Vision-Based Fire Detection

Detecting Fire using a Variety of 2D & 3D Convolutional Neural Networks

Prepared by:
Theodore Psillos
PSLTHE001

Prepared for:
Associate Professor Fred Nicolls
Department of Electrical Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Electrical & Computer Engineering

November 6, 2022

Keywords: Computer Vision, Convolutional Neural Networks, Deep Learning, Fire Detection, Machine Learning

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

November 6, 2022

Theodore Psillos

Date

Terms of Reference

At the start of this honours project, my supervisor and I briefly outlined my expectations for this paper. Professor Nicolls made it clear that I was to investigate the current state of visual fire detection and implement an algorithm(s) that solved one or more of the related computer vision problems within the field. Finally, we agreed that a comprehensive performance evaluation was required. However, due to the nature of the problem to be investigated being so broad, my supervisor gave me some freedom to explore a variety of avenues and to rather use the aforementioned expectations as a guide for this paper.

Acknowledgements

This honours project was a bit of a roller coaster given it was the first one that I have ever written. It did, however, provide me the opportunity to further develop my skills in the machine learning field - a topic that has recently become a growing interest of mine. If there is one thing that I learnt, it is how little we actually know and how much we have to still learn. I feel so incredibly grateful and lucky to have been given this opportunity and hope that I have done it justice.

To my supervisor Associate Professor Fred Nicolls: Thank you for the time you took to listen to my ideas and the guidance that you gave me. It definitely took me some time to get a grasp of what exactly I was wanting to achieve with this honours project so thank you for your patience. The approach you took and advice you gave were invaluable lessons as they not only helped me complete this paper but also developed me as an engineer. I cannot thank you enough.

To my parents: Mom and Dad, only ever wanting the best for me. I will never be able to express in such a short paragraph how grateful I am for your unwavering support through the course of my degree and this honours project. Dad, thank you for reminding me to set high standards for myself and to not settle for anything better. Mom, thank you for the endless hours of listening to me brainstorm and talk about my degree. You both have shown me that hard work and consistency always pays off.

To my sister Alexa: Your willingness to help me despite your busy schedule goes to show how indebted I am to you. Thank you for taking the time to read through this VERY long honours project and correcting my grammatical errors - I know you may not have understood much but I appreciate it.

To those friends I made throughout the course of my degree. I couldn't have chosen a better set of people to share this amazing experience with. Nic, Mill, Davo: The countless projects we worked on, peer-learning and celebrations are unforgettable. I'm glad that we all managed to reach this milestone of finishing in four years.

Abstract

The impact of climate change is notable in the global increase of wild fires, which has lead to the growing need for early fire detection. The current solution of particle-based fire detection systems is flawed due to their high false positive rates, delayed response times and limited application. The technological advancements made in the 21^{st} century have led to the emergence of the vision-based fire detection field, within which a hybrid of deep learning and traditional handcrafted techniques is the most widely applied approach. One of the key issues in this field is the lack of large diverse datasets and unreliable evaluation methods used to measure the performance of video-based interpretations of this approach.

The primary objective of this paper is to develop, evaluate and compare the performance and generalisation of various 2D and 3D CNN models and to provide the best interpretation of the hybrid approach for vision-based fire detection. The models implemented were a 2D CNN, a 3D CNN implemented as a 2D CNN, a 3D CNN with varying depths of 2, 4 and 6, a 3D frame referencing CNN and lastly, a 3D frame differencing CNN. The general structure used for each model is as follows: two sets of convolutional, pooling and batch normalisation layers followed by a flatten, dense, batch normalisation and a dense layer. These models were trained to utilise the spatial and/or temporal information of the limited High Performance Wireless Research and Education Network (HPWREN) dataset, that contains videos of fire scenes in the remote mountainous regions of southern California. Data augmentation was used to increase the dataset size that the k-fold cross-validation technique was used to train the models on. Two unseen videos, separate to the training and testing sets, were used to obtain a reliable evaluation of the performance of each model.

The main performance technique used, focused on the ability of the models to generalise to unseen data. This was separate to the training and the test sets of the cross-validation technique. The confusion matrix and predictions made for each unseen video frame were obtained to aid in the comparison of the models. The top performing model was the 3D CNN with depth 6, which achieved an accuracy of 98.28% and 81.90% for the two unseen videos. The performance of the 3D CNN with depth 2, 3D CNN implemented as a 2D CNN and the 2D CNN was improved by an average of 4.69% when a moving average function was applied to the frames. The frame differencing and reference frame 3D CNN models performed poorly as their interpretations struggled to distinguish features of fire in the presence of a variety of smoke-liking moving objects.

The conclusion drawn, based on the scope of the paper and the datasets available for use in the field of vision-based fire detection, is that the 3D CNN model with depth 6 is the most appropriate interpretation of the hybrid approach for this field. This was supported by this model's accuracy, early detection of fire and low false positive rates. In addition, this model outperformed the rest given its reliable generalisation to unseen data and its ability to perform well using a limited (HPWREN) dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

'The only impossible journey is the one you never begin.'
— *Tony Robbins*

The impact of climate change is far-reaching and enduring. A notable example is the global increase in wild fires, burned areas and frequency of seasons [1]. Early fire detection can be a way to prevent the severity of these fires in order to protect our environment, safeguard our biodiversity and ensure that people's livelihoods are preserved.

1.1 Background

Traditional particle-based fire detection systems, otherwise known as smoke detectors, are widely used throughout the world. Smoke detectors detect fire using the smoke particles produced during a fire's burning process. Once a fire is detected, smoke detectors will raise a fire alarm, which in many cases alerts the relevant authorities [2]. While these devices have saved many lives and prevented damage to people's belongings, they do have notable flaws. Smoke detectors are prone to giving a false alarm. Examples of what often causes false positives are low sensor batteries, close proximity to steam produced through cooking and volatile organic compounds that can often be found in fresh paint [3]. Other shortcomings of smoke detectors include the delayed response time for fire detection, which often occurs when the detector is not in close proximity to the fire and fire detection being limited to small enclosed environments [4]. It has been proven that smoke detectors are ineffective in detecting fire in large enclosed and open spaces. Examples of large enclosed spaces include large airplane hangars or assembly factory spaces, while examples of large open spaces are forests, farms and mountainous regions. As a result of these shortcomings of smoke detectors, this paper will explore alternative and more effective solutions for fire detection.

With the technological advancements made in the 21^{st} century, an alternative solution to smoke detectors was using computer vision to detect fire. A variety of approaches to vision-based fire detection were explored, with many researchers providing their own interpretations of them. An early approach within the field sought to extract fire features using handcrafted techniques with each interpretation of the approach trying to address common issues. Mistakenly classifying clouds or mist as fire due to their smoke-like characteristics is a good example of such an issue. Interpretations also differed based on their application, with some detecting fire using its flame feature, while others used its smoke feature. With more recent development to the memory and computing capabilities of computers, these handcrafted techniques adopted a machine learning approach. This led to deep neural networks

and Convolutional Neural Networks (CNN) becoming the leading interpretations of a new hybrid approach. To date, the success of object classification using CNN's has resulted in its wide use within the vision-based fire detection field due to its ability to extract fire features from image data.

1.2 Objectives

The interpretations of existing fire detection approaches within the field of vision-based fire detection, have various shortcomings. The most notable shortcoming is the lack of a diverse, reliable and large dataset that can be used in the training of deep learning models. With interpretations of traditional approaches becoming outdated and tedious to implement, a new approach for vision-based fire detection, which is a hybrid of traditional approaches and deep learning, has emerged. This paper, the objectives of which are stated below, will investigate and propose a suitable interpretation of the hybrid approach for vision-based fire detection.

1.2.1 Issues to be Investigated

The primary objective of this paper is to develop, evaluate and compare the performance and generalisation of various 2D and 3D CNN models and to provide the best interpretation of the new hybrid approach for vision-based fire detection. These models utilise the spatial and/or temporal information of the limited fire dataset provided by the High Performance Wireless Research and Education Network (HPWREN). Moreover, this paper seeks to investigate the general issue of automated fire detection by exploring the application of the above models in a practical setting. Over and above the primary objectives of this paper, the investigations performed also provide useful context to set the following secondary objectives:

1. To conduct a comprehensive review of the vision-based fire detection field. This is in the form of various researchers' interpretations of common approaches and the current fire datasets available.
2. To provide an overview of the various 2D and 3D CNN model approaches used, along with theoretical background on the key elements of a CNN architecture and the performance metrics used.
3. To compare the performance of the models using traditional evaluation techniques.
4. To analyse and compare the CNN models' ability to generalise to unseen data.
5. To discuss the impact of each model's structure on its generalisation to unseen data.
6. To reach conclusions based on the results obtained and analysis performed.
7. To provide suitable recommendations for future research.

1.2.2 Purpose of the Study

Fire causes significant damage to people, their property and the natural environment. The leading cause of fire is often human error or a system failure, which can result in the loss of life or permanent injuries. As of October 19, 2022, the National Interagency Fire Center (NIFC) reported that a total of 59 547 fires had already burnt 30 000 kilometres-squared of vegetation zones in the United States

of America (USA) [5]. The impact that these remote fires have on animals and the environment is devastating and is due to the lack of detecting techniques and fire services for those regions. In order to avoid such disasters it is important to address these issues by investigating methods that implement early detection of fire with high accuracy.

Traditional vision-based fire detection focuses on fire feature extraction through the use of handcrafted techniques. However, this approach is time-consuming and tedious due to the nature of hand-crafted feature engineering. It struggles to detect the early stages of fire scenes, especially when there are fire-like objects, such as clouds, changing lighting conditions and poor video quality. This led to these handcrafted techniques producing high false positive rates and having a low level of accuracy. Researchers addressed these issues by adopting a hybrid of this approach and deep learning. However, the lack of large diverse datasets has resulted in poorly trained models that fail to fully extract features of fire. This is evident in the various models that use these datasets to classify images of fire as opposed to detecting it. Within this hybrid approach, video-based fire classification has become increasingly popular due to the abundance of temporal information that it can provide. However, the current evaluation methods used are unreliable and fail to test a model's ability to generalise to unseen fire data. This too is an indirect result of the limited dataset sizes available within the field of vision-based fire detection.

1.3 Design of Study

Initial research in the field of vision-based fire detection has produced numerous experiments and solutions for fire detection. Although many experiments were unsuccessful, and their results are therefore not explicitly mentioned in this paper, it is worth noting that they still played a role in the development of the proposed experimental design. This section summarises the approaches that were considered in the experimental stage and the findings that have formed the foundation of the design of this paper.

1.3.1 Original Scope of Study

There are over 2 700 informal settlements in South Africa [6]. Many inhabitants of these closely packed shelter settlements rely on paraffin oil as a fuel source for light, cooking and generating warmth during winter [7]. While paraffin is affordable, it is highly flammable and is the most common cause of roughly 5 000 shack fires per year in South Africa [8]. In light of the severity of the impact of these fires on the low income population of South Africa, the original scope of this paper was to produce a vision-based fire detection solution that implemented the early detection of shack fires. However, a major drawback to this scope was the lack of an image- or video-based shack fire dataset. Although one can find images of shack fires on Google, there is a limited number of shack fire videos, many of which are taken from handheld cameras or news channels. In addition, creating a dataset from publicly available images and videos is known to be a labourious process. In the early stages of this research, the findings regarding the lack of a shack fire dataset resulted in a restructuring of the scope and application of this paper. Constrained by time, a conscious decision was made to broaden the scope of the project to that of vision-based fire detection in a remote mountainous or forest region. Furthermore, in an attempt to ensure the proposed approach produces realistic fire detection, all footage that was handheld

or captured by a close camera has been excluded from the dataset used. This is because a real-life application of fire detection would make use of a stationary camera of remote mountainous or forest regions. Furthermore, stationary cameras are expected to allow for early fire detection as they do not require human intervention unless necessary, for example, if maintenance of the camera is required. As a result of the amended scope of this paper, a new dataset that fit the requirements of the scope needed to be sourced.

1.3.2 Limited Datasets

Similar to the lack of shack fire datasets, the broader vision-based fire detection field also has a limited number of useful datasets. Preliminary investigations revealed a wide variety of image-based fire datasets. However, after careful inspection, many of these datasets were compiled directly from Google images and they contained a wide variety of misclassified fire images, such as cartoon pictures of fire, watermarked fire images and fire clip art. As a result of the unsatisfactory datasets, a number of existing papers and coding based on these datasets focuses on recognising an image of fire as opposed to detecting fire. While these outcomes may sound similar, they are markedly different when compared to the context of this application. These shortcomings of these datasets highlighted the importance of finding a large and diverse dataset that also fitted the scope of this paper.

The spatial information in images is often used by machine learning to classify them. However, due to the lack of large datasets, it became important to extract more than the spatial information of an image. Videos consist of multiple frames (images) that change with time and thus provide temporal information on top of the spatial information of each frame. As a result, the original approach of an image-based method of fire detection moved to that of video-based method. Aside from the large amount of information available in videos, there is also a large amount of publicly available video footage due to technological advancements in recent years increasing the accessibility of cameras and the amount of Closed Circuit Television (CCTV) systems being implemented around the world. Consequently, video-based classification has gained significant traction over the last few years.

A popular deep learning structure used in video classification applications is ResNet, a neural network trained on ImageNet data. This model was implemented during preliminary experiments and produced promising results. However, a further analysis of the dataset used raised questions regarding the reliability of these results. This dataset contains multiple videos captured by the same camera, meaning each video had the same background scene as the other. This resulted in data leakage between the training and testing sets, with frames being more easily classified due to the models exposure to them during training. From this stemmed a common theme of poorly compiled datasets that some researchers used to obtain unreliable results.

1.3.3 Video-Based Classification

An article by Taha Anwar [9] forms the basis of the code that was written for this paper. The original code was greatly enhanced to make it applicable in implementing the video-based fire detection interpretations that are investigated in this paper. A combination of various datasets mentioned in Sections 2.3.2 were used to train a 2-dimensional (2D) CNN. The model focused on attribute identification and was trained to categorise video frames as smoke, fire, clouds or mist. However, the

limited dataset lead to a poorly trained model that struggled to classify frames in the correct category. In light of the limited dataset, narrowing the scope of the paper to classify a fire and no fire scene would provide a more useful interpretation of vision-based fire detection.

1.4 Scope & Limitations

The scope of this investigation is limited to the detection of fire scenes that occur during the day and at a distance from the camera. Furthermore, this paper uses cameras located in remote regions and that are fixed at a positional height that overlooks the mountainous regions of Southern California. The smoke characteristic of fire, along with its spatial and temporal features, is used to detect fire. Considering this, the detection of fire using its flame characteristic, the detection of fire at night time and the detection of fire at a short distances, are out of the scope of this investigation.

The primary limitation of this investigation is the lack of a large, reliable and diverse fire dataset. This limitation is continuously highlighted throughout this paper because an important part of the success of a machine learning model is the dataset that is used to train it. A limited dataset often results in poor optimisation of the model during training, leading to overfitting and poor model generalisation to unseen data. Steps are taken over the course of this paper in the form of data augmentation to help increase the dataset size and minimise this limitation.

Another limitation worth mentioning is that of the technique used to evaluate the performance of models. Often a set of data are split up into training and testing sets, with various methods, such as the k-fold cross-validation technique, being used to prevent data leakage across sets. However, in video-based classification there are often several video frames that are identical to one another. This is relevant to the dataset used in this paper because the background scenes of the fire videos remain unchanged for the most part. To address this limitation, alternative evaluation methods are developed by creating an unseen dataset separate to the train and test datasets. The performance of the models is therefore tested based on their ability to generalise to this unseen data.

1.5 Investigation Outline

The content of this investigation begins with Chapter 2; a literature review of the vision-based fire detection field. This introduces the reader to the various approaches within the field, namely, traditional feature extraction using handcrafted techniques and a hybrid of traditional feature extraction and deep learning. It goes on to highlight the key characteristics of fire, namely, smoke and flame, and to define the static and dynamic features that have been used in various interpretations. A thorough analysis of current fire-based datasets is also provided along with corresponding research that has used them.

Chapter 3 provides an in-depth view into the methodology and implementation of the proposed models of this paper. A summary of the HPWREN video dataset and the decisions that are made to label the dataset, are given. Most importantly, this chapter summarises the functions used to process the dataset correctly and introduces the various CNN models used to detect fire in this paper. This chapter is also supplemented with relevant theory regarding the training and performance metrics that are used to collect results about the models.

Chapter 4 presents the various findings of the tests that were performed on the 2D and 3D CNN models. The chapter begins with a comparison of the performance of each model using the bias and variance metrics. Following this, the CNN models are evaluated on unseen data, which consists of two videos with similar background scenes but different fires. For every model, the predictions made per frame and its confusion matrix are obtained to provide visual and numerical representations of the model's generalisation. These allow for the models to be evaluated based on their false positives, false negatives and accuracy metrics. Following this, a moving average function is applied to asses added value to the performance of the models.

Chapter 5 highlights key elements of Chapter 4 and provides a further analysis of the results. A particular focus is laid on the various models' ability to generalise to the unseen data as comparisons are drawn between the impact that a model's structure has on its performance.

Conclusions are then provided in Chapter 6 with the key findings of this paper being highlighted.

Finally, Chapter 7 puts forward recommendations for further research on this topic. Suggestions are made on ways to improve generalisation of the models to unseen data, as well as expanding the scope of the project to include night scenes.

Chapter 2

Literature Review

'If I have seen further, it is by standing on the shoulders of giants.'
— *Sir Isaac Newton*

2.1 An Overview of Vision Based Fire Detection

Early research on vision-based fire detection focused on the characteristics of flame and smoke to determine which of the two is a better classifier of fire. However, the choice between flame and smoke as classifiers depends on their application. For example, smoke often precedes fire, especially when viewed from afar, however flames are more evident at a closer distance. Thus, a smoke classifier would be more useful in identifying forest fires, whereas a flame classifier would be more successful in identifying fires in an enclosed space. As a result, for each characteristic of fire there are distinct features that researchers have tried to extract and use to identify fire [10]. Traditional interpretations of fire detection approaches focused on extracting multiple features of smoke and/or flame using handcrafted techniques. Due to the technological advancements that have taken place in the last two decades, more recent interpretations have adopted a deep learning approach in addition to the traditional approach [11]. The sections that follow will analyse each of these approaches and their relevant interpretations:

1. traditional feature extraction using handcrafted techniques that apply the static and dynamic characteristics of fire and/or smoke and,
2. a hybrid of traditional feature extraction and deep learning.

2.2 Approach 1: Traditional Feature Extraction Using Handcrafted Techniques

2.2.1 Overview

Flames are produced as a result of the exothermic reaction between oxygen and a fuel [12]. On the other hand, smoke is produced as a result of the incomplete combustion of a material caused by a lack of oxygen [13, 14]. Information about the fire can be obtained from certain features of flame and smoke. The key feature of flame is its colour. Blue coloured flames correspond to a higher burning temperature and orange flames to a lower burning temperature. While a flame's shape and contour can also be used to identify it, a flame is often classified based on its flickering motion. On the contrary, smoke is characterised by its colour, height and density [10]. Similar to flames, the colour of the smoke

can also be used to infer the burning temperature. In terms of height, smoke produced closer to the flame is much darker and often has a higher rate of change compared to smoke that is further away from the flame. The density of the smoke cloud produced can provide information regarding the size of the fire. Therefore, features of flame and smoke are categorised into two classes: static and dynamic.

Static features include the colour, texture and wavelet properties of fire, while dynamic features comprise of the fire's speed, change in motion, direction of motion and flickering motion [11]. Early research focused on using static features of flame and smoke for fire detection, however, these approaches led to high false-alarm rates being produced. Although this led to researchers extracting more features to have additional information to process, it became difficult to combine these characteristics to create useful data. As a result, machine learning was integrated into researcher's interpretations in the form of shallow classifier models, as depicted in Figure 2.1 below.

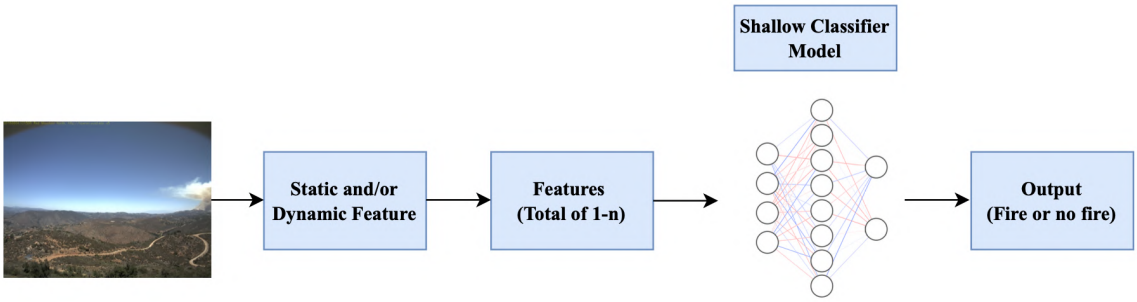


Figure 2.1: An overview of the structure of handcrafted techniques for fire detection

2.2.2 Interpretations

Gómez-Rodríguez et al. [15] use optical flow and wavelet decomposition to detect smoke produced from forest fires. Optical flow techniques are used to approximate the object's motion, surfaces and edges, while wavelets are used to obtain the motion detected at different levels of resolution. This interpretation is successful in extracting features about smoke, such as the height of the smoke column, velocity of displacement, volume of smoke and its inclination angle. However, a major drawback is the high computational cost associated with extracting these features.

Chen et al. [4] utilises both static and dynamic detection techniques for early fire-alarm systems, an interpretation that differs from that of Gómez-Rodríguez et al. This interpretation is based on the assumption that smoke has similar Red Green Blue (RGB) values and that the shade of grey can be better represented using the Hue Saturation Intensity (HSI) colour space. This produces a representation of smoke based on varying intensity components, which Chen et al. uses to create a chromatically-based decision function for static detection. In addition, they focus on the diffusion attribute of smoke, which includes the dynamic features: changeable shape and growth rate. The interpretation of Chen et al. produces a high false alarm rate that was caused by clouds or other grey-coloured shadows. More specifically, the static feature of the colour of smoke is often mistaken for the grey shade of clouds and the dynamic features are mistaken for the shape and movement of clouds. It is therefore evident that colour- and shape-based smoke detection alone is not a reliable interpretation.

Another team of researchers that has applied the interpretation of using the static and dynamic features of smoke is Xu and Xu [16]. Their interpretation is based on extracting features of smoke from greyscale images produced by a stationary camera. While both Xu and Xu and Chen et al. extract growth and disorder features from smoke, Xu and Xu go a step further and also identify the frequent flicker in smoke boundaries and local wavelet energy. Furthermore, they trained a Back Propogation (BP) artificial network to classify images as either smoke or non-smoke. This research is one of the first to introduce the use of neural networks in vision-based fire detection. The interpretation of Xu and Xu is promising due to the associated low computational cost, unlike that of Gómez-Rodríguez et al. Although the results of Xu and Xu are not supported quantitatively, their neural network architecture is informative for the designing of the network used in this paper.

Wang et al. [17] identifies two key characteristics of smoke that help with its classification: the buoyancy that causes smoke to move upward (diffusion feature) and the swaying motion of the smoke in the presence of environmental influences such as wind (swaying feature). To identify the swaying feature, Wang et al. created a swaying identification algorithm that uses the Choquet fuzzy integral to extract dynamic regions of video frames and centroid calculations to determine the point at which an object is most equally distributed. On the other hand, a Grey-Level Co-Occurrence Matrix (GLCM) is used to determine the different combinations of grey pixel levels that were needed for the diffusion feature. Although this approach can differentiate between several types of smoke, it does not translate well to different applications because of the limited range of environment conditions and proximity to the fire on which it was tested. It does, however, provide useful insight into GLCM's.

2.3 Approach 2: Hybrid of Traditional Feature Extraction and Deep Learning

2.3.1 Overview of Deep Learning

There is a variety of deep learning applications such as the detection of objects, audio recognition, anomaly detection and the detection of other such applications with distinct spatial features. The success of Convolutional Neural Networks (CNNs) in machine learning image classification competitions has caused it to gain traction as the best-suited structure for 2-Dimensional (2D) shaped data in object classification [18]. Since its popularisation, many researchers have explored the variety of CNN applications and models. Gonzalez [19] in particular, looked at classifying numerical numbers using a CNN model and explains the workings of the model's architecture. However, with the increased availability of video datasets, CNNs have been restructured to be compatible with 3-Dimensional (3D) shaped data. This means that spatial features of data, as well as temporal features can be learnt over consecutive frames.

One study by Karpathy et al. [20] compares the positive results of image-based CNN models to the performance of a large scale video classification model. The video-based CNN not only extracts the spatial features of a single, static image, but also the temporal evolution of the video. While only modest improvements were found for single-frame models, the spatial-temporal-based models outperformed the spatial CNN models. Another study by Chao et al. [21] looked at an extended application of temporal-action-localisation-based CNNs. Their method not only allows for the robust

classification of the object class but also the start and end time of the object's instance within a video. The importance of temporal information is highlighted by Taha Anwar [9], through his comparison of the actions of standing up from and sitting on a chair. He highlights that these actions can be interchangeably classified as one another since the spatial features of both actions are the same. The only differentiator is the sequence of frames and thus using temporal information would lead to better classification of the video. These papers, as well as some of the research included in this literature review, are initial indicators that a CNN architecture that uses both spatial and temporal information of a video would be an appropriate interpretation for fire detection.

2.3.2 Deep Learning Datasets

The dataset used to train and evaluate a deep learning model directly impacts the ability of the model to generalise to unseen data. Naturally, a diverse and large dataset often results in better performance because the model is exposed to different variations of the same pattern it is trying to identify. Present datasets in this field consist of both images and/or videos, from which spatial and temporal information can be extracted. Fire detection requires a rich dataset due to the numerous characteristics that can be used to classify it. Flame can have a range of colours that vary from blue to red depending on the intensity of the fire, while the colour of smoke varies between black, grey or white depending on the object that is burning. Furthermore, the distance from the camera at which the fire or smoke is identified also plays an integral role.

In light of this, various datasets have been developed and used by academics to train and test their interpretations for fire detection, most of which are publicly available. Despite there being numerous datasets available, there is no standard dataset that is universally used because the datasets differ in type (video or image), dimension of frame, quality (synthesised or real data), application (indoor, outdoor, close up or far away), positioning (fixed or moving) and authenticity (augmented or raw data). The most commonly used datasets for flame and smoke detection are listed below and summarised in the sections that follow:

1. Mivia fire detection dataset,
2. Firenet dataset,
3. Video Smoke Detection dataset,
4. VisiFire dataset,
5. Keimyung University (KMU) dataset,
6. State Key Laboratory of Fire Science (SKLFS) dataset, and
7. High Performance Wireless Research and Education Network (HPWREN) dataset.

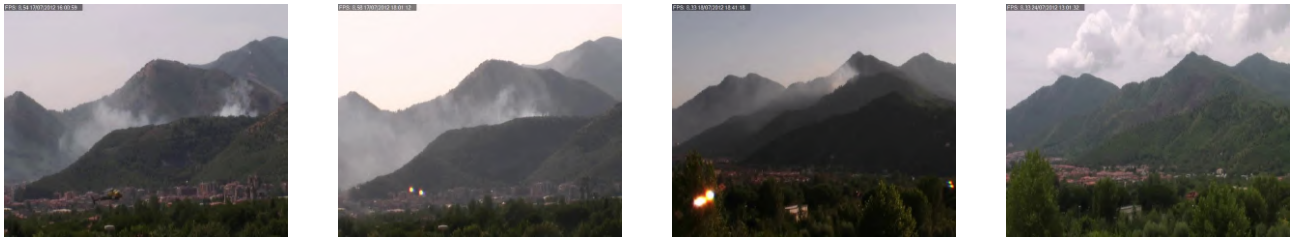
Mivia Fire Detection Dataset

The Mivia Research Lab of the University of Salerno [22, 23] has performed extensive research on flame and smoke detection and produced datasets for both flame and smoke that many researchers such as Muhammad et al. [24] use. The fire dataset consists of 31 videos, with 14 videos characterised

as flame and the remaining 17 videos consisting of general objects or environments with smoke. The smoke dataset, which is significantly larger, contains 149 videos in total. It is split up into various categories such as smoke, mountains, sun, red reflection and other natural environments. Overall, the dataset is predominantly outdoors and consists of raw, authentic data that is of a good quality. The Mivia dataset was used in the initial experiments of this paper, specifically the classification of four attributes.

(a) *Mivia fire 1* (b) *Mivia fire 2* (c) *Mivia fire 3* (d) *Mivia fire 4*

Figure 2.2: Example frames of the Mivia fire dataset.



(a) *Mivia smoke 1* (b) *Mivia smoke 2* (c) *Mivia smoke 3* (d) *Mivia smoke 4*

Figure 2.3: Example frames of the Mivia smoke dataset.

Firenet Dataset

Arpit et al. [25] created the Firenet Dataset as a result of their research on fire detection. It consists of 62 videos and 160 images, with 46 videos being classified as fire and the remaining 16 videos and images classified as no fire. In contrast to the Mivia fire detection dataset, all videos are recorded on a hand held camera and within close proximity of the point of interest. Most videos are not fixed or stationary, unlike the Mivia datasets, and capture fire within controlled environments. As a result, the Firenet dataset may not be useful for applications that seek to create models for the classification of forest fires in an uncontrolled environment. While this dataset is not used in this paper due to differing applications, it still provided insightful information on the types of datasets that previous researchers used.

Video Smoke Detection Dataset

This smoke detection dataset, which was released by Dr F. Yuan [26], is predominately an image dataset. It only consists of six videos split equally between the two categories of smoke and non-smoke. The rest of the dataset consists of different types of images, with some being pre-processed to greyscale and others augmented. The smoke within the videos was produced through controlled experiments, which is common among most fire detection datasets.

VisiFire Dataset

The VisiFire dataset contains flame and smoke video datasets, some of which are the same as the Mivia video dataset. The dataset was compiled by Prof A. Enis Cetin [27] and focuses on four distinct categories, namely, flame, smoke, forest smoke and other. Each category has 13, 21, four and two videos respectively adding up to a total of 40 videos. While the dataset does provide videos of flame and smoke in controlled environments, there are also several raw videos in uncontrolled forest-like environments. This paper uses a combination of certain videos from the VisiFire and Mivia datasets to create a more rich and diverse range of data that were used in the preliminary experiments.

(a) *VisiFire Smoke 1* (b) *VisiFire Smoke 2* (c) *VisiFire Fire 1* (d) *VisiFire Fire 2*

Figure 2.4: Example frames of the VisiFire dataset.

Keimyung University (KMU) Fire and Smoke Dataset

A group of researchers from KMU [28, 29] focused on flame and smoke video datasets that were generated from their own experiments in a controlled environment. This dataset contains some of the same videos as those in the VisiFire dataset and consists of four categories, namely, indoor and outdoor flame, smoke at a close distance, wildfire smoke and smoke- or flame-like objects. In total there are 38 videos. Although the flame videos only consist of small controlled flames and are therefore not useful in certain applications, the wildfire smoke videos are particularly useful in the context of this paper.

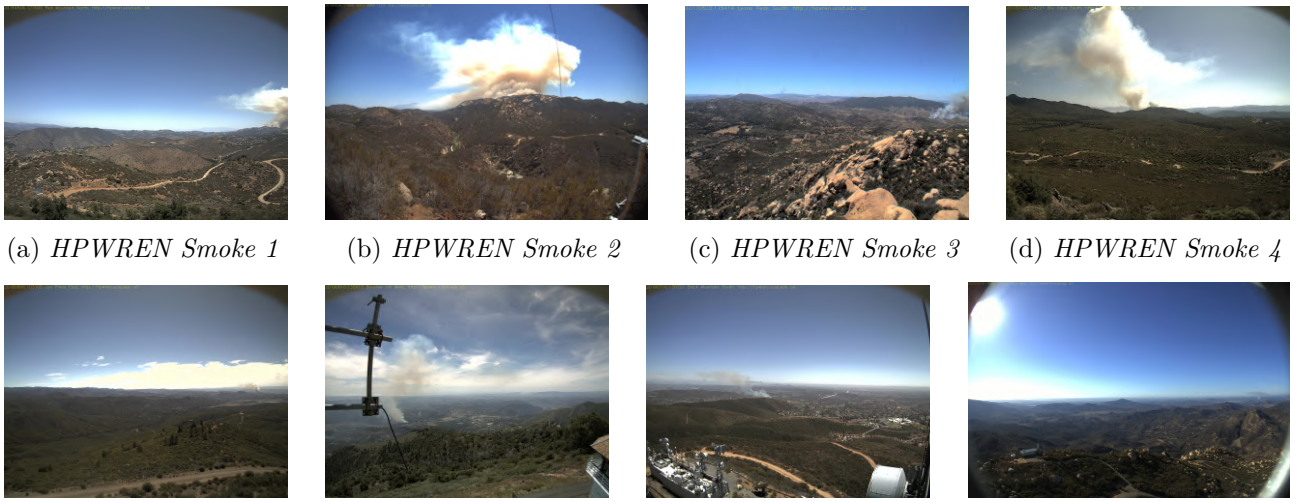
State Key Laboratory of Fire Science (SKLFS) Dataset

The SKLFS [30] dataset is made up of a combination of videos and images. There are 36 104 smoke and non-smoke images, 3 000 synthetic image and video datasets and 3 578 real image and video datasets that on which SKLFS trained their deep learning interpretations. Due to the development of synthetic material, it must be noted that some images in this dataset are more realistic than others. This large dataset allows for valuable training of deep learning models, however, the datasets are not publicly available.

High Performance Wireless Research and Education Network (HPWREN) Dataset

The HPWREN [31, 32] dataset is large, well labelled and consists of 23.72 gigabytes worth of image and video sequences of different fires. They were captured by cameras placed in different geographical landscapes across southern California, with the cameras having a range of light Infra-Red (IR) sensitivity levels and videos consisting of remote scenes before and after the fire. Therefore, this dataset contains diverse data that have been collected from cameras in various fixed positions. The video sequences contain fire scenes that are characterised by the plumes of smoke due to the camera's placement being some distance away. The videos contain many static and dynamic smoke features, for example, differing

colours, density and changes in velocity. In addition, the videos are captured at varying distances from the cameras, for example, close up, in the foreground and near the skyline. It is for these reasons, as well as its suitability to the scope of this paper, that the HPWREN dataset is used in the experimental tests of this project. The Figure 2.5 below display randomly selected frames of certain videos within this dataset.



(a) *HPWREN Smoke 1* (b) *HPWREN Smoke 2* (c) *HPWREN Smoke 3* (d) *HPWREN Smoke 4*

(e) *HPWREN Smoke 5* (f) *HPWREN Smoke 6* (g) *HPWREN Smoke 7* (h) *HPWREN Smoke 8*

Figure 2.5: Example frames of the HPWREN dataset.

2.3.3 Interpretations

As mentioned above, one of the key issues with deep learning is the need for a substantial amount of well labelled data. Aslan et al. [33] attempted to address this issue by creating a robust vision-based detection system using a Deep Convolutional Generative Adversial Network (DCGAN). They focus predominantly on the spatial temporal features, colour and flickering, of fire and grouped video frames to obtain temporal slice images to be processed by the DCGAN structure. In order to make the network more robust, the Stochastic Gradient Descent (SGD) optimiser function is used together with the DCGAN, which is trained on flame data, a noise vector and non-flame data. The major drawback to this interpretation is the lack of utilisation of the temporal features of fire, because Aslan et al. focuses on the flickering motion of flame, which becomes redundant at long distances, thus limiting the effective range of the DCGAN model. Despite this shortcoming, this interpretation requires less memory and has lower computational costs than conventional CNNs. As a result, the methodology of Aslan et al. of grouping video frames is adopted in this paper as it provides valuable temporal information about a set of data.

Gubbi et al. [34] proposed a video-based smoke detection interpretation that uses wavelets and support vector machines. The wavelets are used to decompose the data into different frequency components, which are subsequently used to compare each component with matching resolutions. Following this, vector machines are used to minimise the empirical classification error and maximise the geometric margin. A total of 60 features were extracted and used to train different classifiers, such as K-Nearest Neighbours (KNN) and other non-linear classifiers. Although their method of feature extraction provides the flexibility to use a variety of classifiers, the algorithm itself is slow due to the large feature

dimension that it requires in the vector machine.

The research performed by Muhammad et al. [24] seeks to address this problem by using a computationally efficient CNN to detect and locate fire. This CNN architecture is an adaption from the work of Iandola et al. [35] that proposed a smaller CNN model while still achieving high accuracy. Muhammad et al.'s use of smaller kernel sizes and absence of dense and fully connected layers reduced their CNN model size to three megabytes from 238 megabytes. Their work focuses on the trade off between the accuracy of detection and the efficiency of the network. In addition to being able to localise the fire, the researchers also identified the object that was on fire through segmentation and post-processing. Although the work of Muhammad et al.has high accuracy rates for fire detection, it is based on the dataset mentioned in Section 2.3.2, which consists of experimental fire data. As mentioned above, this limits the model's ability to transpose to real-life and out of control fires. Notwithstanding this limitation, the research of Muhammad et al. research is the most applicable to this paper's experimental implementation because the kernel sizes and order of convolutional layers play an integral role in this paper's interpretation of fire detection.

2.4 Closing Literature Remarks

A significant volume of literature was considered in developing this paper's interpretation of fire detection, with the most notable research being presented. While some research is more useful than others, each played a role in moulding the experiments conducted and the issues that are addressed in this paper. A common issue identified by this literature review is the lack of a reliable and/or diverse fire datasets, as mentioned in Section 2.3.2. As a result, fire detection based on these datasets has led to several interpretations which classify images of fire rather than detecting the fire. Moreover, the application of these interpretations is overfit to the datasets and limits the model's ability to generalise to unseen data. This paper seeks to address the issues related to existing research on fire detection by adopting the successful elements of the above mentioned literature to provide an efficient solution that generalises well on unseen fire data.

Chapter 3

Methodology & Implementation

> 'Take a method and try it. If it fails, admit it frankly, and try another. But by all means, try something.'
>
> — *Franklin D. Roosevelt*

3.1 Methodology Overview

Over the past couple of years, it has become popular to use CNN models for image classification applications. As mentioned in Chapter 1, CNNs are able to identify patterns within images, which are used to identify or classify objects. This paper focuses on implementation of a variety of 2D and 3D CNN models, namely:

1. a 2D CNN,
2. a 3D CNN implemented as a 2D CNN,
3. a 3D CNN with varying depths,
4. a 3D frame referencing CNN,
5. and lastly, a 3D frame differencing CNN

Each model processes the input dataset differently due to the variation in of their CNN architecture. This allows for each model to extract different features and patterns from the same set of data. Although different, the flow of data through each model is inherently the same and can be described succinctly by Figure 3.1 below. Each of the stages displayed in the figure are elaborated in the sections that follow along with the differences in the implementation of models.

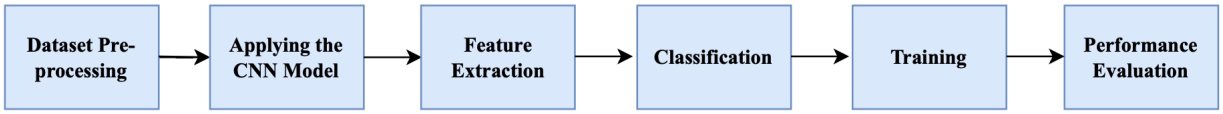


Figure 3.1: An overview of the flow of data for a CNN model.

3.2 Labelling of Dataset

The vision-based fire detection field consists of a small number of diverse datasets, especially those that are applicable to the scope of this paper. Of all the datasets mentioned in Section 2.3.2 of the literature review, the HPWREN dataset is used as it is best suited to the scope of this paper. It is large and diverse, but, the videos are not split into the two categories of fire or no fire. The videos used in this paper were therefore manually split and labelled according to the timestamp of each fire scene. The videos were further split so that the fire dataset would only include the frames where distinct fire characteristics, such as visible smoke.

Initially, the fire and no fire classes contained videos captured from different HPWREN cameras. In other words, no same video sequence would appear in both classes. This limited the total dataset size, leading to the poor performance of the models. Consequently, the videos were equally split up between both classes so that the models were exposed to the fire and no fire scenes of a video sequence. The literature review in Chapter 2 mentions that the HPWREN dataset consists of videos captured by cameras placed all over southern California. Naturally, there are certain cameras that captured several different fires in the same location but the decision was made to only include one fire sequence from each camera so that there are no duplicate no fire scenes in the dataset. The reason behind this decision was to preserve the reliability of the evaluation results for each model. The reliability of results is discussed in greater details in the sections that follow. Example frames of the video sequences used from the HPWREN dataset were shown in Section 2.3.2, with Table 3.1 below provides a summary of the key details regarding the dataset. A full list of all the HPWREN video filenames used in this paper can be found in Appendix A.

Table 3.1: A summary of important dataset metrics.

Video Metric	**Value**
Videos of the Fire & No Fire Classes	44 each
Video Resolution	2048 x 1536 to 3072 x 2048
Average Video Duration	10 seconds
Frames per second (fps)	4 fps
Video Format	mp4

3.3 Hardware & Software Specifications

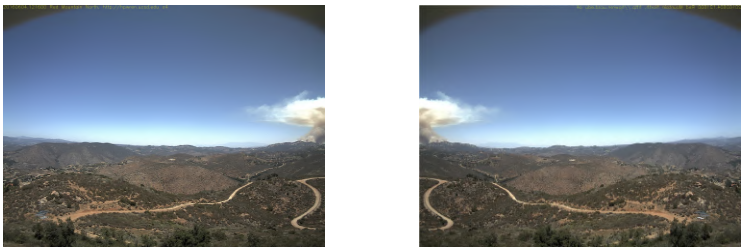
The training of machine learning models often requires significant computation power. Access to more Computer Processing Units (CPUs) and Graphic Processing Units (GPUs) helps speed up the training process of a model. In this paper, a cloud computing product known as Paperspace, was used to implement the experiments and collect results. Paperspace offers access to accelerated computing through high end GPUs and CPUs [36]. The NVIDIA RTX A4000 GPU with 45 gigabytes of RAM, eight CPUs and 16 GPUs was used. TensorFlow is a popular framework that is used for machine learning and artificial intelligence applications. It is particularly popular for its focus on training and deep neural networks [37]. Other than being most familiar with TensorFlow, this framework and the Keras software library were used to implement various 2D and 3D CNN models in Jupyter Notebooks.

3.4 Pre-processing of the Dataset

An important part of machine learning is the pre-processing of the dataset so that the data is uniformly presented to all the models. This processing can range from the manipulation of the data via augmentation and normalization to formatting the data to be compatible with a model's input shape. While a large majority of the processing applied to the frames is the same across all models, there are some differences in the shape of the inputted data. Throughout the data processing, the Open Computer Vision (OpenCV) libraries were used to capture video frames and display predictions on the videos by using its read and write functions. The data pre-processing in this paper is made up of two functions that would extract video frames and create the model's dataset. These functions are elaborated further in the sections that follow.

3.4.1 Function 1 - Frames Extraction

This function extracts each frame from the HPWREN video dataset and applies as set of processing techniques to the frames. Some of these techniques were the resizing of the video frames to a specified height and width, which were 80 x 80 respectively. Additionally, the pixels of the frames were normalised from the Blue Green Red (BGR) scale of 0 to 255 to a range of 0 to 1. This normalisation is key in reducing the complexity of the input that the model is trained on, thereby reducing the computation required to process it. Additionally, data augmentation is applied in the form of a reflection about the y-axis and channel shifting of the image through the addition of a random value to all three channels of the image [38]. This augmentation is important in expanding the size of the dataset to allow for additional training and testing samples. The addition of these two augmentation functions increased the data size by three times with the resulting augmentations visible in Figures 3.2b and 3.3b shown below.



(a) *Original Frame* (b) *Reflected Frame*

Figure 3.2: An example of reflected augmented frames.

(a) *Original Frame* (b) *Channel Shifted Frame*

Figure 3.3: An example of channel shifted augmented frames.

The overall workflow diagram of the frames extraction function and its processes are described by the following workflow diagram shown in Figure 3.4 below.

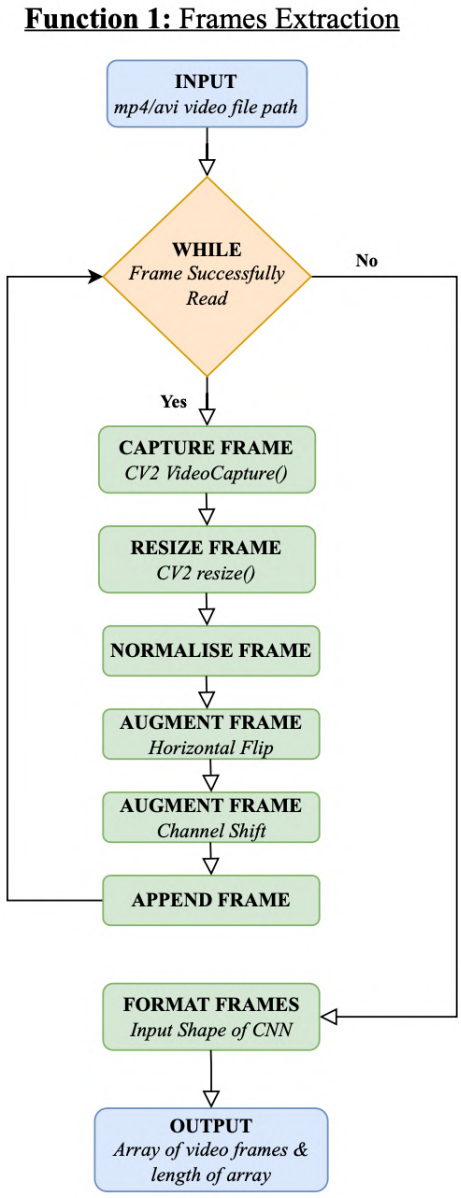


Figure 3.4: The workflow diagram of the frames extraction function.

At the core, all models' frame extraction function follows the above workflow diagram, but they difference comes in where each model incorporates an additional step to format the shape of the video frames to correspond with the input shape of the CNN model. The frames. extraction function used for the 2D CNN model is identical to that of the function shown in Figure 3.4 above and so it does not need to be elaborated further. However, the data shaping and formatting for the other models is explained in more detail in the following sections.

2D & 3D CNN

The key difference between a 2D CNN and a 3D CNN is the increased dimensionality. The input shape for a 2D CNN corresponds to the normalised image height, image width and colour channels (BGR) and is more easily understood by referring to Equation 3.1 below.

$$Input\_Image = image\_height \times image\_width \times image\_channels \tag{3.1}$$

On the other hand, the input shape for a 3D CNN corresponds to the normalised image height, image width, image depth and colour channel. This third dimension is often referred to as different names, but for this paper image depth will be used. The input size for a 3D CNN is better understood when referring to Equation 3.2 below.

$$Input\_Image = image\_height \times image\_width \times image\_depth \times image\_channels \tag{3.2}$$

For all models, the image height and width remain unchanged and are 80 x 80 respectively, as resized in the frames extraction function. Furthermore, image channel refers to the number of different colours that make up the image. In this case, the CV2 function reads in a video frame as a BGR and so for all models, the image channel has a length of three.

As the CV2 function extracts video frames as 2D images, a 3D image can be created by appending the BGR values of corresponding pixels in multiple frames to an array. The depth is then determined by the number of frames that are appended to this array. For example, the BGR values of the pixel at position (0,0) of frame one are appended to a new array along with the BGR values of the pixel at position (0,0) of frame two. Hence, for a depth of four, the first four frames, namely, frames one, two, three and four, are combined to produce the first 3D input. The second 3D input is produced by following the same process for frames two, three, four and five. In light of this, the window slider shown in Figure 3.5 can be used to highlight how the frames are joined to create many 3D inputs.

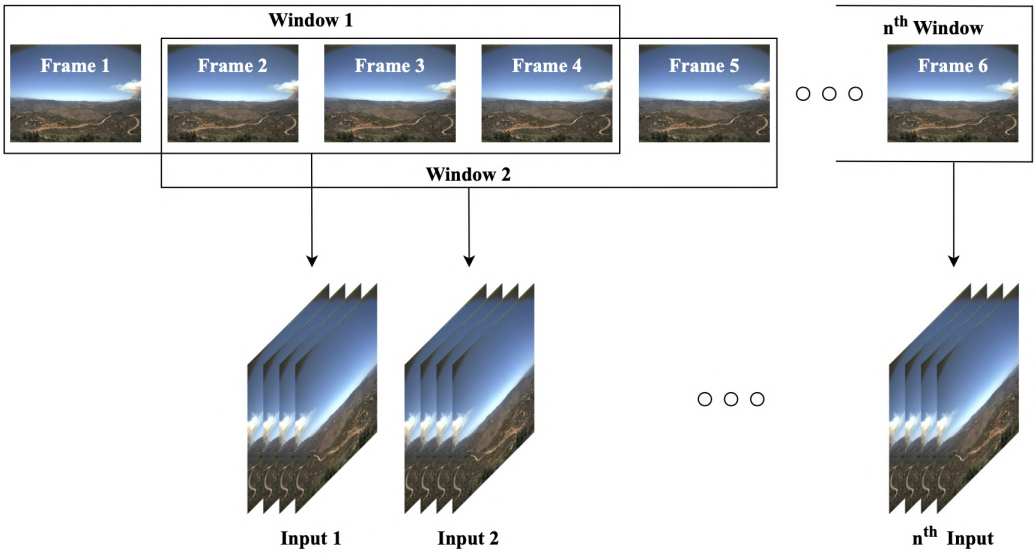


Figure 3.5: Creating a 3D image of depth 4 for the 3D CNN.

The total number of 3D inputs that can be made from a sequence of 2D frames can be calculated using the total number of frames of a video and image depth as shown in Equation 3.3 below. This method and equation mentioned works for differing image depths that are equal to or greater than one.

$$Total\_3D\_Inputs = TotalFrames - ImageDepth - 1 \tag{3.3}$$

3D Frame Referencing & Differencing CNNs

Frame referencing is a technique that joins the reference frame (frame zero) with all the frames that follow. In other words, if there is a total of n frames, frame $n = 0$ is appended to all frames $n > 0$. It is important to note that the reference frame is defined as frame zero of the no fire class and is combined with the rest of the frames from the no fire class as well as all frames from the fire class. The frame referencing method can be better visualised in Figure 3.6 below.

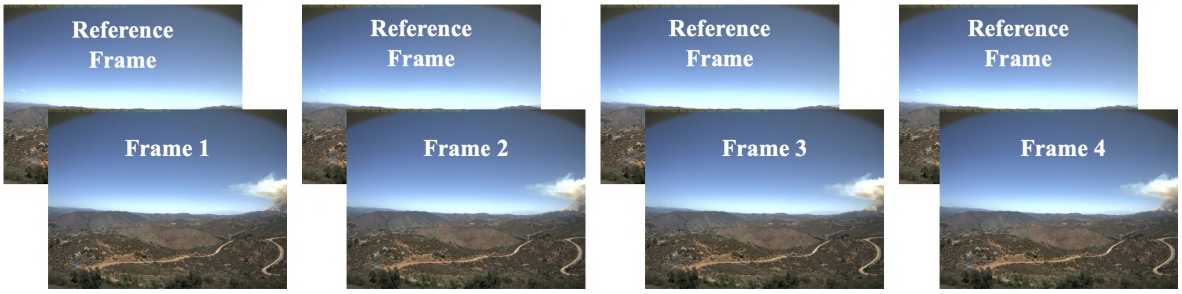


Figure 3.6: Creating a 3D image of depth 2 for the 3D frame referencing CNN.

Considering the above, the frames extraction function has an additional input, which the reference frame (from the no fire class) can be used when formatting the frames of the fire class. Finally, frame differencing adopts the same approach as the one that the frame referencing model follows. The only difference is that instead of appending the the reference frame to all future frames, the difference between a frame and the reference frame is taken. This produces a 3D input shape with an image depth of one and can be better understood when referring to Equation 3.4 below.

$$FrameDifferencing = n^{th}\_Frame - ReferenceFrame \tag{3.4}$$

Equation 3.4 shown above is applied to every corresponding pixel of the current frame and reference frame. The idea behind this approach is that all background pixels similar or equal to those of the reference frame are subtracted to zero meaning that all remaining positive non-zero pixels correspond to pixels of fire. In theory, this significantly simplifies the complexity of the inputted shape and should make the detection of fire easier.

3.4.2 Function 2 - Create Dataset

The second function that is used to create the dataset iterates through the various class directories (in this case, fire and no fire) and calls the frames extraction function for each video within each class directory. Once all frames of a class's videos have been extracted, a maximum number of frames are randomly sampled. This is crucial in creating a diverse dataset that does not contain many identical and consecutive video frames. This random selection of frames also shuffles the order of the frames, decreasing the likelihood of contamination occurring between training and test sets. Following this, a set number of labels equal to the total number of frames sampled are appended to an array. This function can be better visualised through its workflow diagram that is shown in the Figure 3.7 below.

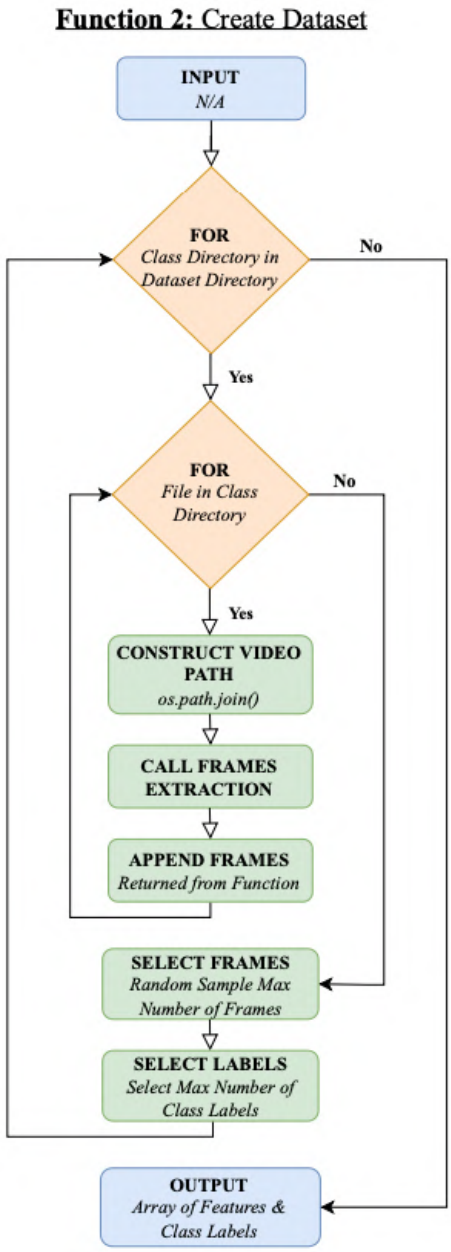


Figure 3.7: The workflow diagram of the create dataset function.

3.5 Applying the CNN Model

The various 2D and 3D CNN models were tested on the same dataset that was extracted using the above functions. For experimental sake, the core structure of these models, such as the number of convolutional layers, pooling layers, filter sizes and batch normalisation locations were kept the same. The difference between each model is its approach and formatting of the shape of the input data. It must be noted that batch normalisation was applied after the first max pooling layer and the first dense layer for each model. The general CNN architecture of each model is shown in Figure 3.8 on the page that follows.

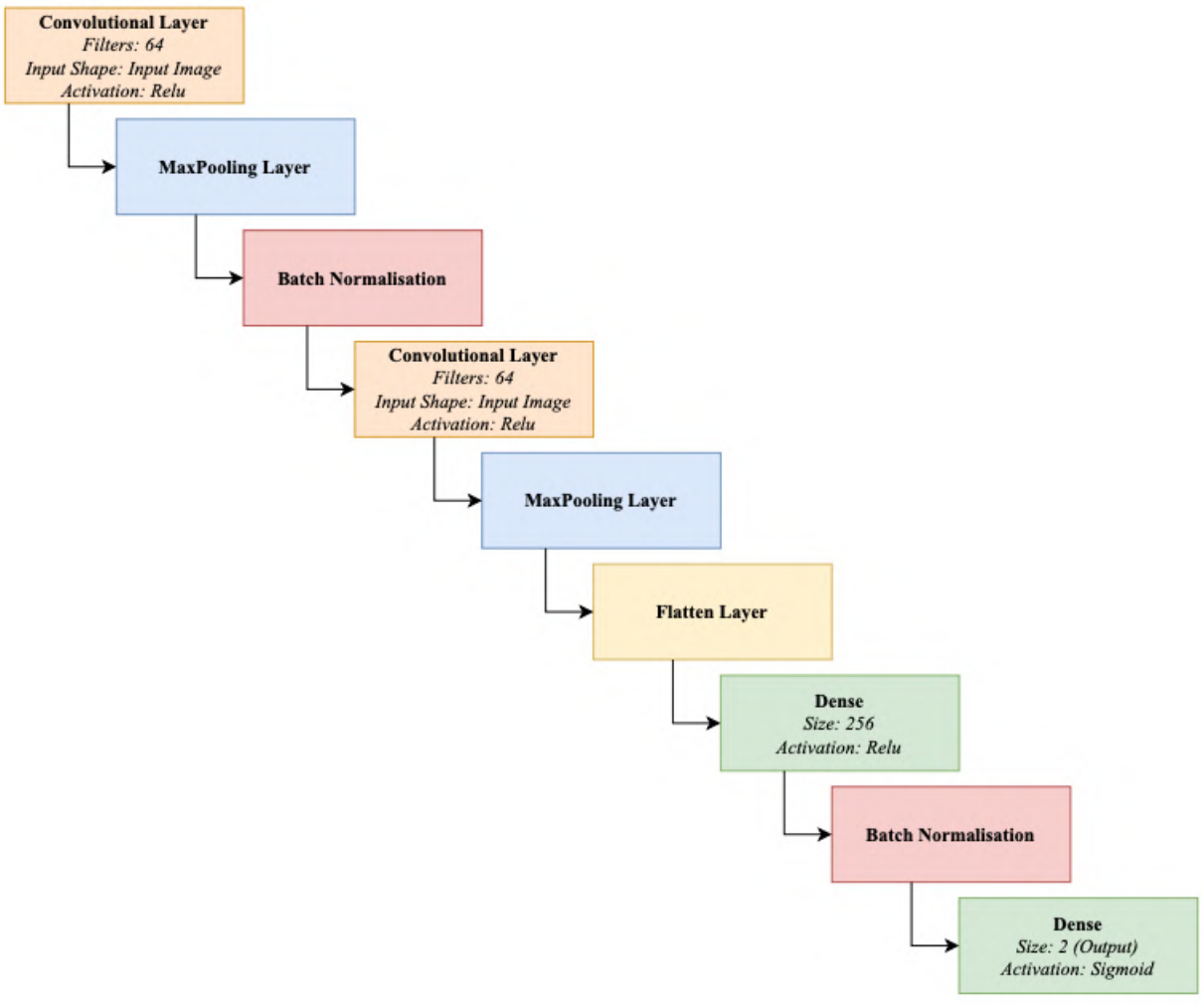


Figure 3.8: The general structure of the CNN architecture used.

3.5.1 General Structure of the CNN Models

The success of CNNs can be attributed to their hierarchical structure and the CNNs' ability to learn strong features from raw input data quickly [24]. The typical CNN architecture contains four key well-known layers.

1. The *convolutional layer*, produces different feature maps after a specified kernel is applied to the input data.
2. The *pooling layer*, preserves the maximum activations of small regions of the previous layer's feature map, while discarding lower less activated pixels. It reduces the number of parameters, and thus dimensionality of the network and ensures translation invarience between layers.
3. *Batch normalisation* re-normalises the activations or weights at the output of a layer to the range of 0 to 1. It does so to prevent any large outliers that could affect the training of the model.
4. The *fully connected layer* often consists of dense and flatten layers. At this stage of the model, a global representation of the model is constructed using high level information of the input data. This layer is often the last and so naturally follows after multiple convolutional and pooling layers.

The *activation* function is used to bind all these layers together by defining how the resulting sum of the weights of a layer are transformed to an output. There are a variety of linear and non-linear activations that can be used but the two most popular activations used in this paper are as follows:

1. The *ReLu Function*, is currently the most used activation function for CNNs. When the input is less than zero the output becomes zero and if the input is greater than zero the input becomes the output [39].
2. The *Sigmoid Function* activation function, takes a real valued input and outputs a value in the range of 0 to 1. This activation is particularly useful when predicating the probability of an output [40].

Finally, the *padding* of a layer is often used when the output and input feature maps need to remain the same dimension size. Several layers of the 3D models in this paper required padding due to the small depth dimension that were used.

2D CNN

The general structure and architecture of the various 2D and 3D CNN models used in this paper can be better visualised in the figures that follow. The batch normalisations that are applied after the first pooling and dense layers are not shown in the model figures that follow. The layer sizes of each model are labelled as well as can be seen in the 2D CNN model displayed in Figure 3.9 on the following page.

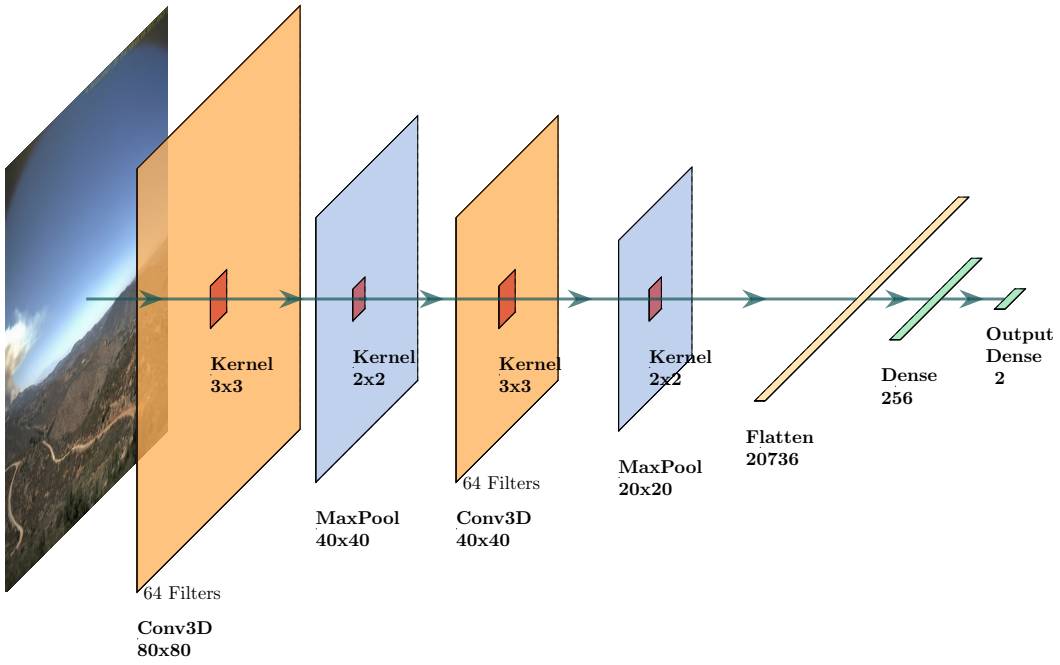


Figure 3.9: The summary diagram of the structure for a 2D CNN.

It is important to take note of the varying kernel and filter sizes that are shown in Figure 3.9 above. For each convolutional layer there were a total of 64 filters with a kernel size of 3x3, while the pooling layers had kernel sizes of 2x2.

3D CNN

Unlike the 2D CNN, the 3D CNN seeks to utilise the temporal information of the data dimensionality, namely the image's depth. Naturally, the kernels that iterate of the pixels of the inputs in the convolutional and pooling layers are also 3-dimensional. In this paper three variations of 3D CNNs were tested with image depths of two, four and six. The kernel sizes of the convolutional and pooling layers for each model are listed in the Table A.1 below.

Table 3.2: A summary of the various kernel sizes used by each of the 3D CNN models with varying depth.

Model Name	**Convolutional Layer Kernel Size**	**Pooling Layer Kernel Size**
3D CNN with Depth 2	3x3x2	2x2x2
3D CNN with Depth 4	3x3x2	2x2x2
3D CNN with Depth 6	3x3x3	2x2x3

The 3D CNN model with depth 4 is shown in Figure 3.10 on the following page.

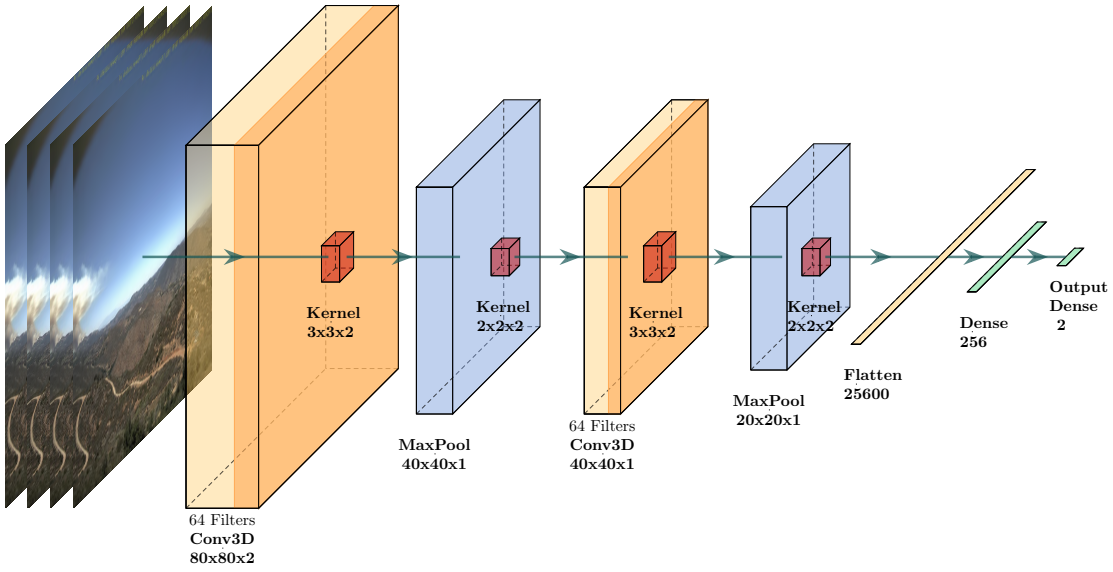


Figure 3.10: The summary diagram of the structure for a 3D CNN.

2D CNN using a 3D CNN

For comparisons sake, a 2D CNN can be created using a 3D CNN model with a set image depth of one. This implies that the convolutional and pooling kernel sizes are thus 3x3x1 and 2x2x1 respectively. This model is displayed in Figure 3.11 shown below.

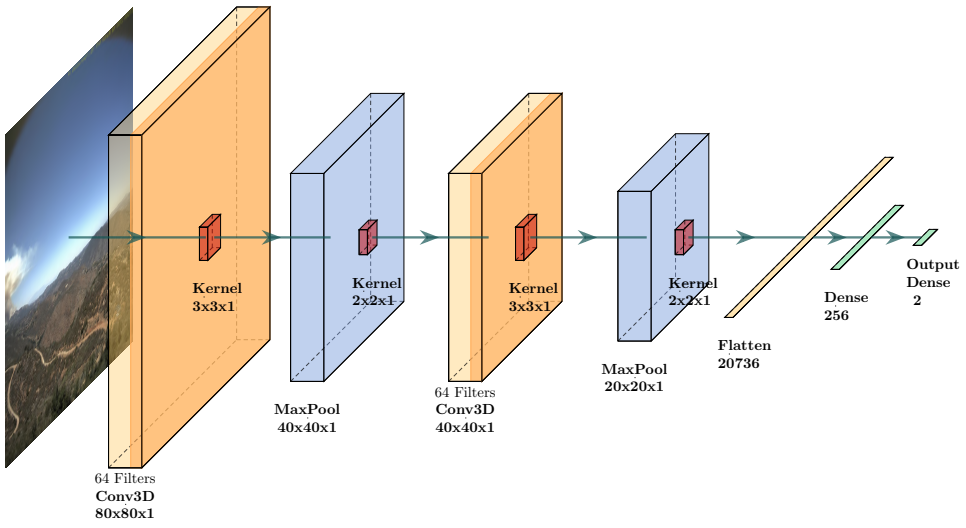


Figure 3.11: The summary diagram of the structure for a 2D CNN using a 3D CNN.

The model displayed above has the same structure as that of the 3D Frame Differencing CNN. As described in the previous pages, the resulting image after taking the difference between the current and reference frames is a 3D image also with depth one.

3D Reference Frame CNN

The frame referencing model is a 3D CNN model implemented with an image depth of two. The structure of this model can be visualised in Figure 3.12 below.

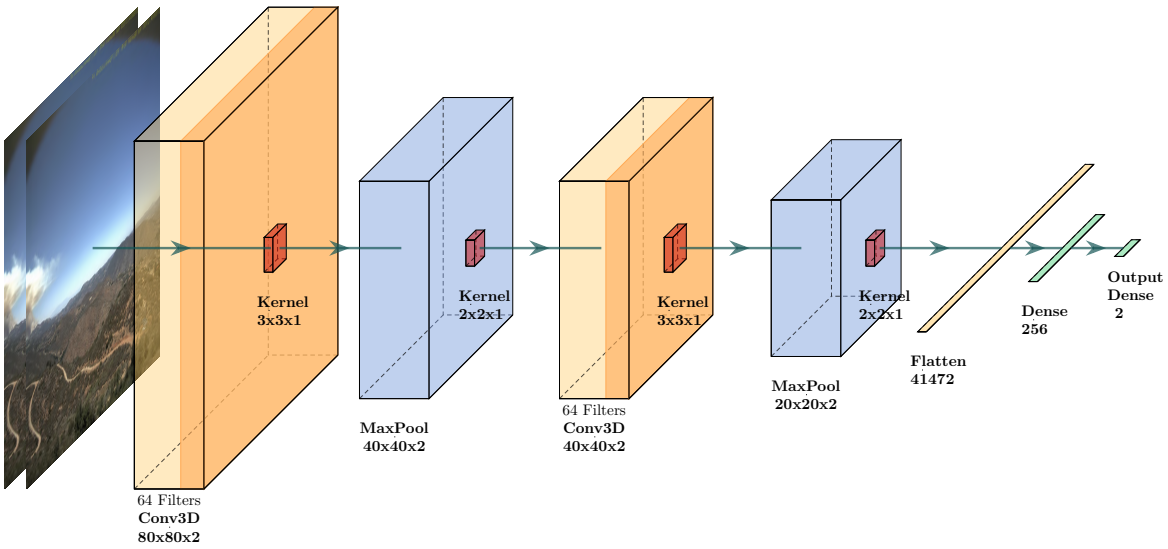


Figure 3.12: The summary diagram of the structure for a 3D CNN using frame referencing.

3.6 Training & Validation

The training of a model helps it learn to identify patterns within objects. There are a myriad of different training techniques as well as ways to split the dataset up into training and testing sets. Considering this, the training of all the models displayed over the past few pages have a data split of 80% and 20%, which is allocated for training and testing respectfully. There was roughly an average of 7 000 frames used for each model that was used to train and test it. The total number of frames used for each model varied as each model often required different input shape sizes and combination of frames. The total number of frames used to train and test each model are displayed in Table 3.3 that is shown below.

Table 3.3: A description of the total number of frames used for each model's training and testing.

Model Name	Total No. of Frames	Total Training	Total Testing
2D CNN	7 500	6 000	1 500
2D CNN using a 3D CNN	7 400	5 920	1 480
3D CNN with Depth 2	7 400	5 920	1 480
3D CNN with Depth 4	6 870	5 496	1 374
3D CNN with Depth 6	6 000	4 800	1 200
3D Frame Differencing CNN	6 600	5 280	1 320
3D Reference Frame CNN	7 380	5 904	1 476

3.6.1 Training & Validation Metrics

Training metrics are important, as they provide information regarding the models performance as it iterates over various batches of the dataset. Some of the more fundamental metrics that were used during training are described below.

1. The *Loss*, is a penalty that is awarded for a bad prediction and indicates how poorly the model predicted a specific batch.
2. The *Accuracy*, is the number of correct predictions made during training.
3. The *Bias*, is the amount that a model's prediction differs to the actual/target value.
4. The *Variance*, is the amount of variability in a model's prediction when compared to predictions made on different training splits.
5. The *Recall*, is similar to the accuracy as it is a measure of the model's ability to detect positive samples.
6. The *Precision*, is the quality of the positive predictions that a model makes.
7. The *F1_score*, is a combination of the *Recall* and *Precision* metrics. It often provides information on the number of false positives and false negatives made.

Another important aspect of training are the loss and optimizer functions used to compile the models. The *binary cross-entropy* loss function and *Adam* optimizer are used for the training of the models. Like its name, the *binary cross-entropy* loss function compares the predicted output to the actual class it belongs to (either 0 or 1) and calculates a score to penalise the model based off the distance away from the expected value. On the other hand, the *Adam* optimizer has an adaptive learning rate that is applied to each weight of the models' neural network and uses default learning rate of 0.001.

3.6.2 K-Fold Cross-Validation

Cross validation is a technique that uses the resampling of data to train and evaluate machine learning models when there is a limited data available. Considering that a diverse and large dataset is one of the underlying issues of the vision-based fire detection field, this method was most suitable for the training of the 2D and 3D CNN models. A total of six folds are made for each model. In other words, the dataset is split into six sub sets of data. Prior to the splitting of data, the dataset is shuffled thoroughly using in-built functions. The cross validation technique is particularly useful as it prevents data leakage between the training and validation sets, which leads to more reliable results. For all the models, a fixed *batch size* of 32 was used, followed with a total of 50 *epochs*. The *batch size* is the total size which the dataset is split up into, while an *epoch* is a full iteration of all the batches. The validation split used for the cross-validation method was 20% and was used to evaluate the model at the end of each *epoch*.

3.6.3 Box & Whiskers Chart

A box-and-whiskers chart is used to compare the performance of the various models using the k-fold cross-evaluation technique. The charts that are produced allows for one to draw conclusions to the bias and variance of certain models. The chart provides insight into the distribution of data points across a selected measure. The ranges of performance metrics such as the median, mode and outlier are displayed on the chart and are useful when determining the best performing model.

3.7 Performance Evaluation & Optimisation

An integral part to the development of models during the training process is their performance evaluation. A variety of techniques were used to validate the models' performance after each fold of the cross-validation method. With each fold, the model was saved along with important performance metrics, namely, accuracy, loss, precision, recall and f1_score. Once trained, each model was evaluated on the test set, allowing for training and validation curves to be plotted. The training metrics were calculated after each batch and so a box-and-whiskers chart could be produced for each epoch. These charts are particularly useful, as they provide insight into the variability of an epochs training and if there were any outliers. On the other hand, the validation metrics, which are determined at the end of each epoch, were plotted on the same graph to provide insight as to whether the model was over-fitting to training data.

3.7.1 Alternative Evaluation Methods

Although it will be discussed in more detail in the results section, one of the issues with video-based classification is the indirect cross-contamination of data between the training and test sets. As videos are split up into several consecutive frames, the probability of frames from the same video being used in both testing and training sets is quite high especially for videos with high frame rates. Significant steps were taken to prevent this, such as the random sampling of the dataset to a fixed amount, the shuffling of data and the use of the k-fold cross-validation technique. That being said, an alternative evaluation method that guarantees the reliable testing of the models is still required.

One of the key objectives of this thesis, is to test the models' ability to generalise to unseen data. Considering this, the models were used to predict the frames of unseen videos that were excluded from the training, validation and test sets. The unseen videos were captured by a HPWREN camera that the model had already been exposed to in training. The unseen aspect of the video is the fire sequence, which the model has not been exposed to at all. This tests the models' ability to detect a new fire within a region in southern California that it is already familiar with. This is also known as the models' ability to generalise to unseen data of fire and this test provides a measurement of how accurately it can predict the video. The accuracy of these unseen videos can be plotted as the predictions made per frame as it provides insight into the model's confidence in its classification of each frame. The *youtube-dl* and *moviepy* libraries were used to download the two unseen videos from my YouTube channel to perform these predictions. The frames extraction method was also used when making a prediction for each frame of the video. The link to the YouTube channel can be found in Appendix C. Table 3.4 on the following page summaries the key details of the unseen video data used.

Table 3.4: Details of unseen videos 1 & 2 used in generalisation tests.

Video Name	Caption	Resolution	Frames	Frame Rate
20171207-FIRE-smer-tcs8-mobo-c	Unseen Video 1	3072 × 2048	120	4
20170711-FIRE-sm-n-mobo-c	Unseen Video 2	2048 × 1536	116	4

Examples of the frames of the two unseen videos used to test the generalisation of the models are shown in Figures 3.13 and 3.14. It is important to note the reason behind the selection of these two videos. The first video contains smoke-like objects in the form of clouds. This unseen video tests the ability of the models to detect fire in the presence of clouds. Furthermore, the split of no fire and fire frames in unseen video 1 is equal, unlike that of unseen video 2 where the majority of the video consists of no fire frames. Unseen video 2 also contains a cloud like object but, for the most part, the sky is clear unlike video 2. Details regarding which frames of each video correspond to fire and which correspond to no fire are provided in the results chapter, namely Chapter 4.

(a) *Sample 1* (b) *Sample 2* (c) *Sample 3* (d) *Sample 4*

Figure 3.13: Sample frames of unseen video 1.

(a) *Sample 1* (b) *Sample 2* (c) *Sample 3* (d) *Sample 4*

Figure 3.14: Sample frames of unseen video 2.

3.7.2 Confusion Matrix

A confusion matrix also provides insightful details regarding the performance of a model. The metrics it provides, namely, the true and false positive and negative rates, compliment the frame prediction graphs mentioned in the previous section. They give a numerical value to the performance of a model on the unseen videos, while the frame prediction graphs are insightful due to the temporal information it provides surrounding a model's predictions. In this paper, the positive case is considered as fire while the negative case is considered as no fire. Given this, a false positive case is when a no fire frame

is incorrectly classified as a fire frame. On the other hand, a false negative case is when a fire frame is incorrectly classified as a no fire frame. The accuracy of a model's prediction for a video can also be determined using the confusion matrix. An example of the confusion matrix used in this paper is provided in Figure 3.15 below.

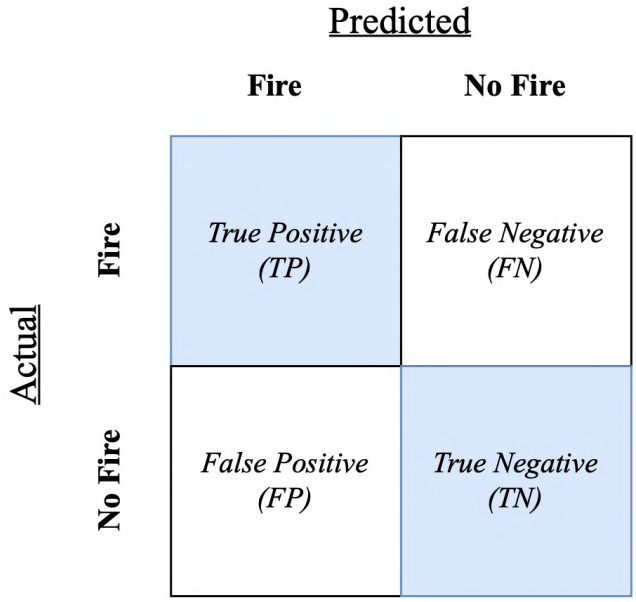


Figure 3.15: An example of the confusion matrix used in this thesis.

3.7.3 Moving Average Function

An optimisation technique often used in the field of vision-based fire detection is that of the moving average function. While traditional approaches focus on single frame classification, moving average takes into consideration a predefined set number of frames. It is most applicable in situations were there is variability in the predictions in the form of outliers. A window length specifies the number of frames that the moving average function takes into consideration when the models make predictions. For a particular frame, its prediction is determined by the maximum predicted class from the average of the predictions of the previous frames. The function is easy to implement and is used to optimise various of the models' performance as will be seen later in the Chapter 4.

3.8 Collection of Results

Prior to various results being collected, model optimisations were made. This was in the form of testing different model architectures, introducing regularizers, using different activations, filter sizes of convolutional layers and changing the batch and epoch sizes. Once these tests were completed, the results of the performance evaluation methods mentioned above were obtained. The only model that was tested multiple times was the 3D CNN as the depth of the 3D input image was varied from two to four to six. In light of this the following results were obtained for each model:

1. The *training* and *validation* performance metrics for each fold.

2. The *testing* evaluation metrics for each fold.
3. The *box & whiskers* chart of each model.
4. The *frames prediction* of the unseen video outside of the dataset.
5. The *confusion matrix*.
6. The percentage optimisation after using the *moving average function*

3.8.1 Code

As mentioned in Chapter 2, the code that was used in this paper was based on a video-based classification example written by Taha Anwar [9]. However, it must be mentioned that a large portion of the code was enhanced/adjusted to suit the scope of this paper. The code used in this experiment can be found on my GitHub repository that is linked in Appendix B, while the code written by Taha Anwar is available at the following citation [9]. Some of the fundamental additions to the code are listed below:

1. A variety of 2D and 3D CNN models were implemented with different structures. This included the use of different optimisers, activation and loss functions.
2. Naturally, the dataset was pre-processed differently to create a set of labels and features that were of the right dimensions for the models above.
3. Additionally, performance metrics, such as precision, recall and f1_score, were obtained during the training and testing of the models.
4. A 6-fold cross-validation evaluation was performed as opposed to a once-off training and testing evaluation.
5. A particular loss history function was defined to obtain training and validation plots per batch as well as per epoch.
6. Predictions made per frame were plotted along with the confusion matrix of each unseen video tested.
7. The moving function used was adjusted to better suit the scope of this paper.
8. Finally, box-and-whisker charts of each model's performance were added so that a comparison could be in the results section that follows.

Chapter 4

Results

'If at first you don't succeed, try, try again'
— *William Edward Hickson*

4.1 Overview of Results

The results of the experiments performed on each of the five models are presented in this chapter. The results will be presented in the following order:

1. 2D CNN
2. 2D CNN using a 3D CNN
3. 3D CNN with varying depths
4. 3D Reference Frame CNN
5. 3D Frame Differencing CNN

Due to the variety of 2D and 3D models, a large volume of results was produced, however, only those most applicable are presented in the sections that follow. The experiments performed were driven by the problems defined in the introduction in Chapter 1. The models were evaluated quantitatively using the k-fold cross-validation evaluation technique and based on their ability to generalise to unseen data. The models are compared based on the confusion matrix, accuracy, false positive and negatives rates they produce.

4.2 Comparison of Performance Evaluation of each Model

The HPWREN dataset as defined in Appendix A was used in the performance evaluation of each model. The results of each model's performance are best expressed using a box and whisker plot, since it visually displays information about the median, lower and upper quartiles, as well as any potential outliers. Furthermore, it provides insight into the spread of the performance, but more importantly the bias and variance of a model's performance.

4.2.1 Loss and Accuracy Performance Comparison

The box and whisker plot of each model is shown in Figure 4.1. It is important to note that the scale of the y-axis is set between 70% and 100% so that the results of the models can be viewed more easily. Furthermore, when comparing the results of the models, models with accuracies in the range of 90% and above will be referred to as having a low bias and those with accuracies lower than 85% will be referred to as having a high bias. All models show reasonably low bias, with the the 3D CNN with Depth 6 being the lowest performing with an accuracy of 79%. On the other hand, the 3D Frame Differencing CNN was the highest performing model and had an accuracy of 94%.

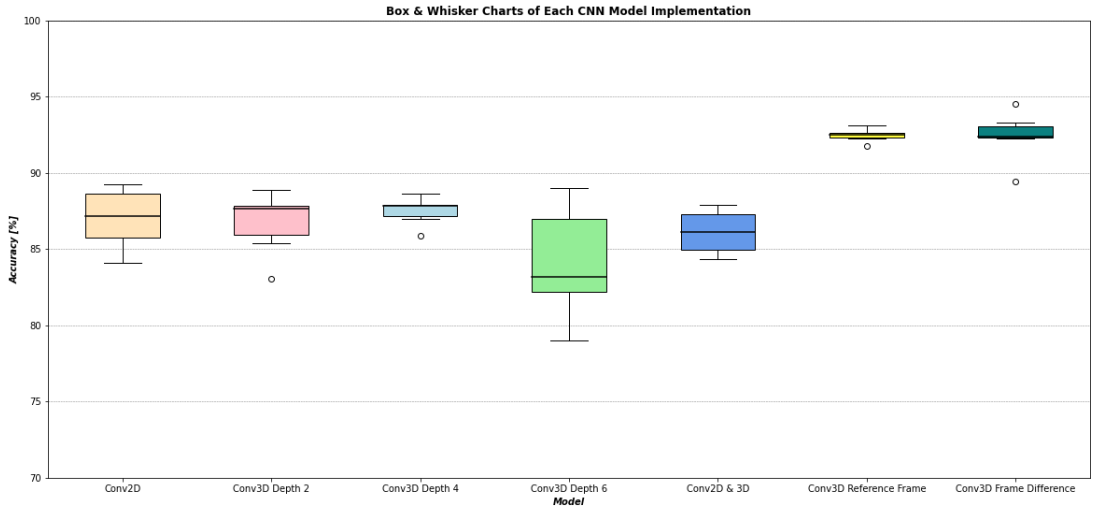


Figure 4.1: A comparison of each models' accuracy.

Conv2D

The accuracy results of the 2D CNN model in the box and whisker plot indicate signs of medium bias (neither low bias nor high bias) because the accuracy values are mostly the same as three other models, namely, Conv3D Depth 2, Conv3D Depth 4 and Conv2D & 3D. On the other hand, it also shows signs of high variance due to the large spread of its accuracy values. It has the second highest variance out of all the models. Therefore, this model is considered slightly underfit to the dataset used.

Conv3D Depth 2

The 3D CNN model with Depth 2, has accuracy values that are similar to the Conv2D model, as well as the other models, and thus indicates medium variance. It shows slightly less variance than the Conv2D model, however, it has an outlier with an accuracy of 83%. In light of this, the model can be considered to be fairly fit to the dataset used.

Conv3D Depth 4

One of the less invariant models, the 3D CNN model with Depth 4, exhibits low variance, which is the third best out of all models shown in Figure 4.1. Similar to the Conv3D Depth 2 model, this model also has an outlier with an accuracy of 86%. On the contrary, the Conv3D Depth 4 model has accuracy

values in the same range as the Conv2D and Conv3D Depth 2 models and thus has a medium bias. It therefore be concluded that the Conv3D Depth 4 model has neither overfitted or underfitted to the dataset used.

Conv3D Depth 6

The lowest performing model based on these evaluation metrics was the 3D CNN model with Depth 6. This model shows instances where the metrics are on par with the metrics of three models mentioned above, however, it is hampered by a larger variance in its results. Although it does not have any outliers, the variance indicates underfitting to the training data, which explains its high bias.

Conv2D & 3D

This model which is similar to the Conv2D model, which means it has a similar performance. Although it has slightly less variance than the Conv2D model, it exhibits a medium level variance and bias. Out of all the models, it has the third worst accuracy which indicates that this model has been underfit to the training data.

Conv3D Reference Frame

The highest performing model was the 3D Reference Frame CNN. The accuracy values of this model are high in comparison to the other models, which indicates a low level of bias. The spread of the accuracy values is low which is an indication of low variance. The model appears to be well-fit to the training dataset that it still has some capacity to learn but not so much that it overfits.

Conv3D Frame Difference

The highest scoring model in terms of accuracy was the 3D Frame Differencing CNN model as it had an accuracy of 94.5%. Much like the Conv3D Reference Frame, the accuracy values of the model are high when compared to the other models. Despite this, there are two outliers (the largest number of outliers out of all models) with each outlier on either end of the model's spread. Considering this, it does not have as low a variance as the Conv3D Reference Frame model, but it is still outperforms most of the other models. The descriptions above regarding each model's bias and variance are neatly summarised by the bullseye plot shown in Figure 4.2 below.

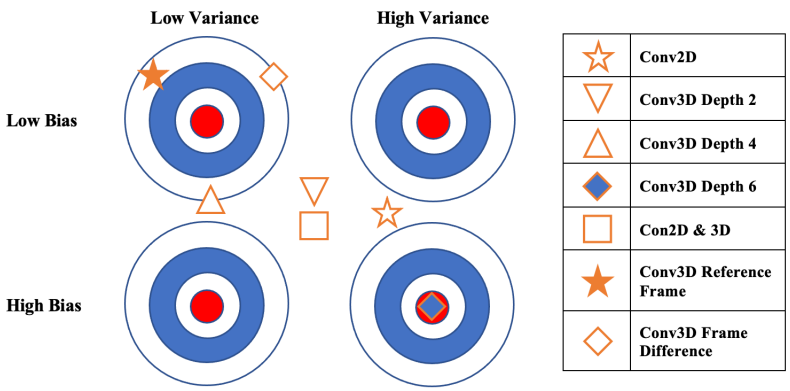


Figure 4.2: The bias and variance bullseye plot for each model.

4.2.2 Loss and Accuracy Performance Comparison of 3D CNN Models

While each model has been discussed in terms of bias and variance, it is important to compare the effect that the variation of each of the 3D CNN models' depth has on their performance. Of the three variations of 3D CNNs, the best performing model for this evaluation had a depth of four. When considering the temporal information, the Conv3D Depth 2 model only utilises the temporal information of the current frame and the next frame. In videos, this can often result in too little temporal information as consecutive frames may be identical for videos with high frame rates. On the other hand, the Conv3D Depth 6 model, has an abundance of frames as an input, which allows the model to be exposed to more temporal information than the other two models. However, this does come at a cost, as the greater the depth of the CNN model, the smaller the dataset becomes, meaning there is less data available to optimise the model during training and validation. This is well-presented by Equation 3.3 where a larger image depth leads to a lower number of windows and thus 3D input images. It is this lack of data that leads to a more variable performance by the model as seen in Figure 4.3 below.

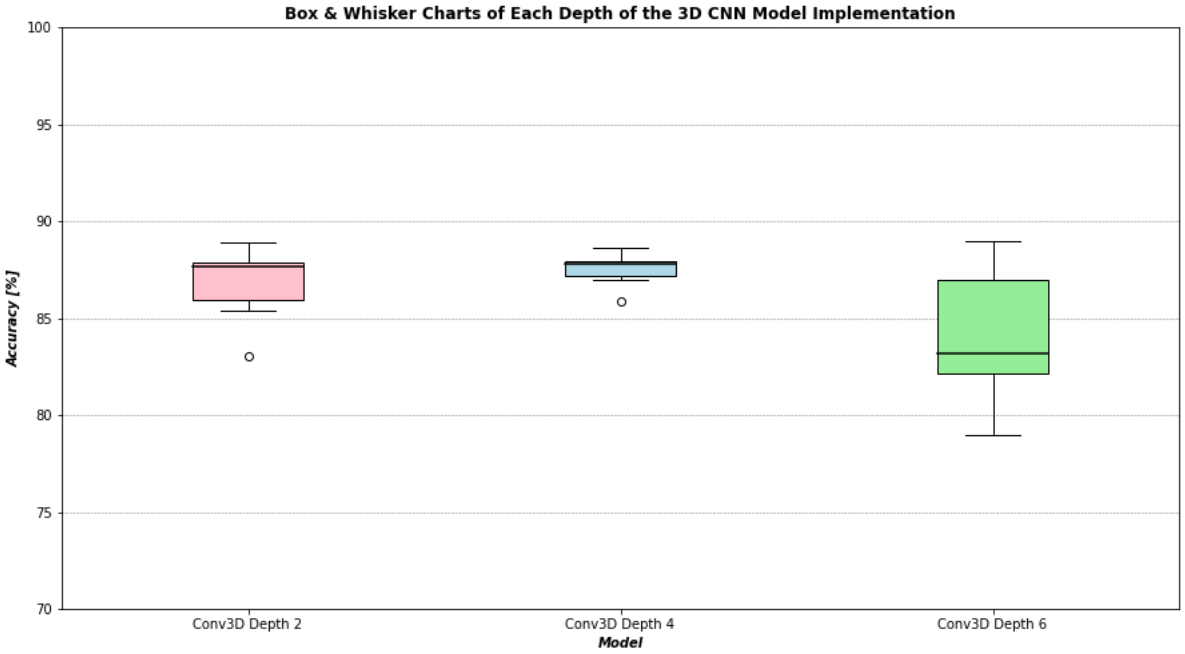


Figure 4.3: A comparison of each models accuracy.

4.2.3 Performance Remarks

While the use of the k-fold cross-evaluation technique is a great way to measure the performance of the models, video based-classification needs alternative methods to get a true sense of the ability of the models to generalise. That being said, the bias, variance and accuracy metrics provide interesting insight into the training and validation evaluation of the models. These metrics were particularly important in the fine-tuning of the various hyperparamters that were mentioned in Chapter 3.

4.3 Generalisation to Unseen Data

Each model's ability to generalise to unseen data can be evaluated based on its accuracy in correctly predicting a frame as fire or no fire, as well as its false positive and negative rates. Generalisation provides a more realistic performance testing method, as each model is placed in a real life environment where fire needs to be detected. The videos described in Table A.1 and shown in Figures 3.13 and 3.14 are used to test each model's generalisation ability.

4.3.1 Summary of the Fold that Generalises Best for each Model

While the six-fold cross-validation technique produced six different performing folds for each model, only the fold that generalised the best was chosen for this evaluation. The selected fold of each model is displayed in Table 4.1 along with the corresponding accuracy and loss for that fold. In addition, the rank of the fold's accuracy and loss in comparison to the other folds for that model are shown as well.

Table 4.1: A summary of accuracy & loss metrics of each model's best generalising fold.

Model	**Fold Number**	**Accuracy**	**Rank**	**Loss**	**Rank**
2D CNN	5	88.24%	3	0.256	4
2D CNN using a 3D CNN	6	84.35%	6	0.264	3
3D CNN with Depth 2	3	87.67%	3	0.235	2
3D CNN with Depth 4	2	87.86%	3	0.221	1
3D CNN with Depth 6	6	89.00%	1	0.279	1
3D CNN using a Reference Frame	3	92.44%	4	0.114	2
3D CNN using Frame Differencing	3	94.54%	1	0.108	1

An notable observation is that the best performing fold of a model does not correspond to the fold that generalises the best. For several of the models, the fold that generalised best often was the second or third best performing fold when evaluated using the k-fold cross-validation method mentioned in Section 4.2. Interestingly, the folds that generalised best had one of the lowest loss values when compared to other folds. The loss of a fold is often associated with the expected variability of a model's results when testing its generalisation. The lower the loss, the more consistently a model generalises to various sets of videos. On the contrary, the higher the loss, the more variable a model's performance is. Table 4.1 shows that the loss of a fold is more important in determining how well it generalises to unseen data than its accuracy. A lower accuracy does, however, indicate that the model did not overfit to the training data, which is ideal because overfitting often leads to the poor generalisation of data.

4.3.2 Comparison of Various Models Performance on Unseen Video 1

The folds of each model, which are mentioned in Table 4.1 were tested on the first of the two unseen videos. The accuracy of the prediction, false positive and false negative rates were calculated for each model. As mentioned in Chapter 3, fire is considered as positive while no fire is considered as negative.

Thus, a false positive occurs when a model predicts a fire frame when the frame is actually a no fire frame, while a false negative occurs when a model predicts a no fire frame when the frame is actually a fire frame. These metrics are summarised in Table 4.2 and the model that performed the best being highlighted in light blue.

Table 4.2: A summary of performance metrics of each model's best performing fold on unseen video 1.

Model	**Fold Number**	**False Positives**	**False Negatives**	**Accuracy**
2D CNN	5	11.67%	56.45%	65.57%
2D CNN using a 3D CNN	6	41.67%	8.20%	75.21%
3D CNN with Depth 2	3	26.67%	3.33%	85.00%
3D CNN with Depth 4	2	35.00%	79.30%	43.22%
3D CNN with Depth 6	6	3.33%	0%	98.28%
3D CNN using a Reference Frame	3	0%	100%	50.41%
3D CNN using Frame Differencing	3	0%	100%	50.41%

The best performing model was the 3D CNN with Depth 6, with other models that also performed well being the 3D CNN with Depth 2, 2D CNN implemented using a 3D CNN and the 2D CNN. The numbers behind these performance results can be further investigated by looking at graphs of the model's prediction for each frame of unseen video one. In addition, the confusion matrix of positives and negatives is also provided for further insight. The graph provides temporal information, a better understanding of the frames that the model may have predicted incorrectly, as well as the confidence in which the prediction is made. On the other hand, the confusion matrix is useful in providing numerical values that support the graphs. Table 4.3 provides the frames of unseen video one that correspond with no fire and fire respectively.

Table 4.3: Corresponding fire and no fire frames of unseen video 1.

Classification	**Frame Numbers**
No Fire	0 - 59
Fire	60 - 118

2D CNN

In the early stages of unseen video one, the 2D CNN correctly classifies the frames as no fire with great confidence. For the first 30 frames, the no fire class is predicted with an accuracy of 76%, while the fire class is predicted with a 24% accuracy. Some interesting changes occur from frames 30 to 60, as the model adapts from predicting no fire frames to fire frames. Although the majority of the frames are still correctly classified as no fire, they are done so with less confidence. In unseen video one, there is a significant amount of cloud coverage throughout these frames, which would explain why the 2D CNN model incorrectly classified them.

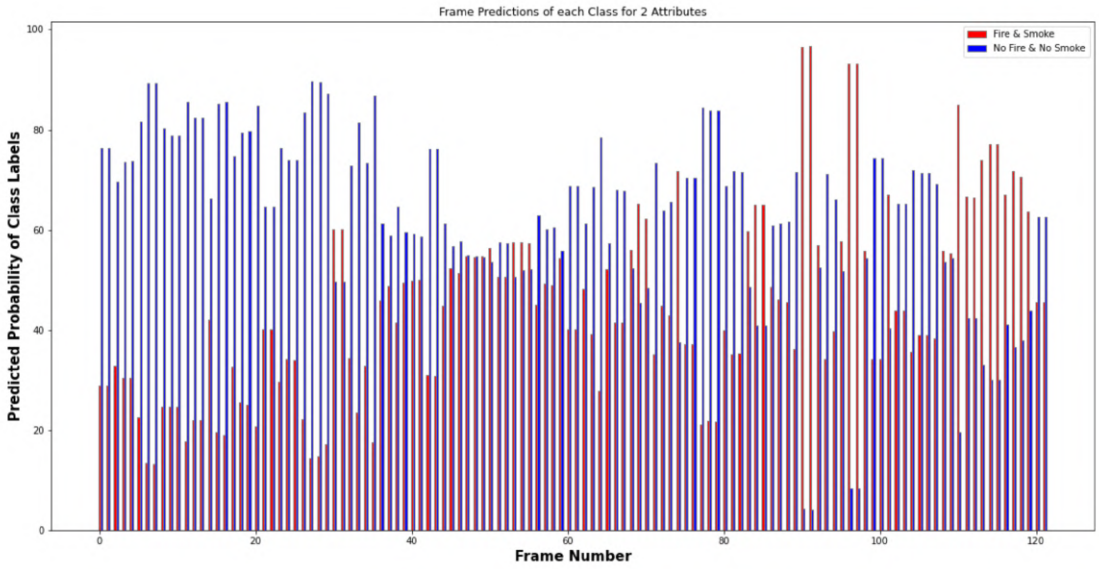


Figure 4.4: Frame predictions made by the 2D CNN model for video 1.

The fire in unseen video one actually occurs from frame 60 onward and Figure 4.4 shows that the model struggles to correctly classify these fire frames. It must be noted that there are some stages within the video where the model performs well and can classify the fire with great confidence, namely frames 85 to 100 and frames 110 to 120. That being said, there is still some variability in the predictions of fire and this is evident in the confusion matrix of the model, which shows that a total of 34 false negatives, i.e. 34 frames, were incorrectly classified as no fire. It can be deduced from the confusion matrix in Figure 4.5, that the 2D CNN model struggles to differentiate between the smoke produced by the fire and the clouds in the sky.

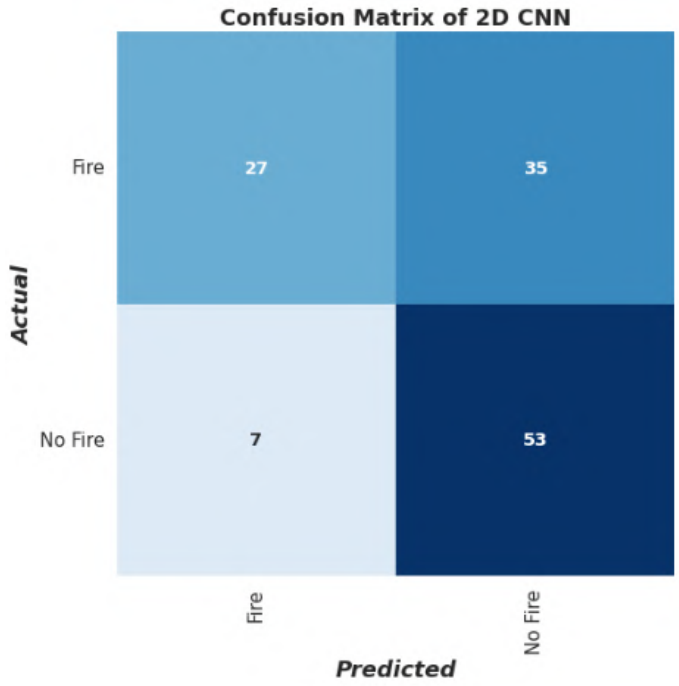


Figure 4.5: Confusion matrix of the 2D CNN model for video 1.

2D CNN using 3D CNN

The 3D implementation of the 2D CNN, was close to 10% more accurate than the 2D CNN. This approach did, however, produce contrasting results to that of the 2D CNN approach. While the 2D CNN confidently predicted the first 30 frames as no fire, the same cannot be said for this model. An average no fire prediction of 66% is made for these first few frames, while the other 34% was predicted as fire. The 2D CNN approach had a delayed response in classifying the fire frames. The converse is true for the 2D CNN implemented using a 3D CNN as displayed in Figure 4.6. Frames 30 to 60, which lead up to the fire frames from 60 onwards, are classified incorrectly. While one of the objectives of this paper is to achieve early fire detection, this detection of fire, although early, is incorrect.

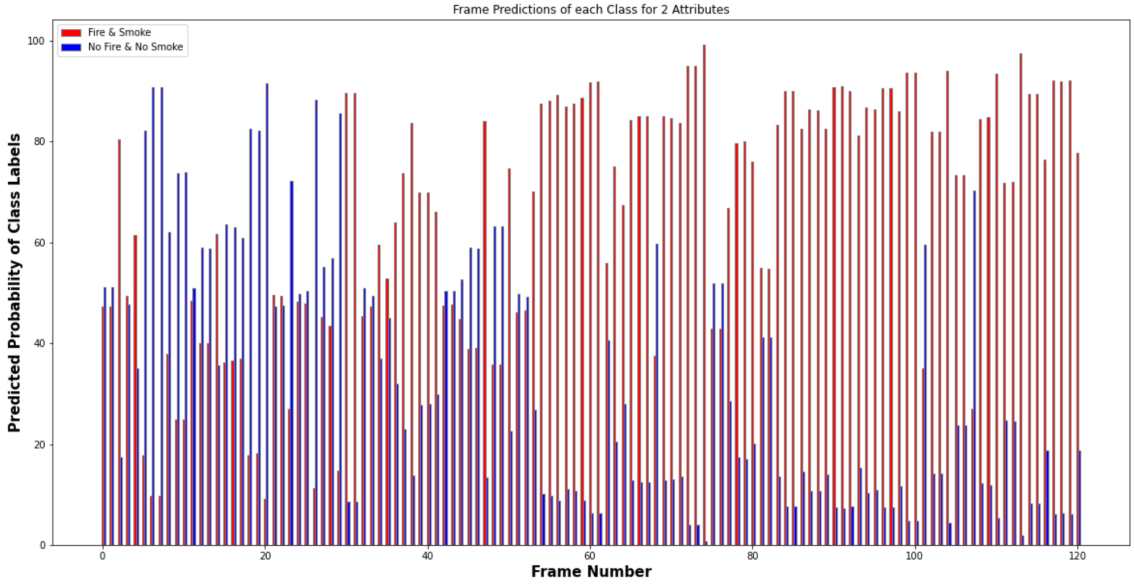


Figure 4.6: Frame predictions made by the 2D CNN using a 3D CNN for video 1.

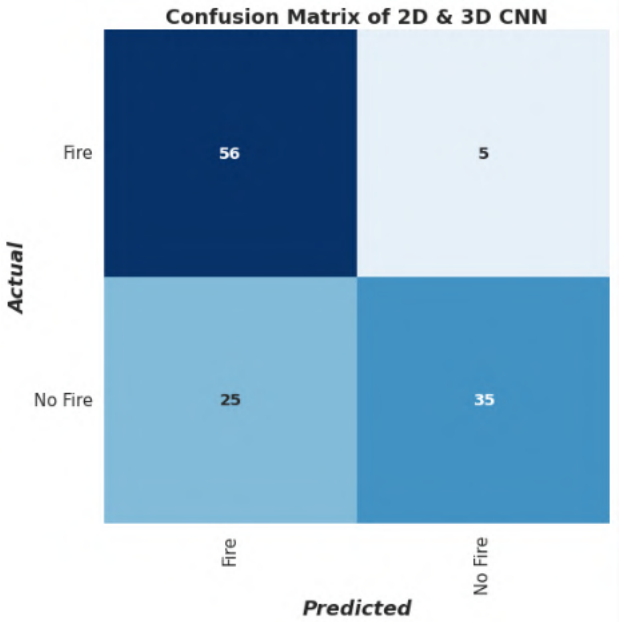


Figure 4.7: Confusion matrix of the 2D CNN using 3D CNN for video 1.

This model is able to classify the fire frames, frames 60 to 120, with greater certainty and less variability than the 2D CNN. Other than an incorrect prediction at frame 100, the model's performance is supported by its classifying accuracy of 75.21%. The confusion matrix in Figure 4.7 better depicts the high number of false positive predictions that were made. It can further be deduced that the 2D & 3D CNN model struggles to differentiate between the no fire frames, frames zero to 60, that consist of smoke-like clouds, and the fire frames, frames 60 to 120, that consist of both the clouds and smoke produced by the fire.

3D CNN with Depth 2

The 3D CNN model with depth 2, was the second best performing model having an accuracy of 85.00%. Its performance in the first 30 frames of unseen video one are similar to that of the 2D CNN model, as the no fire class is correctly classified with an average accuracy of 78%. Even though the model manages to predict the correct class for frames 30 to 50, it does so less convincingly than its prediction for the first 30 frames. Its prediction accuracy is greatly reduced to an average of 57% for the fire class, while an average accuracy of 43% is predicted for the fire class.

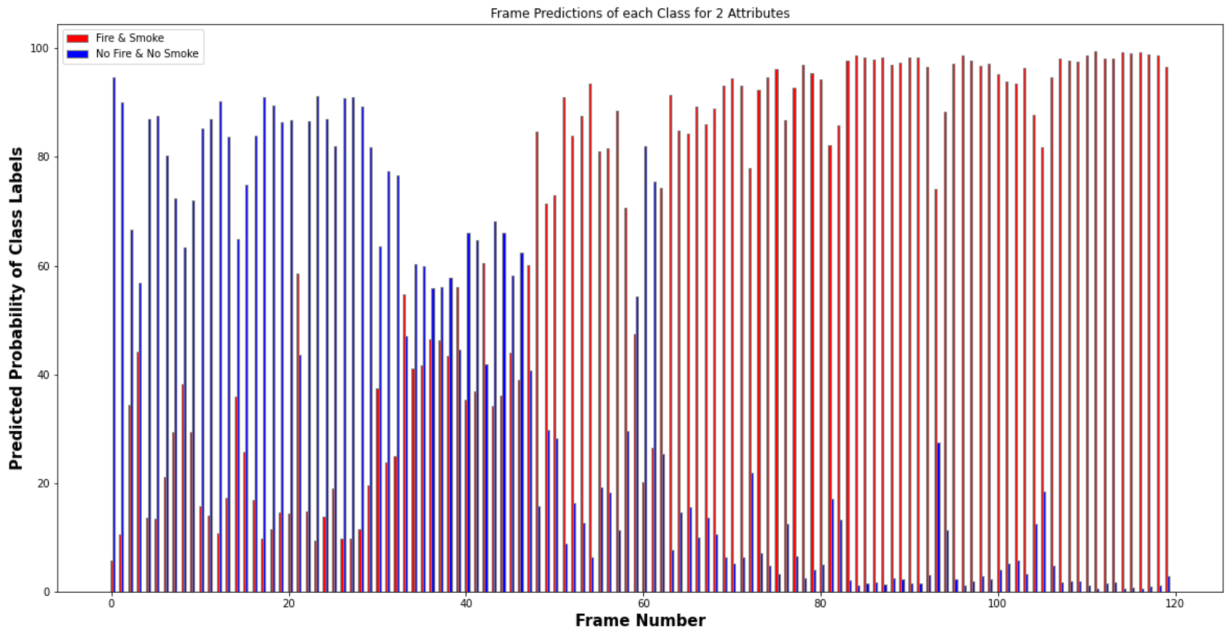


Figure 4.8: Frame predictions made by the 3D CNN model of depth 4 for video 1.

Early signs of fire detection can be seen in frames 50 to 60 of Figure 4.8. For this set of frames, there is a large portion of white cloud cover that dominates the skies, as well as over exposure in the video. These two characteristics are similar to smoke, which explains why the model initially misclassifies these frames as fire, much like the 3D implementation of a 2D model. That being said, the 3D CNN model with depth 2 correctly classifies frames 60 to 120 as fire frames and does so convincingly. The confusion matrix for this model, as seen in Figure 4.9, is similar to that of the previous model. The key differentiator is that this model produced only 16 false positive and one false negative misclassifications. This translates to 16 fewer misclassifications than the 3D model implemented as a 2D CNN.

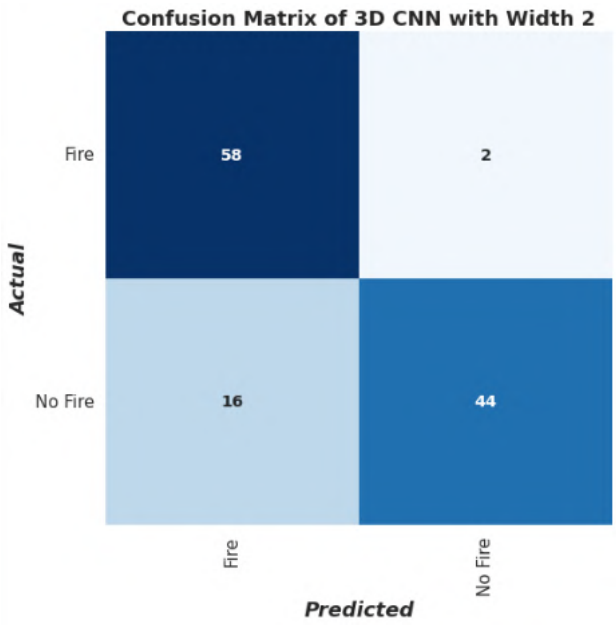


Figure 4.9: Confusion matrix of the 3D CNN with depth 2 for video 1.

3D CNN with Depth 4

Although the overall performance of this model is poor, having the lowest prediction accuracy of 43.22%, the predictions made per frame provide interesting results. Figure 4.10 shows the variability of the model when trying to classify each frame. It is important to observe that, aside from the first few frames, the model classifies frames with confidence, even if it is incorrect. This means that despite its inaccuracy, this model still has an ability to classify the unseen video, and with appropriate tuning its performance can be improved.

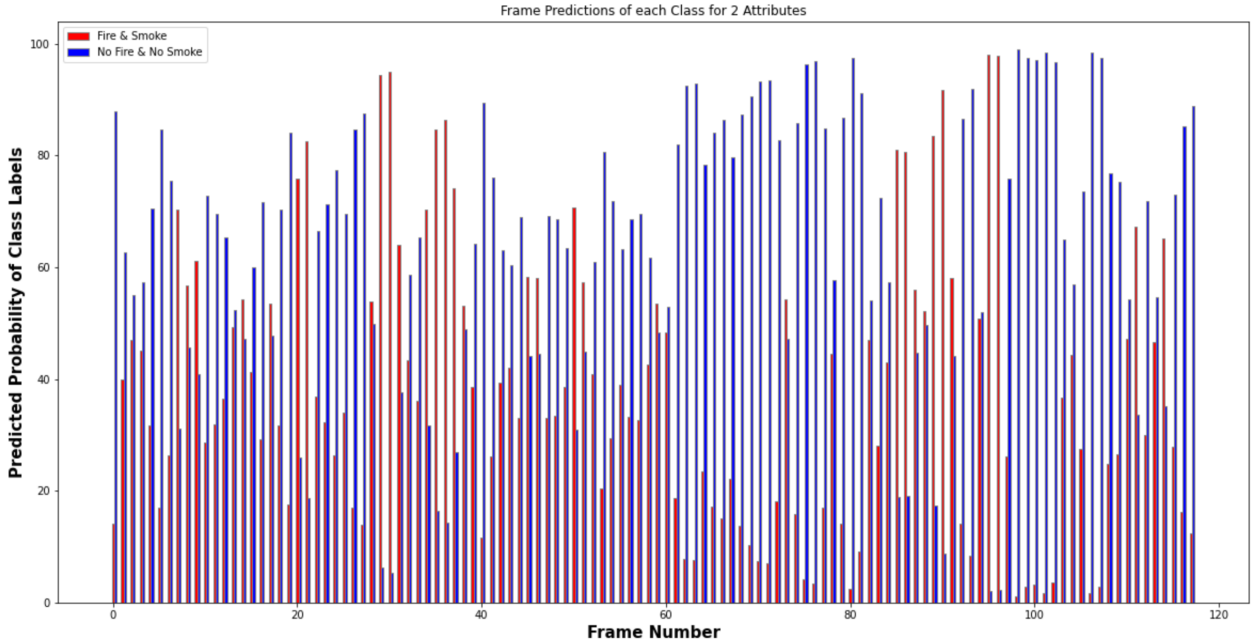


Figure 4.10: Frame predictions made by the 3D CNN model of depth 4 for video 1.

A large portion of the misclassifications came as a result of false negatives, i.e. classifying a fire frame as no fire. The confusion matrix in Figure 4.11 shows the poor performance of this model and supports its low accuracy value.

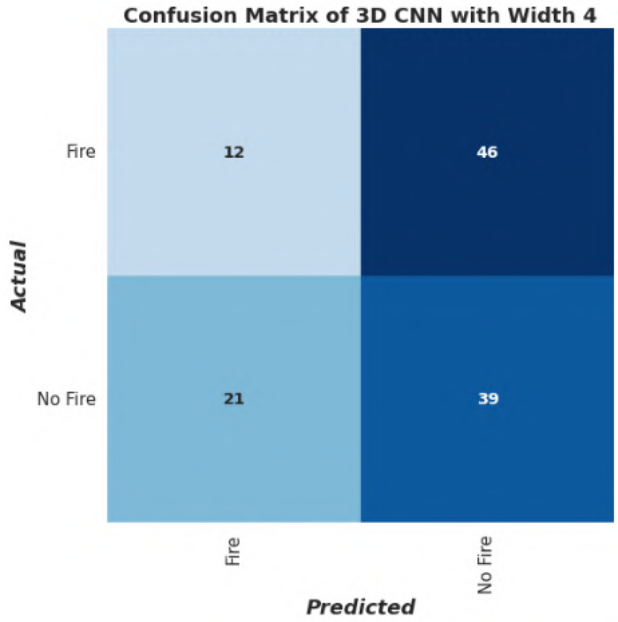


Figure 4.11: Confusion matrix of the 3D CNN with depth 4 for video 1.

3D CNN with Depth 6

The 3D CNN with depth 6 was the most accurate model and thus best performing model when tested on unseen video one. Its accuracy of 98.28% is better represented by the predictions made per frame as shown in Figure 4.12.

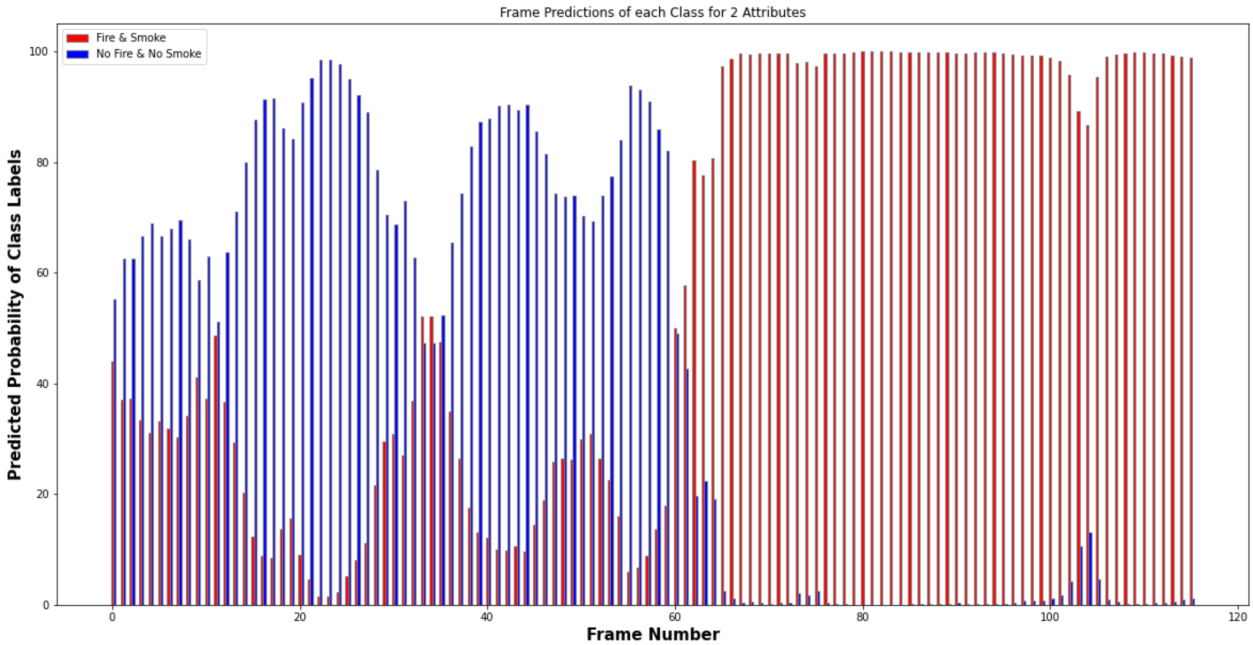


Figure 4.12: Frame predictions made by the 3D CNN model of depth 6 for video 1.

The model did not convincingly classify the no fire class for the first few frames even though it correctly classified them. Furthermore, it not only correctly predicts all of the fire frames, but it does so as soon as the no fire scene of frame 59 changes to a fire scene in frame 60. Considering this, the model shows signs of early fire detection without incorrectly classifying frames, as both the 3D CNN model with depth 2 and the 3D model implementation of a 2D CNN do. The classification of fire is also made with close to 100% confidence. The confusion matrix in Figure 4.13 further shows that the only incorrect classifications were those of three false positives. These false positives occur for frames 16, 34 and 35. The latter two frames are when there is increased cloud coverage in the video. As mentioned above, this caused the 3D CNN model with depth 2 to also falsely classify these types of frames. But unlike that model, the 3D CNN with depth 6 only did so for 2 frames, before correctly classifying the frames that followed.

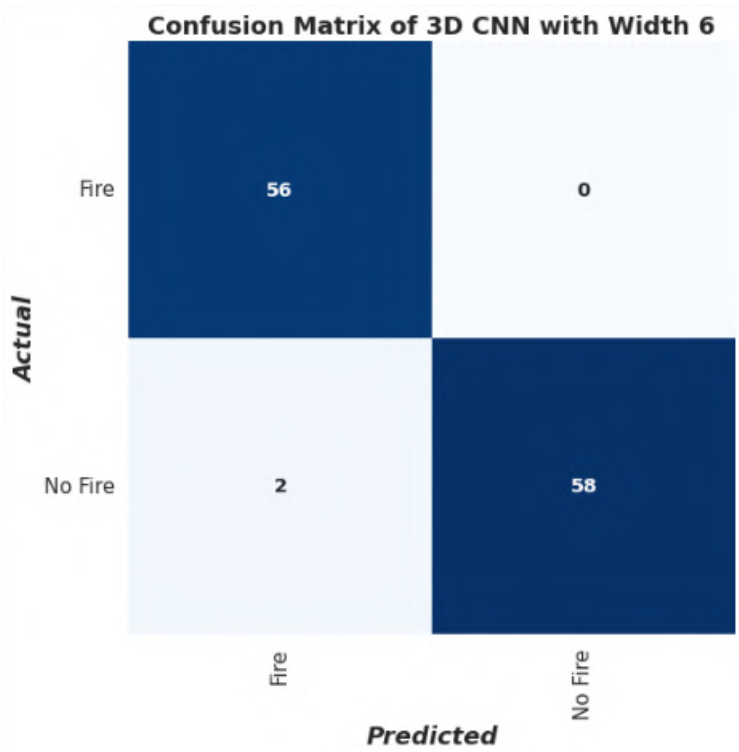


Figure 4.13: Confusion matrix of the 3D CNN with depth 6 for video 1.

3D Frame Referencing & Frame Differencing CNN Models

The two lowest performing models were the 3D Frame Referencing CNN and 3D Frame Differencing CNN models. Interestingly, these were the two models that performed best in the k-fold cross-validation evaluation. Even though their accuracies were not the worst, with the 3D CNN with depth 4 having the worst overall performance, this is not a true representation of the performance of these models. While the 3D CNN model with depth 4 misclassified frames, which resulted in high false positive and negative rates, the Reference Frame and Frame Differencing models had a 100% false negative rate. This means that for every fire frame, the models convincingly classified the frames as having no fire. This is better understood by looking at the confusion matrices of these models, which are shown in Figures 4.14 and 4.15 on the following page.

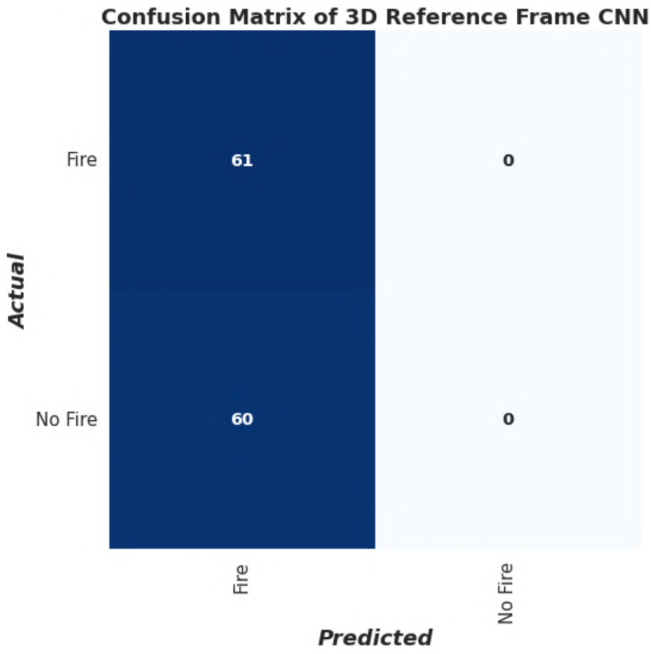


Figure 4.14: Confusion matrix of the 3D frame referencing CNN for video 1.

The confusion matrices above clearly depict that these two models were the worst performers in the context of this paper, as they failed to predict a single fire frame.

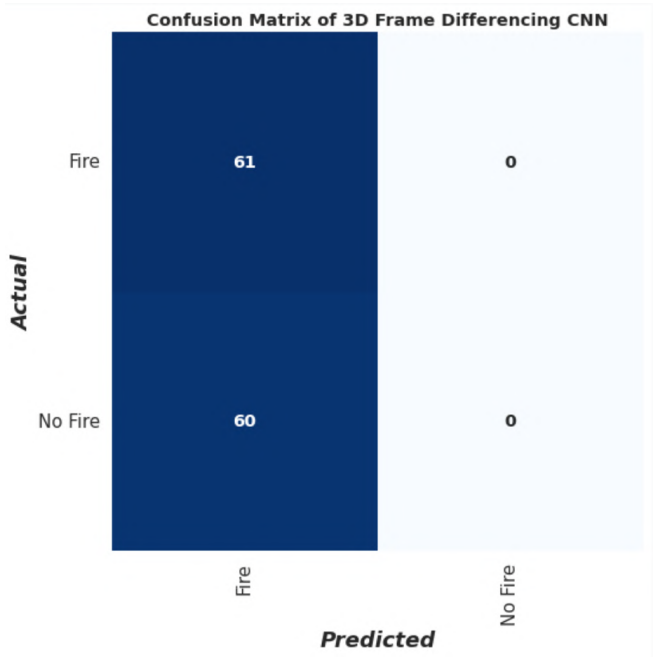


Figure 4.15: Confusion matrix of the 3D frame differencing CNN for video 1.

4.3.3 Moving Average Performance

Moving average is the process of basing the prediction of a frame on two or more consecutive frames. The predictions of each class for those frames are averaged and the class with the higher prediction is classified for that frame. This process helps smooth out the predictions of a video, especially when

there are a few outliers in the form of false positives or negatives. The impact on the performance metrics of applying the moving average is investigated in the sections below.

Models Performance on Unseen Video 1 using Moving Average

For these results, a fixed window size of five was used, which means that the classifying of frames depends on that of the five surrounding frames. It must be mentioned that for the first four frames, predictions were made using a window size that incrementally increased to the specified size of five. The models that were chosen to apply the moving average were the 2D CNN, 2D CNN using a 3D CNN, a 3D CNN with depth 2 and a 3D CNN with depth 4. The other models were disregarded because the moving average would have a negligent effect on improving their performance. The resulting change to the performance metrics of the models to which the moving average was applied is shown in Table 4.4.

Table 4.4: A summary of performance metrics of models using moving average on unseen video 1.

Model	**Fold Number**	**False Positives**	**False Negatives**	**Accuracy**	**Increased Accuracy**
2D CNN	5	**3.33%**	56.45%	69.67%	4.10%
2D CNN using a 3D CNN	6	43.33%	**0%**	78.51%	3.30%
3D CNN with Depth 2	3	16.67%	**0%**	**91.67%**	6.67%
3D CNN with Depth 4	2	20.00%	79.31%	50.85%	**7.63%**

There are notable improvements in the prediction accuracies of all the models. The model with the most significant accuracy improvement was unironically the worst performing model, namely, the 3D CNN with Depth 4. This does, however, support the claims that were made surrounding the variability in the model's initial accuracy prior to the moving average being applied. The 2D CNN had the lowest false positive rate of 3.33%, while both the 3D CNN implemented as a 2D CNN and the 3D CNN model with depth 2 had false negative rates of 0%. The confusion matrix for each model better represents the improvements to the model's accuracy as a result of using the moving average.

2D CNN

Naturally, with an increase in prediction accuracy to 69.67%, it could be assumed this was due to a decrease in the false positive and negative rates. This is the case, as even with the moving average, the 2D CNN model still misclassified an equal amount of fire frames as it did without the moving average. The impact of applying the moving average is seen in the big decrease in the number of false positives with only two frames, rather than seven without the moving average, being misclassified as fire. This is shown in the confusion diagram of Figure 4.16.

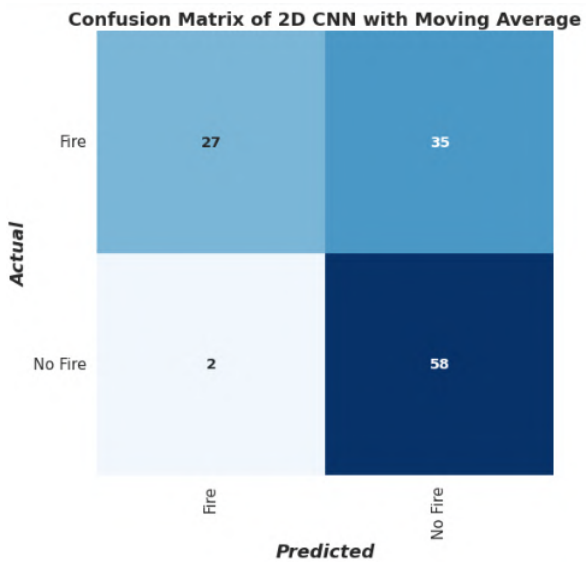


Figure 4.16: Confusion matrix of the 2D CNN with moving average for video 1.

2D & 3D CNN

Although the accuracy for the 2D and 3D CNN model only increased by 3.30% to 78.51%, the use of the moving average was equally as effective as it was for the 2D CNN model. There were no fire frames that were misclassified, as shown by the confusion matrix in Figure 4.17. This led to the reduction of the false negative rate to zero and an increase in overall accuracy. That being said, the moving average did lead to a no fire frame being misclassified and thus an increase to the false positive rate. This can often happen with the moving average method, in particular during the change from no fire frames to fire frames, which occurs at the 60^{th} frame of the unseen video. This model was more confident in its fire predictions than no fire predictions, as shown by Figure 4.6. Considering this, the moving average had an adverse affect to the false positive rate for this model.

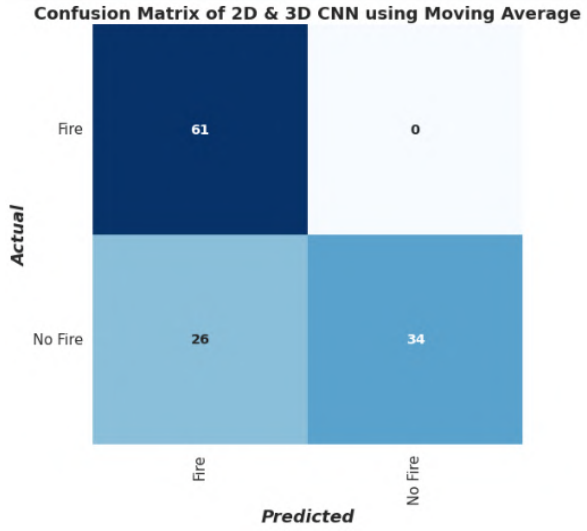


Figure 4.17: Confusion matrix of the 2D & 3D CNN with moving average for video 1.

3D CNN with Depth 2

Unlike the previous model, the 3D CNN with depth 2 produced positive results all round when the moving average method was implemented, as seen in Figure 4.18. A reduction in the false negative rate to 0% and only 10 frames being misclassified as fire means that the overall accuracy improved significantly. While the accuracy of 91.67% was the best of all the models, as mentioned in Table 4.4, it is still outperformed by the 3D CNN model with depth 6 without a moving average.

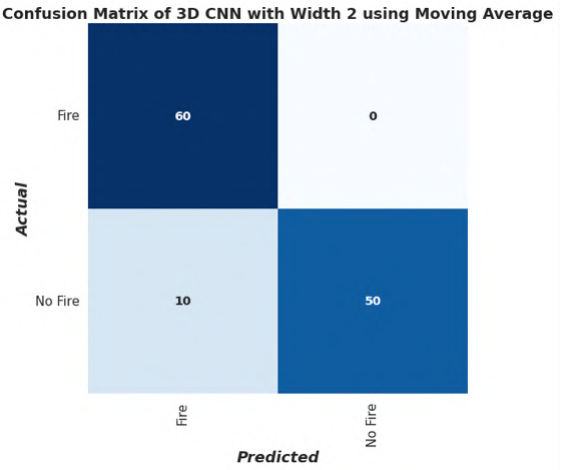


Figure 4.18: Confusion matrix of the 3D CNN Depth 2 with moving average for video 1.

3D CNN with Depth 4

Interestingly, the worst performing model benefited the most when the moving average was used. The 3D CNN with depth 4 saw an accuracy increase of 7.63%. While its overall accuracy of 50.85% meant that this model remained the worst performing of all, the improvements of applying a moving average are significant. The confusion matrix of the model, shown in Figure 4.19, indicates that there were fewer false positives predicted. Nine additional frames were classified correctly when compared to the original model shown in Figure 4.11. Although the false negative rate did not decrease, it did not increase, which in itself is positive.

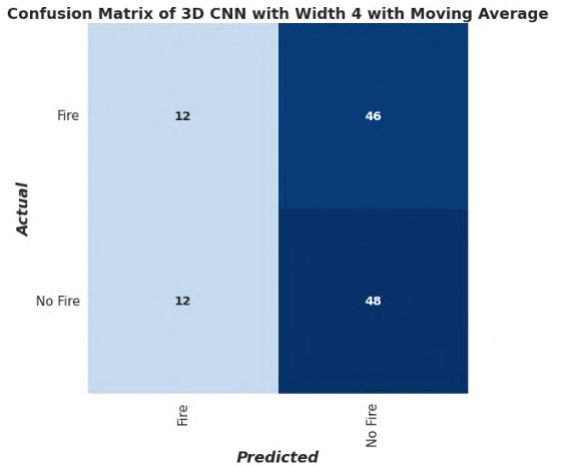


Figure 4.19: Confusion matrix of the 3D CNN Depth 4 with moving average for video 1.

4.3.4 Comparison of the Performance of Various Models on Unseen Video 2

The same comparisons were performed between the performance of the models on the second unseen video, with the specific folds of each model that were used in these experiments mentioned in Table 4.1. All the performance metrics that were calculated and used in the previous section for the performance comparison based on the first unseen model, are summarised in Table 4.5 and the best performing model is highlighted in light blue.

Table 4.5: A summary of performance metrics of each model's best performing fold on unseen video 2

Model	**Fold Number**	**False Positives**	**False Negatives**	**Accuracy**
2D CNN	5	0%	100%	68.03%
2D CNN using a 3D CNN	6	100%	0%	31.40%
3D CNN with Depth 2	3	0%	100%	68.03%
3D CNN with Depth 4	2	0%	100%	68.03%
3D CNN with Depth 6	6	0%	63.64%	81.90%
3D CNN using a Reference Frame	3	0%	100%	68.03%
3D CNN using Frame Differencing	3	0%	100%	50.41%

Similar to the results of the previous section, the best performing model was the 3D CNN with Depth 6. Contrary to the accuracy results displayed in Table 4.1 above, all other models only classified the frames of unseen video two as one class. This observation is further supported by the 100% false positive or false negative rates of models except the 3D CNN with Depth 6. Considering this, the numbers behind the 3D CNN with Depth 6 model's performance results can be investigated further by looking at graphs of the model's prediction for each frame of the video. In addition to this graph, the confusion matrix is also provided for further insight. Table 4.6 below provides the frames of unseen video one, that correspond with no fire and fire respectively.

Table 4.6: Corresponding fire and no fire frames of unseen video 2.

Classification	**Frame Numbers**
No Fire	0 - 81
Fire	82 - 116

3D CNN with Depth 6

The first 70 no fire frames of the video are not only classified correctly by the model, but also with great confidence. The 3D CNN model with depth 6 classifies these frames as no fire with an accuracy of close to 100%. In light of this, the relevant results are rather observed for the latter frames of the video, namely the fire frames. The model has a slightly delayed response in identifying the fire, as it does so four frames after when the fire first appears. However, once identified, it struggles to maintain

confidence in its prediction of the fire class. Frames 90 to 105 of Figure 4.20 show that the model misclassifies the frames as no fire. While the model then proceeds to correctly classify frames 106, 107 and 108, it does not do so with great certainty.

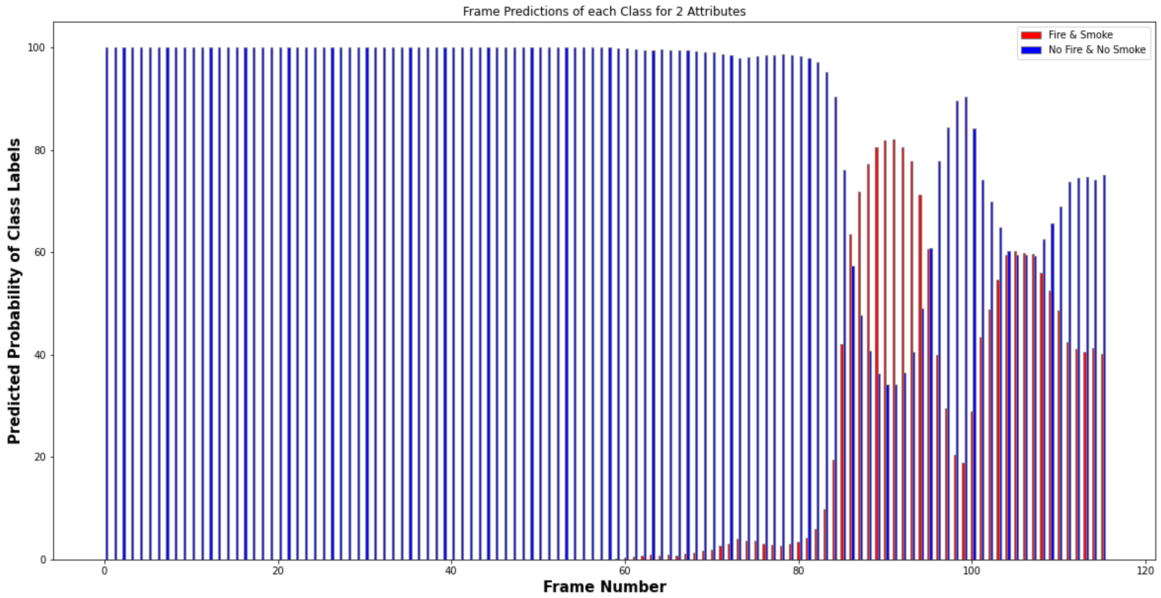


Figure 4.20: Frame predictions made by the 3D CNN model of depth 6 for video 2.

The confusion matrix shown in Figure 4.21 provides more insight on the misclassification of the classes by the 3D CNN model with depth 6. There were a total of 21 false negatives, with only 12 frames being correctly classified as fire.

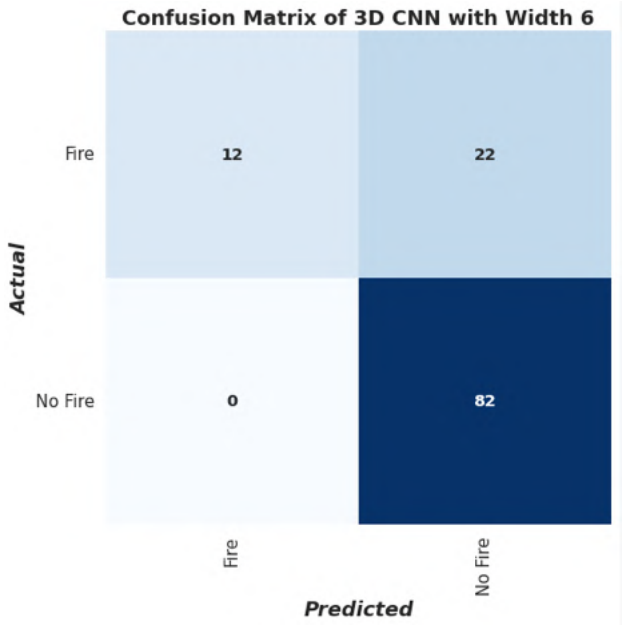


Figure 4.21: Confusion matrix of the 3D CNN with depth 6 for video 2.

Other Models

The frame prediction diagrams of all the models other than the 3D CNN with depth 6 model discussed above are not included in this report, as they do not provide useful information. The other models failed to identify the fire with only one of the classes being classified for every frame of unseen video two. The performance of these models for unseen video two is near identical to that described by Figures 4.14 and 4.15 of the reference frame and frame differencing models for unseen video one. Their accuracy, as shown in Table 4.5 is not a true reflection of their performance, as a model can easily achieve 50% accuracy if each frame is classified as one class only. In this case, the models achieved an accuracy of 68.03% because roughly 70% of the frames were no fire frames, as summarised by Table 4.6.

Chapter 5

Discussion

'It's hard to see things when you're too close. Take a step back and look'

— *Bob Ross*

The problem that is investigated in this paper involves the quantification and comparison of the performance of various 2D and 3D CNN models on a limited HPWREN dataset. This chapter aims to provide an analysis of the results that were presented in Chapter 4. Comparisons between each model's performance on the unseen videos and the model's structure are drawn.

5.1 Performance Analysis of Models

A performance analysis of the various models' ability to generalise to the unseen videos is conducted in the sections that follow. Each section compares the structural differences of every model, which makes it more clear why certain models performed better than others. Some of the differences that are looked at are the input image shapes supplied to the models and the kernel sizes of the models' convolutional and pooling layers.

5.1.1 2D CNN Model

The most commonly used image classifier model is a 2D CNN. Considering this, and that videos are made of consecutive frames/images, using a 2D CNN to classify fire videos is a reasonable solution. This assumption is reflected in its performance results to unseen video one, where it achieved an accuracy of 65.57%. The accuracy achieved is unremarkable, but the 2D CNN model has shown signs of being able to detect video fire frames. When a moving average function was used, this performance increased to 69.67%. While this is a positive improvement to its performance it pales in comparison to the better performing models, like the 3D CNN implementations. The lack of performance can primarily be explained by the lack of data, which impacts the 2D CNN model more so than others, as it is entirely dependent on using spatial information of fire, namely, colour, shape and size, to detect it. This often requires a significant amount of frames/images to train the model to learn these features.

One of the key problems identified with the vision-based fire detection field is the lack of large diverse datasets. While HPWREN is suitable for the scope of this paper, it is very limited with only an average of 7 000 frames being used to train and test the 2D CNN model. Considering this, its lack in performance on unseen video two can be better understood, even though it scored an accuracy of 68.03% for this video. While at face value this is a surprisingly better result than the accuracy

achieved for video one, the other performance metrics tell a different story. The model had a 100% false negative rate meaning that zero fire frames were classified correctly. Coupled with 70% of unseen video two consisting of no fire frames, the effectiveness of the model's ability to correctly predict frames for this video is questioned. To only use the accuracy metric to draw conclusions regarding the model's performance is not sufficient. Nevertheless, this interpretation shows positive signs in being a well-performing fire detection interpretation, provided that it is improved further by using more complex data augmentation methods to increase its dataset size

5.1.2 2D CNN Model Implemented with a 3D CNN

A model which had a similar approach to that of the 2D CNN was the 3D CNN implemented as a 2D CNN. Both models take one image/frame as an input and use the spatial features of the input to classify it as fire or no fire. However, the key difference between the two models is the added layer of dimensionality in the form of image depth in the 2D CNN model implemented with a 3D CNN, resulting in kernel sizes of 3x3x1 and 2x2x1 being used for the convolutional and pooling layers respectively. This added layer not only means that the input shape of the data differs to that of the 2D CNN but it is also interpreted differently, with the possibility of extracting certain temporal features. While a 3D input with a depth of one has fewer temporal features than an input with depth two or more, this subtle change in structure reflected positively in the accuracy results for unseen video one. The 2D CNN model implemented with a 3D CNN, had an accuracy of 75.21% - a roughly 10% increase in accuracy from the 2D CNN's accuracy. Although the 2D CNN had a larger increase in accuracy when the moving average function was applied, the 2D CNN model implemented with a 3D CNN was still the better performing model.

Interestingly, this model had a higher false positive rate while the 2D CNN had a much higher false negative rate. Something that was observed in the 3D CNN models, the classification of the last few no fire frames of unseen video one, namely, frames 50 to 60, as fire frames. Consequently, this model performed the poorest for unseen video two, as it classified all frames as fire. Naturally, with only around 30% of video two's frames actually being fire, it had a 31.4% accuracy. In light of its poor performance for the second video, similar conclusions to the 2D CNN model can be drawn regarding its performance and data availability. The 2D CNN implemented with a 3D CNN also requires a larger dataset, and most importantly more fire data, so as to be trained better.

5.1.3 3D CNN Models

The literature review in Chapter 2 highlighted the importance of utilising temporal information in video-based classification. Spatial information in itself is not sufficient, particularly in an application where the spatial features of fire can often be mistaken for that of clouds or mist. Consequently, a variety of 3D CNN models with different depths were tested to determine the effect that different lengths of temporal information had on its fire detection performance. Other than the differing image depths of the input image shapes for each 3D CNN model, each interpretation differentiates from the next based on the kernel sizes used for the convolutional and pooling layers. The effect that these kernel sizes had on the performance of the various models is discussed separately in the three sections that follow.

3D CNN with Depth 2

The 3D CNN model with depth 2 used convolutional and pooling kernel sizes of 3x3x2 and 2x2x2 respectively. This means that the kernels are applied on the whole depth of the input, namely, on both the current and next frames for a particular point. This interpretation better utilises the temporal information of fire between two adjacent frames, something that the reference frame model does not do as well as it has a different approach. The utilisation of a video's temporal information is supported by the model's performance when generalising to unseen video one. This model had the second best performance with an accuracy of 85%, which was later improved to 91.67% when the moving average function was applied. However, as mentioned in Section 5.1.2, the 3D interpretation detects fire earlier than when it actually occurs in the frames. That being said, this interpretation had a false positive rate of 2.67%, which was significantly lower than that of the 2D model implemented using a 3D CNN. Despite all this, the use of a temporal window of two frames was too small. Although the model learnt the temporal features of fire, the cloud cover present in video one meant the model still struggled to differentiate it from fire smoke, as was the case for several other models. Even though this model's performance on video one is second best, it fails to generalise to video two, scoring an accuracy of only 68.03% for a video that contains 70% no fire frames. This second evaluation shows that the 3D CNN model with depth 2 struggles to extract temporal information from videos that contain a low number of fire frames.

3D CNN with Depth 4

Similar to the previous model, the 3D CNN with depth 4 has kernel sizes of 3x3x2 and 2x2x2 for the convolution and pooling layers respectively. The difference lies in the fact that four consecutive frames are combined to make the 3D input of which the convolutional and pooling kernels iterate over the image depth twice. By doing so, different temporal information about the four frames is obtained from each iteration. Surprisingly, this interpretation was not supported by its performance as seen in Chapter 4. The results showed poor performance for both unseen videos, unlike the model with depth 2, which only underperformed for video two. Of the two videos, the model with depth 4 scored the lowest accuracy of 43.22% for video one and 68.05% for video two. The accuracy for video one did improve the most (by 7.63% to 50.85%) when the moving average function was applied, however, the overall performance of this interpretation is still poor. The figures in Section 4.3.2 show the variability in the predictions that were made per frame. Considering all of the above, the kernel sizes used for this model did not fully utilise the temporal information that was provided. In addition, kernel sizes of 3x3x4 and 2x2x4 could be better suited to increasing the model's performance if one takes into account the performance of the previous model.

3D CNN with Depth 6

The best performing model out of all the interpretations was the 3D CNN with depth 6. Unlike the two previous models with depth's two and four, this interpretation used convolutional and pooling kernel sizes of 3x3x3 and 2x2x3. Much like the 3D CNN with depth 4, there were two iterations of the convolutional kernel on the input imaged. However, six consecutive frames were combined together to make the 3D input image. This decision was consciously made due to the low frame rate of 4 fps, with a wider window frame allowing for the model to be presented with more useful and variable temporal

and spatial information. The 3D CNN with depth 6 scored an accuracy of 98.28% when testing its ability to generalise to unseen video one. Only two no fire frames were misclassified for this video - the lowest false positive rate of all models. While all other models generalised poorly to video two, the 3D CNN with depth 6 was able to do so with an accuracy of 81.90%. Although there were 22 false negatives predicted (fire frames incorrectly predicted as no fire), the model was able to correctly detect the initial fire frames when the fire started around frame 82. While a sustained prediction for all fire frames is ideal, early detection of fire is arguably more impressive. This model showed that an image depth of 6 provided the right amount of temporal information about the fires in videos that had either a majority or minority of fire frames. It shows that for the current augmentation and dataset size, it is best suited for the application of fire detection.

5.1.4 3D Frame Differencing CNN Model

The 3D Frame Differencing CNN model produced great results when evaluated using the cross-validation method, however, it struggled to generalise to the unseen videos. It was one of two models that only predicted one class (no fire) for all video frames, with the structure behind the model providing more insight for this performance. As mentioned in Chapter 3, frame differencing looks at suppressing the information between two consecutive frames that is not relevant or in this context not different. By doing so, only the important features of frames are fed into the 3D CNN as an input. In the context of the CNN structure, there are various pixels that are zero or negative values, which are then set to a value of zero after the activation function is applied to them. However, there are various moving objects or features in the HPWREN video dataset that may be extracted unnecessarily using this method. These include the sun moving, birds flying past the cameras, and more importantly, a variety of cloud coverage. The literature review conducted in Chapter 2 mentions that the spatial features of clouds can often be mistaken as smoke produced from fire. This means the clouds in no fire frames that the model was trained on can often lead to the model misclassifying an unseen video of smoke as no fire. One other such example is the change in brightness of a video. With a low frame rate of 4 fps, frames often have different levels of brightness due to the moving sun. This reduces the overall effectiveness of the frame difference method, as few irrelevant background pixels are minimised. Considering this, the model struggles to extract the important features of fire. This is supported by its performance on unseen data, with it only scoring an accuracy of 50.41% on video one and 68.03% for video two where 70% of the frames are no fire frames.

5.1.5 3D Frame Referencing CNN Model

The 3D Frame Referencing CNN model has a very similar implementation to the 3D CNN model with depth 2. One of the differences of this interpretation to the others is that the frame reference model constantly inputs the first no fire frame to the model combined with all consecutive fire and no fire frames. On the other hand the 3D CNN model with depth 2 differs by taking two consecutive frames: the frame in question and the following frame.

The other key difference is that the model structures only differ in the kernel sizes used for each convolutional and pooling layer. For the reference frame model, kernel sizes of 3x3x1 and 2x2x1 were respectively used, while the 3D CNN model with depth 2 used 3x3x2 and 2x2x2 sized kernels. This means that for the frame referencing model a constant no fire scene (reference or first frame) is fed in

as part of the input. Coupled with the convolutional kernel convolving on each depth of the input, i.e. the reference frame and current frame, the model is continuously exposed to the reference frame for every input. It fails to utilise the temporal information that could be extracted from the input, namely, learning to identify differences between the reference frame and current frame. Its overexposure to the first no fire frame has resulted in the model becoming optimised in such a way that the fire features become difficult to extract and classify correctly. Much like the frame differencing CNN, this is supported by its poor generalisation to unseen data as this model only scored an accuracy of 50.41% on video one and 68.03% for video two.

Chapter 6

Conclusion

'It always seems impossible until its done'
— *Nelson Mandela*

The impact of climate change on the global increase in wild fires is notable and has led to the growing need for early fire detection. The existing and widely-used solution of particle-based fire detection systems, i.e. smoke detectors, is flawed due to the high false positive rates produced, delayed response times and limited application. False alarms require human operator intervention to validate whether the alarm is false or not, which is cumbersome and undesirable. With the technological advancements of the 21^{st} century, an alternative solution to smoke detectors, namely the use of computer vision to detect fire, has been developed. Through this, the field of vision-based fire detection was established, and traditional handcrafted techniques for feature extraction were predominantly used to detect fire. However, many of the interpretations of this approach proposed by researchers are tedious to implement and inaccurate, especially because the interpretations struggle to differentiate between fire-like objects.

As a result of the shortcomings of the traditional handcrafted techniques, a hybrid approach that combines these techniques with deep learning has more recently been proposed for use and applied in this field. The lack of large and diverse datasets within the vision-based fire detection field is one of the leading issues in applying interpretations of this hybrid approach. Furthermore, this paper identifies that more reliable evaluation techniques are needed, especially techniques that focus on a model's ability to generalise to unseen data.

Inspired by the success of CNNs as image classifiers, this paper proposes a variety of 2D and 3D CNN models as possible interpretations to this hybrid approach. These models are namely, a 2D CNN, a 3D CNN implemented as a 2D CNN, a 3D CNN with varying depths of 2, 4 and 6, a 3D frame referencing CNN and lastly, a 3D frame differencing CNN. These models utilise the spatial and/or temporal information of the HPWREN dataset to better detect fire. The HPWREN dataset was selected due to it being the most diverse and relevant dataset to the scope of this paper, however, it was still limited in size relative to what was required for optimal training. Data augmentation was therefore also implemented in this paper's research to increase the size of the dataset and to allow for better model training. Finally, a set of two videos, separate to the training and testing sets, was used to test each model's ability to generalise to unseen data and to obtain a reliable evaluation of each model's performance.

The performance of each model was originally evaluated using the k-fold cross-validation technique. Based on this evaluation technique, the 3D frame differencing and reference frame CNN models were best performing, having the smallest bias and variance. Conversely, the 3D CNN with depth 6 was the worst performing, as it had the greatest bias and variance. Based on the results of applying the cross-validation evaluation technique, the other models had an equal and average performance. However, the high probability of adjacent video frames ending up in both training and test sets meant that conclusions regarding the performance of all the models could not reliably be drawn.

A different technique was therefore used, in an attempt to more reliably evaluate the performance of the models. This technique focused on the ability of the models to generalise to unseen data, which was separate to the training and the test sets of the previous technique. The 3D CNN with depth 6 achieved an accuracy of 98.28% and 81.90% for the two unseen videos and therefore generalised the best out of all the models. Other models that performed well were the 3D CNN with depth 2, the 3D CNN implemented as a 2D CNN and the 2D CNN. The performance of these models improved by an average of 4.69% when the moving average function was used. Despite this improvement, the 3D CNN with depth 6 remained the outperformer. The other models generalised poorly with the 3D frame differencing and reference frame CNN models failing to classify a single fire frame in either of the two unseen videos tested.

A thorough analysis was conducted of the performance results produced by applying the two techniques discussed above. The analysis drew conclusions between a model's performance and its CNN structure. For the 2D CNN, the dataset used was too limited for the model to learn all spatial features of fire. Considering this, its performance was average, but could be improved with the use of a wider variety of data augmentation techniques. The 3D CNN implemented as a 2D CNN outperformed the 2D CNN model due to the added dimension, which allowed the model to learn from both the spatial and temporal features of fire. However, considering this model is based on a 2D CNN it too would require an augmented dataset to improve performance. The frame differencing model was ineffective as it struggled to distinguish features of fire in the presence of a variety of smoke-like moving objects such as clouds. The reference frame CNN also failed to learn to identify the temporal differences between the first no fire frame and all other frames. The kernel sizes used in the structure of the 3D CNN models directly affected their performance. The kernels for the 3D CNN of depth 2 generalised well to videos with many fire frames, however, the short temporal length led to poorer performance for videos with fewer fire frames. The worst performing model, namely the 3D CNN with depth 4, struggled to utilise the temporal information using kernel size of 3x3x2 and 2x2x2. On the other hand, the 3D CNN model with depth 6 outperformed all models, using kernel sizes of 3x3x3 and 2x2x3. This model had the right temporal length to fully take advantage of the limited temporal and spatial information that was provided in both unseen videos.

Based on the scope of this paper and the datasets available for use in the field of vision-based fire detection, it can be concluded that the 3D CNN model with depth 6 is the most appropriate interpretation of the hybrid approach for this field. This conclusion is supported by this model's accuracy, early detection of fire and low false positive rates. In addition, this model outperformed the rest given its reliable generalisation to unseen data and its ability to perform well using a limited (HPWREN) dataset.

Chapter 7

Recommendations

> 'You can't go back and change the beginning, but you can start where you are and change the ending.'
>
> — *C.S. Lewis*

7.1 An Overview

As a result of the conclusion of this paper being specific to its scope and objectives, the author would like to put forward recommendations for wider and further research in this field. The recommendations put forward cover the potential to improve the generalisation of all the models, especially the 3D CNN model with depth 6, in order to widen the applications to which they can be transposed. In addition, the recommendations consider expanding the scope of the project to include fires that occur during the night.

7.2 Improving the Generalisation of Models

The generalisation of the models tested as possible interpretations of the hybrid approach for vision-based fire detection in this paper, most specifically the best performer of these models (the 3D CNN with depth 6), can be further improved to better transpose to different applications, i.e. different scenes to that of the mountainous region of the HPWREN dataset. While better generalisation can be achieved by increasing the size of the dataset to incorporate new video fire scenes, these scenes are currently unavailable.

This poses new challenges, because for the different scenes where fire videos do not exist, the model can only learn features from a no fire video. However, several ways to overcome this issue are proposed. The HPWREN dataset consists of a wide variety of fires that have differing static and dynamic features such as shape, size, colour and speed of elevation. These fires can be extracted from the videos in the HPWREN dataset so that a more complex augmentation than the one used in this paper can be applied to scenes that do not have fire data. Superimposing the fires from the HPWREN videos onto a new scene means that synthetic fire data can be created on which the models can be trained. As a result, the 3D CNN model with depth 6 can be trained to detect fire on a variety of different applications where fire may not yet have occurred before. Applying this complex augmentation technique would also further increase the size of the training dataset. It would be interesting to test whether data augmentation would significantly improve the generalisation of the best performing model from this paper's analysis, namely the 3D CNN model with depth 6. Furthermore, valuable insights could be

created by considering whether data augmentation would improve the generalisation of the average and worse performing models from this paper to such an extent that the 3D CNN model with depth 6 is no longer the best performer for video-based fire detection.

7.3 Expanding the Scope

The scope of this paper is to propose the best interpretation of the hybrid approach for vision-based fire detection based on video footage captured by stationary cameras during the day. However, fires can also occur at nighttime and the scope of future papers on this topic could therefore be expanded to include nighttime fire scenes. The issue with identifying fire at night is the lack of training data that is available; there is an even smaller amount of available datasets of nighttime fire scenes than the datasets of daytime fire scenes used in this paper. As a result, the complex data augmentations that are recommended in the previous paragraph can be used to synthesise new nighttime fire data. Fires at night may include different fire features that are more recognisable than those from the daytime. For example, this paper focused on the smoke characteristic of fire as it is most distinguishable at a far distance and during the day. However, smoke might be difficult to identify at night and so the bright orange flame characteristic of fire may be preferable. This would require the retraining and optimisation of the 2D and 3D models used in this paper, with different pre-processing techniques, such as the infra-red spectrum, needing to be applied to the data.

Bibliography

[1] U. S. E. P. Agency. (2022) Climate change indicators: Wildfires. [Online]. Available: https://19january2021snapshot.epa.gov/climate-indicators/climate-change-indicators-wildfires_.html

[2] S. S. Incorporation. (2022) Are fire alarms connected to the fire department? [Online]. Available: https://www.statesystemsinc.com/blog/are-fire-alarms-connected-fire-department/

[3] M. Chandrashekhar. (2021) Why do smoke alarms keep going off even when there's no smoke? [Online]. Available: https://theconversation.com/why-do-smoke-alarms-keep-going-off-even-when-theres-no-smoke-152526

[4] T.-H. Chen, Y.-H. Yin, S.-F. Huang, and Y.-T. Ye, "The smoke detection for early fire-alarming system base on video processing," in *2006 international conference on intelligent information hiding and multimedia*. IEEE, 2006, pp. 427–430.

[5] C. for Disastar Philanthropy (CDP). (2022) 2022 north american wildfires. [Online]. Available: https://disasterphilanthropy.org/disasters/2022-north-american-wildfires/

[6] P. M. Group. (2022) Informal settlements. [Online]. Available: https://pmg.org.za/page/Informal%20Settlements

[7] D. K. Kimemia and A. van Niekerk, "Energy poverty, shack fires and childhood burns," *SAMJ: South African Medical Journal*, vol. 107, pp. 289 – 291, 04 2017. [Online]. Available: http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S0256-95742017000400010&nrm=iso

[8] Burnshield. (2022) Paraffin is life in south africa. [Online]. Available: https://www.burnshield.com/paraffin-is-life-in-south-africa/

[9] T. Anwar. (2021) Introduction to video classification and human activity recognition. [Online]. Available: https://learnopencv.com/introduction-to-video-classification-and-human-activity-recognition/

[10] S. Geetha, C. Abhishek, and C. Akshayanat, "Machine vision based fire detection techniques: a survey," *Fire Technology*, vol. 57, no. 2, pp. 591–623, 2021.

[11] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboğlu, B. U. Töreyin, and S. Verstockt, "Video fire detection–review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, 2013.

[12] H. Wang, E. R. Hawkes, J. H. Chen, B. Zhou, Z. Li, and M. Aldén, "Direct numerical simulations of a high karlovitz number laboratory premixed jet flame–an analysis of flame stretch and flame thickening," *Journal of Fluid Mechanics*, vol. 815, pp. 511–536, 2017.

[13] D. Rasbash and D. Drysdale, "Fundamentals of smoke production," *Fire Safety Journal*, vol. 5, no. 1, pp. 77–86, 1982. [Online]. Available: https://www.sciencedirect.com/science/article/pii/037971128290008X

[14] J. R. H. K. H. E. K. M. P. J. T. J. M. W. C. W. Morgan J. Hurley, Daniel Gottuk, "Sfpe handbook of fire protection engineering," in *National Fire Protection Association*. Springer New York, NY, 2008.

[15] F. Gómez-Rodríguez, B. Arrue, and A. Ollero, "Smoke monitoring and measurement using image processing. application to forest fires," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5094, 09 2003.

[16] Z. Xu and J. Xu, "Automatic fire smoke detection based on image visual features," in *2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, 2007, pp. 316–319.

[17] S. Wang, Y. He, J. J. Zou, D. Zhou, and J. Wang, "Early smoke detection in video using swaying and diffusion feature," *Journal of Intelligent & Fuzzy Systems*, vol. 26, no. 1, pp. 267–275, 2014.

[18] M. Hussain, J. J. Bird, and D. R. Faria, "A study on cnn transfer learning for image classification," in *UK Workshop on computational Intelligence*. Springer, 2018, pp. 191–202.

[19] R. C. Gonzalez, "Deep convolutional neural networks [lecture notes]," *IEEE Signal Processing Magazine*, vol. 35, no. 6, pp. 79–87, 2018.

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[21] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[22] P. Foggia, A. Saggese, and M. Vento, "Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion," *IEEE TRANSACTIONS on circuits and systems for video technology*, vol. 25, no. 9, pp. 1545–1556, 2015.

[23] R. D. Lascio, A. Greco, A. Saggese, and M. Vento, "Improving fire detection reliability by a combination of videoanalytics," in *International Conference Image Analysis and Recognition*. Springer, 2014, pp. 477–484.

[24] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep cnn-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2018.

[25] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "Firenet: A specialized lightweight fire & smoke detection model for real-time iot applications," *arXiv preprint arXiv:1905.11922*, 2019.

[26] D. F. Yuan. Video smoke detection. [Online]. Available: http://staff.ustc.edu.cn/%7eyfn/vsd.html

[27] P. A. E. Cetin. (2008) Computer vision based fire detection software. [Online]. Available: http://signal.ee.bilkent.edu.tr/VisiFire/

[28] B. C. Ko, J.-Y. Kwak, and J.-Y. Nam, "Wildfire smoke detection using temporospatial features and random forest classifiers," *Optical Engineering*, vol. 51, no. 1, p. 017208, 2012. [Online]. Available: https://doi.org/10.1117/1.OE.51.1.017208

[29] KMU-CVPR. (2012) Kmu fire smoke database. [Online]. Available: https://cvpr.kmu.ac.kr/Dataset/Dataset.htm

[30] Q. Z. (2018) Research webpage about smoke detection for fire alarm: Datasets. [Online]. Available: http://smoke.ustc.edu.cn/datasets.htm

[31] T. H. P. W. Research and E. Network. (2021) The hpwren fire ignition images library for neural network training. [Online]. Available: http://hpwren.ucsd.edu/HPWREN-FIgLib/

[32] ——. (2022) The high performance wireless research and education network. [Online]. Available: http://hpwren.ucsd.edu

[33] S. Aslan, U. Gudukbay, B. Töreyin, and A. Cetin, "Deep convolutional generative adversarial networks for flame detection in video," 12 2020.

[34] J. Gubbi, S. Marusic, and M. Palaniswami, "Smoke detection in video using wavelets and support vector machines," *Fire Safety Journal*, vol. 44, no. 8, pp. 1110–1115, 2009.

[35] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[36] Paperspace. (2022) Cloud computing, evolved. [Online]. Available: https://www.paperspace.com

[37] TensorFlow. (2022) Introduction to tensorflow. [Online]. Available: https://www.tensorflow.org/learn

[38] V. Agarwal. (2020) Complete image augmentation in opencv. [Online]. Available: https://towardsdatascience.com/complete-image-augmentation-in-opencv-31a6b02694f5

[39] J. Brownlee. (2021) How to choose an activation function for deep learning. [Online]. Available: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/

[40] S. SHARMA. (2017) Activation functions in neural networks. [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Appendix A

Detailed HPWREN Video List

Table A.1: A summary of the HPWREN video dataset used in this thesis.

Video Name	**Caption**	**Resolution**	**Frames**	**Frame Rate**
20160604-FIRE-rm-n-mobo-c	Video 1	2048 × 1536	120	4
20160604-FIRE-smer-tcs3-mobo-c	Video 2	3072 × 2048	116	4
0160619-FIRE-lp-e-iqeye	Video 3	1600 × 1200	60	4
20160619-FIRE-om-e-mobo-c	Video 4	2048 × 1536	120	4
20160619-FIRE-pi-s-mobo-c	Video 5	2048 × 1536	120	4
20160718-FIRE-lp-n-iqeye	Video 6	1600 × 1200	60	4
20160718-FIRE-mg-s-iqeye	Video 7	1600 × 1200	58	4
20160718-FIRE-mw-e-mobo-c	Video 8	2048 × 1536	118	4
20161113-FIRE-bl-n-mobo-c	Video 9	2048 × 1536	120	4
20161113-FIRE-bm-w-mobo-c	Video 10	2048 × 1536	118	4
20170519-FIRE-rm-w-mobo-c	Video 11	2048 × 1536	120	4
20170520-FIRE-lp-s-iqeye	Video 12	1600 × 1200	116	4
20170520-FIRE-pi-w-mobo-c	Video 13	2048 × 1536	116	4
20170609-FIRE-sm-n-mobo-c	Video 14	2048 × 1536	120	4
20170613-FIRE-bh-w-mobo-c	Video 15	2048 × 1536	120	4
20170613-FIRE-hp-n-mobo-c	Video 16	2048 × 1536	118	4
20170625-BBM-bm-n-mobo	Video 17	2048 × 1536	60	4
20170627-FIRE-om-e-mobo-c	Video 18	2048 × 1536	120	4
20170711-FIRE-bl-e-mobo-c	Video 19	2048 × 1536	120	4
20170711-FIRE-bl-s-mobo-c	Video 20	2048 × 1536	120	4
20170711-FIRE-sdsc-e-mobo-c	Video 21	2048 × 1536	120	4
20170722-FIRE-bm-n-mobo-c	Video 22	2048 × 1536	118	4
20170722-FIRE-hp-e-mobo-c	Video 23	2048 × 1536	116	4
20170722-FIRE-mg-n-iqeye	Video 24	2560 × 1920	120	4
20170722-FIRE-so-s-mobo-c	Video 25	2048 × 1536	120	4
20170807-FIRE-bh-n-mobo-c	Video 26	2048 × 1536	114	4

Video Name	Caption	Resolution	Frames	Frame Rate
20170826-FIRE-tp-s-mobo-c	Video 27	2048 × 1536	120	4
20170927-FIRE-smer-tcs9-mobo-c	Video 28	3072 × 2048	120	4
20171010-FIRE-hp-w-mobo-c	Video 29	2048 × 1536	120	4
20171010-FIRE-rm-e-mobo-c	Video 30	2048 × 1536	120	4
20171021-FIRE-pi-e-mobo-c	Video 31	2048 × 1536	118	4
20171026-FIRE-rm-n-mobo-c	Video 32	2048 × 1536	116	4
20171026-FIRE-smer-tcs8-mobo-c	Video 33	3072 × 2048	120	4
20180603-FIRE-sm-w-mobo-c	Video 34	2048 × 1536	118	4
20190712-RockHouse-wc-e-mobo-c	Video 35	3072 × 2048	120	4
20190717-FIRE-lp-n-mobo-c	Video 36	3072 × 2048	120	4
20180727-FIRE-wc-n-mobo-c	Video 37	3072 × 2048	120	4
20190728-FIRE-om-n-mobo-c	Video 38	3072 × 2048	120	4
20190809-PinosSouth-pi-s-mobo	Video 39	2048 × 1536	58	4
20190813-Topanga-69bravo-n-mobo	Video 40	3072 × 2048	120	4
20190813-FIRE-69bravo-e-mobo-c	Video 41	3072 × 2048	118	4
20190814-Border-lp-s-mobo	Video 42	3072 × 2048	120	4
20190814-FIRE-om-e-mobo-c	Video 43	3072 × 2048	120	4
20190825-FIRE-sm-w-mobo-c	Video 44	2048 × 1536	114	4

Appendix B

GitHub Repository

I have created a Vision-Based Fire Detection GitHub repository for this thesis, which contains all the Jupyter notebooks that were used during the experiments. The list of videos used, as found in Appendix A, can also be found on the repository along with all the various performance results for this thesis.

Appendix C

YouTube Channel

The unseen videos that were used to test the generalisation of the models can be found on my YouTube channel. Additionally, the various output videos containing the frame predictions and frame classification can be found on the same channel.

Appendix D

Ethics Form

Application for Approval of Ethics in Research (EiR) Projects
Faculty of Engineering and the Built Environment, University of Cape Town

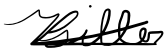
ETHICS APPLICATION FORM

Please Note:
Any person planning to undertake research in the Faculty of Engineering and the Built Environment (EBE) at the University of Cape Town is required to complete this form **before** collecting or analysing data. The objective of submitting this application *prior* to embarking on research is to ensure that the highest ethical standards in research, conducted under the auspices of the EBE Faculty, are met. Please ensure that you have read, and understood the **EBE Ethics in Research Handbook** (available from the UCT EBE, Research Ethics website) prior to completing this application form: http://www.ebe.uct.ac.za/ebe/research/ethics1

APPLICANT'S DETAILS		
Name of principal researcher, student or external applicant		
Department		
Preferred email address of applicant:		
If Student	Your Degree: e.g., MSc, PhD, etc.	
	Credit Value of Research: e.g., 60/120/180/360 etc.	
	Name of Supervisor (if supervised):	
If this is a researchcontract, indicate the source of funding/sponsorship		
Project Title		

I hereby undertake to carry out my research in such a way that:

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

APPLICATION BY	Full name	Signature	Date
Principal Researcher/ Student/External applicant			
SUPPORTED BY	Full name	Signature	Date
Supervisor (where applicable)			

APPROVED BY	Full name	Signature	Date
HOD (or delegated nominee) Final authority for all applicants who have answered NO to all questions in Section 1; and for all Undergraduate research (Including Honours).			
Chair: Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the questions in Section 1.			