the ensemble models, the accuracy of the models used to make up the ensemble was a useful, if not precise, predictor of how the ensemble would perform. At least with the approaches to ensemble models compared in this research, the ensemble models were generally not more accurate than the separate models in every case, but were often more consistent, and in the case of the classification layer models were at times as or more accurate than the models that made them up.

6.1.4 Question 4: What are the implications of the accuracy results of the individual camera data and ensemble models, and how can it be used to design a more effective, low-cost camera sensor?

The data for training the models does need to be varied and representative of the variety of real world cases that would be encountered. However, what representative means for a dataset may be more complicated and harder to predict than simply having enough examples of each of the known classes. The flexibility of machine learning models in allowing data to determine the parameters to evaluate can also be a limitation in that weaknesses may also be more difficult to predict and account

for. An application using machine learning would benefit from careful testing of well varied validation datasets to identify such problems early, and to amend the training data accordingly.

A variety of sensors may help to fill in gaps in the ability of a single sensor, as well as gaps in the training data. This is a known benefit of sensor fusion, and this applies to a machine learning based use case as well. In particular, similar sensor inputs may be prone to similar types of failure for certain data, as was seen in the Valid Data 2 predictions for the visual and ultraviolet data.

Human verification of results, at least for a new system, may be a valuable step, as machine learning based models may not always fail in predictable ways, as again was seen in the results for both the visual and ultraviolet data for Valid Data 2. This need for verification may also provide additional justification for use of a relatively simple ensemble method that can easily be evaluated, such as a majority vote.

Parameters that help individual models also help ensemble models, as was seen in these experiments with model architecture and the use of transfer learning. Testing accuracy with validation datasets on smaller aspects of larger models may be a more efficient way to assess what might give better accuracy for the larger ensemble model as well.

6.2 Future Work

The results of this dissertation imply possible future experiments that would potentially build upon this work, based on the conclusions drawn in the fourth research question in the previous section.

For assessing this specific camera system, or another like it, there are several additional experiments that may help to improve its performance.

A first set of experiments could be additional data collection and analysis with this specific prototype sensor. Additional preprocessing of the data for training and classification could help identify the source of the results of Valid Data 2, or

at least eliminate some possibilities. Collecting a larger and more varied dataset, including additional locations, could determine how well any of these approaches could generalize. The addition of another light source, such as an ultraviolet light or visual light could help characterize how the visual and ultraviolet camera behave in varying light. Conditions where the thermal camera might perform poorly, such as very warm conditions, or with other sources of heat in the camera's frame, should also be included.

Modifications to the sensor could also be made, such as changing out the cameras, adding additional cameras, or additional types of sensors. A comparison of the same classification techniques with a dataset collected from different sensors could provide additional context to the strengths and limitations of the different ensemble approaches compared.

Additionally, more model architectures, both for individual inputs and ensembles, could be compared. With more complex data, more complex models may be useful, both for individual cameras and for ensembles. The more complex models compared in these experiments, ResNet and VGGNet, often performed better than the less complex AlexNet. Limited experiments with the larger ResNet and VGGNet models had diminishing returns with a previous but similar dataset, but a more complex dataset might give more advantage to a more complex model. More complex ensemble models may also be able to give better results. This could include a weighted majority vote, such as was done in [76]. The weights could be determined by training, or, given the information that sunlight may be a predictable indicator that visual and ultraviolet data may be less useful, based on additional sensor data. The classification layer ensembles could also be compared in more detail to evaluate any additional advantages or disadvantages with more complex datasets.

Modifications could also be made to the sensor itself, potentially changing the cameras used, whether cameras from different manufacturers to compare designs or higher cost cameras, or cameras that would allow for more direct control to test the effect of manually setting the focus, gain, and exposure. Additional cameras could

also be used, adding more information to an ensemble. Other types of sensors, such as a microphone to capture the sound of the arc, could also be included, and used as additional classification inputs. As discussed above, the difference in behavior of the thermal camera compared to the visual and ultraviolet helped to identify weaknesses in those two camera systems. A greater variety of sensor information available may help to make the overall sensor more adaptable, and less vulnerable to issues with one specific sensor.

Bibliography

[1] J. B. Campbell and R. H. Wynne, *Introduction to remote sensing*. Guilford press, 2011. 1

[2] E. Alvarez-Vanhard, T. Corpetti, and T. Houet, "Uav & satellite synergies for optical remote sensing applications: A literature review," *Science of remote sensing*, vol. 3, p. 100019, 2021. 1

[3] E. Balestrieri, P. Daponte, L. De Vito, F. Picariello, and I. Tudosa, "Sensors and measurements for uav safety: An overview," *Sensors*, vol. 21, no. 24, p. 8253, 2021. 2

[4] C. Stöcker, R. Bennett, F. Nex, M. Gerke, and J. Zevenbergen, "Review of the current state of uav regulations," *Remote sensing*, vol. 9, no. 5, p. 459, 2017. 2

[5] R. R. Navalgund, V. Jayaraman, and P. Roy, "Remote sensing applications: An overview," *current science*, pp. 1747–1766, 2007. 3

[6] X. Yu and H. Su, "Characteristic band of arcing between pantograph-catenary system in urban rail detection system by uv method," in *Journal of Physics: Conference Series*, vol. 1168, p. 022062, IOP Publishing, 2019. 3, 18

[7] R. Morello, S. C. Mukhopadhyay, Z. Liu, D. Slomovitz, and S. R. Samantaray, "Advances on sensing technologies for smart cities and power grids: A review," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7596–7610, 2017. 4

[8] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, and J. J. Sousa, "Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry," *Remote sensing*, vol. 9, no. 11, p. 1110, 2017. 5, 16

[9] A. Morales, R. Guerra, P. Horstrand, M. Diaz, A. Jimenez, J. Melian, S. Lopez, and J. F. Lopez, "A multispectral camera development: From the prototype assembly until its use in a uav system," *Sensors*, vol. 20, no. 21, p. 6129, 2020. 6, 26

[10] J. Kim, H. Kim, T. Kim, N. Kim, and Y. Choi, "Mlpd: Multi-label pedestrian detector in multispectral domain," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7846–7853, 2021. 6, 16, 26

[11] D. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997. 9

[12] R. C. Luo and M. G. Kay, "Multisensor integration and fusion: issues and approaches," in *Sensor fusion*, vol. 931, pp. 42–49, SPIE, 1988. 10

[13] F. Alam, R. Mehmood, I. Katib, N. N. Albogami, and A. Albeshri, "Data fusion and iot for smart ubiquitous environments: A survey," *Ieee Access*, vol. 5, pp. 9533–9554, 2017. 10

[14] R. C. Luo, C.-C. Yih, and K. L. Su, "Multisensor fusion and integration: approaches, applications, and future research directions," *IEEE Sensors journal*, vol. 2, no. 2, pp. 107–119, 2002. 10

[15] T. Meng, X. Jing, Z. Yan, and W. Pedrycz, "A survey on machine learning for data fusion," *Information Fusion*, vol. 57, pp. 115–129, 2020. 10

[16] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use," *Remote sensing*, vol. 4, no. 6, pp. 1671–1692, 2012. 11

[17] W. Commons, "File:onyxstar fox-c8 xt xender 360.jpg," 2024. [Online; accessed 20-February-2025]. 11

[18] Y. Guo, X. Jia, D. Paull, J. Zhang, A. Farooq, X. Chen, and M. N. Islam, "A drone-based sensing system to support satellite image analysis for rice farm mapping," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 9376–9379, IEEE, 2019. 12

[19] Y. Inoue and M. Yokoyama, "Drone-based optical, thermal, and 3d sensing for diagnostic information in smart farming–systems and algorithms–," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 7266–7269, IEEE, 2019. 12

[20] E. D. McCarthy, J. M. Martin, M. M. Boer, and J. A. Welbergen, "Drone-based thermal remote sensing provides an effective new tool for monitoring the abundance of roosting fruit bats," *Remote Sensing in Ecology and Conservation*, vol. 7, no. 3, pp. 461–474, 2021. 12

[21] E. Alevizos, D. Oikonomou, A. V. Argyriou, and D. D. Alexakis, "Fusion of drone-based rgb and multi-spectral imagery for shallow water bathymetry inversion," *Remote Sensing*, vol. 14, no. 5, p. 1127, 2022. 12

[22] J. Zhang, W. Guo, B. Zhou, and G. S. Okin, "Drone-based remote sensing for research on wind erosion in drylands: possible applications," *Remote Sensing*, vol. 13, no. 2, p. 283, 2021. 12

[23] A. M. Cunliffe, R. E. Brazier, and K. Anderson, "Ultra-fine grain landscape-scale quantification of dryland vegetation structure with drone-acquired structure-from-motion photogrammetry," *Remote Sensing of Environment*, vol. 183, pp. 129–143, 2016. 12

[24] L. Tang and G. Shao, "Drone remote sensing for forestry research and practices," *Journal of Forestry Research*, vol. 26, pp. 791–797, 2015. 12

[25] J. R. Martínez-de Dios, L. Merino, F. Caballero, and A. Ollero, "Automatic forest-fire measuring using ground stations and unmanned aerial systems," *Sensors*, vol. 11, no. 6, pp. 6328–6353, 2011. 12

[26] M. Rutkauskas, M. Asenov, S. Ramamoorthy, and D. T. Reid, "Autonomous multi-species environmental gas sensing using drone-based fourier-transform infrared spectroscopy," *Optics Express*, vol. 27, no. 7, pp. 9578–9587, 2019. 13

[27] S. Carroll, J. Satme, S. Alkharusi, N. Vitzilaios, A. Downey, and D. Rizos, "Drone-based vibration monitoring and assessment of structures," *Applied Sciences*, vol. 11, no. 18, p. 8560, 2021. 13

[28] H.-A. Langåker, H. Kjerkreit, C. L. Syversen, R. J. Moore, Ø. H. Holhjem, I. Jensen, A. Morrison, A. A. Transeth, O. Kvien, G. Berg, *et al.*, "An autonomous drone-based system for inspection of electrical substations," *International Journal of Advanced Robotic Systems*, vol. 18, no. 2, p. 17298814211002973, 2021. 13

[29] D. Waleed, S. Mukhopadhyay, U. Tariq, and A. H. El-Hag, "Drone-based ceramic insulators condition monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021. 13

[30] D. Landgrebe, "Information extraction principles and methods for multispectral and hyperspectral image data," in *Information processing for remote sensing*, pp. 3–37, World Scientific, 1999. 13

[31] J. V. Taranik, "Characteristics of the landsat multispectral data system," tech. rep., US Geological Survey, 1978. 14

[32] W. Commons, "File:solar spectrum en.svg," 2024. [Online; accessed 20-February-2025]. 14

[33] N. Yokoya, C. Grohnfeldt, and J. Chanussot, "Hyperspectral and multispectral data fusion: A comparative review of the recent literature," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 2, pp. 29–56, 2017. 14

[34] R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis, *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (aviris)," *Remote sensing of environment*, vol. 65, no. 3, pp. 227–248, 1998. 14

[35] L. J. Rickard, R. W. Basedow, E. F. Zalewski, P. R. Silverglate, and M. Landers, "Hydice: An airborne system for hyperspectral imaging," in *Imaging Spectrometry of the Terrestrial Environment*, vol. 1937, pp. 173–179, SPIE, 1993. 14

[36] M. T. Eismann and R. C. Hardie, "Application of the stochastic mixing model to hyperspectral resolution enhancement," *IEEE transactions on geoscience and remote sensing*, vol. 42, no. 9, pp. 1924–1933, 2004. 15

[37] R. B. Gomez, A. Jazaeri, and M. Kafatos, "Wavelet-based hyperspectral and multispectral image fusion," in *Geo-Spatial Image and Data Exploitation II*, vol. 4383, pp. 36–42, SPIE, 2001. 15

[38] F. Marinello, "Last generation instrument for agriculture multispectral data collection," *Agricultural Engineering International: CIGR Journal*, vol. 19, no. 1, pp. 87–93, 2017. 16

[39] I. Wahab, O. Hall, and M. Jirström, "Remote sensing of yields: Application of uav imagery-derived ndvi for estimating maize vigor and yields in complex farming systems in sub-saharan africa," *Drones*, vol. 2, no. 3, p. 28, 2018. 16

[40] L. Quebrajo, M. Perez-Ruiz, L. Pérez-Urrestarazu, G. Martínez, and G. Egea, "Linking thermal imaging and soil remote sensing to enhance irrigation

management of sugar beet," *Biosystems Engineering*, vol. 165, pp. 77–87, 2018. 16

[41] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1037–1045, 2015. 16

[42] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu, and A. M. López, "Pedestrian detection at day/night time with visible and fir cameras: A comparison," *Sensors*, vol. 16, no. 6, p. 820, 2016. 16

[43] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37, Springer, 2016. 17

[44] M. Kim, S. Joung, K. Park, S. Kim, and K. Sohn, "Unpaired cross-spectral pedestrian detection via adversarial feature learning," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 1650–1654, IEEE, 2019. 17

[45] H. Zhang, E. Fromont, S. Lefevre, and B. Avignon, "Multispectral fusion for object detection with cyclic fuse-and-refine blocks," in *2020 IEEE International conference on image processing (ICIP)*, pp. 276–280, IEEE, 2020. 17

[46] K. Takumi, K. Watanabe, Q. Ha, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada, "Multispectral object detection for autonomous vehicles," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pp. 35–43, 2017. 17

[47] H. Ha, S. Han, and J. Lee, "Fault detection on transmission lines using a microphone array and an infrared thermal imaging camera," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 1, pp. 267–275, 2011. 17

[48] X.-F. Li and J.-M. Liu, "Edge detection on arc image of low voltage apparatus," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693)*, vol. 5, pp. 2921–2924, IEEE, 2003. 17

[49] Y. Seferi, S. M. Blair, C. Mester, and B. G. Stewart, "A novel arc detection method for dc railway systems," *Energies*, vol. 14, no. 2, p. 444, 2021. 18

[50] I. Aydin, "A new approach based on firefly algorithm for vision-based railway overhead inspection system," *Measurement*, vol. 74, pp. 43–55, 2015. 18

[51] O. Bruno, A. Landi, M. Papi, and L. Sani, "Phototube sensor for monitoring the quality of current collection on overhead electrified railways," *Proceedings of the institution of mechanical engineers, part f: Journal of rail and rapid transit*, vol. 215, no. 3, pp. 231–241, 2001. 18

[52] A. Landi, L. Menconi, and L. Sani, "Hough transform and thermo-vision for monitoring pantograph-catenary system," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 220, no. 4, pp. 435–447, 2006. 18

[53] İ. Aydın, O. Yaman, M. Karaköse, and S. B. Çelebi, "Particle swarm based arc detection on time series in pantograph-catenary system," in *2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings*, pp. 344–349, IEEE, 2014. 18

[54] D. Ma, L. Jin, J. He, and K. Gao, "Classification of partial discharge severities of ceramic insulators based on texture analysis of uv pulses," *High Voltage*, vol. 6, no. 6, pp. 986–996, 2021. 18

[55] S. Pei, Y. Liu, X. Ji, J. Geng, G. Zhou, and S. Wang, "Uv-flashover evaluation of porcelain insulators based on deep learning," *IET Science, Measurement & Technology*, vol. 12, no. 6, pp. 770–776, 2018. 18

[56] J.-R. Riba, Á. Gómez-Pau, and M. Moreno-Eguilaz, "Experimental study of visual corona under aeronautic pressure conditions using low-cost imaging sensors," *Sensors*, vol. 20, no. 2, p. 411, 2020. 18

[57] J.-R. Riba, Á. Gómez-Pau, and M. Moreno-Eguilaz, "Insulation failure quantification based on the energy of digital images using low-cost imaging sensors," *Sensors*, vol. 20, no. 24, p. 7219, 2020. 18, 26

[58] J. Jiang, M. Zhao, Z. Wen, C. Zhang, and R. Albarracín, "Detection of dc series arc in more electric aircraft power system based on optical spectrometry," *High Voltage*, vol. 5, no. 1, pp. 24–29, 2020. 19

[59] C. F. Kumru, A. Lutfi, A. El-Hag, A. Darwish, S. S. Refaat, and H. Abu-Rub, "A study on the ability of different non-intrusive sensors for diagnosing outdoor insulator defects," in *2023 IEEE Electrical Insulation Conference (EIC)*, pp. 1–4, IEEE, 2023. 19, 26

[60] C. R. Turner, A. Fuggetta, L. Lavazza, and A. L. Wolf, "A conceptual basis for feature engineering," *Journal of Systems and Software*, vol. 49, no. 1, pp. 3–15, 1999. 19

[61] N. Ha, K. Xu, G. Ren, A. Mitchell, and J. Z. Ou, "Machine learning-enabled smart sensor systems," *Advanced Intelligent Systems*, vol. 2, no. 9, p. 2000063, 2020. 19

[62] S. Ertekin, J. Huang, L. Bottou, and L. Giles, "Learning on the border: active learning in imbalanced data classification," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 127–136, 2007. 19

[63] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017. 20

[64] A. Abraham, "Artificial neural networks," *Handbook of measuring system design*, 2005. 20, 22

[65] W. Commons, "File:artificial neuron structure.svg," 2024. [Online; accessed 20-February-2025]. 20

[66] W. Commons, "File:colored neural network.svg," 2025. [Online; accessed 20-February-2025]. 20

[67] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021. 22

[68] W. Commons, "File:typical cnn.png," 2022. [Online; accessed 20-February-2025]. 22

[69] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, p. 495, 2017. 23

[70] W. Commons, "File:convolutional neural network, convolution worked example.png," 2024. [Online; accessed 20-February-2025]. 23

[71] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile networks and applications*, vol. 19, pp. 171–209, 2014. 23

[72] D. Müller, I. Soto-Rey, and F. Kramer, "An analysis on ensemble learning optimized medical image classification with deep convolutional neural networks," *Ieee Access*, vol. 10, pp. 66467–66480, 2022. 23

[73] F. Ullah, I. Ullah, R. U. Khan, S. Khan, K. Khan, and G. Pau, "Conventional to deep ensemble methods for hyperspectral image classification: A comprehensive survey," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024. 25

[74] J. Nalepa, M. Myller, L. Tulczyjew, and M. Kawulok, "Deep ensembles for hyperspectral image data classification and unmixing," *Remote Sensing*, vol. 13, no. 20, p. 4133, 2021. 25

[75] S. Dong, W. Feng, Y. Quan, G. Dauphin, L. Gao, and M. Xing, "Deep ensemble cnn method based on sample expansion for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022. 25

[76] B. Cheng, W. Wu, D. Tao, S. Mei, T. Mao, and J. Cheng, "Random cropping ensemble neural network for image classification in a robotic arm grasping system," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6795–6806, 2020. 25, 107

[77] R. Verma and J. Ali, "A comparative study of various types of image noise and efficient noise removal techniques," *International Journal of advanced research in computer science and software engineering*, vol. 3, no. 10, 2013. 25

[78] P. Patidar, M. Gupta, S. Srivastava, and A. K. Nagawat, "Image de-noising by various filters for different noise," *International journal of computer applications*, vol. 9, no. 4, pp. 45–50, 2010. 26

[79] H. Schulzrinne, A. Rao, and R. Lanphier, "Rfc2326: Real time streaming protocol (rtsp)," 1998. 32

[80] J. Newmarch and J. Newmarch, "Ffmpeg/libav," *Linux sound programming*, pp. 227–234, 2017. 32

[81] "Usb4000 data sheet," tech. rep., Ocean Optics. 33

[82] "python-seabreeze." https://python-seabreeze.readthedocs.io/en/latest/. 33

[83] D. Sliney, "What is light? the visible spectrum and beyond," *Eye*, vol. 30, no. 2, pp. 222–229, 2016. 44

[84] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk, "What is the space of spectral sensitivity functions for digital color cameras?," in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 168–179, IEEE, 2013. 44

[85] F. Günther and S. Fritsch, "Neuralnet: training of neural networks.," *R J.*, vol. 2, no. 1, p. 30, 2010. 51

[86] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009. 51

[87] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019. 52

[88] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012. 53

[89] W. Commons, "File:alexnet original block diagram.svg," 2024. [Online; accessed 20-February-2025]. 53

[90] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 53

[91] W. Commons, "File:vgg neural network.png," 2019. [Online; accessed 20-February-2025]. 53

[92] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 53

[93] W. Commons, "File:resblock.png," 2023. [Online; accessed 20-February-2025]. 53

[94] Ž. Vujović *et al.*, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021. 55

[95] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015. 55

[96] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv preprint arXiv:2008.05756*, 2020. 56

[97] J. Opitz, "A closer look at classification evaluation metrics and a critical reflection of common evaluation practice," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 820–836, 2024. 57

[98] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. 128

Appendix A

Data Collection Code

A.1 Spectrometer

The python script used to implement the Seabreeze API to control the Ocean View spectrometer is included here. For each reading, several loops were taken, usually five, to improve the accuracy of the data. This was done after I noticed to first measurement was often lower than the others, or to account for some kind of interference, such as a shadow moving over the input.

```
from seabreeze.spectrometers import list_devices, Spectrometer
import numpy as np
import os
import datetime


def save_spec_loop(save_path, loop_num):
    '''
    Save spectrometer data with a time stamp
    :param save_path: system path of were to save data
    :param loop_num: number of times to read and save data,
    to improve accuracy
```

```
    :return:
    '''
    os.makedirs(save_path, exist_ok=True)
    devices = list_devices()
    device = Spectrometer(devices[0])
    # set integration time
    device.integration_time_micros(100000)

    for i in range(loop_num):
        # generate time stamp
        now = datetime.datetime.now()
        now = (now.strftime('%Y-%m-%dT%H:%M:%S') +
               ('-%02d' % (now.microsecond / 10000)))

        file_name = (save_path + "/" + now + "_" +
                     str(i) + "_" + ".npy")
        wavelengths, intensities = take_spec_data(device)
        save_spectrum(wavelengths, intensities, file_name)
```

A.2 Images

This section includes to primary two functions used to capture image data.

```
def run_save_images(base_path, date, directory, label):
    '''
    This code is used to capture images from the three cameras
     simultaneously when the space bar of a laptop is hit, giving
      control over the timing of the images.
    :param base_path: system path to were the images should be stored
```

```
:param date: date data was captured, or organizing data. Images also
contain a time stamp.
:param label: class names, 'on' or 'off'
:param directory: label of subdirectory to sort data type, train or
 valid, incremented train1, train2... or valid1, valid2...
 if multiple taken on same day
:return:
'''

# values set by system, checked manually with
# v2l2-ctl --list-devices
# manually checked each time to make sure the cameras are
# in correct order
cam_id1 = 8
cam_id2 = 10
cam_id3 = 12

data_path = "{}/{}/valid/{}/{}".format(base_path, date,
                                       directory, label)

path1 = os.path.join(data_path, "visual")
path2 = os.path.join(data_path, "uv")
path3 = os.path.join(data_path, "thermal")

# create path for each camera type
os.makedirs(path1, exist_ok=True)
os.makedirs(path2, exist_ok=True)
os.makedirs(path3, exist_ok=True)
```

```
    save_images(cam_id1, cam_id2, cam_id3, path1, path2, path3,
                            start_counter=0)

def save_images(cam_id1, cam_id2, cam_id3, path1, path2, path3,
                start_counter=None):
    '''
    The function used to capture images
    :param cam_id1: visual camera id
    :param cam_id2: uv camera id
    :param cam_id3: thermal camera id
    :param path1: visual camera save path
    :param path2: uv camera save path
    :param path3: thermal camera save path
    :param start_counter: counter value used to number images
    :return:
    '''
    cam1 = cv2.VideoCapture(cam_id1)
    cam2 = cv2.VideoCapture(cam_id2)
    cam3 = cv2.VideoCapture(cam_id3)

    # only two cameras were shown at once due to machine
    # limitations the ones chosen varied with the light conditions
    # or which cameras were on the end to make sure the lighter was
    # in frame and on
    cv2.namedWindow("thermal")
    cv2.namedWindow("visual")
    # cv2.namedWindow("uv")

    if start_counter:
```

```
    img_counter = start_counter
else:
    img_counter = 0

while True:
    ret1, frame1 = cam1.read()
    ret2, frame2 = cam2.read()
    ret3, frame3 = cam3.read()
    if not ret1:
        print("failed to grab frame cam1")
        break

    if not ret3:
        print("failed to grab frame cam3")
        break

    cv2.imshow("thermal", frame3)
    cv2.imshow("visual", frame1)
    # cv2.imshow("uv", frame2)

    k = cv2.waitKey(5)
    if k % 256 == 27:
        # ESC pressed
        # end session
        print("Escape hit, closing...")
        break
    elif k % 256 == 32:
        # SPACE pressed
        #take image
```

```
            # create a numerical label with leading zeros
            # to sort data later
            count_label = str(img_counter).zfill(3)

            # create timestamp
            now = datetime.datetime.now()
            now = (now.strftime('%Y-%m-%dT%H:%M:%S') +
                   ('-%02d' % (now.microsecond / 10000)))

            # create labels for each image
            img1_name = os.path.join(path1, "visual_{}_frame_{}.png"
            .format(now, count_label))
            img2_name = os.path.join(path2, "uv_{}_frame_{}.png"
            .format(now, count_label))
            img3_name = os.path.join(path3, "thermal_{}_frame_{}.png"
            .format(now, count_label))

            # save images
            cv2.imwrite(img1_name, frame1)
            cv2.imwrite(img2_name, frame2)
            cv2.imwrite(img3_name, frame3)
            print("{} written!".format(img_counter))
            img_counter += 1

# clean up when done
cam1.release()
cam2.release()
cam3.release()
```

```
cv2.destroyAllWindows()
```

Appendix B

Detailed Result Metrics

Additional metrics from both the individual camera models and the ensemble models are included in this appendix.

B.1 Individual Cameras

This section includes the full detailed results from the individual camera models. A summary of results can be found in Chapter 4. Table B.1 shows the accuracy with the use of transfer learning, and Table B.2 shows the accuracy using transfer learning. The full list of precision, recall, and F1 metrics are included in Table B.3 without transfer learning and in Table B.4. MCC results are included in Table B.5 without transfer learning and Table B.6 with transfer learning. The implementation of the metrics used are imported from scikit-learn [98].

Table B.1: Accuracy results for individual cameras without transfer learning.

Accuracy - No Transfer Learning					
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Visual					
AlexNet	0.339	0	0.549	0.705	0.398
ResNet	0.64	0.016	0.534	0.565	0.437
VGGNet	0.573	0.02	0.235	0.603	0.35
Ultraviolet					
AlexNet	0.552	0.285	0.567	0.945	0.583
ResNet	0.87	0.044	0.892	0.962	0.691
VGGNet	0.448	0.494	0.582	0.932	0.611
Thermal					
AlexNet	0.954	1	0.56	0.764	0.814
ResNet	0.912	1	0.922	0.949	0.946
VGGNet	0.962	1	0.944	0.954	0.965

Table B.2: Accuracy results for individual cameras with transfer learning.

Accuracy - Transfer Learning					
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Visual					
AlexNet	0.293	0.092	0.582	0.759	0.432
ResNet	0.707	0.024	0.784	0.608	0.533
VGGNet	0.494	0.233	0.713	0.768	0.553
Ultraviolet					
AlexNet	0.749	0.112	0.866	0.954	0.67
ResNet	0.556	0.378	0.914	0.966	0.706
VGGNet	0.561	0.305	0.918	0.966	0.69
Thermal					
AlexNet	0.929	1	0.888	0.97	0.946
ResNet	0.95	1	0.974	0.97	0.974
VGGNet	0.946	1	1	0.958	0.977

Table B.3: Other metric results for individual cameras without transfer learning

Additional Metrics - No Transfer Learning

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Visual					
AlexNet					
Precision	0.303	0	0.694	0.713	0.373
Recall	0.287	0	0.185	0.713	0.291
F1	0.295	0	0.292	0.713	0.327
ResNet					
Precision	0.572	0.032	0.519	0.542	0.463
Recall	1	0.031	1	1	0.754
F1	0.728	0.032	0.684	0.703	0.574
VGGNet					
Precision	0.659	0.039	0.284	0.577	0.357
Recall	0.235	0.039	0.341	0.861	0.367
F1	0.346	0.039	0.31	0.691	0.362
Ultraviolet					
AlexNet					
Precision	0.569	0.361	0.538	0.936	0.569
Recall	0.287	0.52	1	0.959	0.703
F1	0.382	0.426	0.699	0.947	0.629
ResNet					
Precision	0.868	0.083	1	1	0.706
Recall	0.861	0.087	0.785	0.926	0.659
F1	0.865	0.085	0.88	0.962	0.682
VGGNet					
Precision	0.435	0.502	0.547	1	0.578
Recall	0.496	0.961	1	0.869	0.842
F1	0.463	0.659	0.707	0.93	0.685
Thermal					
AlexNet					
Precision	1	1	0.534	0.696	0.741
Recall	0.904	1	1	0.959	0.968
F1	0.95	1	0.696	0.807	0.839
ResNet					
Precision	1	1	0.925	0.982	0.974
Recall	0.817	1	0.919	0.918	0.916
F1	0.9	1	0.922	0.949	0.944
VGGNet					
Precision	1	1	0.962	0.983	0.985
Recall	0.922	1	0.926	0.926	0.944
F1	0.959	1	0.943	0.954	0.964

Table B.4: Other metric results for individual cameras with transfer learning

Additional Metrics - Transfer Learning					
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Visual					
AlexNet					
Precision	0.292	0.159	0.649	0.715	0.435
Recall	0.33	0.181	0.37	0.885	0.439
F1	0.31	0.37	0.472	0.791	0.437
ResNet					
Precision	1	0.047	0.699	0.571	0.531
Recall	0.391	0.047	1	0.951	0.605
F1	0.562	0.047	0.823	0.714	0.566
VGGNet					
Precision	0.484	0.322	0.637	0.719	0.537
Recall	0.809	0.457	1	0.902	0.794
F1	0.606	0.378	0.778	0.8	0.641
Ultraviolet					
AlexNet					
Precision	0.71	0.174	0.954	1	0.673
Recall	0.809	0.197	0.77	0.91	0.667
F1	0.756	0.185	0.852	0.953	0.67
ResNet					
Precision	0.531	0.435	0.944	1	0.672
Recall	0.678	0.74	0.881	0.934	0.812
F1	0.595	0.548	0.912	0.966	0.735
VGGNet					
Precision	0.529	0.345	0.925	1	0.668
Recall	0.8	0.402	0.911	0.934	0.762
F1	0.637	0.371	0.918	0.966	0.712
Thermal					
AlexNet					
Precision	1	1	0.818	1	0.941
Recall	0.852	1	1	0.943	0.952
F1	0.92	1	0.9	0.97	0.946
ResNet					
Precision	1	1	1	1	1
Recall	0.896	1	0.948	0.943	0.948
F1	0.945	1	0.973	0.97	0.973
VGGNet					
Precision	1	1	1	0.983	0.996
Recall	0.887	1	1	0.934	0.958
F1	0.94	1	1	0.958	0.977

Table B.5: MCC results for individual cameras without transfer learning.

MCC - No Transfer Learning					
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Visual					
AlexNet	-0.327	-1	0.15	0.409	0.387
ResNet	0.419	-0.968	0.177	0.238	0.271
VGGNet	0.162	-0.961	-0.543	0.226	0.223
Ultraviolet					
AlexNet	0.099	-0.498	0.262	0.89	0.885
ResNet	0.74	-0.915	0.803	0.927	0.895
VGGNet	-0.102	-0.102	0.294	0.873	0.824
Thermal					
AlexNet	0.911	1	0.245	0.567	0.595
ResNet	0.836	1	0.843	0.901	0.87
VGGNet	0.927	1	0.889	0.909	0.881

Table B.6: MCC results for individual cameras with transfer learning.

MCC - Transfer Learning					
	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
Visual					
AlexNet	-0.413	-0.83	0.185	0.531	0.532
ResNet	0.5	-0.953	0.628	0.277	0.288
VGGNet	0.013	-0.607	0.518	0.551	0.556
Ultraviolet					
AlexNet	0.504	-0.788	0.746	0.911	0.874
ResNet	0.125	-0.383	0.83	0.935	0.906
VGGNet	0.156	-0.401	0.836	0.935	0.906
Thermal					
AlexNet	0.866	1	0.796	0.943	0.917
ResNet	0.904	1	0.949	0.943	0.917
VGGNet	0.896	1	1	0.917	0.892

B.2 Classification Layer

The detailed results form both Model 1 and Model 2 without transfer learning are included in this section.

Table B.7: Accuracy results for ensemble models, Classification Layer Model 1, no transfer learning.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
AlexNet					
UV & Vis	0.481	0.51	0.504	0.515	0.503
UV & Therm	0.519	0.49	0.519	0.574	0.525
Therm & Vis	0.962	1	0.94	0.945	0.962
All Three	0.519	0.49	0.369	0.675	0.509
ResNet					
UV & Vis	0.975	0.116	0.959	0.97	0.754
UV & Therm	0.866	0.189	0.907	0.966	0.731
Therm & Vis	0.954	0.992	0.918	0.996	0.964
All Three	0.925	0.514	0.959	0.966	0.841
VGGNet					
UV & Vis	0.715	0	0.78	0.7	0.55
UV & Therm	0.941	1	0.825	0.949	0.926
Therm & Vis	0.51	0.161	0.698	0.869	0.559
All Three	0.996	0	0.784	0.544	0.581

Table B.8: Accuracy results for ensemble models, Classification Layer Model 2, no transfer learning.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
AlexNet					
UV & Vis	1	0	0.72	0.473	0.548
UV & Therm	0.958	0.996	0.892	0.937	0.945
Therm & Vis	0.904	0.783	0.933	0.928	0.887
All Three	0.548	0.49	0.362	0.671	0.513
ResNet					
UV & Vis	0.912	0.064	0.97	0.966	0.728
UV & Therm	0.858	0.329	0.944	0.966	0.774
Therm & Vis	0.925	0.912	0.918	0.996	0.937
All Three	0.736	0.51	0.847	0.797	0.724
VGGNet					
UV & Vis	0.912	0	0.709	0.692	0.576
UV & Therm	0.933	0.992	0.966	0.962	0.964
Therm & Vis	0.967	0.699	0.944	0.966	0.893
All Three	0.967	0.916	0.937	0.916	0.934

Table B.9: MCC results for ensemble models, Classification Layer Model 1, no transfer learning.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
AlexNet					
UV & Vis	0	0	0	0	0
UV & Therm	0	0	0.102	0.149	0.05
Therm & Vis	0.927	1	0.881	0.896	0.925
All Three	0	0	-0.318	0.35	0.02
ResNet					
UV & Vis	0.951	-0.79	0.918	0.943	0.509
UV & Therm	0.733	-0.674	0.819	0.935	0.462
Therm & Vis	0.911	0.984	0.837	0.992	0.927
All Three	0.858	0.025	0.918	0.935	0.691
VGGNet					
UV & Vis	0.514	-1	0.624	0.4	0.105
UV & Therm	0.888	1	0.669	0.899	0.853
Therm & Vis	0.059	-0.688	0.398	0.741	0.118
All Three	0.992	-1	0.572	0.18	0.164

Table B.10: MCC results for ensemble models, Classification Layer Model 2, no transfer learning.

	Valid Data 1	Valid Data 2	Valid Data 3	Valid Data 4	All
AlexNet					
UV & Vis	1	-1	0.446	-0.204	0.096
UV & Therm	0.919	0.992	0.802	0.874	0.89
Therm & Vis	0.821	0.62	0.871	0.866	0.774
All Three	0.18	0	-0.328	0.356	0.027
ResNet					
UV & Vis	0.836	-0.879	0.942	0.935	0.457
UV & Therm	0.724	-0.46	0.894	0.935	0.55
Therm & Vis	0.858	0.836	0.837	0.992	0.873
All Three	0.564	0	0.718	0.642	0.53
VGGNet					
UV & Vis	0.836	-1	0.452	0.385	0.155
UV & Therm	0.873	0.984	0.934	0.924	0.928
Therm & Vis	0.935	0.492	0.888	0.933	0.795
All Three	0.935	0.843	0.877	0.833	0.868

Vita

Elizabeth Piersall grew up in Fredericksburg, Virginia. She attended Virginia Tech and graduated with a Bachelors of Science in Physics, during which she interned for two summers at Oak Ridge National Laboratory. After working in software development for a few years, she chose to pursue a Ph.D in Data Science and Engineering at the University of Tennessee in the Bredesen Center, during which she was hired by Oak Ridge National Laboratory, joining the Grid Communications and Security group.