Ahmet K. AĞIRMAN

A Ph.D. Thesis

AGU 2022

NIGHTTIME FIRE DETECTION FROM VIDEO

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Ahmet K. AĞIRMAN
June 2022

NIGHTTIME FIRE DETECTION FROM VIDEO

A THESIS
SUBMITTED TO THE DEPARTMENT OF XXX
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Ahmet K. AĞIRMAN
June 2022

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name-Surname: Ahmet K. AĞIRMAN

Signature :

REGULATORY COMPLIANCE

Ph.D. thesis titled Nighttime Fire Detection from Video has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Ahmet K. AĞIRMAN	Asst. Prof. Kasım TAŞDEMİR

Head of the Electrical and Computer Engineering Program
Assoc. Prof. Kutay İÇÖZ

ACCEPTANCE AND APPROVAL

Ph.D. thesis titled Scene Perception in Low Light Conditions and prepared by Ahmet K. AĞIRMAN has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

30/06/2022

(Thesis Defense Exam Date)

JURY:

Advisor : Assist. Prof. Kasım TAŞDEMİR

Member : Prof. Behçet Uğur TÖREYİN

Member : Assoc. Prof. Ahmet Turan ÖZDEMİR

Member : Assist. Prof. Gülay YALÇIN ALKAN

Member : Assist. Prof. Burcu BAKIR GÜNGÖR

APPROVAL:

The acceptance of this Ph.D. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / …….. and numbered ……..……. .

…….. /…….. / ……..

(Date)

Graduate School Dean
Prof. Dr. İrfan ALAN

ABSTRACT

NIGHTTIME FIRE DETECTION FROM VIDEO

Ahmet K. AĞIRMAN
Ph.D. in Electrical and Computer Engineering
Advisor: Asst. Prof. Kasım TAŞDEMİR

June 2022

With the recent advancements in the field of Computer Vision, the central tasks such as object detection, segmentation or object tracking methods attain all-time high accuracies in natural image sets such as ImageNet, COCO, etc. However, due to the innate downsides of digital images acquired in insufficiently illuminated environments, the conventional methods suffer severely. This specific problem remains unsolved. Especially if the environment is pitch dark and the object of interest is emitting light, the dynamic range of the current digital cameras falls short in this situation and the generated digital image contains almost no perceptible visual texture. One prominent example of this is nighttime forest fire videos. In this thesis, detection of nighttime forest fires from video is addressed as an application of the challenging task, scene perception in low light conditions.

The first contribution of this dissertation is developing a novel object tracking algorithm for glowing object in the dark environments. The algorithm allows to track fire and non-fire objects throughout the video. The second contribution of the thesis is proposal of new handcrafted features which are designed to capture spatio-temporal behavior of the glowing objects since there is little or no visual textures to be processed. The results showed that the features are descriptive enough to distinguish fire from the other deceptive light sources. The third contribution is employing deep learning models to automatically extract spatial features with CNNs, and temporal features from bi-directional Long Short-Term Memory (BLSTM) networks. The empirical test results show that a CNN + BLSTM pipeline can effectively detect fires at night with a high accuracy. Finally, a new comprehensive nighttime fire video dataset comprising 1358 positive videos and 334535 of fire frames is created.

Keywords: SVM, CNN, BLSTM, nightfire, VFD

ÖZET

VİDEODAN GECE YANGIN TESPİTİ

Ahmet K. AĞIRMAN
Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı Doktora
Tez Yöneticisi: Dr. Öğr. Üyesi Kasım TAŞDEMİR

Haziran-2022

Bilgisayarlı Görü alanındaki son ilerlemeler, ImageNet, COCO, vb. doğal görüntü veri setleri üzerinde nesne tespiti, bölütlendirme, nesne takibi gibi merkezi işlemlerde tüm zamanların en yüksek doğruluklarına ulaşmaktadır. Fakat yetersiz ışığa sahip ortamlardan elde edilmiş dijital görüntüler üzerinde özünde var olan dezavantajlardan geleneksel yöntemler ciddi bir şekilde zorluk yaşamaktadır. Bu belirli problem henüz çözülememiştir. Özellikle ortam zifiri karanlık ve hedef nesne ışık yayıyorsa günümüz dijital kameraların dinamik menzili bu duruma yetersiz kalmakta ve elde edilmiş dijital görüntüler neredeyse hiç algılanabilir görsel doku taşımamaktadır. Buna önemli bir örnek, gece yangını videolarıdır. Bu tezde, düşük ışık şartlarında sahne algısı zorlu probleminin bir uygulaması olarak videolardan gece orman yangını tespiti sorunu üzerine gidilmiştir.

Bu tezin ilk katkısı, karanlık ortamlarda parlayan nesnelerin takibini sağlayan bir algoritmanın geliştirilmesidir. Algoritma, yangın ve yangın olmayan nesnelerin video boyunca takibini sağlamaktadır. Tezin ikinci katkısı ise elle oluşturulmuş yeni öznitelikler ile işlenecek görsel doku neredeyse hiç olmadığından parıldayan nesnelerin zamansal ve uzamsal davranışını yakalanmasıdır. Sonuçlar göstermiştir ki bu öznitelikler yangını sahnedeki diğer çeldirici ışık kaynaklarından ayırt etmede yeterince betimleyicidir. Üçüncü katkı ise karanlık videolardan otomatik uzamsal öznitelik çıkarmak için CNN'ler, zamansal davranışı yakalayabilmek için de iki yönlü uzun kısa süreli bellek (BLSTM) kullanılmasıdır. Ampirik deney sonuçları göstermiştir ki CNN+BLSTM düzeneği gece yangınlarını etkin bir şekilde ve yüksek doğrulukta tespit edebilmektedir. Son olarak, 1358 pozitif video ve 334535 alev çerçevesinden oluşan kapsamlı bir gece orman yangını veri seti derlenmiş ve kullanışlı hale getirilmiştir.

Anahtar kelimeler: SVM, CNN, BLSTM, gece yangını, VFD

Acknowledgements

Writing this dissertation is camping one week at the top of the Mount Erciyes. However, climbing to the that point required unutterable experiences which had to take reasonably long time. During this experience, there have always been people around me who pushed me up when had blues. I would like to acknowledge them here.

Before all, I would like to thank my advisor, Assist. Prof. Kasım Taşdemir, for being endlesly positive and supportive during my study. He was always standing next to the desk whenever I switched of the light and look for someone's help. I also would like to thank to Assoc. Prof. Murat AYDIN for helping me find computer stations for experiments. Especially, during the first year of the COVID-19 pandemic, it has been a life-saver. I would like to also thank to Assist. Prof. Gülay YALÇIN ALKAN and Assoc. Prof. Ahmet Turan ÖZDEMİR for guiding and supporting me along with my advisor during my study patiently. I also would like to thank Dr. Surasak Chunsrivirot for his support that at the times I was down, he was never tired saying that 'Yes you can!'.

I would like to thank to my parents, wife and son for supporting me at heart and experiencing each moment of this journey with me. Their support under all conditions, patience for me waiting in the corner of the end, and sacrifice from many aspect of their life is immense. This thesis would be never possible without them.

And finally, I thank to everyone who supported and wished the best for me during my study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AB	AdaBoost
BLSTM	Bi-directional Long Short-Term Memory
BoF	Bag of Features
CC	Creative Commons
CCD	Charge-Coupled Device
CCTV	Closed-Circuit Television
CNN	Convolutional Neural Networks
COCO	Common Objects in Context
CT	Computer Tomography
DOF	Depth of Field
FCC	Fully Connected Convolution
FinD	Fire in Dark
FPGA	Field-Programmable Gate Array
HDR	High Dynamic Range
HoF	Histogram of Oriented Features
IBk	Instance-Based learning with parameter k
ILSVRC	Large-Scale Visual Recognition Challenge
IOU	Intersection Over Union
ISO	International Organization for Standardization
J48	C4.5 algorithm
LIDAR	Light Detection and Ranging
LMS	Least Mean Square
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
MRI	Magnetic Resonance Imaging
MV	Majority Voting
NaN	Not a Number
NIST	The National Institute of Standards and Technology
RADAR	Radio Detection and Ranging
RBF	Radial Based Function
RELU	Rectifier Linear Unit

RF	Random Forests
RGB	Red, Green, Blue
RNN	Recurring Neural Network
SVM	Support Vector Machines
TPR	True Negative Rate
TPR	True Positive Rate
TSY	The Standard YouTube (License)
UAV	Unmanned Aerial Vehicle
VFD	Video Fire Detection
VS	Visual Spectrum
VSD	Video Smoke Detection
WDR	Wide Dynamic Range
YOLO	You Look Only Once

To my dear father, mother, wife, and son

Chapter 1

Introduction

Perception of real-world scenes has been an exciting yet a complex topic of psychology for decades. Visual scene perception is human's understanding of the environment as she views it. As psychologists have been trying to thoroughly understand various aspects of human visual perception, computer science community has been quite busy about perception of environment not by humans but machines.

Computer vision is one subfield of machine perception, extracting information from digital images. It can be contrasted to the human perception where images are captured by eyes and information extraction is done in brain. A computer typically creates an image by capturing a real-world scene through a sensor, then converting it to digital signals, and finally storing it in a digital memory. Even though human visual perception is limited to visual spectrum (VS) of full light spectrum, computer scene perception can work with wide range of electromagnetic spectrum and images acquired from different sensors such as VS cameras, LIDARs, RADARs, IR sensors, ultrasonic sensors, acoustic sensors, MRI scanners, X-RAY receptors, CT scanners, etc.

Automated learning from sensed data and making decisions upon gathered knowledge has been possible after emergence of machine learning methods. After the recent revolution of deep learning, the computation costs have reduced significantly [1]. Computer vision has been an enabler of many emerging technologies: On healthcare, automated drug design, automated diagnosis by IBM's Watson; on precision agriculture, assessing crop health and/or yield state, automated agricultural robot steering; on environmental protection, automated wildfire detection from cameras both on day and nighttime, observing oceans and wild life; on finance, making video conference with virtual assistants; on transportation, using automated vehicles, trucks, and buses, etc. are some limited number of examples can daily be found on news streams.

Artificial intelligence captured momentum in history a couple of times leading humanity to think about the machines entering singularity, that is, machine intelligence

exceeding human intelligence. However, this has never been the case. Today, renaissance of deep learning on computer vision escalated such expectations again [2].

A fundamental example that singularity is not the case at least for decades should be poor computer understanding of images captured from scenes in adverse environments and in low-light conditions. Relevant to orientation of this dissertation, a concrete example is fires taking place at night.

Growing number of fires for each year in the last decade eradicating our forests faster than ever together with all natural and wildlife existing in. When coupled with adverse climate conditions, it is almost impossible to stop these devastating events and firefighting agents are irremediably left with a painful wait that they must watch the fire completely burn down an entire forest ecosystem and finally die out when there is no wood, house, car, and finally memories to burn.

For this reason, responding the fire events timely has been a top priority more than ever. Responding that need, historically, many lookout towers were built and a human lookout living there had to observe forests for long hours especially at fire seasons. This method was effective for some time in the midst of unfavorable human factors. Emergence and later widespread use of surveillance cameras replaced many lookout towers & jobs and made forest observation relatively easier. However, watching too many screens and cameras by a limited number of human operators is also not feasible. This is the moment automatic fire detection algorithms enter the scene. These algorithms are being developed by researchers at least three decades. Today these systems are backbone of the automatic forest surveillance against the fire events.

Today's world liberated the data and the data liberated the artificial intelligence. In other words, deep learning contributed immensely to video fire detection (VFD) with more than three hundred fifty research papers to the date. However, a limited number of them address the detection of nighttime fire events. Fire videos at daytime can deliver many useful spatial, color, and texture features for an effective detection. Besides flames, smoke is a key indicator and target object to detect fire events at daytime. Unfortunately, nighttime fires do not include rich color and texture features compared to daytime fires. What's worse, smoke is not effectively visible, thus not perceptible at night. In nighttime fires, detection mostly relies on flame movement and binary-like color information against the background. However, movement of flames can easily be confused with headlights and other light sources. Therefore, specialized algorithms should be developed for fire detection at night. Furthermore, building effective deep learning models require

well-organized representative datasets. Today's world not only can generate fire data from fixed CCTV cameras, but also from various other sources like mobile phone cameras, UAV cameras, on-vehicle patrolling cameras, etc. Analysis of videos recorded from a fixed CCTV camera is relatively easy since, the camera is stationary, and it is straightforward to eliminate background to work on target objects in the scene. These cameras also generate almost same distant views which is not the case for non-stationary cameras. Therefore, algorithms should also adapt the new nature of the data intended to be used for model building.

1.1 Structure of the Dissertation

This work is organized as following. In Chapter 1, significance of fire disasters is addressed. Fundamental questions that are investigated in this thesis are given and contributions are presented.

In Chapter 2, the relation between nighttime visual environmental scenes and capturing them via digital imaging devices, i.e., cameras, are discussed. In Chapter 2.1, camera features and their adjustment for the nighttime are given. In Chapter 2.2, what attributes night fires show on digital images both due to camera capabilities and environmental conditions are illustrated. This chapter points out the challenges of fire detection at night in terms of these attributes.

In Chapter 3, the methods frequently used for fire detection problem and their underlying theory are explained. For crafting features by hand, conventionally, SVM had been a popular method in the literature. In Chapter 3.1, SVM is briefly explained and showed that it can classify linearly not separable data by mapping it to an upper space. When useful features are desired to be extracted automatically, then deep learning methods are practical. In Chapter 3.2, theory of deep learning methods that are frequently applied to VFD problems are provided. For spatial feature analysis, CNNs and for temporal feature analysis, RNNs are explained. In chapter 4, a literature revie of nighttime VFD techniques are given.

In chapter 5, two fire datasets are proposed. The first dataset, FinD Dataset Set1, is generated by the author and Assist. Prof. Kasım TAŞDEMİR, by starting controlled natural fires by burning wood and fodder at night. The second dataset, compiled by the author, is a collection of videos recoded for real world fire disasters.

In Chapter 6, a feature set for night fire detection is designed and used for automatic fire detection by training an SVM model. These features are mostly based temporal behavior of the fire blobs through a video. Later, further models like RF are used to test the proposed features. The test results showed that these temporal features are useful for nighttime fire detection.

In Chapter 7, deep learning methods are used for automatic feature extraction, instead of designing them by hand. To accomplish this, both spatial and temporal feature extracting methods are used in a pipeline. For spatial feature extraction, a CNN model and for temporal analysis, a RNN model is used. The test results showed that this pipeline can attain a high accuracy in a considerably short time.

Finally, in Chapter 8, conclusions drawn from this dissertation are given, expected social impact is discussed and comments on potential future directions are given.

1.2 Fire Disasters

From 1988 to date, 63480 forest fires occurred and roughly 313.000 hectares of forest destroyed in Turkey. Steady decrease of hectares destroyed per fire from 13 to 4.9 throughout the years shows an effective fire response, but number of fires per year is on a slight increase [3]. US also has similar statistics that decline of its rate of destroyed area per fire is on a stall (Figure 1.1) [4]. Besides enhancements on firefighting techniques and development of related technology, this progress is apparently due to efficient surveillance techniques and timely reports. Furthermore, 20% of reported forest fires in Turkey and 48% of city fires in Istanbul occurred at no daylight conditions since July 2020 and January 2020, respectively [5, 6].

Wildfire is a significant threat worldwide and among the significant devastating natural disasters that can have immediate and long-term effects on the environment, the people, and the economy [7]. In favorable conditions, the spread speed of bushfires can attain as high as **24** km/h which makes its suppression extremely hard [8]. Therefore, one of the most crucial steps in firefighting is early detection of the fire after the ignition. As one of the most common ways of early detection, analyzing live videos for a possible wildfire helps mitigate the severity of the aftermath of forest fires, as the statistical data indicates [3, 4].

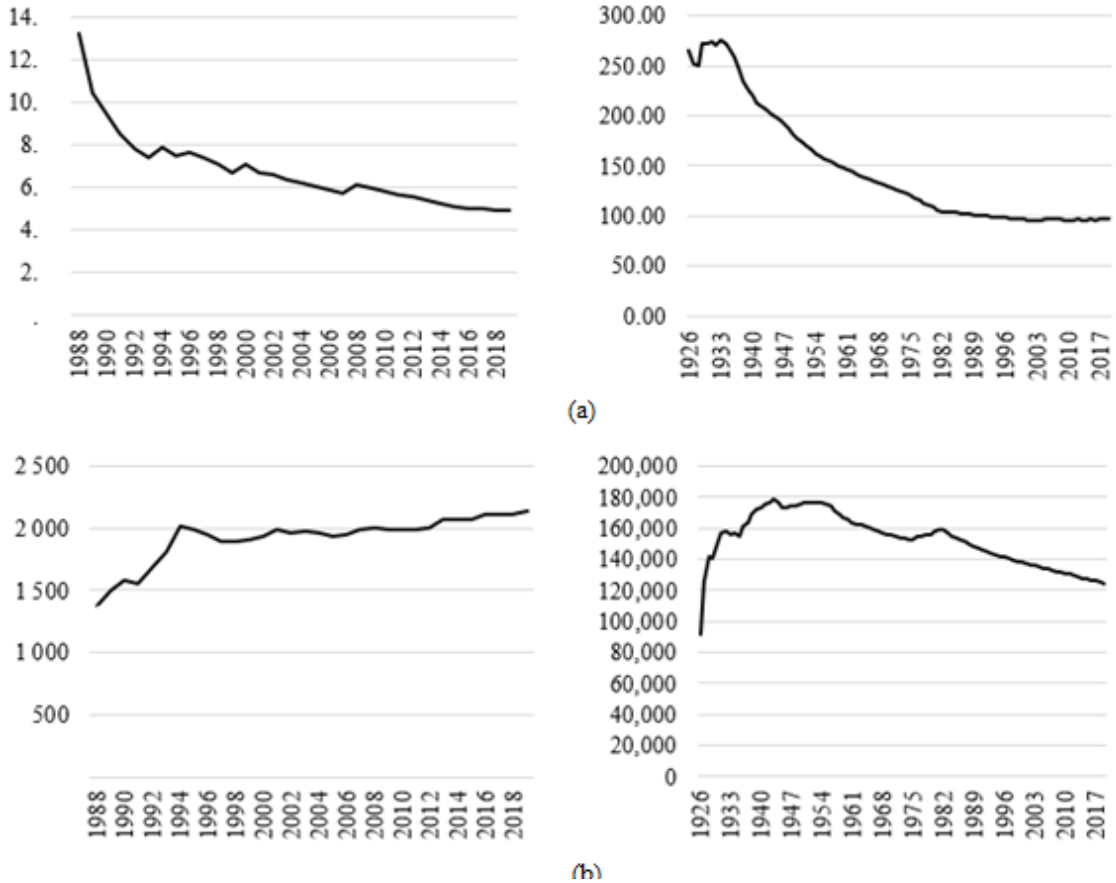

Figure 1.1 a) A running average of hectare area destroyed per fire in Turkey (left) and acre area in USA (right). b) A running average of number of fires per year in Turkey (left) and in USA (right)

If fires are not controlled soon after ignition in conditions of strong dry wind, high temperature and abundant continuous fuel, they will travel at high speed for several hours, burning out huge areas. Control of the head fire is impossible, and the area that is burnt largely depends on the time a fire starts, the period that elapses before a change of wind direction and the weather conditions after the change [8].

1.2 Problem Statement and Goal

Automated scene perception is a challenging task when the environment is under adverse conditions. Video fire detection (VFD) at night is a special sub-problem of the scene perception at low light conditions which also includes other popular detection problems of vehicle, pedestrian, object, and so on. Night-time object detection with visual spectrum (VS) cameras is a challenging task compared to its daytime counterpart since there is no sufficient light reflecting back from the target object to the camera lens. Even

if an object is detected, it is even more difficult to determine or identity of such an object in low light conditions. The fundamental reason for this is the physics of underlying mechanism of vision. It is impossible to perceive non-luminous objects with VS cameras without a distinct light source in a dark environment. Compared to non-luminous objects, fire itself is a light emitting object. Therefore, it seems detecting fire should be as trivial as detecting a light source in the dark. However, this is not always the case. Conventional day-time fire detection techniques use color and texture features effectively for smoke and fire detection which are mostly not present in night-time images which requires targeted methods for night-time fires.

For fire event detection, smoke is not a reliable target object for raising fire alarms since its features are not detectable at night. In the event of fire, depending on amount of the light emitted from the fire, density and spread of the smoke, event distance from the camera, availability of the reflective objects in surroundings like dense tall trees, etc., smoke is mostly not visible as an integral object and behaves as light diffusing agent like air in the scene. These factors do not make it a reliable target but a challenge in detecting fire events.

The only reliable object in detecting a non-urban fire event scene is the flame object which also brings its own detection challenges. These challenges can be defined based on different types of scenes, i.e., fire contour, other visible objects in the scene, event distance from the camera, etc. For example, a faraway forest fire or a forest fire recorded via an aerial vehicle will have a union of convex and concave lines creating a contour mostly following shape of the land field. However, this is not the case for close or mid-range forest fires. In the close or mid-range forest fires, there is sufficient light that other stationary or moving objects are also visible under evident effect of the smoke. This requires differentiating such objects from fire object. Another example is distinguishing fire from other light sources especially in a smokey environment. In the event of non-urban area wildfires, flames are accompanied most of the time by dense smoke which diffuses light from any source and makes surroundings of these light sources not easily detectable (Figure 2.6). Such light sources can be listed as revolving, flashing, or continuous head lights, city lights, road lights, hand or head-held lights, moon light, lightening, etc. Color and texture features of night fires are limited, therefore, besides spatial features, temporal features are central in detecting night-time fires. Some of the distinguishing temporal fire features are flickering, pointing high into the sky, dying down or flaring up of the flames, temporal disappearance of flame due to smoke accumulation.

Object detection in low-light conditions is sometimes possible but challenging depending on the light source's flux, the contrast in the scene, the objects' morphology, reflection and distance from the camera, and the light-source type which affects contrast against background. It is difficult to distinguish objects (i.e., road signs) from the background in a low contrast image without manually enhancing it, i.e., by changing color curves.

In this dissertation, we propose methods that help robust scene perception from VS cameras under low-light conditions specific to VFD at night.

The dissertation investigates prospective answers to the following questions:

- What spatio-temporal features should be used for the measurement of temporal changes in a video to detect fire using prominent machine learning algorithms, i.e., SVM, RF, etc.?
- Can an end-to-end CNN based deep learning model be used instead of hand-crafting features? And can this model automatically generate descriptive features for the optimum detection performance?
- Is temporal analysis useful for night-time VFD?
- Is the data used in the night-time VFD research is satisfactory?

1.3 Contributions

This dissertation contributes to the scientific literature in the following ways:

- It develops a method that designs features to be analyzed for fire detection via machine learning algorithms, i.e., SVM, RF, AB, IBk, etc.
- It proposes an object tracking method between consecutive frames of a video.
- It develops a pipeline consisted of CNN and BLSTM to use both spatial and temporal features for an automated feature generation and nighttime fire detection.
- It proposes a challenging real-life dataset that can be used in training, testing, and benchmarking robust night-time VFD models.

Chapter 2

Nighttime Imaging and Fires at Night

In this chapter, we evaluate limits of nighttime digital imaging in terms of sensing and sensed parts. As a sensing device, cameras, have limitations which require adjustment of its capabilities carefully. On the other hand, independent of camera capabilities, the sensed environment also has its own challenges. Added contribution of adverse effects of fire for nighttime imaging is also discussed in this chapter.

2.1 Night-Time Imaging

Capturing quality images at night is troublesome compared to day-time imaging. One can improve night-time video quality by adjusting several parameters each with its own limitations.

In night-time imaging, the fundamental item required is the light itself which turns out to be at a low amount naturally. Therefore, either natural or artificial, any light source is welcomed during a video shooting in the dark. When the light is insufficient, then the camera means should be utilized maximally to receive uttermost performance. A camera itself should embody competent features for the best results. One important feature is sensor size. The larger the sensor is, the more amount of light the camera will be able to

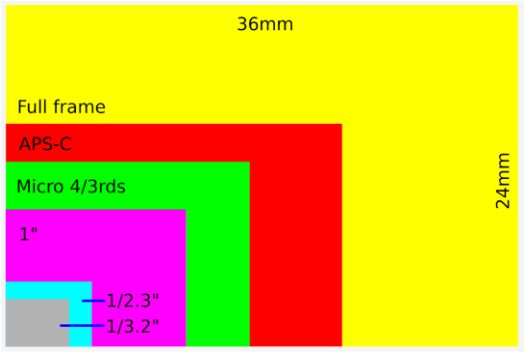

Figure 2.1 Common sensor sizes in use. 36mm x 24mm sensor is known as full frame sensor [83].

capture at an instant. A full frame (35mm) sensor is always preferred against other sensors (Figure 2.1).

Another feature is the maximum ISO gain the camera can deliver. To increase light sensitivity of the camera, the max ISO gain available should also be high. Lens type is also an important feature since it will control how much light can be received at once. A lens with a higher max aperture will let the camera sensor receive more light at an instant which in turn require a shorter stutter speed. This type of lenses is called as quick or fast lenses. The final important feature for a camera is stabilization. Camera should include stabilization features like Optical Image Stabilization or requires use of a gimbal.

Other than choosing a competent set of features for a camera, adjusting the exposure during a shooting is other side of the coin. Aperture is size of the opening pupil of a lens. It directly controls the amount of light will land onto camera sensor at an instant. A larger aperture refers to a larger opening pupil and a higher intensity of light. As an example, doubling the light intensity is doubling the pupil area, in turn increasing the pupil diameter by a factor of $\sqrt{2} \approx 1.4$ which is known as f-number or f-stop. In a low-light environment, the sensor requires more amount of light to generate a brighter image. Then increasing aperture in such condition can be preferable. In this case, one should be careful about the depth of field. Depth of the field (DOF) is the distance between closest and farthest planes that are in focus (Figure 2.2). In a shallow DOF, this distance is short, i.e., these planes are closer to each other, and the image can only show the area very close to the plane of focus sharp and remaining area blurry. On the other hand, in a deep DOF, the image can show not only plane of the focus sharp, but also a variable amount of distance at behind and front of the plane of focus. Then, when the DOF is shallow, which is the case when the aperture is wide; one should be cautious that the interested scene should be in a required distance to let the narrow DOF field can contain it. This implies that a moving object can easily go out of DOF and become blurry when the camera is not moving correspondingly.

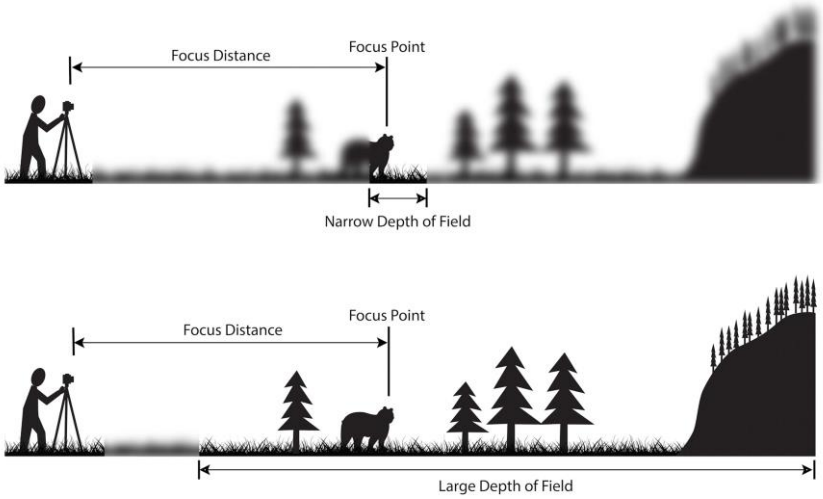

Figure 2.2 Depth of field of a camera [85]. If target object is not within DOF, then it will be blurred in the frame. At the top, DOF is narrow, therefore a limited volume will be clear in the image. However, at the bottom, DOF is wide, and an extended volume will be seen clear in the image.

Shutter is either mechanical or electronic mechanism that is used to allow light land onto sensor for a determined length of time. The speed the shutter opens, and closes is termed as shutter speed; thus, the lower the shutter speed, the longer the exposure time, and then the higher amount of light accumulates onto the sensor during exposure time. For a video recording, shutter speed implies that the amount of time the light is sampled during a one frame interval. In the event of night-time imaging, one wants the sensor is exposed to more light to get a brighter image. By means of shutter speed, this can be achieved by letting the shutter open as long as needed. However, longer exposure times (i.e., lower shutter speeds) will make video recordings blurry in case of motion in the scene. In order to balance the motion-blur, the exposure time can be chosen to half of the one frame interval and keep the camera stationary on for example a tripod. At the time of hand-holding a camera, an important problem is the camera shake. This subsequently requires a balance between shutter speed and focal length of the lens. The general rule is using a shutter speed equal to the focal length; however, in night-time imaging, these speeds can be yet not enough, and even faster speeds may be required in the event of moving scenes.

ISO refers to degree of the sensitivity of the camera to the light. Roughly, it is the amplifying gain applied to voltage levels of the sensor pixels. Therefore, when there is insufficient amount of light accumulated onto the sensor, by using ISO, the image can be

enhanced in terms of brightness as if it is constructed in a high amount of environmental light (Figure 2.3).

Figure 2.3 Contribution of ISO on image visibility at insufficient light [84]. In scarcity of light, ISO helps capturing more detailed images.

However, the fact that the image contains noise, higher the ISO higher the amplification, in turn, larger the grains in the images. This may require a post-processing means to eliminate noise in the image (Figure 2.4)

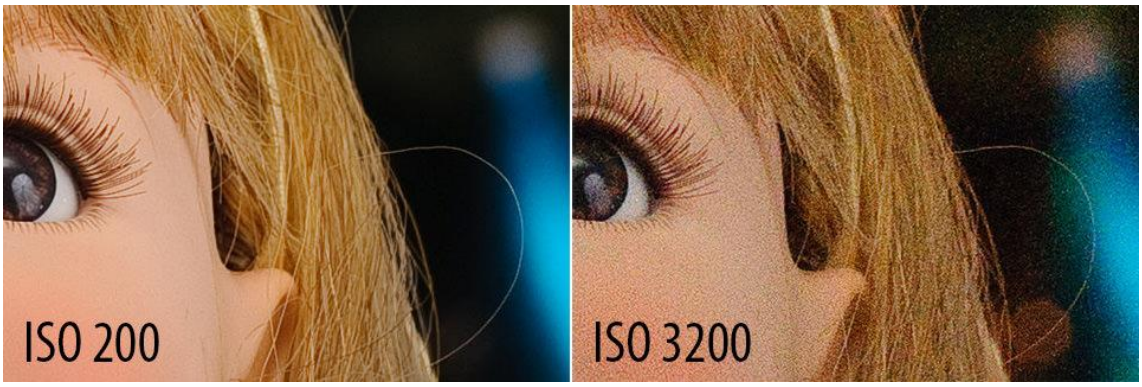

Figure 2.4 Noise contributed by large ISO [84]. High ISO values also magnifies the noise (right) while enhancing the image (left). Thus, it should be used cautiously.

Brightness of a point in a scene can be measured by luminous intensity per unit area (cd/m^2) or luminance. Most of the time not all points in a scene have the same luminance, indeed variable from a minimum value to a maximum. The amount of this variability or contrast is related to concept of dynamic range. Assuming the contrast ratio between brightest (max luminance) and darkest (min luminance) points in a scene is c (sometimes also referred to as c:1), then the dynamic range is defined in terms of stops by

$$\log_2 c \tag{2.1}$$

or in terms of decibels by

$$20\log_{10} c \tag{2.2}$$

Roughly ~0.166 times the decibels value will give the stops value of the dynamic range of the scene. Moving a stop one up or down will double or halves the brightness, respectively. A scene with 24 stops dynamic range implies that the contrast ratio is above 16,000,000:1.

Besides the scenes, cameras also have a dynamic range. Most modern high-end camera dynamic ranges are around 14 stops. Considering real life scenes can easily have much higher stops, this implies that a camera can register the light as it is only if its luminance lies within dynamic range of the camera. When a luminance value coming from a scene and hitting the camera sensor photosite (pixel) is less than the minimum luminance value of camera dynamic range, then photosite is considered as in black color and when a luminance value hitting the photosite is higher than maximum value of camera dynamic range, this time color of corresponding photosite is considered as white. This limiting constraint makes it difficult to get quality photos or videos of scenes having high contrast. In order to overcome this problem, multiple shots of the scene each focusing on different mean brightness of the scene can be used to generate a better image. This

Figure 2.5 HDR applied to a scene [86]. When dynamic range of a camera cannot cover contrast ratio of scene, it should capture either darkish (left) or brightish (right) image at a time. To overcome this, these two images can contribute their quality parts and generate an image that can illustrate dark and bright areas in great detail at the same time (middle).

technique is known as high dynamic range (HDR) and tries to enhance the image via post-processing algorithms (Figure 2.5). By employing HDR, a limited dynamic range camera can stich an image that can show texture of dark areas (i.e., shadows) better but bright areas worse and another image that can show texture of bright areas (i.e., sky and clouds) better but dark areas worse. This tries to get the best of both worlds and generate better looking images.

Another technique is wide dynamic range (WDR) and mostly employed in CCTV cameras. In this technique, the camera has dual sensors; one is capturing an image focused on dark areas in the scene and other focused on bright areas. These images then combined by an image processor for final outcome. When there is a single sensor on the CCTV camera, then camera captures the same scene at different shutter speeds multiple times. In this case, in order to realize texture of bright areas, a high shutter speed (short exposure time) lets the bright areas not blow out. Similarly, the dark areas require a slow shutter speed (long exposure time) to capture enough light to generate sufficient texture. Then the camera processor combines these images. In general, WDR technique is more successful in imaging dark scenes when compared to HDR.

A rule of thumb for capturing quality images at night is using a tripod for the camera in use, or alternatively capturing images from a fixed camera. A fixed camera makes capturing images in manual mode sound which lets adjusting exposure setting for desired results.

Even though visible range cameras are a popular option for VFD, to get non-blurry, bright, and sharp shootings at night require a stationary camera and an adaptive adjustment of shutter speed, aperture, and ISO settings for the night environment. Furthermore, the contrast of the environment shouldn't exceed the dynamic range of the camera. Nevertheless, a stationary camera can be an option for only forest fire watch towers and security cameras but not for moving land or aerial surveillance vehicles as well as mobile devices with video recording capability. Therefore, detecting fire from moving camera recordings is a great challenge for VFD, especially for the night.

2.2 Nature of Night Fires

Fire is a chemical process that takes place when a combustible agent and oxygen react in suitable conditions [9]. Fire process can emit color depending on amount of oxygen content as in Bunsen burner and combustible agent as in chemicals used for

colorful fireworks. The spectral interval of fire light is in the range from 0,4 to 14 μm while the visible spectrum lays between 0,4 and 0,7 μm [10]. This makes visible spectrum cameras a useful and budget option for fire sensing; however, the wide range of fire colors also make it difficult to train neural networks based only on image color information.

Nighttime object detection is a challenging task compared to its daytime counterpart. It is impossible to perceive non-luminous objects with RGB cameras without a distinct light source in a dark environment. Object detection in low-light conditions is sometimes possible but challenging depending on the light source's flux, the contrast in the scene, the objects' morphology, reflection and distance from the camera, and the light-source type which affects contrast against background. In Figure 2.6a, the left picture is a dark image with a low contrast against the background and the right picture is the same image manually enhanced by changing color curves. It is difficult to distinguish objects (i.e., road signs) from the background in the left image.

Another challenge is the insufficiently visible texture which makes identifying objects in their surroundings difficult. When images are blurry or in indistinguishable texture due to the camera, heavy smoke, or fog in the environment, then CNN filters will generate similar feature maps. As a result, it misleads the network to an incorrect classification. In Figure 2.6b, two pictures of scenes with an insufficient texture are given. Such images may have a fairly similar texture that makes it difficult to distinguish them from each other. The image on the left shows a fire object around a house with a significantly reduced texture. The image on the right shows a vehicle headlight. The texture of both vehicle and headlights are significantly reduced.

Figure 2.6 Challenges of nigh-time fire detection a) A dark image with a low contrast against the background (left) and the same image manually enhanced by changing color curves.; b) Images with insufficient texture.; c) Noisy night-time images.; d) An aerial image of a vehicle and the corresponding black-white image.

The dark images contain a considerable amount of noise. This noise also brings a challenge to the training process. In Figure 2.6c, the noise is mainly due to the heavy smoke coming from fires.

Nighttime images contain minimal color information compared to daytime images. This makes them close to binary images; thus, color and texture analysis becomes harder In Figure 2.6d, an aerial image of a vehicle and the corresponding black-white image are given. Smoke detection is possible with daytime videos. However, due to the lack of rich color information, this is impossible primarily for night-time videos. Cameras have a limited dynamic range. When the parameters such as exposure and ISO are set for the foreground object region, the remaining part of the scene becomes near black.

Moreover, the light sources visible in a frame introduce further challenges due to the camera's dynamic range shift. Those effects are explained in the experimental results at Section 7.4.

In some cases, the scene includes only bright or dark regions making the video akin to a binary video. Binary images give limited information about an object, including its shape and position in the frame. It lacks color and texture information which are central in object detection tasks. For example, a streetlight or a freshly ignited flame might look indistinguishable in night images (Figure 2.7).

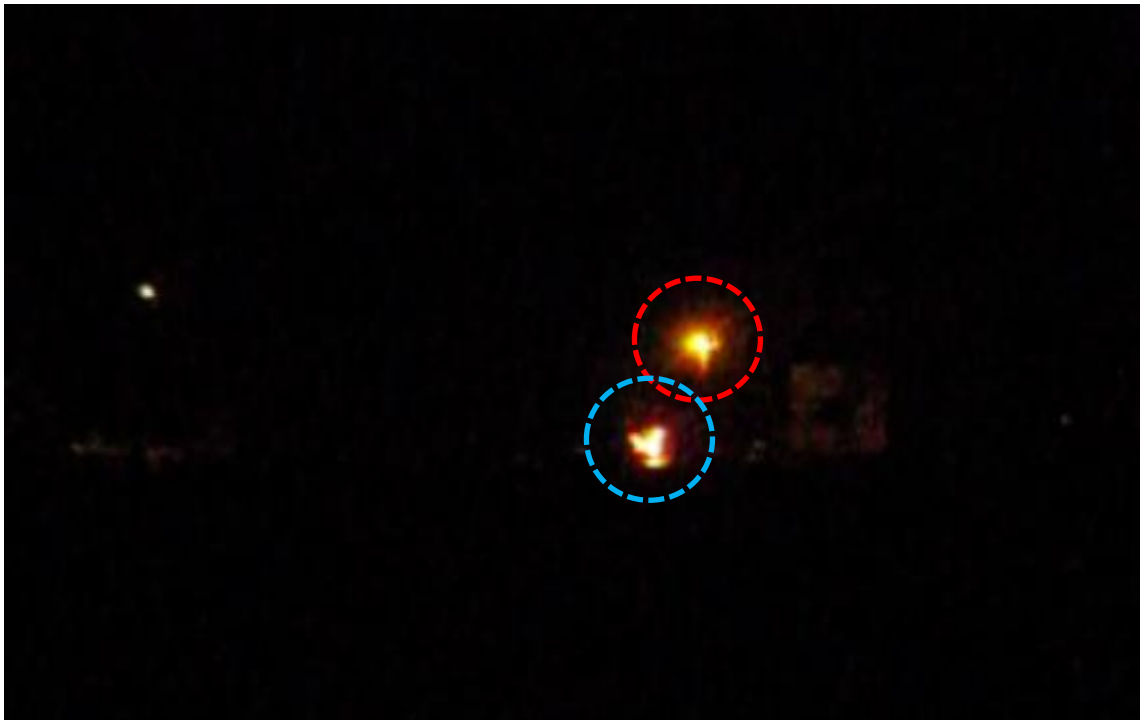

Figure 2.7 A streetlight (red circle) and freshly started fire object (blue circle). They are almost indistinguishable from each other. Note that land line is also indistinguishable that position of both objects that are relative to each other is also insignificant.

Nonetheless, a binary video offers descriptive clues about the investigated object, such as its motion behavior throughout the video. In night-fire videos, the flame has a distinct motion behavior such as flickering, shooting high into the air, dying down or flaring up, and temporal disappearance due to smoke occlusion. Therefore, this study aims to benefit from these temporal behavioral characteristics of a fire object.

The challenging properties of nighttime fire videos can be summarized as:

- No texture or color information: The acquired digital image looks more like a binary-colored image because of the insufficient dynamic range of the cameras,
- All light emitting objects look alike: Especially under the heavy influence of smoke, car headlights, fire, streetlights, etc. are formidably distinguishable from each other,
- High noise in the image: The camera sets the ISO value to the maximum in order to compensate the low light. This causes a significantly high noise in the image,
- Insufficient dynamic range against high contrast: Nighttime fire scenes includes almost pure dark areas due to no sufficient light source around and very bright areas due to fire object as a powerful light emitting source. This leads a high contrast ratio where cameras cannot adopt. This eventually causes the camera work in a contrast range close to either dark or bright areas depending on focus,
- Low contrast against background: In the night images, due to low contrast of objects against the background, it is difficult to distinguish such objects from the background effectively.

Chapter 3

Underlying Theory of Fire Object Detection

3.1 Conventional Methods

In two-class classification problems, any data point in a dataset is expected to belong into either the first class, i.e., positive or the second class, i.e., negative. This requires the dataset be divided into two disjoint groups. This pre-divided dataset can help one to determine class of a never-seen-before data sample. The algorithms do this task for us in an automated way by learning rules for assigning a new data sample to pre-existing classes with the help of mathematical models they are designed to optimize.

Support Vector Machines (SVMs) have been a successful example of such algorithms and used not only for binary classification problems but also for multi-class classification problems for many years. It is also used for nighttime VFD analysis in this thesis.

SVM uses hyperplanes to separate data into regions (see Figure 3.1). Assume the linear model of a two-class classification problem [11]:

$$y(\vec{x}) = \vec{w}^T \phi(\vec{x}) + b \quad \textbf{(3.1)}$$

where notations are denoted as following:

$\vec{x}$: D-dimensional input vector. Each dimension corresponds to a feature,

$\vec{w}$: D-dimensional weight vector. Each value corresponds to weight of corresponding input at that dimension,

b: Scalar bias term,

y: Output,

ϕ: A function that transforms the feature space to another space.

The input vector $\vec{x}$ takes a class from $k \in \{-1,1\}$ and sign of the output $y(x)$ contributes to this assignment via its sign, i.e.,

$$\vec{x} \rightarrow k_x = -1 \Rightarrow y(x) < 0 \quad \textbf{(3.2)}$$

and

$$\vec{x} \rightarrow k_x = 1 \Rightarrow y(x) > 0 \quad \textbf{(3.3)}$$

or

$$k_x y(x) > 0. \quad \textbf{(3.4)}$$

The separation hyperplane is defined at $y(\vec{x}) = 0$. Its rotation behavior is determined by orthogonal weight vector, $\vec{w}$, while translation behavior is controlled by magnitude of bias, b. For a given separating hyperplane, we can measure distance of all data points to the hyperplane. The minimum distance is of importance which is termed as the margin. Margin is defined by two boundaries defined by

$$k_x y(x) = 1. \quad \textbf{(3.5)}$$

By definition, it is expected at least one data point to lie on one boundary of the margin (Figure 3.1). These data points are termed as support vectors, and they satisfy equation (3.5). On the other hand, no data point should lie within the margin. When this is the case, i.e., $k_x y(x) < 0$, then incorrect classifications have been made.

A data scattered in D-dimensional feature space requires a $(D-1)$-dimesional hyperplane to get divided by. When there is at least one $(D-1)$-dimesional hyperplane dividing the feature space into two disjoint regions, then the data is called as linearly separable data in the D-dimensional feature space. When this is not the case, then the data is linearly not separable in the D-dimensional feature space which requires different techniques for the classification task.

Assume that the data is linearly separable in the feature space. However, it is possible to be more than one hyperplane that can divide the binary data into two disjoint regions. In that case, the hyperplane with the maximum margin is the optimum hyperplane that leads to the least generalization error.

If the data is not linearly separable in current feature space, where here $D = 2$, then the dataset is mapped to a higher feature space, i.e., a space with a $D > 2$ dimension, for the hope that the data is linearly separable by an optimum corresponding hyperplane, i.e.,

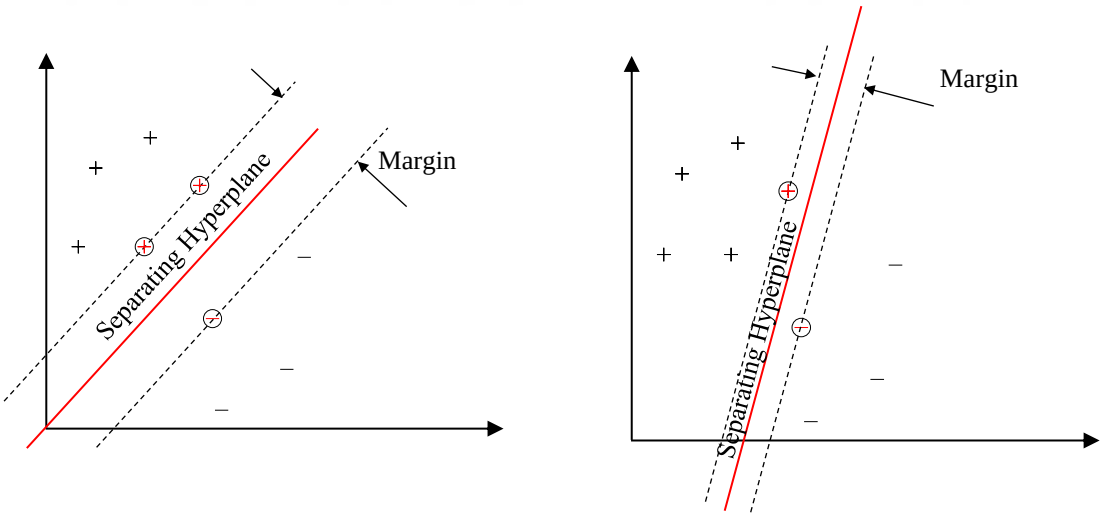

Figure 3.1 A 2-dimensional data scattered in space and separating hyperplanes (red lines).

by a plane. The function maps the current feature space to a higher dimensional space is called the kernel function. The kernel function generates a higher dimensional feature space with transformed data point and SVM find a hyperplane dividing the new space into two disjoint regions (Figure 3.2). Then that hyperplane is projected back to the original space leading a nonlinear separation boundary.

When classes and features are attributed in a nonlinear way and the number of features is not very large compared to the number of samples, then Gaussian (or radial bases function, RBF) kernel is useful to separate the higher space as defined by

$$\mathrm{K}(\vec{x}, \vec{x}') = \exp(-\gamma \|\vec{x} - \vec{x}'\|^2), \gamma > 0. \quad \textbf{(3.6)}$$

When SVM is allowed to make error, penalizing it is a good practice. Then one wants to minimize

$$\frac{1}{2}\vec{w}^T\vec{w} + C\sum_{all\ i}\xi_i \quad \textbf{(3.7)}$$

where $C > 0$ is the error penalty term and ξ_i is indicator variable in that $\xi_i > 1$ implies an incorrect classification. LIBSVM library [12] conducts a grid search of

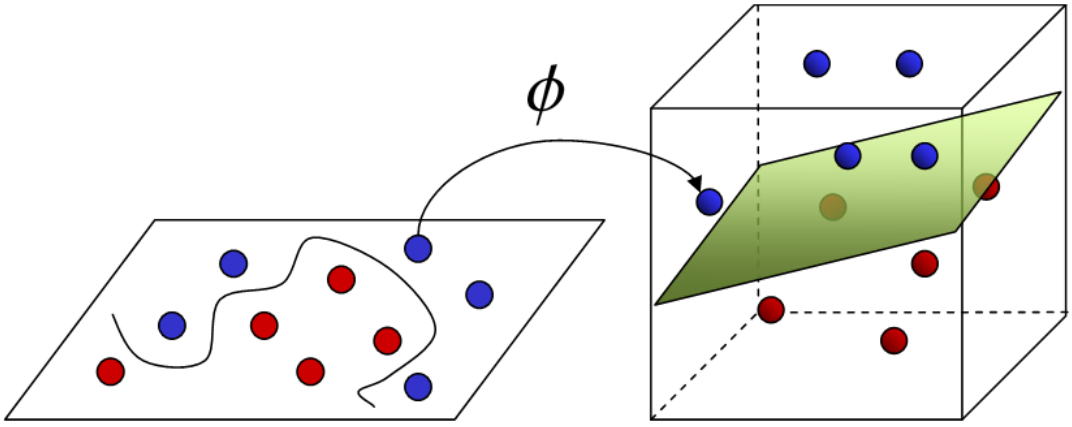

Figure 3.2 Feature space transformation [81].

(C, γ) pairs using cross-validation in the manner that $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$.

3.2 Deep Learning Methods

For computers, learning from experience became possible via hard work of researchers in the broad discipline of artificial intelligence (AI). AI generally aimed to perform human-like cognitive capabilities via computer programs or algorithms. A computer program that can count from 1 to 100 should not be considered as an act of intelligence. Yet, intelligence is associated with discovering patterns and making decisions or concluding automated results based on those patterns. Therefore, machine learning (ML) is another, but more constrained, term used interchangeably with AI implying little human intervention during pattern discovery and automatic decision-making process. Today we have the term deep learning which implying the following qualities: use of neural networks, use of immense amount of data, employing from simpler to more complex hierarchy of concepts, and full automated decision making with no human intervention in the process.

Building block of a neural network is artificial neuron. Artificial neuron is frequently compared to an actual neuron we have in our brains. Even though they have some similarities, they differ in terms of topology, size, propagation speed, adaptive topology and learning scheme (Figure 3.3). An artificial neuron receives a number of inputs, amplifies each with a corresponding weight, sums the amplified inputs and finally outputs a value if the sum meets certain conditions. This is definition of a perceptron and when multitude of them is used for the same inputs, we call it as single-layer network.

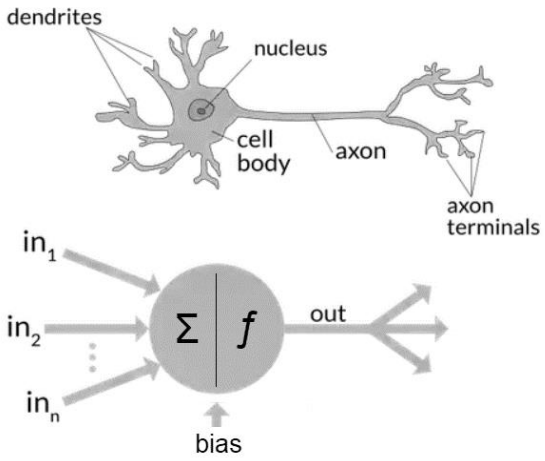

Figure 3.3 Actual and artificial neuron [13].

When output of a single layer is given as input to another layer, and so on, then this is a multi-layer perceptron (MLP) with an input and an output where at least one hidden layer is in the middle (Figure 3.4).

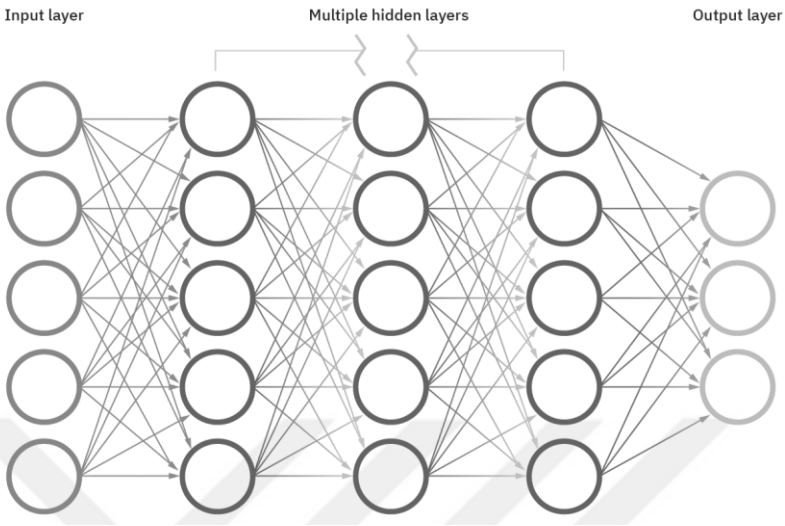

Figure 3.4 A neural network structure [87].

In Figure 3.4, the network receives one dimensional data. When two or more-dimensional topological data is required for the input, then convolutional neural networks (CNNs) are convenient.

CNNs are the neural network models that can work on images, learn from them, and execute desired deep learning tasks, i.e., object detection, image segmentation, image classification, etc. In general terms, a CNN consists of an input layer, a number of hidden layers, and a classification layer (Figure 3.5). If we need to process RGB images, then in the input layer, an $(H \times W \times 3)$ image is given as an input, where H, W, and 3 is the height, the width, and the number of channels of the image, while preserving its spatial grid-like structure.

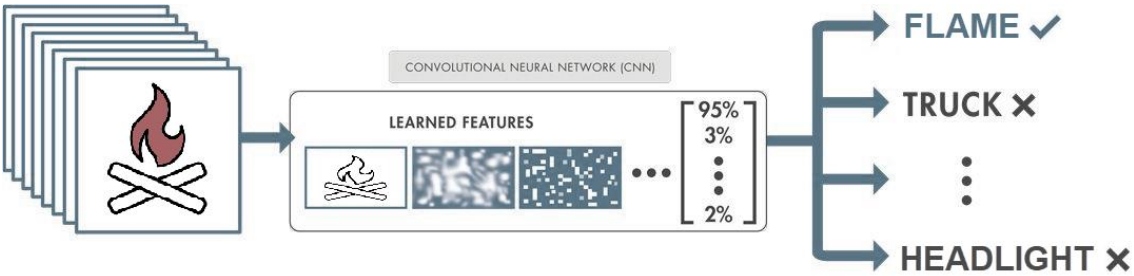

Figure 3.5 A standard convolutional neural network structure (partly [88]).

A standard neural network, as shown in Figure 3.3, connects each node of previous layer to each node of current layer. We term this structure as full connection. Contrary to this idea, a CNN does not connect each node of previous layer to the nodes of current layer, i.e., it is not necessarily fully connected. It is selective. To achieve this, a very useful operation, convolution is performed onto the previous layer.

Convolution is performed in a type of the hidden layers that is called the convolutional layer. This layer receives an input image and scans a $(k \times k \times 3)$ filter over it which is termed as convolution process as mentioned above. The filter and its projection onto the image matrix are element-wise multiplied then summed to get a weighted sum. Values of elements of the filter are termed as weights which should be optimized during training (Figure 3.6).

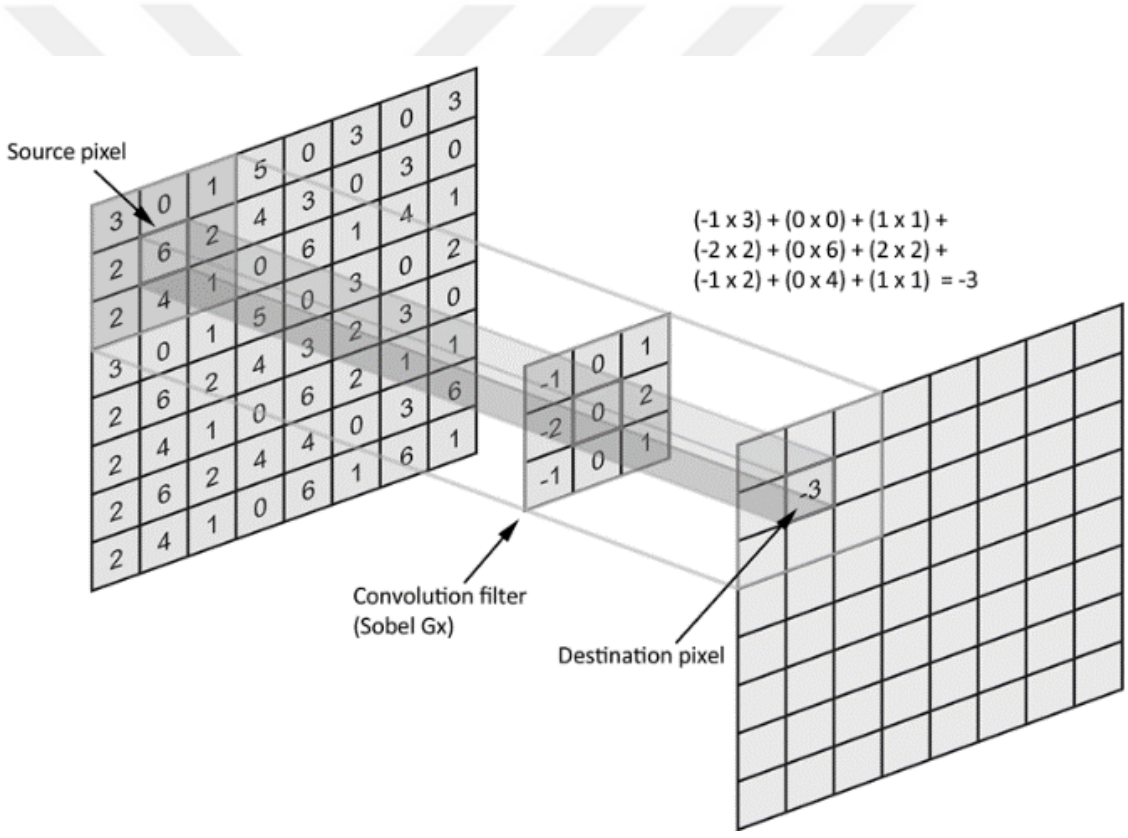

Figure 3.6 A 3x3 kernel (convolution filter) is projected on an image and after convolution, the scalar is registered on the corresponding cell in a feature map (destination cell) [89].

At each projection, the $(k \times k)$ kernel gets an abstraction or summary of the $(k \times k)$ image region via convolution. The number of scans corresponds to the number of abstractions the kernel in use performed onto the input image. One should realize that a $(k \times k)$ region is mapped to a scalar number. This implies that k^2 number of inputs are not mapped to the next layer individually, as expected in a fully connected layer. Instead,

they are only mapped to a single neuron after the convolution operation. Then, other neurons in the current layer should not get any input data from this region. A convolution result for each region of the image is stored into corresponding matrix cell of a convoluted matrix, of which is also spatially related as in the input image (Figure 3.7).

Using more than one kernel is a common practice where each kernel abstracts the entire image in a different way. For example, one kernel may focus on certain line attributes and other on certain color attributes, so on. Depending on size of the kernel and the method preferred for scanning, size of the feature map varies. Scan methods are distinguished by their stride and padding settings. Stride means, at each shift, how many cells the filter will move in one direction. In Figure 3.7, the left quaternary group shows a 1-stride operation in both directions and the right group, a 2-stride operation. Padding is adding zero values to borderlines of an image. For example, a $(H \times W)$ image will be $(H + 2 \times W + 2)$ after padding by setting the added first and last columns and rows to zeros.

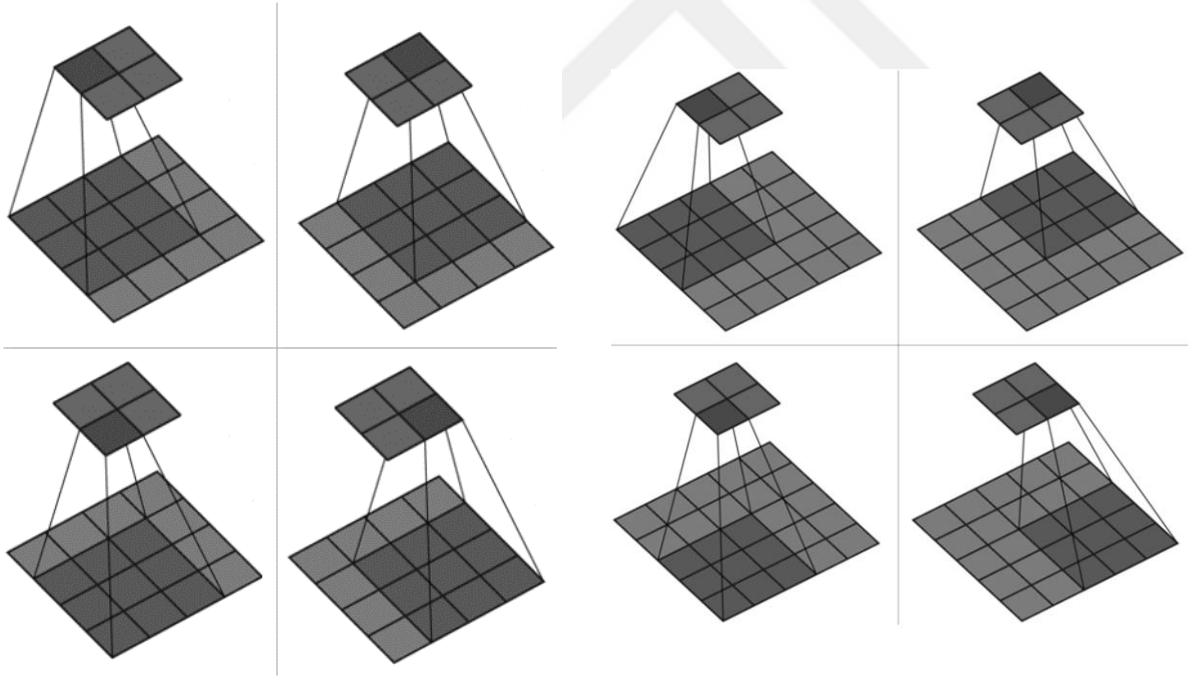

Figure 3.7 Two common types of stride operation. 1-stride on the left group and 2-strides on the right group (partly [90]).

The size of the convoluted matrix that the convolution operation generates can be computed as following.

$$(1 + \frac{h + \sum p_i - f}{s}) \times (1 + \frac{w + \sum p_i - f}{s}) \quad \textbf{(3.8)}$$

where h and w are height and width of the original image, p_i is size of padding from one side of the image, i.e., when the size of padding is 2 from left and 3 from right, then total padding for width is 5, f is kernel size on one dimension, i.e., for a 3×3 kernel $f = 3$, and s is the stride step along one dimension.

Now, the convoluted matrix with $(M \times N)$ dimensions will be input to an activation layer. In the activation layer, an activation function is applied elementwise to the convoluted matrix to determine which cells to fire. A common activation function is RELU among many others and is required to make negative elements of convoluted matrix zero and add non-linearity to the network. The activation process generates an $(M \times N)$ activation or feature map which is then input to a pooling layer. A pooling layer summarizes the most important information in a feature map. A common pooling method is max pooling which gets the max value of a projection sub-matrix onto the feature map and finally generates an $(m \times n)$ matrix. This convolution, activation, pooling sequence can take place numerous times depending on the desired architecture which, in general, constitutes depth of the hidden layers.

Assuming the final output of the hidden layers is an $(m \times n)$ matrix, the first layer of classification layer, i.e., flatten layer, converts it to an $(1 \times mn)$ vector to make it a useful input to a conventional multi-layer perceptron (MLP). Final layer values of MLP are passed to a Softmax layer which is another type of activation specifically preferred to be used in output layer for classification problems. Softmax computes probabilities of each label that the input image belongs to. An argmax function finally picks the max probability class among others and delivers a prediction for the class of the image with the corresponding probability.

A BLSTM cell uses two coupled LSTM cell engines, as summarized in Figure 3.8. The coupled engine receives the first elements of a sequence and time-reversed version of it, and then produces an output. This procedure continues until all elements of (sequence, reversed version) pairs are processed by the engine. This allows the network to learn both from past and `future' simultaneously and gives more accurate results in classifying a scene.

CNN features extracted from the first stage are fed to a BLSTM network stack for training. The rolled network structure is given in Figure 7.1. The BLSTM stack receives each sequence with the size $1024 \times N$ where N is the number of frames per sample video.

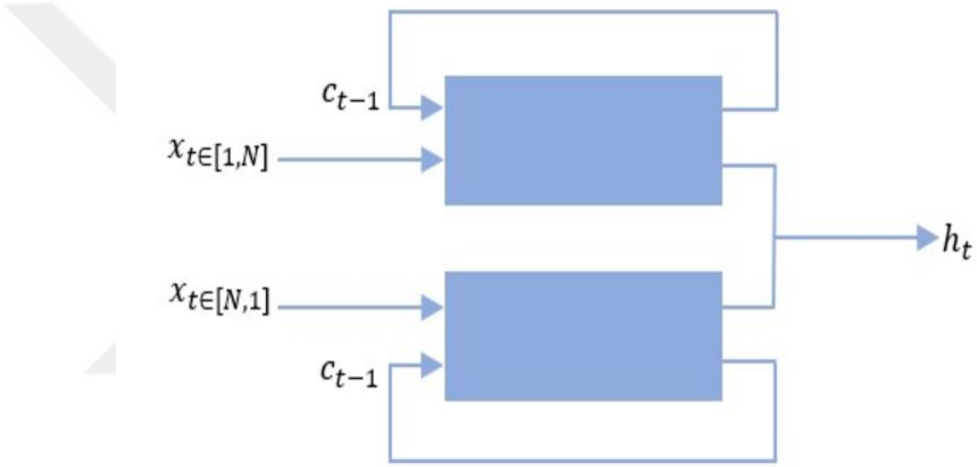

Figure 3.8 BLSTM cell engine.

Stacked BLSTM performs better than single-cell counterparts in accuracy, ability to learn at different time scales, and ability to manipulate parameters with increased non-linear operations [14]. A dropout layer is required to avoid overfitting for long stacks. Final conventional layers are a fully connected layer output of two for (fire, non-fire) classes, a Softmax layer for probability computations, and a classification layer for cross-entropy loss computations.

Chapter 4

Related Work

Fire event monitoring and detection methods rely on a number of systems operated by mostly miscellaneous governmental authorities. Conventionally systems include manual observation on watch towers by fire lookouts or periodic patrols. Stuff employed as lookout needs to have an exceptional eyesight for long distances, no color bias, and ability to distinguish details in depth. The observation of the forest field will be typically 12 hours/day especially on fire seasons and on stormy days with lightnings. Therefore, a lookout should be in excellent mental and physical conditions for a lifestyle with loneliness and monotonous routines [15]. One alternative to on-site manual monitoring is surveillance cameras deployed at observation sites. These cameras send observational video data to forest administration centers and human operators watches multiple monitor screens for fire or smoke catching. This effort includes 24 hours/day observation which requires shifting working hours [16]. These two surveillance systems require human lookout either on-site or in-office both of which are subject to human factors and human error.

An alternative to manual detection, automatic fire detection is carried out by technological systems that include sensors, devices, and underlying algorithms. Sensors of these systems may be installed on stationary points, on mobile land or aerial vehicles or on satellites. Stationary points frequently include on-site watch towers while mobile and aerial vehicles include patrolling vehicles and UAVs, respectively. The data acquired from these sensors are transferred to computing devices to instantly process it through decision making algorithms and finally generate required alert messages.

Time of a fire event is also important in designing detection methods. The nature of fire at nighttime has already been discussed in Section 2.2. Due to these complexities, specific algorithms are required for nighttime fire detection.

Evidence used to make a fire event decision and a corresponding sensor to detect that evidence also require specialized methods. Heat is a common evidence of fire

detection; thus, it requires heat detectors. These detectors can be mounted on certain buildings, trees, or other entities and may have data and power lines. They may also be deployed from air or by hand to the ground and powered by batteries with wireless connectivity for data transfer. Another evidence frequently used is existence of odor, certain gases like CO2 or CO, or smoke in the field of interest. For the specific gas, a specific sensor is required. These sensors can be stationary or mobile. The last evidence will be illustrated is image of smoke or flame in terms of various electromagnetic spectrum, i.e., visual spectrum cameras, IR cameras, etc. Image based algorithms uses useful features derived from images to detect fires. When these features are designed by hand, machine learning tools are required for automatic decision making. On the other hand, when automatic feature extraction is desired, then deep learning tools are used. Consequently, specialized sensors, algorithms, or devices should adequately be combined to sense the fire evidence on target and raise alerts.

In this thesis, methods of interest include visible spectrum-based land or aerial fire detection methods from videos.

In order to overcome fire detection problem from video surveillance systems, many techniques have been proposed [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28]. Front runner techniques include detection of fire, smoke or both depending on spatial and temporal features of objects and colors in a video and using different spectral or physical range cameras [24]. After processing video sequences, those techniques decide if a pixel, frame, sequence or the whole video contain a fire. This requires use of a decision-making process. To date, logistic regression, adaptive decision fusion, correlation or covariance descriptors, Bayesian models, neural networks, LMS and SVM have been frequent tools used for decision making process.

However, very limited number of these works considered the fire detection in dark videos. Recent works shows evidence of semi or full daytime fires [29, 30, 31, 32, 33, 34, 35] . The most relevant studies are briefly explained below.

Tasdemir's work at [29] proposes a method on distant night fire detection. Since the fire event is assumed to be at far distance, fire is considered as a slow-moving object. Even though this approach is fine for distant fires, it may not be correct for short or mid-range ones.

Gunay et al. proposed a set of hand-crafted features for night-fire detection [30]. They developed a decision-making system that fuses decisions of sub-algorithms. These sub-algorithms make decisions based on detecting slow-moving objects, bright regions,

periodic regions, and moving region interpretation. Ho and Chen used a CCD camera and a laser light to detect smoke at night [31]. They analyzed spectral, diffusing, and scattering characteristics of the trajectory of the laser beam with a fuzzy reasoning system to detect fire smokes. Gomes et al. proposed a rule-based fire detection system tested on night-time fires besides indoors, rural, and urban fires [32]. They used two parallel working pipelines, one for fire detection and the other for fire confirmation, to make the final fire decision.

Park and Ko proposed a multi-staged night-time fire classification method using a modified YOLOv3 architecture and Random Forests (RF) [34]. They first analyzed the videos with ELASTIC-YOLOv3 to detect candidate fire regions per frame. Then, they generated fire tubes from fire frames based on a rule that joins fire object candidates in successive frames. They generated a histogram of oriented features (HoF) from the fire tubes, then transferred them to a bag of features (BoF) with a code-book mapping. Finally, they used a bag of feature histograms as features to train an RF classifier. In the process of fire tube generation, a threshold that allows adding a frame to the fire tube needs to be manually set. This threshold should be adjusted according to the distance between the fire object and the camera. A dataset containing a diverse set of real-world examples is not practical to decide on a global threshold covering all samples. In addition, they extract a histogram of features from each frame of an object tube. The motion behavior of the flame, such as flickering, cannot be captured because the indexes of frames are lost when the features are put into a bag of features set. The limited generalization capacity of the method makes it suitable only for scenarios where the dataset distribution is not diverse.

Pan et al. developed a pruned CNN via Fourier analysis to detect wildfire and tested its performance on a limited number of fire videos besides daytime videos [35]. They used MobileNetv2 and pruned redundant low-energy kernels and similar kernel pairs by calculating their DFTs, thus letting them save approximately 7% time and 22% storage.

In its problem nature, fire detection is a subdomain of object detection. Evaluating performance of a network pipeline in object detection, a well-defined set of metrics based on a ground truth method should be employed. Two domains of ground-truths can be defined for fire datasets. Spatial ground-truths are generated at pixel-, region-, or frame-level. A pixel-level ground-truth identifies the label of each pixel in a frame. Therefore, it gives the densest ground-truth information about a frame. However, it does not give any neighboring information between pixels. A region-level ground truth refers to a region of interest where a certain area of the frame is labeled as positive or negative. It

can still divide the entire frame pixels as positive or negative and can give neighboring information between cells. A frame-level ground-truth implies that the target object is contained in all pixels of the frame. Temporal ground-truths, on the other hand, are generated at frame-, interval-, or video-level. A frame-level temporal ground-truth implies which frame at what time instance contains the target object, and an interval-level ground-truth implies the target object is contained at all frames in a certain interval of the video. It is noted that none of the temporal ground truths can give spatial information about the labeling. For example, let a 10-seconds video contains fire objects at only 2nd to 4th seconds, then only this interval is labeled as fire. Finally, a video-level temporal ground truth implies that each frame of the video contains the target object, then the video is labeled as fire or none of them contains the target object, then the video is labeled as non-fire.

Ground-truth depth and domain is important for the method used in fire detection. For example, consider that a video-level temporal ground-truth of a video is fire; however, in the same video, some of the frames do not contain a fire object. Also consider that a temporal deep learning method will be used for the analysis. Then it should be considered that the algorithm will also learn from the frames without fire as if they are fire and this will affect the training process.

Depending on ground-truth type employed for the data, a performance metric should be selected. In [36], authors give a comprehensive review of spatial ground-truth performance metrics based on intersection over union (IOU) for both images and videos. With IOU, a bounding box or a closed boundary line around a target object is required to calculate intersection and union of actual bounding box (or blob generated by the closed boundary line) and predicted bounding box (or blob). In the case of temporal ground-truthed data, there are no bounding boxes or boundary lines. In that case, IOU is not suitable, and the whole image or video is considered to belong to a class. Accuracy and F1 scores are two useful performance metrics chosen for our experiments as also by recent VFD studies including [32, 34, 35].

Chapter 5

Proposed Night Fire Datasets

5.1 Introduction

5.1.1 Importance of a dataset

Wildfires cause tremendous damage to the natural life, economy, and society more than ever nowadays. In the USA, the burned area increases by approximately 180,000 acres per each year. In 2020, the burned area due to wildfires is slightly more than 10 million acres which is a land area greater than that of Maryland [37]. There are many important reasons for the increasing fire trend in USA compared to world [38]; however, the scope of this chapter will be limited to detection means of fires occurring in the wild, rural/exurban, and suburban areas, collectively we will call as non-urban areas. Non-urban areas have potential to create wildfires since their environment contains plant agents that are prone to wild, fast growing, and large fires compared to urban/city areas. In the case of such fire events, fire can spread very quickly in these areas and rough land structure gives a limited mobility and accessibility to the fire regions. Therefore, accessing, controlling, and extinguishing non-urban fires are more difficult compared to urban fires and their occurrence locations. This brings importance of early detection of non-urban fires to front. Video fire detection (VFD) techniques have been an effective response to this need that successful project were implemented to realize VFD techniques [28, 39]. VFD techniques use video data to detect fires where machine learning methods have been central. Neural networks are state-of-the-art methods for detecting fires in videos and these networks require a ground truth video data that will be used to train a model which is expected to raise true positive fire alarms on the never-seen video streams.

5.1.2 Purpose of preparing a new dataset

Even though a ground truth dataset is essential for training a neural network, it is also essential in comparing performance of existing neural nets offered by researchers. There are many video fire datasets used and made available by research community. A review of these sets is given in Section 5.2. However, these datasets are almost for day-time fire detection tasks, i.e., smoke or flame detection. In this work, we introduce a comprehensive and challenging night-time fire dataset, Fire in Dark (FinD) which is expected to help researchers develop and compare machine learning models for night-time non-urban fires.

5.2 Literature Review of Fire Datasets

To the best knowledge of the authors, Neal et al. published the first work on image-based fire detection with neural networks in 1991 [40]. Since then, there have been over 3 hundred research works on the problem of VFD. These publications used image or video data in order to verify performance of their work for hand-crafted features or to train their neural networks for automatic feature extraction and then verify performance of the trained models. The data of the majority of these works are not accessible due to lack of access links, broken links, or in-accessible links from other countries. Even though, the dataset used is created from a combination of data from other accessible known datasets, access links to the mixture of final dataset mostly not given in those works. Therefore, it is not possible to replicate these methods with their original data.

The papers with open access data either includes direct working links or controlled access by registering to the database or signing a license agreement of the providing institution. Since the data is available, it is possible to replicate the original work with the corresponding data. Some of these works provide a citation format to their data when other researchers intend to use them in their own work and some of them only have access links for citation purposes. When the sets in the access links given below assessed, it will be seen that researchers borrowed data from other datasets in creating a dataset according to their needs. In this study, we only review and list popular open access datasets to make research community save time in searching fire datasets.

The most used dataset for model development and comparison in the literature is VisiFire dataset [41] from Bilkent University, Turkey. It includes 14 positive fire videos, 23 positive smoke videos and 2 negative smoke videos. Only 2 of these videos are

negative fire or smoke videos.[1] Since this dataset dates to 2005, video resolutions are not HD. Another most used dataset is KMU Fire \& Smoke Database from Keimyung University, Korea [42]. This dataset includes 22 positive fire videos, 6 positive smoke videos, and 10 negative fire or smoke videos. This dataset includes only 7 positive smoke videos and 1 negative smoke video at night.[2]

The MIVIA Fire Detection Dataset from University of Salerno, Italy [43] includes 14 positive & 17 negative fire videos and 149 positive smoke videos.[3]

Video smoke detection (VSD) dataset from University of Science and Technology of China includes 3 positive smoke videos, 3 negative smoke videos, 6323 positive smoke images, and 74989 negative smoke images without checking duplications for the images.[4]

The ViSOR dataset from University of Modena and Reggio Emilia, Italy [44, 45] includes 14 positive smoke videos among other provided sets.[5]

FIRESENSE dataset [28] includes 11 positive fire videos, 16 negative fire videos two of which are night-time videos, 13 positive smoke videos, and 9 negative smoke videos.[6]

Corsican Fire Database from The University of Corsica Pasquale Paoli [10] includes 500 images in visible spectrum (VS), 100 pair of images in both VS and NIR, and 5 sequences of pair of images in both VS and NIR[7]. Since we do not have direct access to this dataset, we cannot give exact content of night video in the set.

FiSmo dataset from University of Sao Paulo, Brazil [46] and the RESCUER project [39] includes a collection of datasets: FiSmo-FireVid dataset contains 27 positive fire videos only one being night-time fire video, FiSmo-RESCUER dataset contains 61 positive fire videos, FiSmo-BoWFire dataset contains 199 positive fire images, 107 negative fire images, 80 positive smoke images, and 80 negative fire images, FiSmo-Flicker-Fire dataset contains 984 positive fire images, FiSmo-FireSmoke dataset contains 1077 positive fire, 369 positive smoke, 527 positive fire and smoke, 3583 negative fire

[1] http://signal.ee.bilkent.edu.tr/VisiFire/Demo/SampleClips.html
[2] https://cvpr.kmu.ac.kr/Dataset/Dataset.htm
[3] https://mivia.unisa.it/datasets/
[4] http://staff.ustc.edu.cn/~yfn/vsd.html
[5] https://aimagelab.ing.unimore.it/visor
[6] https://zenodo.org/record/836749##.YNhUxugzYdU
[7] http://cfdb.univ-corse.fr/

and smoke images, and FiSmo-SmokeBlock dataset includes 832 positive and 834 negative smoke images.[8]

DynTex dataset from La Rochelle University, France is a wide and great collection of texture videos and suitable for neural network model training. The DynTex contains 29 positive fire/flame videos, 20 positive smoke videos and 630 negative fire and smoke videos.[9]

The National Institute of Standards and Technology (NIST) has an online repository of fire videos in different environments that has been also popular in research community.[10]

Furg Fire dataset from Federal University of Rio Grande, Brazil [47] contains 17 positive fire videos and 6 negative fire videos.[11]

State Key Laboratory of Fire Science (SKLFS) from University of Science and Technology of China [48] contains 20 positive and 10 negative smoke videos.[12]

ImageNet [49] is a very popular benchmark dataset in neural networks research community and used for fire detection research. Since it does not contain any fire or smoke images, images from it can be added to negative fire or smoke images.[13]

Even though it is rarely used by the research community, we should also mention the ALERT Wildfire observation camera network[14] which is very useful for extracting landscape view of both fire and non-fire sequences from long distances at various points of west of the US.

The dataset Anton used in his research [50] contains 10 positive and 10 negative smoke videos\footnote.[15] The Ultimate Chase website\footnote[16] and its YouTube channel also have 14 positive fire videos used multiple times by the fire researchers. Only one video from The Ultimate Chase is of night fire.

[8] https://github.com/mtcazzolato/dsw2017
[9] http://dyntex.univ-lr.fr/index.html
[10] https://www.nist.gov/video-category/fire
[11] https://github.com/steffensbola/furg-fire-dataset
[12] http://smoke.ustc.edu.cn/datasets.htm
[13] https://www.kaggle.com/c/imagenet-object-localization-challenge/data
[14] http://www.alertwildfire.org/
[15] https://disk.yandex.com.tr/d/q97BQ9v58WNRD
[16] http://www.ultimatechase.com/Fire_Video.htm

FESB department at University of Split offers a collection of some datasets on smoke detection and one dataset\footnote[17] provides 10 positive smoke videos [51].

(YUP++) Dynamic Scenes Dataset from York University, Canada [52] provides 30 stationary & 30 moving camera positive fire videos and 570 stationary & 570 moving camera negative fire videos with different contexts\footnote.[18] 9 of these videos can be considered as night-time fires.

From the most cited and popularly used datasets reviewed above show that they do not contain sufficient amount of night-time fire videos both for positive and negative cases. Furthermore, these datasets should also be examined in terms of if deceptive negative videos are in existence in the set. For example, when the smoke detection is the task, then obviously positive smoke videos should contain smoke produced by fire. However, the neural networks should also distinguish smoke-like objects from the smoke in the scene. Therefore, it is desirable to have deceptive smoke-like videos as negative samples to sufficiently train the model against deceptive objects.

Park et al. developed a new night-fire dataset which contains 10 positive and 10 negative fire videos collected from both KMU dataset and from YouTube. There is no access link to this dataset as for now to give information about nature of videos, however from their work [34] we deduce that the positive and negative videos belong to urban areas. Pan et al. also used both daytime and night-time videos for their algorithm [35] however these videos are also not openly accessible. Nevertheless, sample images in their work show that they belong to non-urban area fire events.

5.4 Preparing A Fire Dataset

Well-designed datasets are a backbone stage of developing automatic fire detection systems based on computer vision. In the literature, it is difficult to find an extended preprocessing information given about the dataset used for the video fire detection research. This section intends to give a framework on fire dataset preparation for researchers based on objective experience during creation of the FinD dataset. Besides general dataset preparation requirements, a video fire dataset may include requirements regard to the specific problem of video fire detection.

---

17 http://wildfire.fesb.hr/index.php?option=com_content&view=article&id=65&Itemid=53

18 http://vision.eecs.yorku.ca/research/dynamic-scenes/

5.4.1 Data retrieval and acquisition

Deep learning model development requires ready-to-use structured data. If the structured data is not in hand, then it should either be acquired from other structured data sources, be collected from unstructured data sources, and then converted to structured data or generated from scratch. In either case, lack of dataset in hand for VFD research brings us to initial step of preparing a dataset, data retrieval.

Structured data retrieval is more trivial since it frequently requires searching the source, registering the institutions service, requesting access, requesting permission, and downloading and storing the data. Publicly listed datastores for structured data frequently requires its own terms and conditions. Some of them even impose transferring terms and conditions, i.e., copyleft license. Data under a copyleft license can be downloaded, adapted, and shared; however, any derivative work generated from this data is also under the same original terms and conditions of the original data that made creating of these derivatives possible. Structured data is expected to require little to no effort for use in model development.

In the case of unstructured or raw data, that is the data that cannot readily be used for a deep learning pipeline and potentially requires multiple strenuous processing steps beforehand, a search should be conducted across public video sharing services like Google, YouTube, or Vimeo or data repositories that are specifically constructed to hold desired data. Most of the time the data of interest is publicly available, however it does not mean that the data is immediately allowed for downloading. In other words, one should be cautious for collecting data from publicly accessible data sources in terms of data licenses and permissions. For example, videos that will be collected from the YouTube videos will have either The Standard YouTube (TSY) License or Creative Commons (CC) License. The Standard YouTube License gives the YouTube rights to stream the content of an owner/uploader and requires the third parties access this media only on YouTube website. This implies that videos under TSY license is not allowed to be downloaded, adapted, or shared. At the first instant this constraint seems troublesome. However, Section 107 of the Copyright Act of USA defines four pillars of fair use that data collection for scientific research can be based on:

1- *The purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes:* In terms of this criterion, the desired dataset should not directly be taken from the source as a

whole. Instead, videos should be cropped, trimmed, resized, etc. Furthermore, the samples can include only a fraction of a short time compared to the corresponding original video lengths and sound is advised to be eliminated. Therefore, these small video pieces should not and cannot satisfy an audience in any way that the corresponding original, long, uncropped, untrimmed, and unresized video with the sound would.

2- *The nature of the copyrighted work:* In terms of this criterion, the desired raw data (i.e., YouTube videos) should be factual work, in other words, they should be merely and naturally occurring physical events and not include any fictional work. In that regard, the dataset being compiled from raw videos should also be factual work without any fictional element.

3- *The amount and substantiality of the portion used in relation to the copyrighted work as a whole:* In this criterion, one should be clear that she/he does not use the raw data as a whole, in fact it should not subject to direct or whole use. Thus, the raw data should be trimmed (took only a small-time interval of the video) for scientific use. Consequently, the amount of these small video pieces should not be comparable to the amount of corresponding original video in terms of wholeness.

4- *The effect of the use upon the potential market for or value of the copyrighted work:* In terms of this criterion, researchers should not conduct a business activity that rely on the dataset they generated. They should have no intention to market any opportunity that one may seek from the work. They also should not have intention to put this dataset to any video streaming service in the way that original video uploaders did. Therefore, the researchers should not be a potential competitor or market killer for the original video uploader. The authors should only seek scientific contribution to advance common good in the event of adverse effects of the problem being studied.

In summary, during the process of generating a dataset, the researchers should be within the boundaries of "fair use" in terms of Section 107 of the Copyright Acts and the video service’s terms and conditions for fair use if the data is not licensed under CC. Giving reference to the original creators of the data is also required regardless of the Section 107 of the Copyright Acts.

If the data desired to be collected is licensed under CC, then a researcher is free to download, adapt, and share the data or even use it in commercial applications. It should be noted that giving reference to the original creators of the data is also required.

Data sources often include a search engine and "fire, wildfire, smoke, video, flame, burn, forest, video, explosion" are some useful keywords that can be considered for finding desired video fire data.

Sometimes, preexisting data is not enough or does not fit the problem, then data generation from scratch is required. Data generation is a more involved method for data acquisition. It requires environment set up, data generator instrument selection (i.e., camera, sensor, etc.) and adjustment of the instrument settings.

After gathering raw data from various sources, it should be accepted to the dataset after it meets a set of data accepting rules. Data accepting rules ensure that the data added to the database meets standards and ready to be used for subsequent steps. Different data accepting rules can be developed for different preprocessing steps. For example, a set of data accepting rules that will be used after collecting raw data should be different than the rules that will be used after data cleansing. Fundamental rules of thumb for raw data acceptance are validity, quality, quantity, variety.

Each candidate video should be evaluated in terms of these dimensions to determine its potential contribution to the VFD research. Data validity refers to checking class or label, type, size, time stamp, and uniqueness of the data. Selection bias is a problem in data validity. When accepted data does not represent form, appearance, or version of the target object, then the model built on this data will not be able to make adequate predictions.

Data quality refers to checking image resolution, object interpretability, and selection bias. For example, videos record decades ago may have higher resolutions but bad image quality, i.e., insufficient details of colors and texture, or pixelated video, etc. Distribution of video resolutions in the accepted set is also considered under data quality. Choosing RGB videos contrary to black and white or grayscale videos is also considered under video quality check.

Data variety refers to how representative the whole dataset is for target classification problem. Different than selection bias, the accepted data is a representation of the target object; however, the whole accepted dataset is based on a limited number of forms, appearances, or versions of the target object. For example, if the dataset is based on very short-range fires at macro level, then a model built on this set will not be able to

detect far-range fires effectively. High variety in terms of fire scenarios and scenes will contribute to developing more robust fire detecting pipelines.

Data quantity refers to how the number of accepted videos is distributed in the dataset in terms of data variety. It also includes total number of accepted videos in the set.

5.4.2 Data cleansing

Most of the time, the raw video data includes many unwanted data fragments in terms of spatial regions or temporal intervals. A researcher may want to extract these parts from the raw data and refine it for the next preprocessing step. For example, news outlets have very useful fire data embedded in their broadcasts. However, these broadcasts frequently include logos of the news outlet, subtitles, supertitles, other irrelevant video embeddings, etc., along with desired video footage in the same frame. Therefore, one will want to eliminate these spatially unwanted regions or conversely will only extract the desired region in the frame along with the video. Then editing the frame spatially is called cropping and lets one to eliminate or extract regions per frame along the video.

Alternatively, a video can include an unwanted video interval, or it can be longer than a predetermined time length. Then, eliminating the unwanted video length is performed by trimming or cutting. Trimming is merely shaving the video from the beginning and end to make its length shorter. Cutting is detaching a time interval of a video either for later use or for dropping.

There are many software tools that implement cropping, trimming, and cutting steps effectively. A couple of them are Adobe Premiere Pro, Final Cut Pro, Filmora, etc.

After obtaining a video part that is spatially and temporally acceptable, i.e., it does not include any unwanted region or frame, resizing can be the next step for data cleaning. Depending on the pipeline input size, the video data can be resized, or as a matter of choice, it can be stored as is. A researcher may want to extract these parts from or shorten the raw data before using it. It is strongly advised that keeping geographical region, city, denominated fire call name, incident date for each video will be very helpful in preventing data duplicates. Accessing fire data was limited a decade ago, but as of today, there is an immense amount of fire videos, numerous of them are rebroadcasted by many news outlets. Therefore, it is likely to add the same fire scene to the dataset multiple times. If that is the case, then removing or checking for duplicates will be an added step for data cleaning.

5.4.3 Data annotation

5.4.3.1 Fire/Non-Fire annotation protocol

In the video fire detection research, the fundamental goal is classifying a data sample as either fire or non-fire. Interpretation of such classification should be defined clearly.

When an automatic fire detection system alarms a true positive fire event, then this is an alert that should call attention of firefighting services and possibly requires an immediate intervention. This is a common set of actions that one expects to be taken in case of a fire alert. This expectation is a definitive key about what a fire event is.

From the deep learning point of view, however, each of these steps should be defined carefully. In other words, deep learning algorithms should agree on the meaning of fire and non-fire labels via a well-defined annotation protocol. For example, if the assignment rule for fire label is determined as *"a flame object implies the fire label"* then a fire smoke object should not be attributed to the fire label. On the other hand, if the assignment rule for fire label is determined as *"a flame and/or smoke object implies a fire event which implies the fire label"* then this time a smoke object should also be attributed to the fire label.

In the real-world examples, video smoke detection is another effective method for raising fire alarms during daytime. The most common difficulty for video smoke detection is distinguishing fire smokes from fogs, clouds, and other smoke like sources. In the nighttime fire events, smoke object is not useful evidence for fire detection due to low light conditions, therefore, researchers use flame object as evidence for a fire event even though the smoke can coexist with the flame object in the scene.

Annotating the data with a high confidence is a difficult task which determines quality of the annotation step, which in turn has a direct impact on the prediction performance of deep learning models.

Data annotation protocol should also be designed in terms of annotating frames as time instances or annotating successive frames as transitions. In that regard, the first annotation technique is made by annotating the data frame by frame by assigning either fire or non-fire labels to each frame of the video. With annotation of instances, one can get existence of fire in the frames. On the other hand, the second technique, transitional annotation, labels a transition, T_t, from frame I_{t-1} to I_t labeled as fire if fire is propagated from time I_{t-1} to I_t and as non-fire otherwise. Transitional annotation lets segmentation

of fire events across the video compared to object segmentation in single instances [46, 53].

5.4.3.2 Selection rules of objects for annotation

Some temporal analysis methods require uniquely identifying fire objects from the beginning of a video and label them with the same labels. Keeping track of unique objects is easier after converting frames to black and white images. Then the rules developed in Section 5.4.3 is used for object annotation.

5.4.3.3 Ground-truth depts

In the literature, two domains of ground-truths used for training and test fire datasets. Spatial ground-truths are generated at pixel-, region-, or frame-level. A pixel-level ground-truth identifies label of each pixel in a frame. Therefore, it gives the densest ground-truth information about a frame. However, it does not give any neighboring information between pixels. A region-level ground truth refers to a region of interest that a certain area of the frame is labeled as positive or negative labels. It can still divide the entire frame pixels as positive or negative and can give neighboring information between cells. A frame-level ground-truth implies that the target object is contained in all pixels of the frame. Temporal ground-truths, on the other hand, are generated at frame-, interval-, or video-level. A frame-level temporal ground-truth implies which frame at what time instance contains the target object, and an interval-level ground-truth implies the target object is contained at all frames in a certain interval of the video. It is noted that none of the temporal ground truths can give spatial information about the labelling. For example, let a 10-seconds video contains fire objects at only 2nd to 4th seconds, then only this interval is labeled as fire. Finally, a video-level temporal ground truth implies that each frame of the video contains the target object, then the video is labeled as fire or none of them contains the target object, then the video is labeled as non-fire.

Ground-truth depth and domain is important for the method used in fire detection. For example, consider that a video is labelled as fire assuming video-level temporal ground-truth scheme; however, some of the frames in fact do not contain a fire object. Also consider that a temporal deep learning method will be used for the analysis. Then it should be considered that the algorithm will also learn from the frames without fire as if they are fire and this will affect the training process.

5.4.3.4 Annotation framework

For supervised learning tasks an annotation framework is useful in understanding the behavior of deep learning model. Presence of a certain object may alter decision of the network on target object in a way that it is not expected to conclude. For, example, training an RNN model on videos frequently including both fire and fire fighters together my lead the same model make fire prediction on videos that contains fire fighter but fire. Therefore, further annotating videos for such deceptive objects, events, and states will be useful in training data selection stage as well as understanding performance of the model and the miss-classifications. This framework can be based on events, objects, or states and corresponding sub-features to be labelled. A table of annotations that are useful for fire datasets are given in Table 5.1.

Table 5.1 A summary of proposed fire dataset video annotations

Caption Group	Captions
Object in fire :	tree fire, brush fire, forest fire, vehicle fire, exterior building fire, interior building fire, structure fire
Other light sources :	head light, city light, road light, hand-held light, moon light, lightning
Objects in scene :	fire truck, other vehicle, fire fighter, reporter, other people, pole
Events in scene :	structure collapse, tree collapse, vehicle pass, human movement
Fire contour :	Ground view: V shape, Λ shape, / shape, \ shape; Aerial view: S shape, C shape, water drop, free line
View of fire :	aerial view, ground view, direct view, vehicle-drive view, indirect view (through car/building windows)
Camera motion :	stable, include waggling, include tilts, include displacements, include zooms
Record time :	day, night, semi (heavy smoke like night)
Distance :	macro, short-range, mid-range, far-range
Stage:	pre-fire, beginning, matured, end, post-fire

Fundamental objects that lead to fire event prediction on day and night-time fire alerts are *fire* and *smoke* objects. The fire object in the scene can be in a directly visible form, partially or completely occluded by opaque or transparent objects or there can be no fire object in the scene at all. When a fire object is directly visible to camera, then its color, texture, and temporal features are useful in fire event prediction. If an opaque object partially blocks the fire object, true form of the fire object is altered by the blocking object. If a complete occlusion by an opaque object is the case, color, texture, and temporal features will be even more limited for fire event prediction depending on amount of environmental illumination due to reflection and refraction of the light. When a transparent object occludes the fire object, then an altered and limited color, texture, and temporal features will be available.

The fire event scenes sometimes do not include any visible smoke, which can be termed as *smoke-free* scene. When environmental illumination is not sufficient to distinguish darkness from smokes, then the smoke level is *light*. When smoke is visible with distinguishable gray tones under sufficient illumination and there is no cloud like smoke objects, then the smoke level is *moderate*. When smoke is visible, shady with gray tones, and partially blocks the fire and other objects with visible smoke clouds, then the smoke level is *heavy*. Sometimes, smoke can completely block fire object and other light sources, which is termed as *no-vision*. When artificial smoke is generated by computer vision means, this is termed as *artificial*.

Distance between the camera and fire event is also important in designing fire detection algorithms. The distance determines contour and flickering behavior of the fire. Furthermore, color composition will change due to reflection, scattering, and refraction. When a fire is close enough to the camera or camera is zoomed enough to get a macro shooting that a clear texture of the fire object is visible, then this seen is termed as *macro*. When a person can reach the fire area with a couple of steps, then it is in a *short-range*. When a fire is far enough that a landscape view is available, i.e., from an aerial vehicle or a lookout tower, then it is in *far-range*. Other than these, fires are considered as in *mid-range*.

If fire event is recorded from an aerial vehicle, then this is *air-to-land view*. This type of videos is frequently subject of fire detection techniques employing UAVs. If video is recorded on the ground, then this is termed as *land-to-land view*. This is the most frequent fire video recording mode. If there is no semi- or full-transparent object in between fire and recording device, then this is termed as *direct view*. If there is a semi- or full-transparent in the middle, then this is *indirect view*. This type of videos is frequently recorded behind a car or building windows. If the video is recorded from a moving vehicle, then this is termed as *vehicle-drive view*. In these videos, background is effectively changing.

In nighttime fire events, the target object is fire or flame object rather than smoke object. The object in fire determines motion, color, and contour characteristics of a non-urban fire. These objects are frequently *tree*, *brush*, *forest*, *vehicle*, *interior*/*exterior*/*window of a building*, and *other flammable structures*.

Other than the fire object as a light source in night, there can be other light sources behavior of which can be challenging for fire event detection. These light types are *revolving*/*flashing*/*continuous headlights*, *city lights*, *road lights*, *hand/head-carried*

lights, *moon light*, *fireworks*, *lights emitted from hot molten metal*, *stars*, *volcanoes*, *sunset*, and *sunrise*.

In non-urban night fires, when the illumination is sufficient, labelling data with frequently seen objects are useful in measuring their effect in training process. These objects can be *fire truck*, *other vehicles*, *fire fighter*, *reporter*, *other people*, *pole*, *road sign*.

Other than chemical burning process, there are other frequent event occurring in fire related scenes. These can be listed as *structure collapse*, *tree collapse*, *vehicle pass*, and *human movement*.

Depending on geographic topography, distance, and view angle, the fire contour can be in a couple of shapes which can be listed as V, Λ, *forward/back-slash*, S, C, *water drop*, and *free line* shapes. These shapes become important when the data includes videos recorded from both land and air.

In recorded night fire videos, camera is not always stationary which requires adapting methods other than methods developed for specifically stationary cameras. The moving camera can include the motion characteristics of *waggling*, *tilts*, *displacements*, and *zooms*.

The obvious record time for nighttime fires is *night*; however, depending on amount of smoke discharged to air, sometimes night-like vision is possible at *daytime*. Therefore, using videos recorded at that time sometimes can be an option at training or test processes which can be termed as *semi-night*.

Scene belonging to stages of the fire can be categorized as *pre-fire*, *beginning* of the fire, *matured* fire, *end* of the fire, and *post-fire*. Finding the data for some of the stages of the fire can be relatively difficult, i.e., beginning of the fire.

Finally, the difficulty of the scenes is labelled as *easy*, *moderate*, and *difficult*. The difficult scenes include fire events even difficult for a human to interpret directly by detecting a fire object. It should be emphasized that, for a human, detecting a fire event from a scene by using other evidence is easy, however the difficulty arises when a fire object is difficulty visible. Easy scenes are the ones if a human can easily detect a fire object in it. The moderate scenes are considered in between easy and difficult scenes.

In the literature, data annotation and labelling are conducted by manually or automated by software tools. In manual techniques, researchers, experts, or MTurk workforce is used to complete these tasks by hand. Software tools let the tasks done by in a more automated way. There are plenty of tools and services doing simple labelling,

bounding box, or polygon annotations. Popular software tools include Darwin, CVAT, VoTT, Supervise.ly, SuperAnnotate, and many. State-of-the-art data annotator tools and their features are given in Table 5.2.

Table 5.2 SOTA data annotator tools

	Free	Paid	Labelling	Annotating	Bounding Box	Polygon	CODE	Video/ Image/ Both
Alegion								
Coco-Annotator				X				i
CVAT	X			X			GitHub	b
DataLoop Playment		X						
Dataturks				X				b
Deepen								
Diffgram	X			X				
hasty.ai		X						
Heartex								
Hive Data		X						
Image Tagger			X					
ImgLab	X							
Labelbox		X	X					
Labelimg	X		X					
LabelMe	X		X			X	GitHub	i
Make-Sense				X				i
Prodigy				X				
RectLabel			X					
Scale AI		X						
Super Annotate		X						b
Supervise.ly		X		X				b
V7's Darwin		X						
VGG Image Annotation Tool (VIA)	X			X			GitHub	b
VoTT	X			X	X	X	GitHub	b

5.5 Introducing the FinD Dataset, Set1: A Synthetic Outdoor Night Fire Dataset

This dataset is the initial set generated for nighttime VFD research by me and Asst. Prof. Dr. Kasım Taşdemir. A decent amount of dry bush and wood pieces are used as combustible agent during pure nighttime. The fires are ignited from a single point at it started to develop from the ignition point until its full size. Sketch of place that fire videos were recorded is given in Figure 5.1. The maximum distance a camera can see in the night is 1km and distance between fires and cameras changes from *30m* to *100m*. Besides fires, in 360^{o} sight of the cameras there are a series of streetlights, bright and dark roads, city lights, flashing tower lights, moving vehicle headlights in low- or high-density traffic, short distance house and streetlights.

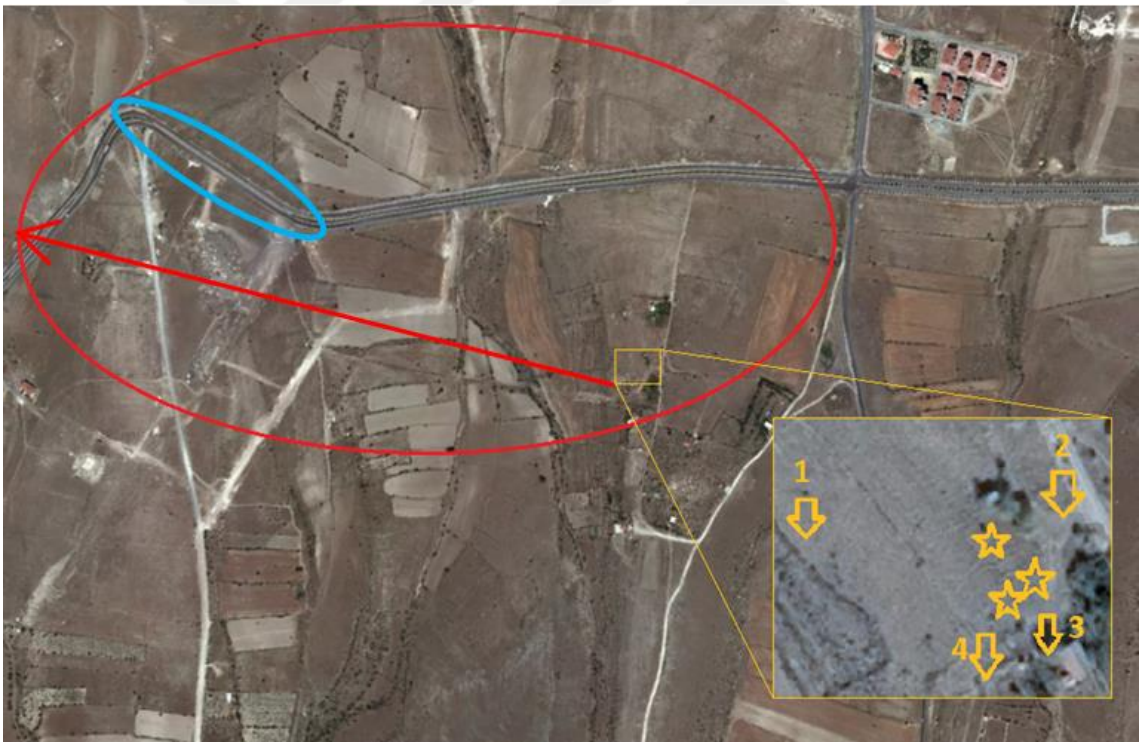

Figure 5.1 Dataset videos are intentionally taken from a place where possible negative light sources appear in the scene such as city lights or car lights. Location of test fires (stars) and cameras (arrows). Sight of the scene is shown in red circle. Maximum distance of sight from cameras is around 1 km.

Four different cameras are used recording fires usually in 640x480 resolution: Casio Exilim EX-Z350, Nikon D3200, Samsung S850, and Samsung WB100. In total, 15 night

fires are recorded. In Table 5.3, some characteristics of the videos are tabulated. Screenshots of each video is given in Figure 6.2 and 6.3.

Table 5.3 Properties of the samples in the video dataset.

Video	Duration	# of frames	# of negative samples	# of positive samples
1	9:41	17439	99	178
2	16:09	29091	265	148
3	21:23	38518	27	307
4	29:00	52231	17	671
5	04:00	6009	505	175
6	11:04	16608	2044	471
7	12:46	19158	530	405
8	20:00	30003	6	1343
9	12:05	21746	81	1508
10	14:14	25608	89	1608
11	18:10	32688	1208	927
12	20:00	35977	266	1876
13	08:34	15435	2681	1113
14	13:17	23913	7758	1839
15	03:25	6145	222	624

5.6 Introducing the FinD Dataset, Set2: A Natural Non-Urban Area Night Fire Dataset

Well-designed datasets are a backbone stage of developing automatic fire detection systems based on computer vision and machine learning.

In this work, a novel video dataset is created in response to scarcity of data sets particularly prepared for night-time fire events occurring at wild, rural, or suburban areas. The dataset contains night-fire videos collected from a number of public online video services for VFD research purposes.

Videos are collected from public video sharing services like Google, YouTube. A combination of search terms like fire, flame, night, wildfire, forest fire, disaster, night drive, lightning, firework, headlight, animation, flicker, etc., was used in search engines and video services to list candidate data. After beginning the search, automatic suggestions have been very helpful in accessing more useful and diverse data. It is important to note that searching for not only fire videos, but also fire-like videos is crucial

in creating a challenging dataset that can represent real life situations. Furthermore, videos of hand-made fires were discarded to let the algorithms trained on complex fire scenarios rather than comparatively simple experimental environments.

Each candidate video was evaluated in terms of data acceptance rules defined in Section 5.4.1. Data validity checks included if the candidate data is a video (i.e., not a GIF file) with any video format, has at least 20 frames, recording time is at night or almost at night with heavy smoke for fire videos and any time for deceptive non-fire videos, includes videos from natural fire events for fire class and includes videos strongly from fire events and fire-like deceptive events for non-fire class, and is not duplicate. Data quality included checking if the candidate data preferably has a higher resolution with a corresponding image quality. Data variety aimed by selecting various types of fire scenarios and scenes while accepting a video candidate. The scenario variety is summarized in Table 5.1.

The publicly available videos are mostly made available by news channels on their accounts at YouTube; therefore, they include many unwanted video fragments which should be handled at preprocessing step. All fire scenes were chosen from real-life fire incidents from 2013 to 2019. During data collection, videos were organized in terms of the incident country, location in the country, fire name, and incident time as much as possible to keep track of incident origins and prevent duplicate samples.

Creating the proposed dataset required several data preprocessing steps. The raw data collected from open sources in .mp4 format was generally not suitable to be readily used in model development. Thus, the data was cleaned from unwanted data fragments by cropping, trimming, cutting, resizing, removing duplicates, etc. The software tool used for this purpose was Adobe Premiere Pro®. The accepted data is then stored in .mp4 format with H.264 compression codec. Furthermore, the video data was organized for each video as either containing fire in all the frames or none of the frames. This lets all frames of a video be labeled as either fire or non-fire and prevent the network getting affected from counter labels during learning about a certain label. For example, a fire labeled video cannot contribute to a non-fire training process since the fire video contains the fire event in all frames. After completing these steps, all videos were added to the dataset in common video formats. In order to reduce computer work and accelerate analysis, a mat file version of the video files was created.

The dataset includes various human-interpreted captions. For instance, videos are captioned in terms of objects of interest that are being burned, such as tree fire, brush fire,

forest fire, vehicle fire, exterior building fire, interior building fire, and structure fire. Another example includes captions in camera movement, such as stable, including waggling, tilts, displacements, or zooms. These captions are extensively explained in Section 5.4.3 and summarized in Table 5.1.

A couple of daytime videos exist only for non-fire videos where tanker aircraft deploy fire extinguisher materials to the land area. They are added to challenge the network with fire-like objects. In total, 1835 videos comprise 1358 night-fire and 477 non-fire videos in the base dataset. Log-scale histogram charts that show frame number frequency of the videos are given in Figure 5.2. 90% of the videos are in 720x1080 resolution and minimum resolution (240x432) videos are only 2.2% of the dataset. A montage of fire and not fire samples are given in Figure 5.3 and 5.4.

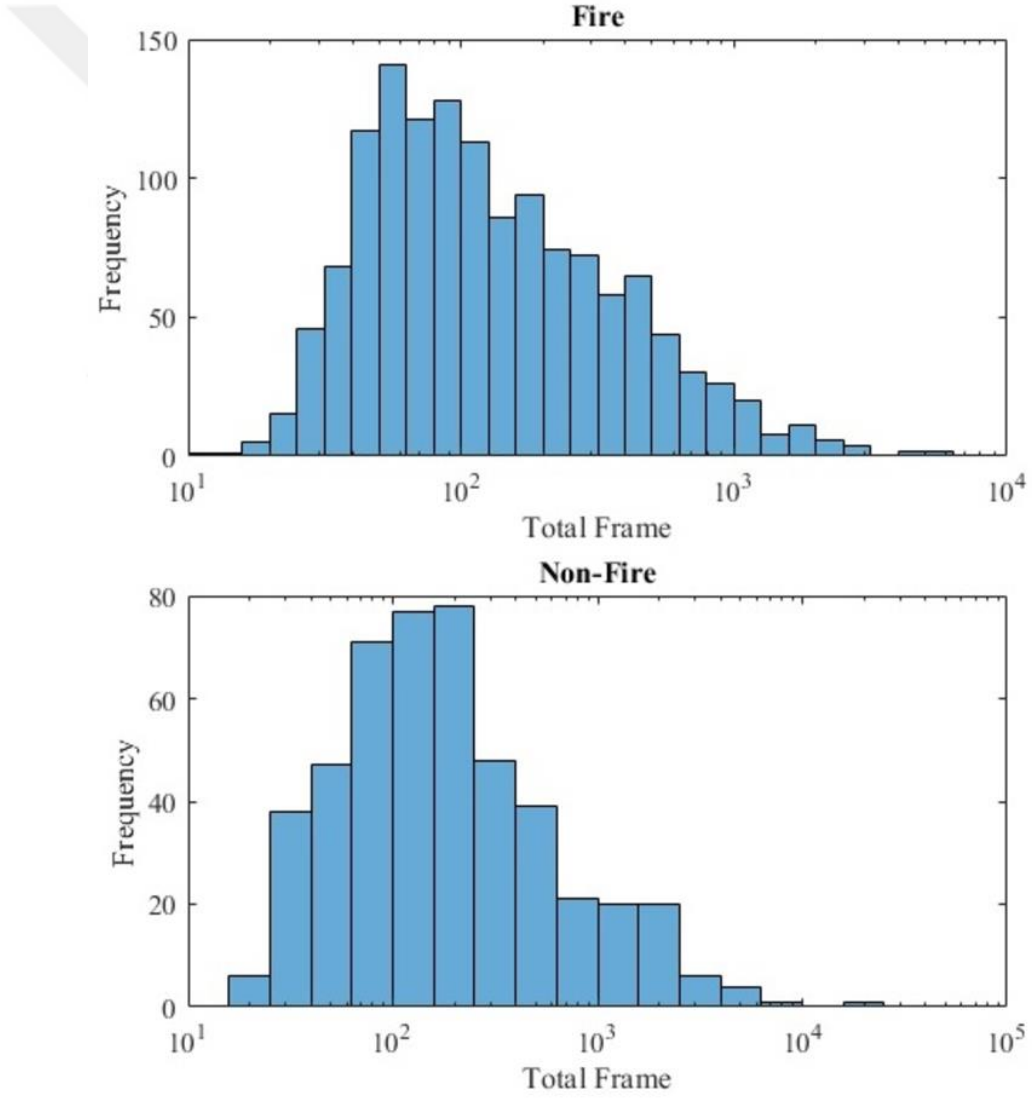

Figure 5.2 A log-scale distribution of the number of frames in both fire and non-fire videos

Figure 5.3 A montage of selected fire images from videos.

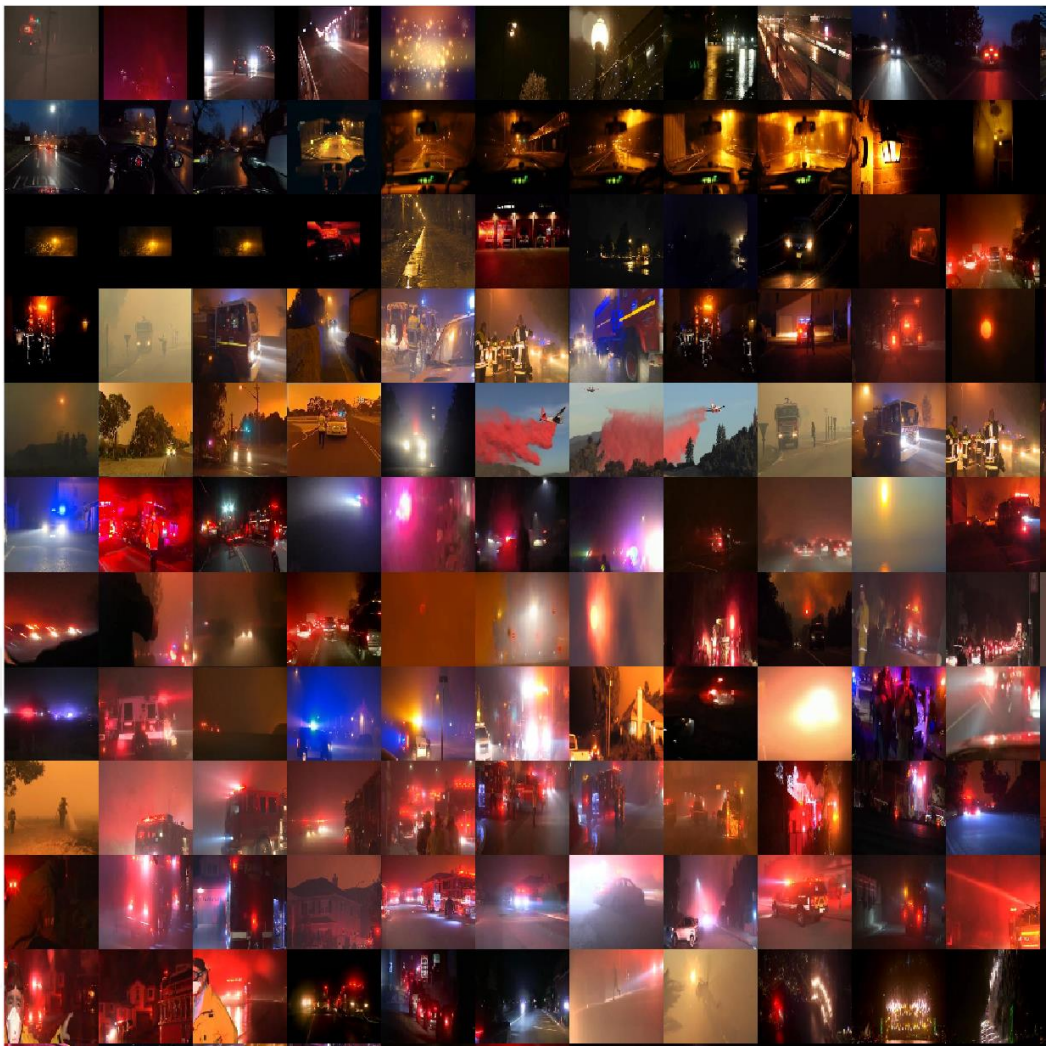

Figure 5.4 A montage of selected non fire images from videos.

Chapter 6

Night Fire Detection Using Hand-Crafted Features

In this section, a wildfire detection algorithm from dark videos is proposed. Unlike the daytime wildfires, in the dark videos, neither the fire nor its surrounding has visually clearly perceptible texture. Its unique visual characteristics make it challenging to extract descriptive object features. This section addresses the challenging problem by tracking the glowing objects in the darkness and extracting features based on the spatio-temporal behavior of them. It is experimentally shown that the proposed features are descriptive enough to classify wildfires with over 90% accuracy even there exists deceptive light sources such as city lights, flashlights, car headlights and reflections in the scene. Moreover, we investigate several conventional machine learning algorithms such as ensemble and kernel-based methods on the same spatio-temporal feature set. Comprehensive empirical test results demonstrate that the most accurate detection is obtained when the spatio-temporal feature set is classified using Random Forest.

6.1 Introduction

Beginning from 1900s, watch towers have been an important part of fire detection across the world. However, due to human factors, fire announcement procedures didn't work properly all the time which increased forest loses especially at rural areas. Employing surveillance cameras instead of lookouts made forest observation relatively easier. However, watching too many cameras by a limited number of staff is also not an easy task. For this reason, computer vision based automatic fire detection methods have

been an important welcome to the fire as well as forestry departments since they do not require any sensor deployment to the fire risk wild areas and besides a quick yes-no alarm, they support information of a fire via monitoring systems.

In this section, we propose a fire detection method that is able to detect short and mid-range fires while overcoming false alarm sources, such as city lights, car headlights, streetlights, etc.

Contribution of this section can be counted as three folds:

- A spatio-temporal feature extraction method including object tracking in dark video is proposed,
- Comprehensive comparison of ensemble and kernel-based classification methods on wildfire detection in dark videos are demonstrated,
- A final wildfire detection method which is robust against common source of false alarm sources in dark videos such as city lights or car headlights is proposed.

6.2 The Proposed Wildfire Detection Method

6.2.1 Extraction of Foreground Objects in Dark Videos

One challenging part of working on light emitting objects on the dark videos is they have limited visual features to be tracked or make any in depth visual analysis. For that reason, instead of visual ques of the object, we target to investigate its temporal behavior. However, we need to track an object throughout the video despite of the challenge. Light-emitting objects appear, disappear, flicker, move and even intersect with others or unmerge from the others in the video. All these cases are handled by the proposed object extraction and tracking algorithm.

Contrary to daytime counterparts, night-time videos contain very limited color information. They are very akin to digital binary images. Thus, without any color processing, each frame is converted to a black & white image with a threshold of τ_0 by using Otsu's method [54]. As a result, the dark pixels are represented by 0 and bright ones by 1. Binary blobs in each frame is detected with 8 connectivity adjacency rule. This eliminated disconnected or isolated foreground pixels. Blobs having fewer pixels than τ_1 are discarded to reduce number of blobs considered as noise. The reason 8 connectivity is used instead of 4 is the nature of a fire which has a very fragmented structure, thus,

when 4 connectivity is used there will be many small blobs belonging to same fire flame which makes analysis difficult. Let $b_{n,k}$ be k'th fire candidate blob of n'th video frame and o_m be mth object in the video. While in one frame o_m can be represented by kth blob, $b_{n,k}$, in succeeding frame it can be represented by $k+1$th blob, $b_{n+1,k+1}$. Then a tagging procedure should be implemented for each blob in each frame to uniquely index each object across the video with an ID. A tag will have a lifetime; a tag is born, lives for a while, and then dies as the object disappears from the video. Basically, light blobs not only appear and disappear from the video, but they also move, intersect or unmerge. For that reason, we need to have an algorithm to track these light emitting objects. If the subsequent frames have intersecting blobs, then it is considered that they are the same objects and so tagged with the same ID. In other words, if tagging function, $b_{n,k} \rightarrow o_m$, is known, tagging procedure is performed as follows:

$$b_{n+1,i} \rightarrow \begin{cases} o_m, & b_{n,k} \cap b_{n+1,k} \neq \emptyset \\ o_{m+1,} & otherwise \end{cases} \quad \textbf{(6.1)}$$

The equation indicates that if two object in consecutive frames are spatially intersecting, they are the same objects and they need to have the same ID. Initially, blobs in the first frame also tagged with their blob numbers. However, this approach has some difficulties. For example, if both $b_{n,k}$ and $b_{n,k+1}$ intersect with $b_{n+1,i}$, are those all the same objects? Another difficulty is if both $b_{n+1,i}$ and $b_{n+1,i+1}$ intersect with $b_{n,k}$, which objects are to be as separate?

In Figure 6.1, both difficulties given above are represented. Assume the first frame $n = 1$ contains seven blobs drawn in black circles and the second frame $n = 2$ contains eight blobs drawn in red circles. Blobs in the first frame take their blob numbers as ID tags, i.e., $b_{1,1}$ gets tag 1, $b_{1,2}$ gets tag 2, etc. Now consider $b_{1,1}$ and $b_{2,4}$ intersect most, then $b_{2,4}$ gets the tag 1. Next, $b_{1,2}$ intersects with $b_{2,2}$ most, thus $b_{2,1}$ dies, $b_{2,2}$ pairs with $b_{1,2}$ and gets the tag 2. Third, $b_{1,3}$ intersects with $b_{2,2}$ most, and thus both $b_{2,3}$ and $b_{2,5}$ die, $b_{2,2}$ pairs with $b_{1,3}$ and gets the tag 3. Fourth, $b_{1,4}$ intersects only with $b_{2,6}$ and $b_{2,6}$ gets the tag 4. Fifth, $b_{1,5}$ intersects only with $b_{2,6}$ and $b_{2,6}$ this time gets the tag 5. In similar way $b_{2,7}$ gets the tag 7, $b_{1,6}$ intersects with no one and dies, $b_{2,8}$ intersects with no one and is born by getting a new tag 8.

This operation made the second difficulty apparent: $b_{2,2}$ and $b_{2,6}$ have two distinct tags transferred to them. This conflict is resolved in a similar way: $b_{2,2}$ intersects with $b_{1,3}$ most when compared to $b_{1,2}$, thus gets the tag 3 and $b_{1,2}$ dies; $b_{2,6}$ intersects with $b_{1,4}$ most, gets the tag 4, and $b_{1,5}$ dies. In summary, tags 2, 5, and 6 dies, tags 1, 3, and 4 survives, tag 8 is newly born, however $b_{2,3}$ and $b_{2,5}$ are stillbirths.

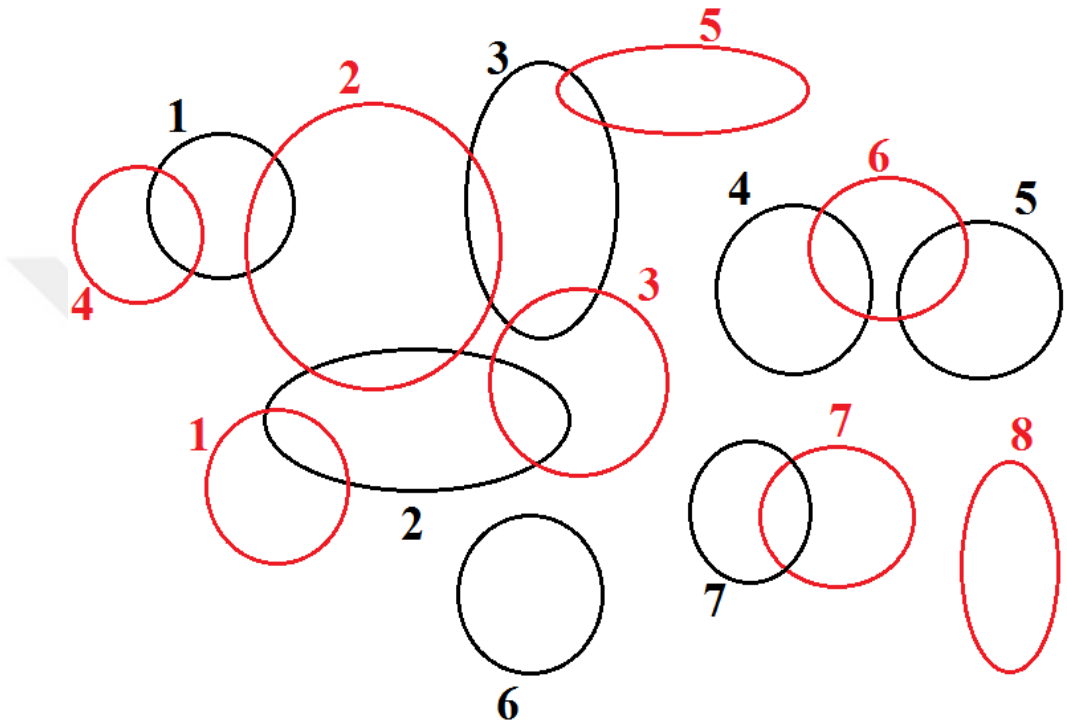

Figure 6.1 The figure shows possible scenarios that might come up during glowing object tracking in a dark video. Black and red circles indicate the foreground object location in the nth and the next frame, i.e., (n+1)'th frame. Since the flame has limited visual ques, their spatio-temporal locations are used to track the objects throughout the video.

6.2.2 Extracting Features

In order to capture the temporal behavior of the flickering flame, the features are extracted from a number of video sequences. Size of the temporal window is a tradeoff between detection time of fire alarm and its accuracy. In order to extract features from same number of frames, a tag that is not apparent along a full window is discarded from computations.

Features of a full-window tag are extracted from change in various motion variables of the tagged object. Thirty features are derived from these 6 variables which can be listed

as: pixel area of the object in frame, 2D position of the mass center of the object, height, width and area of smallest bounding box (BB) of the object.

By using these variables, we can realize distinctive characteristics of a night fire that are flickering and motion behavior. The variables are followed along a window and various features are extracted as explained presently.

While an object's variance of pixel area is large along a window, it is small for moving vehicles and fixed sources such as street, city, or house lights since area of such non-fire light sources does not change suddenly along a video. However, due to flickering motion of a fire, the area will change rapidly. Similarly, variance of height and width of BB will usually be large for fire objects and small for others. It is for this reason, mean and variance of height and width of a BB as well as their first and second order derivatives will be distinctive between fire and non-fire objects.

Let ψ_n be value of a variable at nth frame in a window with N number of frames. For many of these variables, mean and variance is computed as follows, respectively:

$$\mu_0 = \frac{1}{N}\sum_{n=1}^{N}\psi_n \tag{6.2}$$

$$\sigma_0^2 = \frac{1}{N-1}\sum_{n=1}^{N}|\psi_n - \mu_0|^2 \tag{6.3}$$

Mean and variance of first and second order derivative of some variables are computed as in (8.4) & (8.5) and (8.6) & (8.7), respectively.

$$\mu_1 = \frac{1}{N-1}\sum_{n=2}^{N}(\psi_n - \psi_{n-1}) \tag{6.4}$$

$$\sigma_1^2 = \frac{1}{N-2}\sum_{n=2}^{N}|(\psi_n - \psi_{n-1}) - \mu_1|^2 \tag{6.5}$$

$$\mu_2 = \frac{1}{N-2}\sum_{n=3}^{N}(\psi_n + \psi_{n-2}) \tag{6.6}$$

$$\sigma_2^2 = \frac{1}{N-3}\sum_{n=3}^{N}|(\psi_n + \psi_{n-2}) - \mu_2|^2 \tag{6.7}$$

For a fire object, variance of center of mass (CoM) is higher in vertical axis than in lateral axis. For a car moving horizontally, variance of CoM of headlights in vertical axis is very small compared to fire. In the same manner, variance of CoM of fixed light sources

in both axis is negligibly small. These are the reasons we used variance of CoM as a feature.

Here, it is important to note that horizontal and vertical location of CoM is not considered as features since a fire can take place anywhere in the video. Otherwise, the system can be trained for a specific location that fire is expected to start. That's why position free features (i.e., mean and variance of first and second order derivatives) are used. In Table 6.1, variables and features are summarized.

Table 6.1 Extraction of Features from Variables

	Feature		**Feature 1st Der**		**Feature 2nd Der**	
	Mean	*Var*	*Mean*	*Var*	*Mean*	*Var*
Pixel Area	x	x	x	x		
CoM x axis			x	x	x	x
CoM y axis			x	x	x	x
BB width	x	x	x	x	x	x
BB height	x	x	x	x	x	x
BB area	x	x	x	x	x	x

If a feature belongs to a greater interval than other features, impact of small-bounded ones may be reduced. Normalization is the solution to avoid such a problem. Min-max normalization has the ability to preserve relation between elements of a feature vector, thus it is chosen. Let $\delta_{i,j}$ be value of jth feature at sample i. Then, min-max normalization is defined as

$$\bar{\delta}_{i,j} = \frac{\delta_{i,j} - min\delta_{i,j}}{max\delta_{i,j} - min\delta_{i,j}{}_{i,j}} \quad \textbf{(6.8)}$$

In real-time applications, video stream may be continuous. Therefore, after adding a new window, normalization should be implemented throughout up-to-date data.

6.2.3 Training the Model

In this work, as a base classifier, Support Vector Machines (SVM) is used. Besides SVM, majority voting, Random Forests, AdaBoostM1, IBk, and J48 classifiers used, and their performance are compared to SVM. First a classifier model is constructed and then the model predicts class of any test instance it is supplied. Here, we used LIBSVM library with radial based function (RBF) kernel since our data set has a nonlinear classification characteristic.

In order to get most accurate classification, the best $c \in \mathbb{I}^+$ cost and $\gamma \in \mathbb{I}^+$ impact range parameters should be found. If $c_2 > c_1 > 0$ and $\gamma_2 > \gamma_1 > 0$ are predetermined

intervals, then optimization requires a $[c_1, c_2] \times [\gamma_1, \gamma_2]$ size grid search for the best (c^*, γ^*) pair [55]. In our tests, an accelerating intervention to optimization saved time and gave a better (c^*, γ^*) pair compared to pairs obtained when not intervened. The intervention is simple: after at least ten trials, if the last five trials produce a mean absolute deviation of accuracy no greater than 1.5, halt the search and use current pair as (c^*, γ^*).

Classes of instances in training set is determined by a professional for fire objects as 1 and for non-fire objects -1. With (c^*, γ^*) pair, a model is constructed in SVM and class of all instances from a distinct test set is predicted from the set $\{-1,1\}$. Accuracy and elements of confusion matrix (i.e., true positive rate, false negative rate, true negative rate, and false positive rate) are used as performance measures.

While SVM gives satisfactory results of predictions, majority voting (MV) improves these results significantly. In a test set, MV is implemented between distribution of fire or non-fire classification of an object tag. Then, class of the object is redetermined according to result of MV.

6.3 Setup of Experiments

Experiments implemented on a video dataset curated by the authors as depicted in Section 5.5. The global image threshold is experimentally determined to be $\tau_0 = 0.5$, thus objects with low luminance and noise can be eliminated. Furthermore, objects having pixels fewer than $\tau_1 = 16$ also discarded even fire objects since an event size lower than 16 pixels is not considered significant.

Analysis implemented in MATLAB® environment for window sizes of 5, 10, 20, 50, 100, and 200. SVM parameter optimization intervals are experimentally determined to be $c_1 = 5$, $c_2 = 9$, $\gamma_1 = 4$, and $\gamma_2 = 8$.

When a video is chosen for test, remaining ones are used for training (leave-one-out). For a total number of 90 experiments, average training and test set sizes are 7437 and 5058 instances, respectively. It should be noted that number of instances in a training set is limited to a maximum 10,000 while no restriction applied to test sets. Average distribution of fire and not-fire instances over 6 windows are tabulated in Table 6.2 and number of instances per window size is given in Table 6.3.

Table 6.2 Distribution of Not-Fire Classes Among Videos

Video	# of Average Not-Fire Classes (%)	Video	# of Average Not-Fire Classes (%)
1	56,31	9	0,09
2	66,86	10	0,4
3	63,89	11	10,35
4	62,2	12	1,36
5	3,34	13	92,54
6	8,89	14	91,23
7	8,1	15	18,32
8	0,08		

Table 6.3 Number of Instances Among Windows

Window Size	# of Instances
5	280,381
10	102,588
20	44,584
50	16,243
100	7,758
200	3,723

Representative screen shots of videos in not-fire mode and fire mode are given in Figure 6.2 and 6.3, respectively. Some fires were able to reach up to $ 3m $ height under low wind conditions. In videos 1, 2, 3, and 4, a very deceptive streetlight is apparent. From video 9 to 15, very deceptive city lights combined with semi-intense traffic are apparent. Besides these not-fire objects, a torch is also used to create false objects (Figure 6.3, video 3).

Figure 6.2 Camera screen shots showing both fire and not-fire objects. (Videos are numbered from first left to right then up to down)

Figure 6.3 Camera screen shots showing not-fire objects. (Videos are numbered from first left to right then up to down)

6.4 Experimental Results

In this section, performance of our method evaluated, and experimental results are analyzed. The measures we use for evaluation are accuracy, true positive rate which implies state of "true alarm", and false negative rate which implies state of "false alarm". Other measures can be false negative rate or "missed alarm" and true negative rate or "true silent".

6.4.1 SVM Results

Selected performance measurements of the proposed method is shown in Table 6.4. When proposed features are used for a night fire, SVM is able to classify new instances correctly with an accuracy of usually over 90%. Implementing MV after SVM classification boosts accuracy rates usually over 95%. TPR values are over 94% on average, however in some videos TNR values are low due to reasons given as follows.

Table 6.4 SVM Test Results

Video	Accuracy (SVM)	Accuracy (SVM+MV)	TPR (SVM)	TNR (SVM)
1	0,90	0,97	0,91	0,89
2	0,88	0,99	0,78	0,92
3	0,96	0,99	0,92	0,98
4	0,98	0,99	0,97	0,99
5	0,84	0,84	0,84	0,69
6	0,92	0,95	0,95	0,45
7	0,93	0,93	0,99	0,30
8	0,98	0,99	0,97	0
9	0,95	0,98	0,95	0,22
10	0,96	0,99	0,97	0,15
11	0,92	0,93	0,99	0,19
12	0,97	0,99	0,97	0,52
13	0,98	0,99	0,99	0,97
14	0,97	0,99	0,99	0,97
15	0,81	0,81	0,99	0,02

In videos 5 and 6, false alarm generating frame region is a sharp turn which is part of road in the scene (Blue circle in Figure 5.1). This part of the road extends from front to aft in the scene which makes vehicles move not quite linearly. Since traffic is semi-intense or intense during the recording time, vehicles slowed down and overhead lights

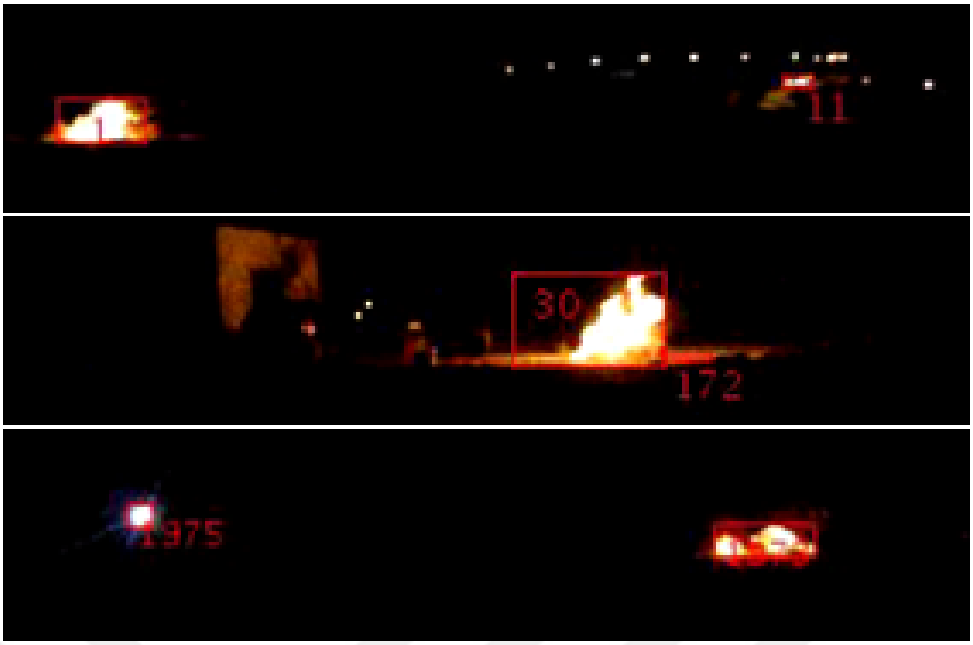

Figure 6.4 Error examples. Top: Independent of fire, 1: fire, 11 (right) not-fire, Middle: Fire dependent, 30: fire, 172 (bottom right) not fire, Bottom: Independent of fire, 1975 (left) fire, 1873 (right) not-fire.

clustered to form fire-like moving objects. In Figure 6.4 on top, a sample frame is shown. Even tough tag 11 is a not-fire object, it is detected as fire and boxed.

In videos 7, 8, 10, 12, and 15, reflections at not-fire regions on the ground up to half or one meter close to fire origin and intense luminance on objects very close to fire cause an error. In Figure 6.4 on middle, tag 30 is a fire objects a predicted as fire, however though tag 172 is a reflection and not a fire, it is classified as fire. Notice that, since these types of errors appear when a fire is the case, they do not really cause a false alarm.

In videos 9 and 11, a moving torch initially turns to the camera and then turns quickly back which makes the illuminated area first grow and then suddenly shrink, eventually causes an error. In Figure 6.4 on bottom, both tags 1873 and 1975 predicted as fire while tag 1975 is a not-fire object.

Window size also has an effect on accuracy. When window size increases, more evidence per window is collected for decision process which allows better predictions. For pre-processing, which includes tagging procedure, more computation is required. However, for SVM runs, less computation is the case. When window size decreases less evidence per window is collected, less pre-processing computation and more SVM computation is required. Table 6.5 shows performance measures for two windows: $N = 5$ and $N = 200$. When $N = 5$, average accuracy is 89.47% and when $N = 200$, accuracy also increases to an average of 96.63%. An increase in window size also decreases false

alarm rates. For example, in Figure 6.3 and video 1, the street, the house, and a torch light are predicted as fire objects. When window size is 200, at the beginning of the video house lights very short time, the torch never and due to move of the camera at the end of the video the streetlight very short time are predicted as fire. Even though this encourages us to use longer windows (preferably with high fps cameras) due to heavy work of pre-process, alarm response time will eventually decrease. In Table 6.5, NaN corresponds to existence of no not-fire objects in the video. In videos 11 and 15, TNR value is 0 due to misclassification of intense luminance of a vehicle standing very close to fire (Figure 6.2).

Table 6.5 Comparison of window sizes of N=5 and N=200

Video	Accuracy (%) (SVM)		TPR (SVM)		TNR (SVM)	
	N=5	N=200	N=5	N=200	N=5	N=200
1	88,28	91,09	0,96	0,87	0,81	0,93
2	82,98	91,32	0,77	0,8	0,85	0,96
3	92,25	98	0,88	0,94	0,94	1
4	97,43	98,78	0,95	0,97	0,98	0,99
5	79,03	91,89	0,79	0,91	0,78	NaN
6	86,61	100	0,93	1	0,61	NaN
7	88,03	100	0,99	1	0,12	1
8	95,87	99,20	0,95	0,99	0	NaN
9	89,59	100	0,89	1	0,43	NaN
10	94,90	98,039	0,95	0,98	0,17	NaN
11	87,68	98,47	0,99	1	0,36	0
12	95,10	98,78	0,96	0,98	0,51	NaN
13	95,66	99,34	0,94	0,98	0,95	0,99
14	94,61	99,54	0,98	1	0,94	0,99
15	73,95	85	0,96	1	0,01	0

Apart from errors explained above, the proposed method successfully does classify street or city lights, headlights of vehicles and many other not-fire objects.

6.4.2 Other Results

SVM is a standard tool for image classification problems. However, there exist some other tools performs equally, some even better. In this section, we implement Random Forests (RF), AdaBoostM1 (AB), IBk and J48 machine learning tools on our data. Performance measures are the same as we used for SVM at previous section. The platform used for implementation is Weka data mining software by The University of Waikato. Default parameter set up is used for the tests. Contrary to SVM experiments,

training data is not limited, and full training set is used for building a model (see Table 6.3). In addition to SVM, 360 more tests are implemented, one test per video per window size, and per machine learning tool.

Accuracy results are given in Table 6.6. In the table, except videos 11 and 15, RF showed the best performance among other tools. Second the best performance belongs to J48. AB showed the best performance for video 15 and IBk for video 11. Videos 3, 4, 6, 7, 8, 10, 11, 12,13, and 14 show a robust performance under any machine learning tool while videos 1, 2, 5, 9, 15 shows unstable performance. In Table 6.7, TNR values are tabulated. On average, IBk gives lowest average false alarm rate of 32.01% and SVM gives the highest average rate of 44.75%. Most robust videos in terms of TNR value are videos 3, 4, 13, and 14. After all, all these analysis shows us in terms of fire catch RF performs the best, however in terms of false alarm avoidance IBk performs the best.

Table 6.6 Accuracy comparison of SVM, Random Forests (RF), AdaBoostM1 (AB), IBk and J48

Video	SVM	RF	AB	IBk	J48
1	90	94,4	83,9	91,5	84,7
2	88	94	83,1	89,6	87,9
3	96	98,6	92,8	94	96,5
4	98	98,4	95	95,6	97,4
5	84	88,3	72,8	78,4	85,3
6	92	92,6	86,3	87,6	88,7
7	93	95,1	92,8	93,8	95
8	98	98,8	95,5	94,4	98
9	95	97,2	85,4	86,5	93
10	96	98,9	96,8	94	98,3
11	92	93,8	93,6	94	93,1
12	97	98,6	95,6	93,1	97,9
13	98	99,3	98,1	97,8	98,6
14	97	98,3	97,3	97,2	96,8
15	81	84,6	88	79,3	82,5

Table 6.7 TNR comparison of SVM, Random Forests (RF), AdaBoostM1 (AB), IBk and J48

Video	SVM	RF	AB	Ibk	J48
1	10,4	8,38	17,2	9	22,72
2	7,7	5,55	21,26	3,9	10,59
3	1,9	0,76	7,58	1,5	2,9
4	1,1	1,12	4,29	1,3	2,4
5	30,8	13,31	15,77	18,86	16,4
6	54,5	53,91	51,19	58,1	56,4
7	69,9	68,9	84,5	78	70,3
8	100	83,3	100	0	33,3
9	77,2	54,7	80	29,7	36,6
10	85,3	89,2	93,9	85,52	94,5
11	80,2	57,12	60,8	48,6	62,2
12	47,7	4,84	56,6	44,2	28,8
13	2	0,72	1,7	1,9	1,36
14	2,8	1,86	2,83	2,9	3,38
15	99,8	68,9	37,72	96,7	78,7

6.5 Concluding Remarks

In this section, a video-based wildfire detection method for under-illuminated environments is proposed. The experimental results show that temporal behavior of the flickering flame in a dark video has a distinct characteristic, and it is well suited for flame and fire detection in low light conditions. This temporal behavior of the fire allows us to extract descriptive spatio-temporal features from a fire video even the visual texture of the objects in the dark video are not visible. The proposed object features are taking advantage of temporal flickering motion of a night fire. The classification method can distinguish the deceptive false alarm sources such as city and streetlights, vehicle headlights and flickering reflections.

It is experimentally verified that the fire detection accuracy of the proposed method is over 90% on the average.

The method is tested with various hyper-parameters such as temporal window size. It is shown that when the temporal window size is increased to include 200 consecutive frames, over 95% accuracy on average was obtained.

The proposed object features are tested with various classification methods such as SVM, Random Forests, AdaBoostM1, IBk and J48. The comprehensive comparison

shows that Random Forests classification attains the highest accuracy on the extracted features. It is also shown that the detection accuracy of IBk is comparable to the most accurate model. Moreover, among all tested machine learning algorithms, IBk gives the smallest false alarm rate, 32.01%, while SVM gives the highest. Therefore, when the reduction of the false alarm rate is more critical, IBk can be employed.

Chapter 7

BLSTM Based Night-Time Video Fire Detection

Distinguishing fire from non-fire objects in night videos is problematic if only spatial features are to be used. Those features are highly disrupted under low-lit environments because of several factors, such as the dynamic range limitations of the cameras. This makes the analysis of temporal behavior of night-time fire indispensable for classification. To this end, a BLSTM based night-time wildfire event detection from a video algorithm is proposed. It is shown in the experiments that the proposed algorithm attains 95.15% of accuracy when tested against a wide variety of actual recordings of night-time wildfire incidents and 23.7 ms per frame detection time.

Moreover, to pave the way for more targeted solutions to this challenging problem, experiment-based thorough investigations of possible sources of incorrect predictions are discussed.

7.1 Introduction

Fire videos can be categorized as daytime and night-time. Night-time wildfires have a considerable percentage among fire incidents as indicated in Section 1.2. Nevertheless, detection of night-time wildfires from videos has not been used effectively due to its challenges.

There are several challenges to wildfire detection from a night-time video. Some are related to the nature of the fire, and others are more related to the camera's limitations.

Compared to its low cost, RGB cameras bring their own challenges to the task of fire detection, especially when night fires are in question.

Historically, previous studies that worked on wildfire detection with night videos employed hand-crafted features until the 2013s [56]. Since then, the paradigm has shifted from hand-crafted features to hand-crafted networks. Today, the recent approach generates features out of the networks, which has been possible by convolutional neural networks (CNN).

CNN-based methods have shown their effectiveness on object detection tasks. However, it is challenging to detect night fires from RGB cameras with well-known object detection algorithms for night-time fire detection in video. One particular reason is that it is not benefiting from the temporal relation of the frames. To alleviate this limitation, employing Recursive Neural Networks (RNNs), which can model a video as a data sequence, has been an option.

2D CNNs can be used for extracting spatial features and RNNs for extracting temporal features. Cascaded CNN+RNN structure is a well-known approach used in various fields such as video description extraction, action recognition, etc., and its effectiveness is shown in multiple studies [57]. Using both CNNs and RNNs in a pipeline has the potential to increase detection performance. This study is the first time that approach has been used against night-time wildfire detection problems to the authors' best knowledge.

The present work proposes a two-stage approach combining spatial and temporal information of an object appearing in a night-time video. The first stage (CNN) computes spatial features, and the second stage (RNN) makes the temporal analysis depending on these features. The CNN stage employs transfer learning on a pre-trained GoogLeNet [58] to reduce the training time of the overall network. The second stage employs the bidirectional long short-term memory (BLSTM) network and is trained with feature maps obtained from the first stage for each video frame. After the network pipeline is trained, it can readily be used for detection in for example watchtowers that are equipped with CCTV cameras. The network can be deployed in two ways. First, it uses the weights determined with the initially training-test procedure and they are not updated in response to different fire or non-fire samples events. Second and the adaptive one is the pipeline continuously updates itself by simultaneous re-training iterations. By doing so, the pipeline always becomes up to date for changing environmental conditions.

As mentioned before, the night fires have a unique nature. Due to the low-lit environment and extremely bright fire objects appearing in the same scene, the physical limitations of the camera, such as the dynamic range, give a unique digital video. Investigating the typical features of night-time wildfire videos, discussing possible sources of incorrect classifications and possible solutions are essential for developing well-targeted solutions. These are also thoroughly investigated and discussed in this study.

Therefore, the novelties and contributions of this study can be summarized as:

- The proposed method incorporates both the spatial and temporal behavior of a night-time wildfire by using a CNN+RNN based network and detects fire at min of 23.4 ms per frame.
- It employs BLSTM for capturing both forward and backward temporal relationships in the night-time wildfire video,
- It uses decisions from spatial and temporal networks to employ majority voting to improve the prediction accuracy,
- The data samples which give the most failure in night-time wildfire detection tasks is identified and carefully investigated, and the nature of night-time wildfire videos is discussed. It is revealed in CNN+BLSTM networks that a non-fire event that is seen on fire scenes has potential to suppress the fire event and revert the decision as “non-fire” instead of “fire” or vice versa.
- A novel night-time wild, rural, suburban area fire detection dataset is proposed to push night-time video fire detection (VFD) research forward.

This chapter is organized as follows: In Section 7.2, the proposed method is explained. In Section 7.3, the experimental setup is illustrated. In Section 7.4, the results of the experiments are discussed, the performance of the GoogLeNet+BLSTM network is evaluated, and majority voting introduced to improve prediction performance. In Section 7.5, misclassifications are discussed in detail and finally in Section 7.6, the findings are summarized, and the conclusions are drawn.

7.2 The proposed method

Distinguishing fire from non-fire objects in night videos is problematic if only spatial features are to be used. Those features are highly disrupted under low-lit

environments because of the physical limitations of the camera and other reasons, as discussed in Section 2.1. This makes the temporal behavior of the bright object indispensable for classification.

To capture the temporal behavior of a fire object along with its spatial features, a coupled spatio-temporal behavior analysis is crucial. To this end, a spatio-temporal network structure consisting of CNN and RNN is proposed. The proposed network first extracts the spatial features of fire candidate videos of various lengths with the help of a pre-trained GoogLeNet CNN network, as explained in Section 7.2.1. Second, temporal learning is performed using a BLSTM RNN network, as explained in Section 7.2.2. In Section 7.2.3, the cascaded CNN+RNN model is demonstrated (Figure 7.1).

7.2.1 The first stage: spatial feature extraction

The first stage of the proposed network is spatial feature extraction. Since the detection of fire will be conducted on sequences of images; the model should be able to process image data and obtain spatial characteristics that will be essential in understanding objects in a scene.

A pre-trained GoogLeNet architecture is picked for spatial feature extraction. The GoogLeNet architecture set a new state of the art for object detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC 2014). In this work, the network was pre-trained on the ImageNet [49] and is available from MATLAB®.

GoogLeNet has been used in previous studies [59]. Their results show that it has a high detection accuracy of 96.7% on a subset of ImageNet with flame, smoke, and other flame-like labels. In another study, GoogLeNet's performance is tested and compared with other well-known models [60]. It reports that GoogLeNet attains 99% accuracy, which is the highest among AlexNet, VGG13, and modifications of them when tested on wildfire videos taken from UAVs. Researchers designed their own models inspired by GoogLeNet because it has higher accuracy than models like AlexNet and is easily adaptable to field-programmable gate array (FPGA) platforms [61] and [62]. They received a 94.43% on BowFire and MIVIA Fire Detection datasets and 93% accuracy on Furg Fire Dataset, respectively, with their modified GoogLeNet network on respective fire detection video datasets. Finally, it was reported that Inception-v3 leads to 2.5% more computational cost than GoogLeNet (Inception-v1) [63]. With the findings on day-time fire datasets mentioned above in the literature and extensive performance comparisons

[64], it is conjectured that GoogLeNet is a reasonable choice for less computational complexity, less model complexity, and relatively high accuracy.

GoogLeNet approximates dense layers by employing local sparse units. These units (Inception modules) can be repeated spatially in the architecture. An Inception module receives input from a previous layer, processes convolutional layers with different kernels in a parallel fashion and concatenates all parallel outputs depth-wise into one tensor. To reduce network resolution, max-pooling layers are used. The Average-pooling layer is used instead of an extra fully connected layer, leading to additional over-fitting [65]

We implement transfer learning with the GoogLeNet network that is pre-trained on ImageNet. Each frame is fed to the pre-trained GoogLeNet CNN network, and a corresponding feature map is extracted from the final average pooling layer. Thus, a video, as an image sequence of size $(H \times W \times 3 \times N)$, is converted to a tensor of size $(1 \times 1 \times 1024) \times N$ where H, W, and N is the height, the width, and the number of frames of the video, respectively. The sequence of these vectors is used for further temporal analysis, using the BLSTM network [66], as explained in the following section. In Figure 7.1, the dashed blue box shows a standard GoogLeNet network structure. GoogLeNet receives images or sequences of images in 224 × 224 size.

7.2.2 Temporal analysis

Long short-term memory (LSTM) is a special kind of recurrent neural network that can learn from sequentially related data without losing essential features throughout time [67]. In other words, LSTM networks can learn from past events and use this knowledge to classify present events. In order to keep track of the past, it requires a useful summary of the past carried to the present. This is accomplished by an updating cell state also termed as long-term memory. The long-term memory is updated by dropping insignificant information and keeping the significant one by distinct internal neural networks. There is another state known as hidden state and is required to update short-term memory and generate an output prediction. Short-term memory is also obtained by another internal neural network. In the end, the LSTM network makes predictions for a given input by keeping track of long-term and short-term 'past experience'.

This property makes them a prominent candidate for video captioning [68]. The building block of an LSTM network is a cell engine that receives input of the current time step along with the cell state and output of the previous time step (hidden state) to generate the current cell state and output. Then these are fed to the next cell iteration.

LSTM is also applied to daytime fire detection problems in [69]. The obtained accuracies are 97.92% and 93.3%, respectively.

However, LSTM models require a longer training time than CNN models since they cannot be run in parallel. On the other hand, LSTM architecture can infer results only in a feed-forward direction, i.e., from past to present. It cannot generalize predictions with the valuable knowledge from the `future.' For example, when an LSTM network is to predict the next word of the statement "I like to make ...", there are numerous options to

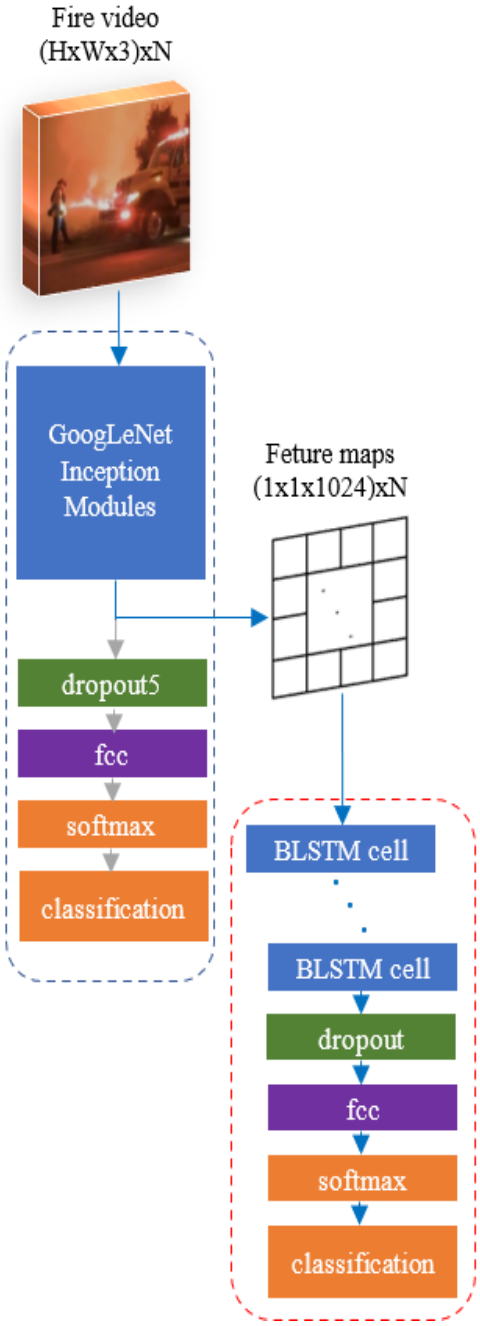

Figure 7.1 GoogLeNet+BLSTM classifier architecture. Both the GoogLeNet and BLSTM are trained networks. They are connected to each other by pruning final four layers of the GoogLeNet. Pruned CNN version outputs feature maps which are input to BLSTM.

choose from. However, if the network were to have a subsequent statement of "I believe melodies heal people" then the prediction would likely be "music" rather than i.e., "cake". In the similar way, predicting the fire events in a video not only via the past experience but also with respect to knowledge of the "future" can be accomplished by a bidirectional LSTM (BLSTM) [70].

Considering that the night-time fire video scenes are inadequate in terms of spatial features compared to daytime videos, the contribution of temporal features to the decision process becomes increasingly important. To capture the temporal behavior of the candidate object and its advantages over LSTM, the BLSTM network is used as the second stage of the proposed method. In Figure 7.1, dashed red box, shows a standard stacked BLSTM network structure in rolled form. A BLSTM network receives a series of data in 1024 × 1 size. If N number of such vectors is the case, then input is in 1024 × N size. Blue arrows starting from the input video and ending at the classification box show the flow diagram of the proposed method.

7.2.3 Model architecture and pipeline

Training the overall network normally includes training of two sub-networks: CNN and BLSTM. However, when it is available, adopting a pre-trained network is useful in order to reduce overall network training time and obtain a working final classifier as soon as possible. In Section 7.2.1, it is mentioned that a pre-trained model is adopted only for the CNN model, which is also termed as transfer learning. Therefore, spatial feature extraction is obtained by using a pre-trained CNN network given in Figure 7.1. In the figure, the full-stack trained CNN network receives never-seen images and generates a prediction in the end. In this work, the full stack trained CNN is not used as is, in fact is only used to extract features of the video frames from the final pooling layer's output of the GoogLeNet network. This implies that final dropout5, FCC, and Softmax layers are excluded from the overall classifier. In Section 7.2.2, these features are used to train a BLSTM network given in Figure 7.1.

Finally, these two stages are connected to each other, as shown in Figure 7.1. Connecting the two stages requires two trained models adapted to a pipeline model with the following steps. First, since the pipeline will receive videos, the image input layer of CNN is replaced by a sequence input layer and input videos are converted from sequences of frames to a tensor of images to let the CNN convolutional layers receive video data image by image. Second, CNN is not expected to output predictions but only generate

feature maps; as a result, dropout5, fully connected, Softmax, and classification layers are unnecessary and truncated from the CNN structure letting the last CNN layer be pool5 which is the final global average pooling layer of GoogLeNet. Third, the pool5 layer will be the input layer of the BLSTM architecture, so the input layer of the BLSTM layer is dropped and the remaining structure is kept as it is. Finally, the adjusted and truncated CNN is connected to truncated BLSTM to obtain the end-to-end classifier pipeline.

In summary, an N frame-long video of size $H \times W \times 3$ is given as an input to the first layer in the figure. The data is processed through the CNN layers until the average pooling layer. Here, the final feature maps are generated as a $1024 \times N$ tensor, and it is fed as input to the first cell of the stacked BLSTM layers. The fully connected final layer outputs the probabilities of the two classes. The class with the highest probability in the Softmax layer is finally labeled to be fire or non-fire.

7.3 Experimental setup

7.3.1 Preprocessing

In the preprocessing step, the data is organized for network training and test. Steps performed in the preprocessing step are shown with gray arrows in Figure 7.2.

Since base videos are in various sizes and a CNN network accepts the input in only a specific size, all fire and non-fire RGB video frames are resized to $224 \times 224 \times 3$. This yields the resized base dataset with 1835 videos, each of which is the size of $224 \times 224 \times 3 \times N$.

We repeated the experiments for various video lengths to investigate the proposed method's fire detection speed and accuracy. We name the video lengths as window size, N, referring to the number of frames in the temporal window. Since the detection speed from 2/3 to 2 seconds suffices for near real-time detection, smaller temporal window sizes at around 60 frames are preferable in a practical sense.

The base videos are sub-sampled with various time windows, N. Assuming that the videos are in 30 fps, the video length, $N \in \boldsymbol{N},\ \boldsymbol{N} = \{20, 30, 45, 60, 75, 90, 120\}$, would give 2/3 to 4 seconds detection latency. We picked max N sequential frames for a window size, N, starting from a randomly determined time position in each base video. In this way, we construct a new intermediate dataset with 1835 shorter videos corresponding to the window size, N. From each base video, only one sub-sample is extracted to ensure

that BLSTM blocks do not memorize the similar scenes that belong to the same base video. This random subsample dataset generation step is repeated for each fold in the experiments. That is, for each fold, a new intermediate sub-sample dataset is generated randomly.

As mentioned in Section 7.2, A CNN network is not trained from scratch, and a pre-trained GoogLeNet architecture is used. The pre-trained GoogLeNet network model was used on the sub-sample dataset for spatial feature extraction. Thus, each video yielded a feature map of size $1024 \times N$. This map was used in the following step, the BLSTM network.

We trained a BLSTM network with a feature map set constructed for each given N window size. We repeated the same training process for $k = (1,2,\ldots,5)$ fold for each N and network settings. Given a window size N, train, validation, and test sets are generated randomly from the corresponding feature maps for each fold k.

There are 477 non-fire negative videos in the base video set. To have balanced positive and negative samples in both train and test sets, we picked 477 feature maps randomly out of 1358 positive samples. This constituted randomly picked 477 fire and 477 non-fire intermediate feature maps set for each (k, N) pair. The intermediate feature maps set was randomly split into three disjoint sets: 70% for training, 10% for validation during training, and 20% for testing. It should be noted that test, train, and validation sets are taken from completely different scenes. If a sample video taken from a base video is in the train set, another sample video taken from the same base video at different intervals cannot be in any of the train, the validation, or the test sets. In this way, we have a more reliable testing scenario because the train and test sets have entirely different scenes. To this end, for each (k, N) pair, $35 = |\boldsymbol{N}|\boldsymbol{max}(k)$ intermediate datasets each of which contains its own training, validation, and test sets are constructed (See Figure 7.2).

7.3.2 Model construction and experiments

This method requires CNN and BLSTM parts trained separately, and then the pre-trained networks are concatenated to construct an end-to-end classifier network. Since a pre-trained GoogLeNet network is used instead of training the CNN from scratch, the only part left to be trained is the BLSTM network based on the extracted features from the pre-trained CNN (Figure 7.2). These feature sets taken from the CNN are given as input to the BLSTM block for temporal behavior analysis.

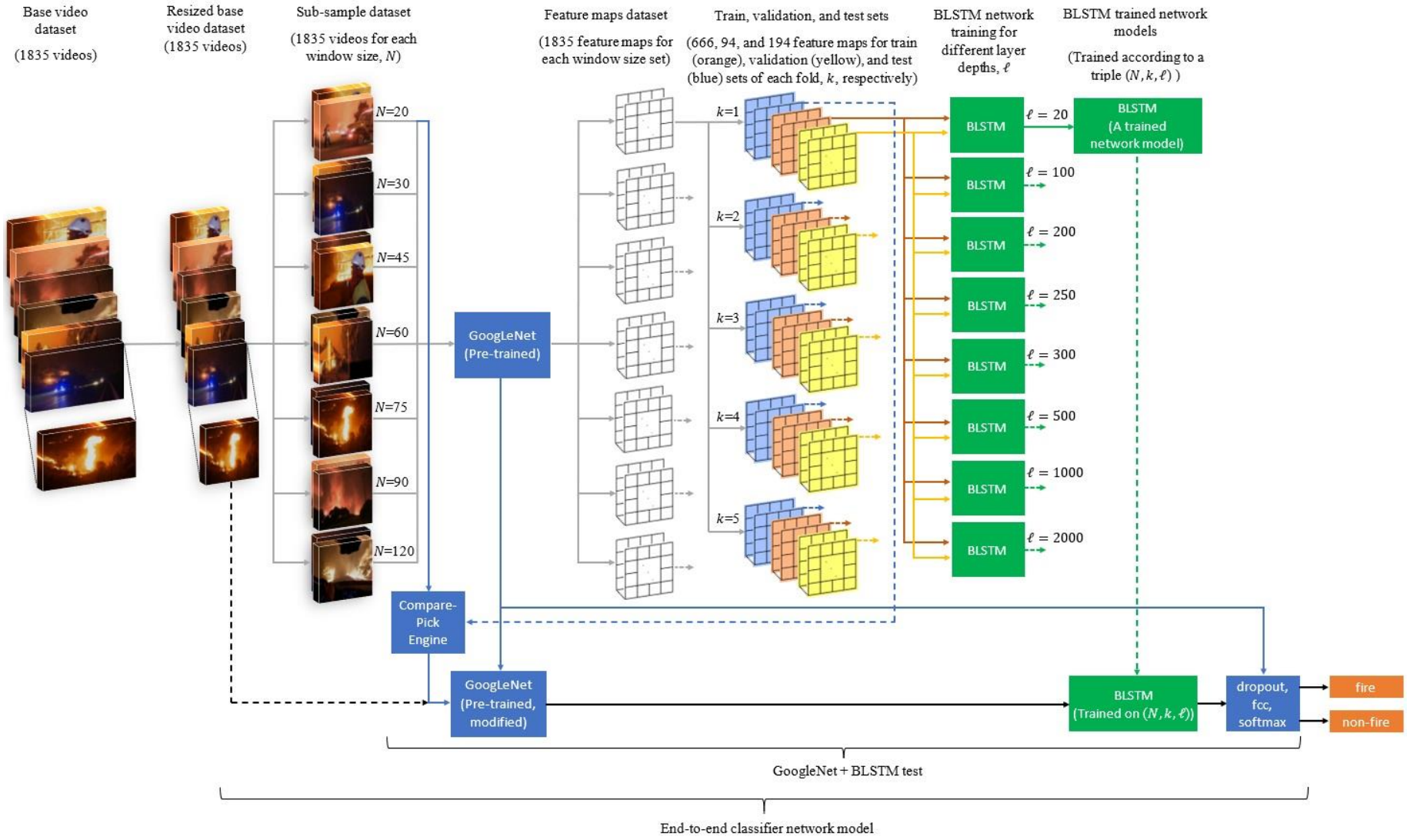

Figure 7.2 Stages of the proposed method. First, both networks should be trained separately and later connected to each other for classification. In this scheme, a pre-trained GoogLeNet used. Trained models are generated for all (N, k, l) triples.

The experiments are conducted in the MATLAB® environment on the Intel® Xeon® CPU E5-2620v2 2x2.1GHz 96GB memory hardware set.

The experiments are performed for various BLSTM network depths, $\ell \in \boldsymbol{L}, \boldsymbol{L} = \{20, 100, 200, 250, 300, 500, 1000, 2000,$ on train, validation, and test sets. Moreover, each of these experiments repeated for various window sizes N for k fold. During the training, the batch size is set to be 16, and the dropout rate is 50% to prevent overfitting. The training initially continued for 30 epochs at the first three folds. In these experiments, it is observed that no improvement occurred in validation accuracy and loss after 12 epochs, i.e., the accuracy and loss graphs stalled. Therefore, to save training time, we decided that the initial setup of 30 epoch is not a good fit. and 12 epochs would suffice for the remaining experiments. Additionally, after experimenting with greater learning rate values, 0.0005 and 0.001, the learning rate was finally set to 0.0001. Since our dataset has large and many data samples, it was important to conduct experiments in viable time and computing power without conceding accuracy; therefore, Adam optimizer was chosen rather than SGDM with recommended parameters in [71]. In RNN network training, gradients can easily explode to unstable values which require limiting gradients not exceeding a threshold. The threshold value for gradients is set to be 2. In the experiments, there were 666 training samples per a training session with a batch size of 16. This makes 10 samples out of 41 batches if batches would not be shuffled after each epoch. In order to make the network see all training samples and prevent the network stuck in a local minimum, batches were shuffled at every epoch. After a training session is completed for an epoch, a validation was performed on the validation set.

In short, we investigated the effect of window size and network depth on various performance scores amounting to $56 = |L| \times |N|$ test scenarios, s_i as given in Eq. 10.1, by selecting sub-sample video lengths from 20 to 120 frames and network depth from 20 to 2000 stacked cells. Each scenario was tested with the same hyper-parameters, optimized with a smaller video set in the previous steps.

$$s_i \in \boldsymbol{S}, \boldsymbol{S} = \{\ell_i, N_i\}_{i=1}^{|L|\times|N|} \quad \textbf{(7.1)}$$

Then the scores for the experimented scenario, s_i, are obtained by averaging the five-fold validation results. A chart of averaged validation accuracy values and F1 scores for all scenarios is given in Table 7.1 and 7.2. The plots of 5-fold average accuracy values and F1 scores for all scenarios are given in Figure 7.3. The average accuracy and F1 score

of all scenarios are 92.2% and 92.1%, respectively. The maximum values of 5-fold average validation accuracy and F1 scores have been 94.5% and 94.4%, respectively, observed at $N = 30$ and $\ell = 100$. The validation results in Figure 7.3 and Table 7.1 & 7.2 shows that deep layers and long video samples do not contribute to training more than shallow layers and short videos.

Table 7.1 5-fold averages of the validation accuracies of the proposed model for various window sizes and layer depths. The highest score is in boldface

		Window Size						
		20	**30**	**45**	**60**	**75**	**90**	**120**
Layer Depth	**20**	91.7	93.6	91.1	91.5	89.4	91.1	89.4
	100	93.2	**94.5**	92.1	91.7	90	92.3	90.2
	200	93.2	94.3	93	91.7	91.3	92.1	91.3
	250	94.3	93.4	92.6	92.6	91.1	91.9	91.9
	300	92.8	93.8	91.9	91.9	91.1	92.6	91.1
	500	93.2	94	92.8	92.8	91.1	92.1	91.5
	1000	94.3	94.3	93.2	93.2	91.5	92.1	92.3
	2000	93.4	94.3	92.6	92.6	90.6	92.6	91.3

Table 7.2 5-fold averages of the validation F1 scores of the proposed model for various window sizes and layer depths. The highest score is in boldface

		Window Size						
Layer Depth		**20**	**30**	**45**	**60**	**75**	**90**	**120**
	20	91.6	93.6	90.5	91.6	89.1	90.9	89
	100	93.1	**94.4**	91.8	91.8	89.4	92.3	90
	200	93	94.1	91.9	91.8	90.9	91.9	91
	250	94.1	93.3	92.8	92.5	90.7	91.9	91.6
	300	92.6	93.8	92.3	92	90.5	92.3	90.7
	500	93	94	92.5	92.8	90.6	92.1	91.3
	1000	94.1	94.2	91.7	93.2	91.1	92.1	92.2
	2000	93.3	94.2	92.1	92.7	90.4	92.5	91.1

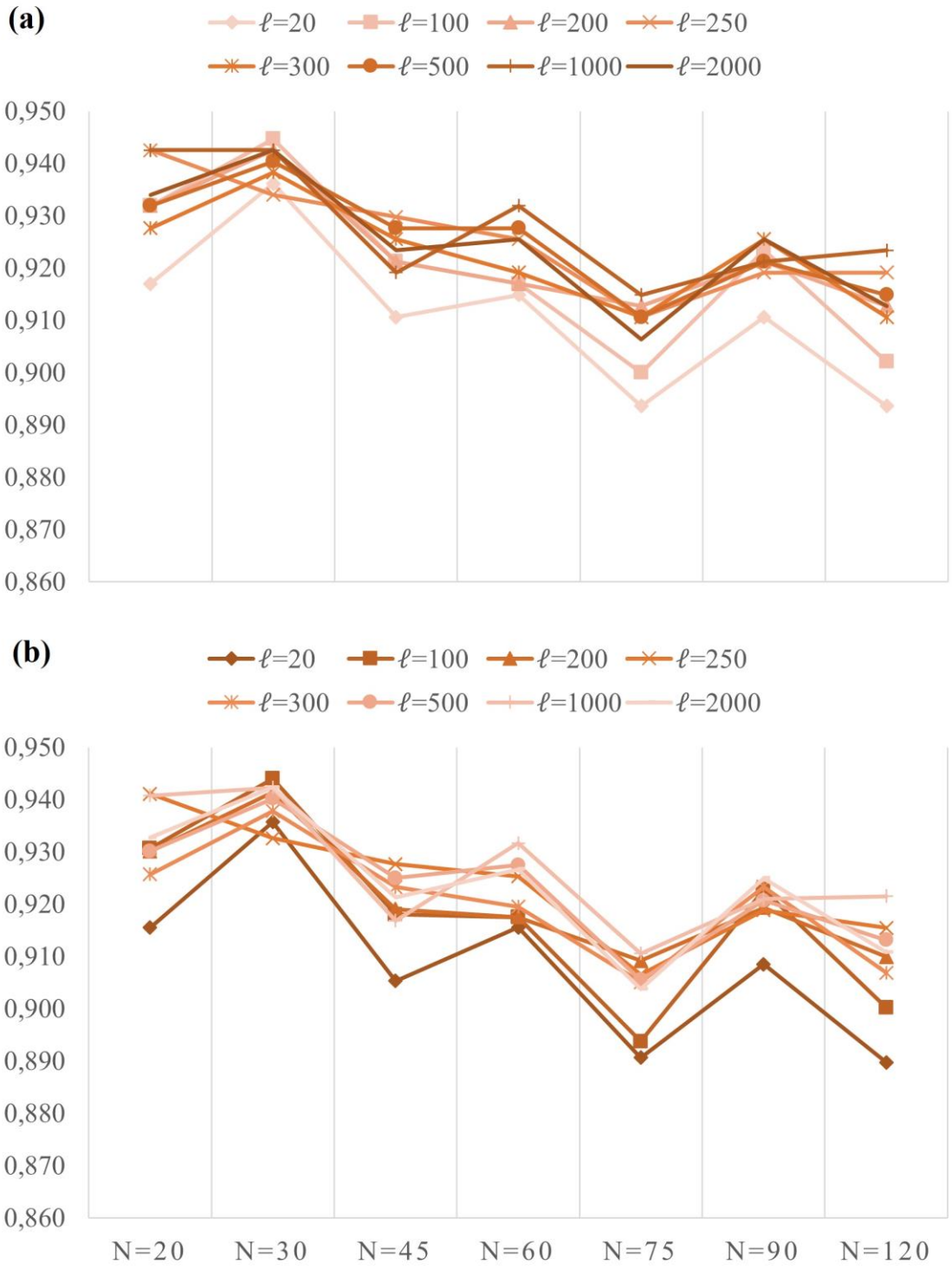

Figure 7.3 5-fold average results of validation sets (a) Accuracy (b) F1 score

7.4 Test results

To show the proposed network's performance, trained models should be tested on never-seen videos in an end-to-end fashion.

These trained classifier models are generated for $7 = |\boldsymbol{N}|$ different window sizes, $8 = |\boldsymbol{L}|$ different layer depths, and for each (N, ℓ) pair, a 5-fold training is performed. That gives $7 \times 8 \times 5 = 280$ models to be trained and tested. After generating and training 280 classifier models for each (N, ℓ, k) triplet, the models are tested against a never-seen night-time fire video data set. A selection of snapshots of videos for early detection results on the test set is given at Figure 7.4. Video versions can also be watched on [72]. At the 6th to 12th seconds in [72], reflection of light from an object, diffused street and headlights, and firefighters with their spatial and temporal behavior were influential in mispredictions.

The test dataset used for each model is different from that of the training and validation sets because of the random selection, as illustrated in Figure 7.2. The randomly selected test set used in each fold contains 194 video clips. Each 280 trained classifier model is tested with 194 respective videos, which makes $280 \times 194 = 54320$ predictions.

As mentioned in Section 7.2, we employed accuracy and F1 measure as the performance metrics. In the case of unbalanced data, i.e., the number of positive and negative samples are not equal, F1 score is a robust indicator for network performance. Given that true positive (TP) is "predicted positive is also actual positive", true negative (TN) is "predicted negative is also actual negative", false positive (FP) is "predicted positive is in fact actual negative", and false negative (FN) is "predicted negative is in fact actual positive"; accuracy and F1 score are defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.2}$$

$$\mathrm{F1} = 2 * \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{7.3}$$

where precision and recall are defined as

$$Precision = \frac{TP}{TP + FP} \tag{7.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{7.5}$$

From the equations 7.2 and 7.3, it is evident that both accuracy and F1 score can get values between [0,1]. The better the accuracy and F1 score is the better performance.

Predictions of the models are obtained by giving the videos as an input to the overall network (Figure 7.2). The average test accuracy and F1 score for each scenario, s_i, are given in Table 7.3 and 7.4. Their plots are given in Figure 7.5. The highest average values of test accuracy and F1 score came out as both 94.7% with 0.0132 and 0.0134 standard deviations, respectively. The (min, max) standard deviation values of Table 7.3 and 7.4 are (0.0039, 0.0263) and (0.0032, 0.0262), respectively. The maximum observed accuracy among the 5-fold attained to 96.9%. The average values of test accuracies and F1 scores have almost been the same as validation measurements. This indicates that there was not an over or underfitting problem with the tests.

Table 7.3 5-fold averages of the test accuracies of the proposed model for various window sizes and layer depths. The highest score is in boldface

		Window Size						
		20	**30**	**45**	**60**	**75**	**90**	**120**
Layer Depth	**20**	92.7	92.2	91.6	92.2	92	90.3	90.1
	100	93.2	93.5	93.3	93.3	92.6	92	91.5
	200	93.8	93	93.4	94.1	93	91.4	92.5
	250	94.5	93.5	93.2	93.6	93	91.4	92.9
	300	93.8	93.4	93.9	93.5	93	92.2	93.1
	500	**94.7**	93.4	93.8	93.6	92.7	92.6	93.1
	1000	94.6	93.7	94.4	93.8	93.1	92.8	92.9
	2000	94.6	93.1	93.4	93.4	92.3	92.5	92.6

Table 7.4 5-fold averages of the test F1 scores of the proposed model for various window sizes and layer depths. The highest score is in boldface

		Window Size						
		20	**30**	**45**	**60**	**75**	**90**	**120**
Layer Depth	**20**	92.7	92.2	91.5	92.1	92	90.3	89.9
	100	93.1	93.5	93.2	93.3	92.5	91.8	91.5
	200	93.8	93	93.4	94.1	93	91.4	92.4
	250	94.5	93.5	93.2	93.6	93	91.4	92.8
	300	93.8	93.4	93.9	93.5	93	92.1	93
	500	**94.7**	93.4	93.8	93.7	92.7	92.6	93.1
	1000	94.6	93.7	94.4	93.9	93.2	92.8	92.8
	2000	94.6	93.1	93.4	93.5	92.4	92.6	92.6

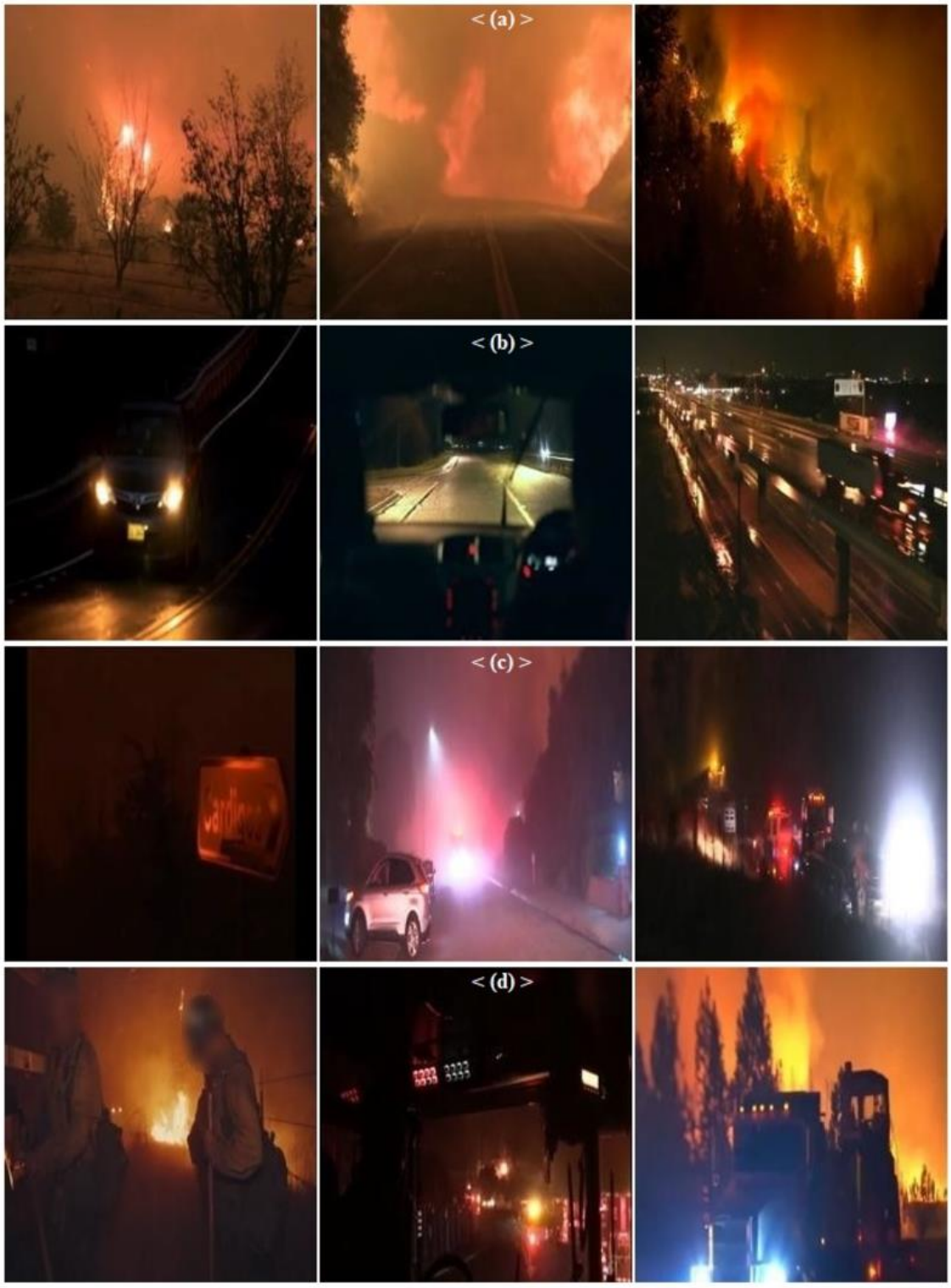

Figure 7.4 Selected images from early test results. In (a) and (b), predictions are correctly fire, and non-fire, respectively. In (c) and (d), predictions are incorrectly fire, and non-fire, respectively. (Faces are blurred in response to privacy concerns.)

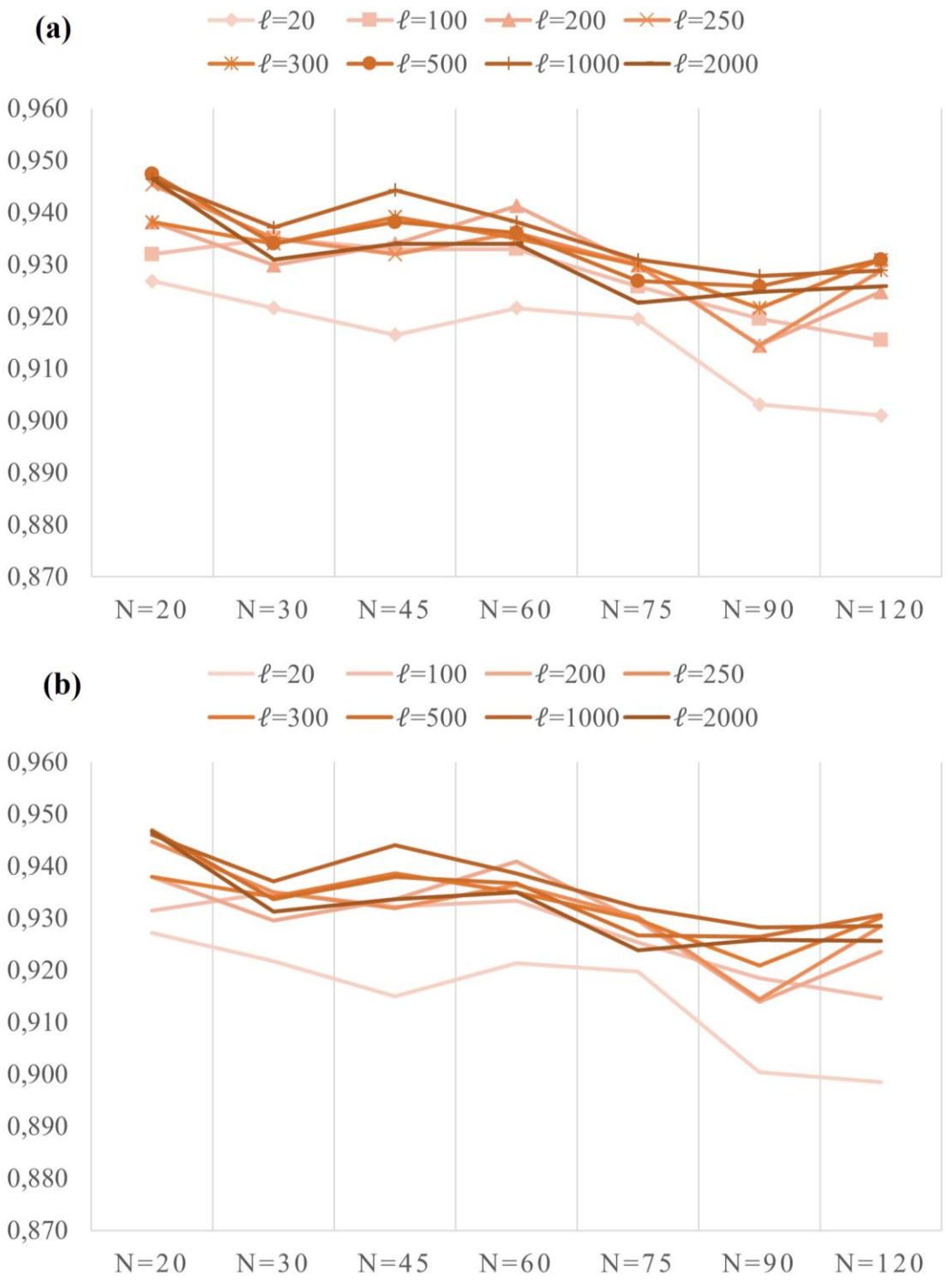

Figure 7.5 5-fold average results of test sets given in the Tables 3 and 4. (a) Accuracy (b) F1 score

The worst accuracy appears when $(N, \ell) = (120,20)$. This is followed by the $(90,20)$ pair. (See the Table 7.3). The table shows an inverse correlation between the window size N and performance measures, accuracy, and F1 scores, for every layer depth

ℓ with few exceptions such as $(30, 1000)$ pair. However, the performance measurements peak when ℓ is around the 300-500 range. The optimal window size and layer depth parameter ranges are $N = 20$ and $\ell = [300 - 500]$.

We observe that shallow networks and long windows do not lead to improved results from the overall results. Indeed, when the BLSTM network has 500 stacked cells fed with 20 frame-long videos, the accuracy rises to the highest. Furthermore, 250 stacked cells give as good accuracy as 500 cells. A lower window size means a lower detection time. Fortunately, the proposed method gives the highest accuracy in the smallest window size, 20. Our method reached a detection duration of 23.4 ms per frame for $(N, \ell) = (20,250)$ and for the best accuracy, i.e., $(N, \ell) = (20,500)$, the detection duration is 23.7 ms per frame. This shows that, contrary to [34], a window size of 20 frames, i.e., two third of a second, would only contribute a delay less than a second. This detection performance can sufficiently be considered as real-time detection.

For a typical video with 30 frames per second (fps) recording, 20 frames would take less than a second. Since the detection time is a significant concern for the first responders in the field, the proposed method significantly contributes.

In summary, a pre-trained CNN was used to extract feature maps which in turn were used to train a BLSTM network. Finally, trained CNN and BLSTM models were interconnected and given never-seen videos to show the pipeline's real-life performance. The CNN was trained on ImageNet which does not include the fire object as a class; instead, it includes a few wildfires event-related classes which are 'fire engine and fire truck'. We tested this pre-trained GoogLeNet on our fire images and as one expects, observed zero percent accuracy.

Obtained results above were used to design, train, and test an improved network. The improved network used a ground-up trained CNN, tCNN, which was trained on our novel dataset rather than a pre-trained CNN (Step 2 of Stage 1 in Figure 7.6. Disjoint video sets for training, validation, and test were determined. In training and validation sets, one frame from each video was randomly sampled to construct an image dataset for CNN training (Step 1 of Stage 1 in Figurer 7.6). After training the CNN, feature maps were extracted through the tCNN and they were used to re-train the BLSTM network which led to a trained BLSTM model, tBLSTM (Steps 5-9 of Stage 3). The best performing window size and layer depths from the model pipeline proposed at Section 7.2 were $N = 20$ and $\ell = [300 - 500]$, thus these parameters were chosen to be $N = 20$ and $\ell = \{250, 500, 1000\}$ for the improved model.

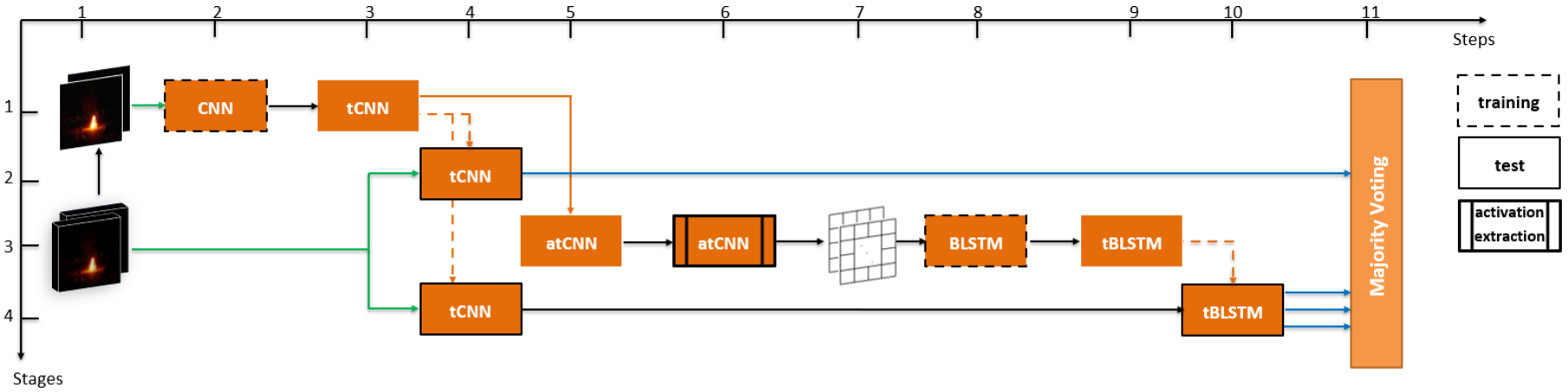

Figure 7.6 Training CNN from ground-up and employing majority voting.

Four prediction results were acquired from the improved network. First result was obtained from tCNN at Step 4 of Stage 2 as shown in Figure 7.6. All frames of a test video were given to the model sequentially to see full performance of the model on the whole video. Therefore, the tCNN model gave 20 predictions per video that overall performance was computed by averaging the number of correct predictions (true positives and true negatives) over the number of all predictions. Second, third, and fourth results were obtained from tCNN + tBLSTM pipeline at Steps 4 and 10 of Stage 4 in Figure 7.6. Video frames were given as input to the pipeline and a single decision was obtained at the output per each layer depth tested. These four results, finally, were given as inputs to a majority voting module (Step 11) to improve the effect of spatial features on decision making. This led to the best average accuracy increase from 94.7 % to 95.15 %. If only the pre-trained CNN were to be used as a classifier, zero accuracy is obtained. If only the tCNN were to be used as a classifier, 94.33 % accuracy is obtained. When the tCNN is coupled with the BLSTM and majority voting, the accuracy increases to 95.15 %. All tests were performed in a five-fold manner as described before.

7.5 Investigating the misclassifications

To further improve the detection performance of the night-time forest fire detection algorithms, sources of misclassifications ought to be investigated. This gives an essential insight into the false classifications and might pave the way for more targeted methods.

We investigated the most frequently misclassified videos in the hope of revealing patterns that lead to wrong predictions. In that regard, the 34 most misclassified test videos, which comprise 44% of the total misclassifications, are considered sufficient for that purpose. We applied masks to these videos while preserving the original video size to see any performance improvement and possible deceptive patterns. The masks are simple black regions that cover either fire or non-fire objects depending on their shape in the scene. When an object is masked by a black region with an arbitrary shape, then that object cannot be considered in the decision process during testing a classifier model.

Adobe Premiere Pro® was used to create such dynamic-shape masks and the masked videos were used for re-testing. The tests were repeated with the respective classifier that gave an error initially on the original non-masked video (As an example, see Figure 7.7).

The number of masks applied to a misclassified video can be different depending on the content of the video (Figure 7.7). Therefore, multiple test videos might be generated from a single video. In total, 76 masked videos were generated from 34 most erroneous videos. In [73] an illustration of how a mask was applied to a misclassified video is given. In the video, only two masks were used, one for fire objects and another for suspected non-fire objects as a whole.

When the masks are applied to fire videos to hide the fire portion, since no fire object is visible in it anymore, its ground truth class is converted from fire to non-fire, respectively. With the updated ground truth labels, re-tests were conducted on masked videos. This resulted in a 50.9% improvement in correctly classifying videos by using the original respective classifier models. This shows that multiple objects in a scene confuse the decision mechanism of the network. If the objects are shown to the models individually, the detection accuracy potentially changes.

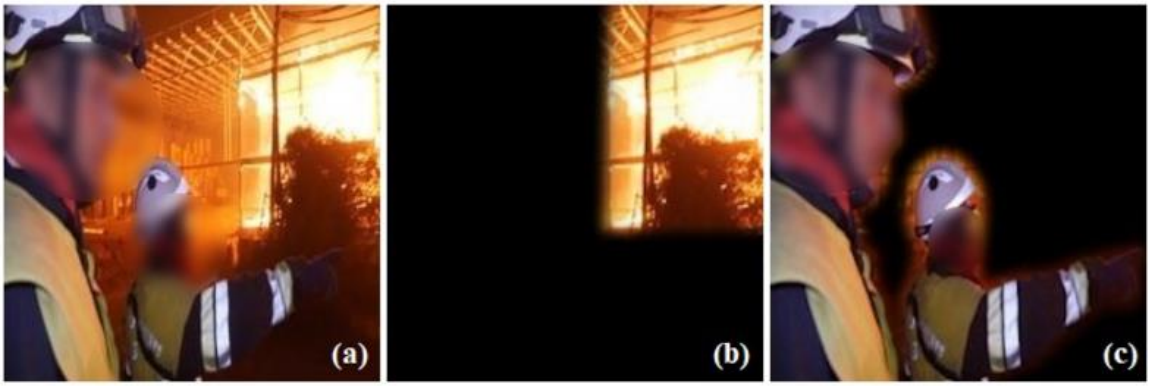

Figure 7.7 A misclassification of a fire scene as non-fire due to firefighters [73]. An originally misclassified video in (a), two firefighters with yellow jackets are blacked out with mask M1 in (b), and all fire objects are blacked out with mask M2 in (c). The video in (b) correctly classified as fire 12/30 times, and the video classification in (c) remained as non-fire for 29/30 times. (Faces are blurred in response to privacy concerns.)

[73] is an example of a missed detection (Figure 7.7). When only a fire object is left, and remaining firefighter figures are masked out, as at the 1st second in [73] then the original respective classifier model makes the new decision correctly as 'fire.' When only firefighters are left in the video, as at the 3rd second in [73] the original model decides 'non-fire' incorrectly 'non-fire' for the original video given in [73].

The presence of people in firefighter-like jackets (i.e., yellow jackets, Figure 7.8) also leads to missed detection as some examples are shown in [74]. In the dataset, firefighters and reporters are frequently seen in these jackets in non-fire videos and less

frequently in fire videos. It indicates that the network learns 'yellow jackets are more related to non-fire class than fire class.' However, the network is expected to learn fire objects as fire, as well. Therefore, when 'fire object' and 'yellow jacket' are in the same scene, we observed that the fire object is in a weak appearance, not flickering, fully or partially occluded by other objects or scene borders. However, when firefighters' jacket is not visible as yellow but mostly dark/shady as at the 2nd second or with a visible yellow jacket with a highly flickering fire object in the scene as at the 3rd second in [74] the video is successfully classified as fire.

Figure 7.8 Yellow jacket is considered evidence of non-fire class due to its frequent presence in the non-fire dataset [74]. (a) and (b) are misclassified as non-fire due to competition between fire and non-fire (yellow jackets) objects. (c) is correctly classified as fire since the yellow jacket is not perceptible. (d) is correctly classified as fire since there is a highly flickering fire object at the back of the reporter's right arm and shoulder. (Faces are blurred in response to privacy concerns.)

Similarly, video in [75] was misclassified as `non-fire even though it is a `fire' video (Figure 7.9). At the 1st second in [75]mask M1 was applied to the video, and vehicle headlights were blacked out; the ground truth label is still `fire.' The same respective

model could predict this video as `fire' this time correctly. At the 4th second in [75], deceptive objects (headlights) were left only, and fire objects were blacked out; now, the ground truth label is `non-fire.' Then the same respective model correctly classified it as `non-fire.' However, this is not strong evidence that "the model predicted `non-fire' correctly since it was already giving the same prediction. In other words, it is a possibility that the model is replicating its incorrect decision meanwhile the ground truth label was changed. This decision pattern implies that a negative object in the scene can manipulate decision-making when a model is trained with scenes containing both negative and positive objects.

Figure 7.9 A misclassification of a fire scene as non-fire due to headlights [75]. An originally misclassified video in (a). When vehicles with headlights are blacked out with mask M1 in (b), the video is correctly classified as fire. When all fire objects are blacked out with mask M2 in (c), then the decision is non-fire.

The video shown in [76] is recorded in a fire event, although there is no fire object in the scene (Figure 7.10). The trained model classified it as `fire' due to vehicle headlights as numbered at the 1st second in [76] Headlights 1, 2, and 3 are flashing while 4 is not. When flashing headlight 3 is masked out from the scene with mask M1 as shown in [76] the prediction is still incorrectly `fire.' However, when all flashing lights are masked out with another mask, M2, as at the 2nd second in [76] then the model correctly predicts the scene as `non-fire.' Similarly, in [77] an artificially flickering electric light was predicted as `fire' (Figure 7.11). When the center and reflected halo environments were masked out separately as given at the 1st and 2nd seconds in [77], respectively, the model could predict them correctly as `non-fire. These two pieces of evidence strongly indicate that the flickering effect of non-fire light sources is a potential deceptive pattern for RNNs

Figure 7.10 A misclassification of a non-fire scene as fire due to flashing headlights [76]. An originally misclassified video in (a) with headlights 1, 2, and 3 flashing while 4 non-flashing. Flashing headlight 3 is blacked out with mask M1 in (b); however, the prediction is still fire. Vehicles with all flashing headlights are masked out with mask M2, and non-flashing headlight 4 of another vehicle is preserved as it is in (c). Now, the prediction changed from fire to correctly non-fire.

and possibly for other temporal analysis algorithms. This finding is also parallel with the reports found in the literature [21].

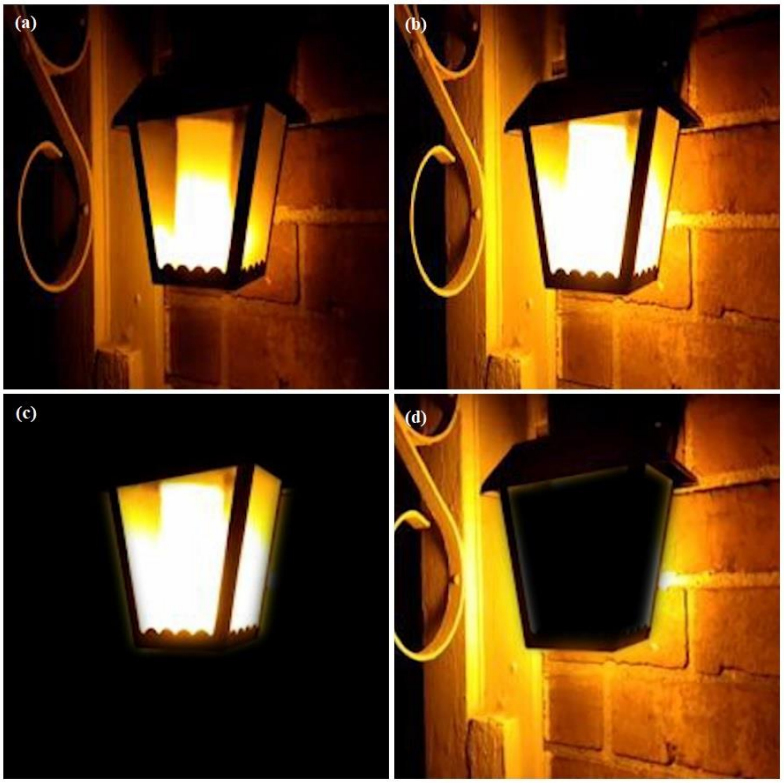

Figure 7.11 A misclassification of a fire scene as non-fire due to a flickering electric light [77]. An originally misclassified video in (a) and (b) includes a flickering electric light. The light source environment is blacked out with mask M1 in (c), and the light source center is blacked out with mask M2 in (d). Videos in (c) and (d) are correctly predicted as non-fire while original video in (a) and (b) not.

Flickering frequency of light at high or low rates have the potential to puzzle predictions (Figure 7.12). In short-range, a fire object's flickering characteristic is apparent. If the fire object is large and strong enough in mid-range, its flickering characteristic is still evident. In long ranges, flickering characteristics become less critical in fire motion characterization. However, if the flickering rate is very high in short or mid ranges, the algorithm cannot relate its motion to average fire motion. High-rate flickering occurs when a burning element contains agents with high flame propagation speed or during strong winds. Low-rate flickering occurs in matured fires that remain primarily in ember form or videos recorded/edited in slow motion [78].

Occlusion is another problem as it is for other object detection tasks, too. If other objects obstruct the fire object, this reduces the chances of correct prediction. When the indefinite form of a fire object is visible, its contour helps the algorithm in the prediction.

Figure 7.12 The flicker rate of fire dramatically changes depending on the fire's combustible agent, wind, and stage [78]). a) A fire object with a high flickering rate. b) A very similar video, but with a steady flickering rate. Only this video was correctly classified among the four. c) A fire object with a very high flickering rate due to explosion. d) A fire object in slow motion video.

In some cases, fire objects are obstructed in certain ways by other objects. The contour of an occluding object becomes the contour of a visible fire object. In the first two seconds of [78] it is seen that fire appears behind a window (Figure 7.12), and at the 1st and 3rd seconds in [79], fire is occluded by solid objects (Figure 7.13). Thus, the typical form of fire disappears, and it creates a fixed-contour object with fire color characteristics. It is conjectured that the algorithm incorrectly learned that fire could have a rigid shape. Therefore, this results in false positives.

Figure 7.13 Due to thick smoke, the turbulent nature and contour of the flame is diminished [79].

When the wildfire size is immense and discharges a large amount of smoke into the environment, the light rays in the environment diffuse into smoke or fog and create a halo effect around the source (Figure 7.13). Illuminated smoke leads to a diminished visible flame contour, and the nature of the fire seems smoother than it should be. In the 1st and 3rd seconds in [79] the videos were misclassified as non-fire. The fire objects in the videos have reduced visibility of flickering and smooth behavior due to dense smoke. Considering the network is trained on videos containing other light sources in dense

smoke, fire objects are indistinguishable from other light sources in such foggy environments. On the other hand, a quite similar video at the 2nd second in [79] includes fire objects with evident flickering. Therefore, this video was correctly classified as fire. Since fire and other light sources become less distinguishable in dense smoke environments, it prevents the network from learning fire motion. In this case, other sub-events in the scene become more important in decision making. At the 3rd second in [79], the presence of a fire-fighter in a yellow jacket has been crucial in classifying the video as non-fire.

Sometimes, objects that are not a natural light source can also resemble fire objects and mislead the network (Figure 7.14). Red fire extinguisher substance discharged from a fire-tank aircraft tricked the network 98 times in our experiments [80] During the substances' landing on the ground and plant area, its spread and motion resemble a fire object, and the network classifies it as fire.

Figure 7.14 An aircraft fire-tanker is discharging red fire extinguisher liquid [80].

In this section, we have investigated the sources of misclassifications. The tests are repeated with various modified videos to unmask the misleading parts in the contents and vulnerabilities of the networks. These additional tests showed that:

- Multiple light sources: The decision accuracy increases if the scene has a single bright object. When both positive and negative objects appear in the scene, the model's prediction accuracy decreases.
- Flickering: The fire has a distinct flickering behavior. The flickering frequency diminishes as the fire gets further away. This makes it hard for the network to learn a single flickering frequency. Moreover, some other light sources, e.g.,

lanterns, also have a flickering nature. This is one of the unsolved challenges of these tasks.

- Fog: A visible flame has sharp and quickly altering edges. When fog is introduced, these details, which are valuable indicators for the model, vanish.
- Occlusion: When an object occludes the fire, a halo appears around the blocking object. This halo looks quite similar to a flame in terms of color. However, it stays still and has a somewhat different shape than fire. During the training, this might force the model to learn these features incorrectly.
- Strobe lights: As reported, a common misclassification source is the strobe lights attached to the vehicles in the field [29, 30]. Those lights cause problems in two respects: first, they have similar periodicity features as fires. Second, they both appear in non-fire scenes, e.g., typically in traffic and in forest fires. Those two features make the predictions less accurate if they appear in the scene.

The night-fire classification problem's challenges can be countered by designing targeted approaches such as preprocessing filters or cascaded models. For example, in an experiment, the authors first detect and mask the possible strobe lights before the prediction [29, 30]. These experiments intended to show and identify the challenges that exist in the night-time fire videos.

7.6 Concluding Remarks

Night-time forest fire videos lack some important spatial information such as color, sharp edges, etc., due to the physical limitations of the camera. For some cases, distinguishing a night fire from artificial light from a single frame is highly challenging even for human-level classification. This makes the night-time fire classification harder than its daytime counterpart. To alleviate this, temporal information is incorporated in the proposed method. Thus, night fire's natural flickering and motion behavior could be captured and involved in the analysis.

In this study, GoogLeNet + BLSTM based network architecture was used to analyze the spatio-temporal information of fire object candidates and detect fire events in night-time videos. The tests were performed for a wide range of parameter sets. The video lengths used in training and tests ranged from 20 to 120 frames. The BLSTM network depths ranged from shallow 20 layers to deep 2000 layers. It is shown in the pre-trained

experiment that the shortest videos, with 20 frames and 500 BLSTM layer-deep-network gave the best parameter combinations with 94.7% accuracy and F1 score at 23.7 ms per frame. Even though the longer videos have more information and deeper networks have more adaptation capacity, they did not have the best parameters. These results were used to tune a majority voting module and the highest accuracy of 95.15% was reached.

For a typical 30 FPS video, the proposed algorithm requires less than a second to accumulate 20 frames and detect the night fire event. Since the response speed of first responders in the field is crucial, this method makes a significant contribution by reducing the response time.

The study also contains a thorough investigation and discussion of possible sources of misclassifications in night-time wildfire detection tasks. Multiple light sources, the flickering of artificial lights, strobe lights, fog, smoke, and occlusion are the primary sources of incorrect predictions.

Several problems remain unsolved. It is conjectured that by designing targeted solutions such as image preprocessing or cascaded decisions, the effects of the aforementioned false classification sources can be mitigated. Moreover, distant fires look significantly different than close ones. Instead of a unified algorithm, multiple algorithms targeting these situations separately can be designed as shown in the majority voting module.

Chapter 8

Conclusions and Future Prospects

8.1 Conclusions

The thesis showed that distinct temporal behavior of glowing objects in a dark video can reveal the identity or the class of them even if there exists little or no visually perceptible textures in the scene. With this, a hand-crafted feature set and an end-to-end deep learning method is used to benefit from the temporal behavior of the objects. Moreover, the largest night fire video dataset to the date is prepared and curated. The details of the individual conclusions drawn from each chapter is given below.

In Chapter 1, impact of fire disasters to our environment is illustrated and importance of timely and automatic fire detection is emphasized. In Chapter 2, it was made clear that how taking images at night requires a careful setup of environment, selection of image taking device and adjusting setting of that device before capturing such images or videos. In Chapter 3, a portrait of nature of nighttime fires and challenges of capturing them via imaging means are given. In Chapters 4, 5, and 6, the theoretical background of the methods used in this dissertation is given. More precisely, working principles of support vector machines, convolutional neural networks, and bidirectional long-term short-term memory networks are presented then related literature is summarized.

In Chapter 7, a novel dataset, FinD, is proposed for nighttime VFD research. This dataset includes two sets. FinD Set1 is developed for hand crafted features and the background is almost black except limited deceptive patterns like streetlights, headlights, hand-held lights, etc. The smoke is not visible in the videos of this set. Subtracting the background is relatively easy for FinD Set1 since the recording cameras were stationery. FinD Set2 is compiled from various videos that were recorded during fire disasters

occurred across the globe. These videos contain challenging scenes including smoke's adverse effects rather than its favorable contribution to daytime fires. These videos mostly recorded from mobile cameras of such as mobile phones, UAVs, helicopters, patrolling vehicles, firetrucks, etc. Negative videos are also collected from fire environments without flame object, or from fire-like videos of nonfire events, i.e., fireworks, volcanos, etc.

In Chapter 8, useful features for nighttime fires are developed and used for machine learning model development with SVMs, RFs, and other algorithms. These features allowed successfully detect fires with 95.53% average accuracy. These features are based on movement of flame object. Therefore, some deceptive objects like exact reflection of fire from a reflective surface, slowly moving vehicle headlights are also considered as fire. In Chapter 10, deep learning-based nightfire detection method is proposed. In a deep learning model, features are not designed by hand anymore, the network discovers useful features by itself and optimizes their weights. In this method, a pipeline of CNN and BLSTM is used. The purpose of using this model is employing both spatial and temporal features of a video for robust fire detection at night videos. Spatial features are extracted via CNN model and used to train BLSTM model. First training both CNN and BLSTM and then connecting them constructed the final classifier model which receives videos at the input and gives predictions at the output. This model attained 95.15% accuracy at 2/3 seconds videos. However, this model is also prone to mispredictions. The model is trained on the FinD Set2 which labels the data video by video as "fire" or "not fire". Then, if a fire video includes the flame object together with i.e., firefighter, then it is observed that on a not fire video including a firefighter could also be predicted as fire.

8.2 Societal Impact and Contribution to Global Sustainability

The proposed research relates to one of the United Nation's 17 Sustainable Development Goals, Goal 15 which states "Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss". Fires are considered part of natural life cycle in terms of eliminating unhealthy forests, stimulating tree growth. However, uncontrolled, frequent, and large fires (collectively unnatural fires) result in severe

ecological, societal, economical, and sustainability problems. Forest fire disasters aggravate greatly the broken carbon cycle, air pollution, natural habitat degradation, respiratory and other health problems. Each uncontrolled wildfire destroys people's houses, habitat, biotope, and destabilizes natural carbon inventory. Therefore, wildfires are at greatest importance in societal impact and global sustainability.

The forest fire detection from video methods developed in this dissertation are expected contribute actual firefighting efforts at early detection and alerting steps. Department of Forestry has automatic fire detection systems running on lookout towers scattered on the forests. However, they are designed to detect only the daytime fires. With the help of the proposed methods, the developed models can be integrated to Department of Forestry's live camera systems and make detection for both daytime and nighttime forest fires. This will let the detection task will extend to 24/7. Furthermore, since the environment of each forest field may demonstrate different visual appearance, the models can be adapted to these environments to make improved predictions and reduce false alarm rates. For this purpose, the models can be iteratively re-trained with continuously flowing video data to realize highest adoption level.

Reducing the number of forest fires by early detection methods has a direct impact on the biodiversity of the natural environments. The forest fires are known to be one of the most disastrous incidents which eliminates large number of living organisms from a wide range of species. When the range of the regions destroyed exceeds a reasonable number of areas, the negative effect of it on the biodiversity might become irreversible. Eventually this chain of incidents might yield desertification of natural and fertile regions in a country.

Moreover, the forests are one of the most valuable assets for countries. Every year dozens of hectares of areas are lost to these disasters. The proposed approaches are believed to lessen this effect by assisting the officials to detect the forest fires earlier.

8.3 Future Prospects

The thesis showed that distinct temporal behavior of glowing objects in a dark video can reveal the identity or the class of them even if there exists little or no visually perceptible textures in the scene. In the thesis, a hand-crafted feature set and an end-to-end deep learning method is used to benefit from the temporal behavior of the objects. However, there are a lot more aspects of it to be studied. The current model does not take

the inter-correlation of the objects in the scene. In future, a Transformer based model can be employed to not only capture the spatio-temporal movements of the objects but also their inter-relations.

The methods proposed can be further optimized for real-time applications. First, the CNN+BLSTM portions can be combined to a single stage similar to single stage segmentation methods such as MaskRCNN. Then the model can be further quantized to small integer format (8-bit integer) to give smaller model in size. This will help the users to install the model into embedded devices such as Raspberry Pi or ARM based microcontrollers.

Apart from that, the proposed dataset is expected to be highly useful to the community. It is hoped that the thoroughly prepared and labeled night fire video dataset pave the way for an academic competition to reach higher or faster detection accuracies.

BIBLIOGRAPHY

[1] J. Janai, F. Güney, A. Behl and A. Geiger, "Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art," *Foundations and Trends® in Computer Graphics and Vision,* vol. 12, no. 1-3, p. 1–308, 2020.

[2] K.-T. Tan and P. Lim, "The artificial intelligence renaissance: deeplearning and the road to human-Level machine intelligence," *Industrial Technology Advances,* vol. 7, 2018.

[3] DGF, "Forestry Statistics (1988-2019)," General Directorate of Forestry of Turkey, Ankara, 2019.

[4] NIFC, "Wildland Fires and Acres (1926-2019)," The National Interagency Fire Center, Boise, 2020.

[5] DGF, "State of Forest Fires," Directorate General of Forestry, [Online]. Available: https://www.ogm.gov.tr/Lists/GuncelOrmanYanginlari. [Accessed 29 January 2017].

[6] IFD, "2011-2016 Statistics," İstanbul Municipality, İstanbul, 2017.

[7] S. Harris, A. Wendy, K. Musa and F. Liam, "The relationship between fire behaviour measures and community," *Natural Hazards,* vol. 63, pp. 391-415, 2012.

[8] P. Chenney and A. Sullivan, Grassfires: Fuel, weather and fire behaviour, Collingwood, Australia: Csiro Publishing, 2008.

[9] D. Drysdale, An Introduction to Fire Dynamics, West Sussex: John Wiley & Sons Ltd, 2011.

[10] T. Toulouse, L. Rossi, A. Campana, T. Çelik and M. A. Akhloufi, "Computer vision for wildfire research: An evolving image dataset for processing and analysis," *Fire Safety Journal,* vol. 92, pp. 188-194, 2017.

[11] C. M. Bishop, Pattern Recognition and Machine Learning, New York: Springer Science+Business Media, LLC, 2006.

[12] H. Chih-Wei, C. Chih-Chung and L. Chih-Jen, *A Practical Guide to Support Vector Classification,* Taipei: Department of Computer Science, National Taiwan University, 2016.

[13] P. Kedia. [Online]. Available: https://medium.com/mlearning-ai/what-is-artificial-neural-network-fd38dd2ec121. [Accessed 21 June 2022].

[14] H. Sak and A. B. F. Senior, *Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition,* Arxiv, 2014.

[15] "Alberta Agriculture and Forestry," The Government of Alberta, 7 January 2022. [Online]. Available: https://wildfire.alberta.ca/recruitment/lookout-observer.aspx. [Accessed 07 June 2022].

[16] D. B. M. B. L. Stipanicev, "Integration of Forest Fire Video Monitoring System and Georgraphic Information System," in *51th International Symposium ELMAR-2009,* Zadar, 2009.

[17] B. U. Toreyin and A. E. Cetin, "Computer vision based forest fire detection," in *2008 IEEE 16th Signal Processing, Communication and Applications Conference*, Aydin, 2008.

[18] K. Dimitropoulos, K. Köse, N. Grammalidis and E. Cetin, "Fire Detection and 3D Fire Propagation Estimation for the Protection Of Cultural Heritage Area," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. XXXVIII, 2010.

[19] O. Günay, B. U. Töreyin and A. E. Çetin, "Online adaptive decision fusion framework based on projections onto convex sets with application to wildfire detection in video," *Optical Engineering,* vol. 50, no. 7, pp. 1-13, 2011.

[20] Y. H. Habiboglu, O. Günay and Ç. A. Enis, "Real-time wildfire detection using correlation descriptors," in *19th European Signal Processing Conference (EUSIPCO 2011),* Barcelona, 2011.

[21] O. Gunay, B. U. Toreyin, K. Kose and A. E. Çetin, "Entropy-Functional-Based Online Adaptive Decision Fusion Framework With Application to Wildfire Detection in Video," *IEEE Transactions on Image Processing,* vol. 21, no. 5, pp. 2853-2865, 2012.

[22] Y. H. Habiboğlu, O. Günay and A. E. Çetin, "Covariance matrix-based fire and flame detection method in video," *Machine Vision and Applications,* vol. 23, p. 1103–1113, 2012.

[23] F. Erden, T. B. Ugur, E. B. Soyer, I. Inac, O. Günay, K. Köse and A. E. Çetin, "Wavelet based flickering flame detector using differential PIR sensors," *Fire Safety Journal,* vol. 53, pp. 13-18, 2012.

[24] S. G. Kong, D. Jin, S. Li and H. Kim, "Fast fire flame detection in surveillance video using logistic regression and temporal smoothing," *Fire Safety Journal,* vol. 79, pp. 37-43, 2016.

[25] S. Verstockt, S. V. Hoecke, T. Beji, B. Merci, B. Gouverneur, A. E. Çetin, P. De Potter and R. V. de Walle, "A multi-modal video analysis approach for car park fire detection," *Fire Safety Journal,* vol. 57, pp. 44-57, 2013.

[26] K. Dimitropoulos, O. Gunay, K. Kose, F. Erden, F. Chaabene, F. Tsalakanidou, N. Grammalidis and E. Cetin, "Flame Detection for Video-Based Early Fire Warning for the Protection of Cultural Heritage," in *Progress in Cultural Heritage Preservation. EuroMed 2012,* Berlin, 2012.

[27] T. Toulouse, L. Rossi, T. Çelik and M. Akhloufi, "Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods," *Signal, Image and Video Processing,* vol. 10, pp. 647-654, 2016.

[28] K. Dimitropoulos, P. Barmpoutis and N. Grammalidis, "Spatio-Temporal Flame Modeling and Dynamic Texture Analysis for Automatic Video-Based Fire Detection," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 25, pp. 339-351, 2015.

[29] K. Tasdemir, O. Gunay, B. U. Toreyin and A. E. Cetin, "Video based fire detection at night," in *2009 IEEE 17th Signal Processing and Communications Applications Conference,* Antalya, 2009.

[30] O. Günay, K. Taşdemir, B. U. Töreyin and A. E. Çetin, "Video based wildfire detection at night," *Fire Safety Journal,* vol. 44, pp. 860-868, 2009.

[31] C.-C. Ho and M.-C. Chen, "Nighttime Fire Smoke Detection System Based on Machine Vision," *International Journal of Precision Engineering and Manufacturing,* vol. 13, no. 8, pp. 1369-1376, 2012.

[32] P. Gomes, P. Santana and J. Barata, "A vision-based approach to fire detection," *International Journal of Advanced Robotic Systems,* vol. 11, no. 9, 2014.

[33] A. K. Ağırman and K. Taşdemir, "Short to mid-range night fire detection," in *25th Signal Processing and Communications Applications Conference (SIU),* Antalya, 2017.

[34] M. K. C. B. Park, "Two-step real-time night-time fire detection in an urban environment using Static ELASTIC-YOLOv3 and Temporal Fire-Tube," *Sensors,* vol. 20, no. 8, p. 2202, 2020.

[35] H. Pan, D. Badawi and A. E. Çetin, "Computationally Efficient Wildfire Detection Method Using a Deep Convolutional Network Pruned via Fourier Analysis," *Sensors,* vol. 20, p. 2891, 2020.

[36] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto and E. A. B. da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," *Electronics*, vol. 10, no. 3, p. 279, 2021.

[37] NIFC, "Total wildland fires and acres (1983-2020)," National Interagency Fire Center, 2020. [Online]. Available: https://www.nifc.gov/fire-information/statistics/wildfires. [Accessed 27 June 2021].

[38] A. Voiland, "Building a long-term record of fire," The Earth Observatory, [Online]. Available: https://earthobservatory.nasa.gov/images/145421/building-a-long-term-record-of-fire. [Accessed 27 June 2021].

[39] R. Consortium, "Rescuer: Reliable and smart crowdsourcing solution for emergency and crisis management," EU-Brazil research and development Cooperation, [Online]. Available: http://www.rescuer-project.org/. [Accessed 27 June 2021].

[40] J. Neal, C. Land, R. Avent and R. Churchill, "Application of artificial neural networks to machine vision flame detection," Air Force Engineering and Services Center, Virginia, 1991.

[41] Y. Dedeoglu, B. U. Toreyin, U. Güdükbay and A. E. Çetin, "Real-time fire and flame detection in video," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, Philadelphia, 2005.

[42] B. C. Ko, K.-H. Cheong and J.-Y. Nam, "Fire detection based on vision sensor and support vector machines," *Fire Safety Journal*, vol. 44, pp. 322-329, 2008.

[43] 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, 2015.

[44] R. Vezzani and R. Cucchiara, "ViSOR: Video Surveillance Online Repository," *Annals of the BMVA*, vol. 2010, no. 2, pp. 1-13, 2010.

[45] R. Vezzani and R. Cucchiara, "Video Surveillance Online Repository (ViSOR): an integrated framework," *Multimedia Tools and Applications*, vol. 50, pp. 359-380, 2010.

[46] C. M. T., A. L. P. S., C. D. Y. T., R. J. S., d. S. J. A., R. J. J. F. and T. A. J. M., "FiSmo: A Compilation of Datasets from Emergency Situations for Fire and Smoke Analysis," in *Proceedings of the satellite events*, Uberlândia, 2017.

[47] C. R. Steffens, R. N. Rodrigues and S. S. d. C. Botelho, "Steffens, Cristiano Rafael and Rodrigues, Ricardo Nagel and Silva da Costa Botelho, Silvia," in *2th Latin American Robotics Symposium and 2015 Third Brazilian Symposium on Robotic*, Uberlandia, 2015.

[48] G. Lin, Y. Zhang, Q. Zhang, Y. Jia, G. Xu and J. Wang, "Smoke detection in video sequences based on dynamic texture using volume local binary patterns," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 11, 2017.

[49] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-89, 2017.

[50] A. Malenichev and O. Krasotkina, "Real-Time Smoke Detection in Video Sequences: Combined Approach," in *International Conference on Pattern Recognition and Machine Intelligence*, Heidelberg, 2013.

[51] M. Bugaric, T. Jakovcevic and D. Stipanicev, "Adaptive estimation of visual smoke detection parameters based on spatial data and fire risk index," *Computer Vision and Image Understanding*, vol. 118, pp. 184-196, 2014.

[52] K. G. Derpanis, M. Lecce, K. Daniilidis and R. P. Wildes, "Dynamic Scene Understanding: The Role of Orientation Features in Space and Time in Scene Classification," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, 2012.

[53] L. P. Avalhais, J. Rodrigues and A. J. Traina, "Fire Detection on Unconstrained Videos Using Color-Aware Spatial Modeling and Motion Flow," in *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, San Jose, 2016.

[54] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[55] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines*, Taipei: Department of Computer Science, NTU, 2021.

[56] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboglu, B. U. Töreyin and S. Verstockt, "Video fire detection – Review," *Digital Signal Processing*, vol. 23, p. 1827–1843, 2013.

[57] A. Peris, M. Bolanos, P. Radeva and F. Casacuberta, "Video Description Using Bidirectional Recurrent Neural Networks," in *Artificial Neural Networks and Machine Learning – ICANN 2016*, Barcelona, 2016.

[58] C. Szegedy, L. Wei, J. Yangqing, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, 2015.

[59] O. Maksymiv, T. Rak and O. Menshikova, "Deep convolutional network for detecting probable emergency situations," in *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*, Lviv, 2016.

[60] L. Wonjae, K. Seonghyun, L. Yong-Tae, L. Hyun-Woo and C. Min, "Deep neural networks for wild fire detection with unmanned aerial vehicle," in *2017 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, 2017.

[61] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho and S. W. Baik, "Convolutional Neural Networks Based Fire Detection in Surveillance Videos," *IEEE Access*, vol. 6, pp. 18174-18183, 2018.

[62] A. J. Dunnings and T. P. Breckon, "Experimentally Defined Convolutional Neural Network Architecture Variants for Non-Temporal Real-Time Fire Detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Athens, 2018.

[63] R. P. Sadewa, B. Irawan and C. Setianingsih, "Fire Detection Using Image Processing Techniques with Convolutional Neural Networks," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019.

[64] S. Bianco, R. Cadene, L. Celona and P. Napoletano, "Benchmark Analysis of Representative Deep Neural Network Architectures," *IEEE Access*, vol. 6, pp. 64270-64277, 2018.

[65] M. Lin, Q. Chen and S. Yan, "Network In Network," *Arxiv*, 2014.

[66] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2005.

[67] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[68] Z. Wu, T. Yao, Y. Fu and Y.-G. Jiang, "Deep Learning for Video Classification and Captioning," in *Frontiers of Multimedia Research*, ACM Books, 2017, pp. 3-29.

[69] B. Kim and J. Lee, "A Video-Based Fire Detection Using Deep Learning Models," *Applied Sciences*, vol. 9, no. 14, p. 2862, 2019.

[70] S. Siami-Namini, N. Tavakoli and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, 2019.

[71] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference for Learning Representations*, San Diego, 2015.

[72] A. K. Ağırman, "Some examples for a CNN+BLSTM pipeline predictions," YouTube, 18 February 2022. [Online]. Available: https://www.youtube.com/watch?v=RvZgITUZ1WE. [Accessed 20 February 2022].

[73] A. K. Ağırman, "A misclassification of a fire scene as non-fire due to existence of firefighters in the scene," YouTube, 18 February 2022. [Online]. Available: https://youtu.be/CLVVpLXgp04. [Accessed 20 February 2022].

[74] A. K. Ağırman, "Effect of a yellow jacket on predictions. Misclassified as nonfire.," YouTube, 18 February 2022. [Online]. Available: https://www.youtube.com/watch?v=aYPXNcOYvds. [Accessed 20 February 2022].

[75] A. K. Ağırman, "A misclassification of a fire scene as non-fire due to headlights," YouTube, 18 February 2022. [Online]. Available: https://youtu.be/db-83tnPttw. [Accessed 20 February 2022].

[76] A. K. Ağırman, "A misclassification of a non-fire scene as fire due to flashing headlights," YouTube, 18 February 2022. [Online]. Available: https://www.youtube.com/watch?v=r T6hBPYsDo. [Accessed 20 February 2022].

[77] A. K. Ağırman, "A misclassification of a fire scene as non-fire due to a flickering electric light," YouTube, 18 February 2022. [Online]. Available: https://www.youtube.com/watch?v=uSvOlDr3ZsY. [Accessed 20 February 2022].

[78] A. K. Ağırman, "The flicker rate of fire dramatically changes depending on many factors," YouTube, 18 February 2022. [Online]. Available: https://www.youtube.com/watch?v=fXQbqxAum0w. [Accessed 20 February 2022].

[79] A. K. Ağırman, "Effect of smoke on vision of flame," YouTube, 18 February 2022. [Online]. Available: https://www.youtube.com/watch?v=Ge7ElgG11U0. [Accessed 20 February 2022].

[80] A. K. Ağırman, "Other fire-like objects deceptive predictive systems," YouTube, 18 Fabruary 2022. [Online]. Available: https://youtu.be/kYt_Z29qeZ0. [Accessed 20 February 2022].

[81] D. Wilimitis, "The Kernel Trick in Support Vector Classification," Towards Data Science, 12 December 2018. [Online]. Available: https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f. [Accessed 07 June 2022].

[82] B. U. Toreyin and A. E. Cetin, "Online Detection of Fire in Video," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, 2007.

[83] P. Ferreira, "Thoughts about digital camera sensor in a world of smartphone cameras," [Online]. Available: https://armpauloferreira.blogspot.com/2019/09/thoughts-about-digital-camera-sensor-in.html. [Accessed 21 June 2022].

[84] N. Mansurov, "What is ISO? The Complete Guide for Beginners," [Online]. Available: https://photographylife.com/what-is-iso-in-photography. [Accessed 21 June 2022].

[85] E. Gray, "Understanding Depth of Field – A Beginner's Guide," [Online]. Available: https://photographylife.com/what-is-depth-of-field. [Accessed 21 June 2022].

[86] Adobe. [Online]. Available: https://www.adobe.com/creativecloud/photography/discover/hdr.html. [Accessed 21 June 2022].

[87] E. Kavlakoğlu, "AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?," [Online]. Available: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks. [Accessed 21 June 2022].

[88] Mathworks, "What Makes CNNs So Useful?," [Online]. Available: https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html#how-they-work. [Accessed 21 June 2022].

[89] D. Cornelisse, "An intuitive guide to Convolutional Neural Networks," [Online]. Available: https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/. [Accessed 21 June 2022].

[90] Mathworks, "2-D Convolutional Layer," [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.convolution2dlayer.html. [Accessed 21 June 2022].

CURRICULUM VITAE

2001 – 2006	B.A., Avionics, Anadolu University, Eskişehir, TURKEY
2010 – 2013	M.Eng., Systems Engineering, Boston University, Boston, MA, USA
2014 – 2022	Ph.D., Electrical and Computer Engineering, Abdullah Gül University, Kayseri, Turkey
2006-2009	Safety Auditor, Directorate General of Civil Aviation, Ankara, Turkey
2013-cont	Lecturer, Erciyes University, Kayseri, Turkey

SELECTED PUBLICATIONS AND PRESENTATIONS

J1) A.K. Agirman, K. Tasdemir, BLSTM based night-time wildfire detection from video published in PLOSONE (May 2022).

C1) A.K. Agirman, K. Tasdemir, Short to Mid-Range Night Fire Detection in IEEE 2017 25th Signal Processing and Communications Applications Conference (SIU) (May 2017).