REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

THE ROLE OF SENTINEL-1 AND SENTINEL-2 DATA FOR OBSERVING WILDFIRES

Deniz Gaye DENİZOĞLU

MASTER OF SCIENCE THESIS

Department of Geomatics Engineering

Remote Sensing and GIS Program

Supervisor

Prof. Dr. Fusun BALIK ŞANLI

Co-Supervisor

Prof. Dr. Martin WERNER

October, 2022

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

THE ROLE OF SENTINEL-1 AND SENTINEL-2 DATA FOR OBSERVING WILDFIRES

A thesis submitted by Deniz Gaye DENİZOĞLU in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE is approved by the committee on 17.10.2022 in Department of Geomatics Engineering, Remote Sensing and GIS Program.

Prof. Dr. Fusun BALIK ŞANLI
Yıldız Technical University
Supervisor

Prof. Dr. Martin WERNER
Technical University of Munich
Co- supervisor

Approved By the Examining Committee

Prof. Dr. Fusun BALIK ŞANLI, Supervisor
Yildiz Technical University _____

Prof. Dr. Bülent BAYRAM, Member
Yildiz Technical University _____

Assoc. Prof. Dr. Saygın ABDİKAN, Member
Hacettepe University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of The Role of Sentinel-1 and Sentinel-2 Data for Observing Wildfires supervised by my supervisor, Prof. Dr. Fusun BALIK ŞANLI. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Deniz Gaye DENİZOĞLU

Signature

Dedicated to my mother

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor Prof. Dr. Fusun BALIK ŞANLI and my co-supervisor Prof. Dr. Martin WERNER for their continuous support, invaluable guidance, and feedback. I will forever be thankful to my colleague Gabriel DAX for always being there when I needed support and reviewing my progress. I am grateful to Srilakshmi NAGARAJAN for her unconditional help. I am also thankful to Onur Can BAYRAK for his contributions.

A special thanks to Prof. Dr. Bülent BAYRAM for his sincere guidance and support since my undergraduate. I am also thankful to Assoc. Prof. Dr. Saygın ABDİKAN for his invaluable feedback. Thanks, should extend to Dr. Hao LI and Stephanie TUMAMPOS for their suggestions, feedback, and support.

Lastly, but most importantly I would like to thank my mother for her unconditional support and encouragement.

Deniz Gaye DENİZOĞLU

TABLE OF CONTENTS

LIST OF ABBREVIATIONS vi

LIST OF FIGURES vii

LIST OF TABLES viii

ABSTRACT ix

ÖZET xi

1 INTRODUCTION 1

1.1 Literature Review 1

1.2 Objective of the Thesis 4

1.3 Hypothesis 4

2 STUDY AREA AND DATASET 5

2.1 Study Area 5

2.2 Dataset 6

3 METHODOLOGY 10

3.1 Dataset Generation 10

3.1.1 Image Preparation 10

3.1.2 Labeling 12

3.1.3 Patch Creation 13

3.1.4 Classification 14

3.2 Deep Learning 14

3.2.1. VGG 15

3.2.2 ResNet 50 – ResNet101 – ResNet152 16

3.2.3 Inceptionv3 18

3.2.4 Xception 20

3.2.5 MobileNet 20

3.2.6 DenseNet121-DenseNet169-DenseNet201 22

3.3 Evaluation Metrics 23

3.4 Findings 23

4 RESULTS AND DISCUSSION 33

BIBLIOGRAPHY 34

PUBLICATIONS FROM THE THESIS 42

LIST OF ABBREVIATIONS

Adam	Adaptive Moment Estimation
AFD	Active Fire Detection
ALOS	Advanced Land Observing Satellite
CNN	Convolutional Neural Network
DL	Deep Learning
EO	Earth Observation
EW	Extra Wide Swath
FN	False Negatives
FP	False Positives
GOLI	Global Operational Land Imager
GRD	Ground Range Multi-Look Detected
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IR	Infrared
IW	Interferometric Wide Swath
ML	Machine Learning
MODIS	Moderate Resolution Imaging Spectroradiometer
MSI	Multi-Spectral Instrument
OLI	Operational Land Imager
ReLU	Rectified Linear Unit
SAR	Synthetic Aperture RADAR
SLC	Single Look Complex
SM	Stripmap
SNAP	Sentinel Application Platform
SVM	Support Vector Machines
SWIR	Short-wave Infrared
TN	True Negatives
TP	True Positives
UAV	Unmanned Aerial Vehicles
VIIRS	Visible Infrared Imaging Radiometer Suite
WV	Wave

LIST OF FIGURES

Figure 2.1 Composite of Sentinel-2 bands 12,11, and 8 of the study area 5

Figure 2.2 Representation of the study area where the red polygon marks the Dixie fire.....6

Figure 2.3 RGB (B04,B03,B02) Sentinel-2 image of the study area 6

Figure 2.4 Sentinel-1 constellation revisit and coverage frequency (taken from [52]) 7

Figure 3.1 Visualization of the processing chain to generate and classify the wildfire patches 10

Figure 3.2 Generated Sentinel-2 band combinations to detect wildfire. From left to right: (B04,B03,B02), (B08,B04,B03), (B02,B08,B12), (B12,B08,B02), (B12,B11,B04), (B12,B11,B08) 11

Figure 3.3 Captured wildfire from Sentinel- 2 image (B12,B11,B8) 11

Figure 3.4 Sentinel-2 image with sample labels, where the red bounding box shows active fire, the green represents burnt area, and the gray box includes unaffected area.....12

Figure 3.5 The study area during wildfire captured from Sentinel-1 (VV+VH) (above), and Sentinel-2 (below).....13

Figure 3.6 Patch samples for each class. From left to right: burnt area, active fire, no fire 13

Figure 3.7 Fire patches in size:128x128 (left) and 32x32 (right) 14

Figure 3.8 A residual function. Left: a building block (on 56×56 feature maps) for ResNet-34. Right: a bottleneck building block for ResNet50-ResNet101-ResNet152 (taken from [65]).....17

Figure 3.9 Network in network architecture (taken from [66]) 18

Figure 3.10 Inception modules. (a): 5×5 convolution is replaced by two 3×3 convolution, (b): after the factorization of the n × n convolutions, (c): with expanded the filter bank outputs (taken from [68]) 19

Figure 3.11 Xception architecture. (a):enrty flow, (b):middle flow, (c):exit flow (taken from [44]).....20

Figure 3.12 Left: Standard convolutional layer. Right: Depthwise Separable convolutions (adopted from [69]) 21

Figure 3.13 Model curves of each architecture.(a):DenseNet 121,(b):DenseNet169, (c):DenseNet201, (d):Inceptionv3, (e):MobileNet,(f) :ResNet101, (g):ResNet152, (h):VGG16, (i):VGG19, (j)Xception 32

LIST OF TABLES

Table 2.1 Sentinel-1 IW Mode Characteristics (adopted from [51]) 8
Table 2.2 Spectral bands of Sentinel-2 (adopted from [55]) 9
Table 3.1 VGG architecture (adopted from [64]) 16
Table 3.2 ResNet Architecture(adopted from [65]) 17
Table 3.3 Inceptionv3 architecture (adopted from [68]) 19
Table 3.4 MobileNet Architecture adopted from [69] 21
Table 3.5 DenseNet architecture adopted from [42] 22
Table 3.6 Accuracy assessment of all results 25
Table 3.7 Confusion matrix results 26
Table 3.8 DenseNet121 predictions 27
Table 3.9 DenseNet169 predictions 27
Table 3.10 DenseNet201 predictions 28
Table 3.11 Inceptionv3 predictions 28
Table 3.12 MobileNet predictions 29
Table 3.13 ResNet101 predictions 29
Table 3.14 ResNet152 predictions 30
Table 3.15 VGG16 predictions 30
Table 3.16 VGG19 predictions 31
Table 3.17 Xception predictions 31

ABSTRACT

The Role of Sentinel-1 and Sentinel-2 Data for Observing Wildfires

Deniz Gaye DENİZOĞLU

Department of Geomatics Engineering

Master of Science Thesis

Supervisor: Prof. Dr. Fusun BALIK ŞANLI

Co- supervisor: Prof. Dr. Martin WERNER

Wildfires are one of the most destructive natural disasters. The effects of wildfires on human populations, wildlife, the environment, air pollution, deforestation, the economy, and climate change are significant. Therefore, wildfire detection and mapping are essential for both early disaster response and fire forecasting. One of the most effective methods for detecting wildfires even in unsettled areas is satellite imagery, which also provides accurate and fair assessments of the fire size and effects of fires in near real-time. In recent years deep learning methods have been used frequently in remote sensing image analysis due to their ability to automatically extract features. The objective of the thesis is to use deep learning networks with different depths and different link transfer strategies to detect active-fire, no-fire, and burnt forest areas. For this purpose, Dixie fire, the second biggest wildfire that took place in the Plumas National Forest, Lassen National Forest, Lassen Volcanic National Park, Butte, Plumas, Shasta, Lassen, and Tehama counties in California was investigated. A manually labeled dataset was generated using Sentinel-2 images. VGG16, VGG19, Xception, ResNet101, ResNet152, MobileNet, Inceptionv3, DenseNet121 DenseNet169, and DenseNet201 deep

learning architectures were used to detect fire, no-fire, and burnt areas. As a result, among the 10 different deep neural network architectures, Xception outperformed with an accuracy of 98%.

Keywords: Wildfire detection, Sentinel-2, Sentinel-1, deep learning, synthetic aperture RADAR.

ÖZET

Orman Yangınlarının SENTİNEL-1 ve SENTİNEL-2 Verileri ile İzlenmesi

Deniz Gaye DENİZOĞLU

Harita Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Prof. Dr. Fusun BALIK ŞANLI

Eş-Danışman: Prof. Dr. Martin WERNER

Orman yangınları en yıkıcı doğal afetlerden biridir. Orman yangınlarının insan nüfusu, yaban hayat, çevre, hava kirliliği, ormansızlaşma, ekonomi ve iklim değişikliği üzerinde çok önemli ve geniş etkileri vardır. Bu nedenle, orman yangınlarının tespiti ve haritalanması, afetlere erken müdahale ve gelecekteki yangınların tahmin edilmesi için çok önemlidir. Uydu görüntüleri, yerleşimin olmadığı bölgelerde bile orman yangınlarını tespit etmek ve yangın boyutu ve etkisinin neredeyse gerçek zamanlı olarak doğru ve adil bir değerlendirmesini yapmak için en kullanışlı araçlardan birini sunar. Son yıllarda derin öğrenme yöntemleri, öznitelikleri otomatik olarak çıkarabilmeleri nedeniyle uzaktan algılama görüntü analizinde sıklıkla kullanılmaktadır. Tezin amacı, aktif yangın, yangından etkilenmemiş alan ve yanmış orman alanlarını tespit etmek için farklı derinliklere sahip derin öğrenme ağlarını kullanmaktır. Bu amaçla Kaliforniya'nın Plumas Ulusal Ormanı, Lassen Ulusal Ormanı, Lassen Volkanik Ulusal Parkı, Butte, Plumas, Shasta, Lassen ve Tehama ilçelerinde meydana gelen ve Kaliforniya'da meydana gelen en büyük ikinci orman yangını olan Dixie yangını incelenmiştir.

Sentinel-2 görüntüleri kullanılarak manuel olarak etiketlenmiş bir veri seti oluşturuldu. Aktif yangın, yangından etkilenmemiş alan ve yanmış alanları tespit etmek için VGG16, VGG19, Xception, ResNet101, ResNet152, MobileNet, Inceptionv3, DenseNet121 DenseNet169 ve DenseNet201 derin öğrenme modelleri kullanıldı. Sonuç olarak,10 farklı derin öğrenme modeli arasında Xception %98'lik doğrulukla diğer modellerden başarılı oldu.

Anahtar Kelimeler: Orman yangını tespiti, Sentinel-2, Sentinel-1, derin öğrenme, sentetik açıklıklı RADAR.

1 INTRODUCTION

1.1 Literature Review

Wildfires are one of the most important natural hazards. They have several effects on many areas. Human habitations, wildlife, the environment, air pollution, deforestation, the economy, and climate change are all significantly and widely impacted by wildfires. They have a negative effect on air quality by releasing carbon [1],[2],[3] harm wildlife and biodiversity [4]. They don't only destroy the environment but also cause economic loss. In 2021 wildfires cost more than $4.4 billion in the USA destroyed [5].

Throughout the past 5 decades, spatial technological advancement, in particular satellite remote sensing technology, has offered advanced detection and research methods for earth observation. These methods have the advantages of being large-scale, detailed, rapid, and precise [6]. Remote sensing techniques have made an important contribution to wildfire monitoring, both in terms of their accessibility and capabilities [7]. The orbital features of satellites allow for the consistent and repeating acquisition of imagery of the whole planet, including regions that are typically inaccessible and quite remote, such as numerous polar, mountainous, desert, and forested areas [8]. Remote sensing sensors provide ideal conditions for time-series analysis since they provide global information through Earth Observation (EO) [9].

In recent years, the application of deep learning methods to remote sensing analysis has increased [10]. Wildfire detection is a crucial task since it can be challenging to identify the source of ignition in a short period, which frequently causes the intensity of fire incidents to increase. The development of precise forecast solutions has sparked major attention among researchers and practitioners due to this issue, and recent advances in deep learning for challenging visual interpretation tasks have created new study areas [11]. But the classical threshold approaches can be affected by tiny forest clearings, and the

omission error hidden by dense smoke makes reliable and rapid detection of wildfires difficult. Robust flexibility, good educational capacity, and high adaptability traits of deep learning made many researchers use it in an almost timely manner to solve the challenge of detecting wildfires in satellite imagery [12].

Active fire and burnt area detection using satellite imagery dates back to the 1970s. In [13] IR scanners from an aircraft using 3- μm to 4-μm and 8.5- μm to 11- μm bands were tested. In [14] active fires with 3.8-μm and 11.0-μm data from the NOAA-6 satellite were detected. In 2003 [15] authors extended the algorithm on the previous work of [16] studied with MODIS (or Moderate Resolution Imaging Spectroradiometer) instrument with 1 km resolution fire channels at 4 and 11. This instrument had a lower false alarm rate and could detect smaller fires. In [17] an active fire detection algorithm that uses the brightness temperature obtained from the 1-km MODIS with 4- μm, 11- μm, and 12-μm was explored. In [18] authors derived data from Landsat-8 operational land imager (OLI) to detect active fires extending the previous studies using multi-temporal analysis. Authors of [19] proposed a hotspot detection method with Landsat-8 OLI data and succeeded to detect 80% of hot pixels. A global operational land imager (GOLI) algorithm that uses a threshold derived from reflectance in Landsat-8 band 4 (0.66 μm), and band 7 (2.20 μm) was developed by [20]. In [21] authors modified U-NET [22] architecture to detect active fires automatically through images acquired from Landsat-8 with outputs gained from well-established, handcrafted algorithms of [18], [19], and [20]. An early wildfire detection model with 3D convolution patch-level classification was introduced by [23] and got a 94% F1 score. In [24] authors analyzed how the Sentinel-2A satellite can be used to map burnt areas. They compared the performance of different bands and spectral indices in an effective way to differentiate between unburnt and burnt areas. The results showed that the most suitable multi-spectral instrument (MSI) bands for detecting burnt areas are the 20 m near-infrared, short-wave infrared, and short-wave bands. Spectral indices from Sentinel-2A were employed by [25] to assess burn severity.

A neural network to classify burnt areas from time series of MODIS imagery was used in [26]. Assessing Sentinel-2 MSI reflectance values in the short and near-infrared wavebands and using the active fires observed by Terra and Aqua MODIS sensors as inputs, in [27]authors suggested a locally-adapted multitemporal two-phase burnt area algorithm. In the first phase, a basic burnt area map is made, and for the second step, tile-dependent data are collected from it. The study found that the product derived from Sentinel-2 imagery at 20-m spatial resolution was more accurate particularly in detecting small fires than MODIS burnt area product (500 m for MCD64A1 c6). In [28] authors demonstrated a super-resolution data fusion method based on a convolutional neural network (CNN) to progress towards the 10-m spatial resolution of the short-wave infrared (SWIR) band to create more comprehensive active fire detection (AFD) maps using Sentinel-2 images. In [29] authors described a weight-selection strategy to address the imbalanced classification of wildfires from MODIS imagery using deep neural networks. Burnt areas were detected using multi-temporal series of three different spectral indices that represent wildfire disturbance with the suggested methodology producing commission and omission errors of around 25% and 40%, respectively from Sentinel-2 imagery in [30].

In [31] the authors used U-Net for burnt area segmentation with a recall of more than 90% across all regions. [32] the authors reached a classification result with an average F1 score of approximately 96% from U-Net and ResUNet [33] architectures by using deep convolutional autoencoders to classify burnt areas using Landsat-8 OLI + imagery. The study of the burnt area segmentation from Sentinel-2 imagery using a U-Net architecture was carried out by [34] reported an overall accuracy of 98%. A deep learning method to detect burnt areas from visible infrared imaging radiometer suite (VIIRS) imagery presented by [35] achieved competitive results. In the study conducted by [36], active fire detection and progression monitoring with Sentinel-1 (synthetic aperture RADAR) SAR time series was performed by using convolutional neural networks. In [37] the authors utilized support vector machine (SVM) to detect burnt areas on pixel-basis in Sentinel-2 imagery and reached a classification accuracy of more than 92%. In [38] authors applied convolutional neural networks to a combination of Sentinel-

1 and Sentinel-2 data to detect burnt areas. A multi-criteria approach was developed by [39] to detect active fire based on the reflectance of the B4, B11, and B12 bands of Sentinel-2.

One of the recent studies by [40] proposed a method to detect active fire and burnt area using U-Net from Sentinel-1, Sentinel-2, Sentinel-3, and MODIS imagery. The classification of wildfires using EfficientNet-B5 [41] and DenseNet201[42] was undertaken by [43] and they had an accuracy of 85%.

In this study, 10 deep learning architectures were utilized to identify active fire, burnt, and unburnt regions in wide forest areas of California. For this purpose, publically available Sentinel-1 and Sentinel-2 images were manually labeled based on target classes. However, due to the labeling problem of Sentinel-1, Sentinel 2 was preferred for further processing. Experimental results presented that the Xception [44] architecture outperformed all other architectures with an overall accuracy of 98%. Others have might have higher accuracies but they tend to overfit. This thesis consists of four chapters. The first chapter covers the introduction and literature review. The second chapter presents study area and data. The third chapter presents methodology. In the fourth chapter, the results and discussion are given.

1.2 Objective of the Thesis

Using deep learning networks with different depths and different link transfer strategies to detect active-fire, no-fire, and burnt forest areas.

1.3 Hypothesis

Deep learning networks are used to detect active fire, no fire, and burnt forest areas using optical/SAR images.

2
STUDY AREA AND DATASET

2.1 Study Area

The Fig.2.1 shows the study area of the Dixie Fire that was observed in this thesis. Dixie Fire is the second biggest wildfire that took place in the Plumas National Forest, Lassen National Forest, Lassen Volcanic National Park, Butte, Plumas, Shasta, Lassen, and Tehama counties in California. It started on 13/07/21 and contained 25/10/21. Drought combined with hot weather, strong winds, and extremely dry vegetation caused the fire to intensify [45]. It was active for 103 days and burned 963,309 acres, destroying 1,329 structures [46]. The Dixie Fire, which cost $637.4 million to put out, was the most expensive wildfire in American history and this amount excludes damage and insurance claims [47]. In Fig.2.2 the study area is seen where the red polygon marks the Dixie fire.Fig.2.3 is a RGB (bands: 4,3,2) composite Sentinel-2 image of the study area where smoke plumes can be observed.

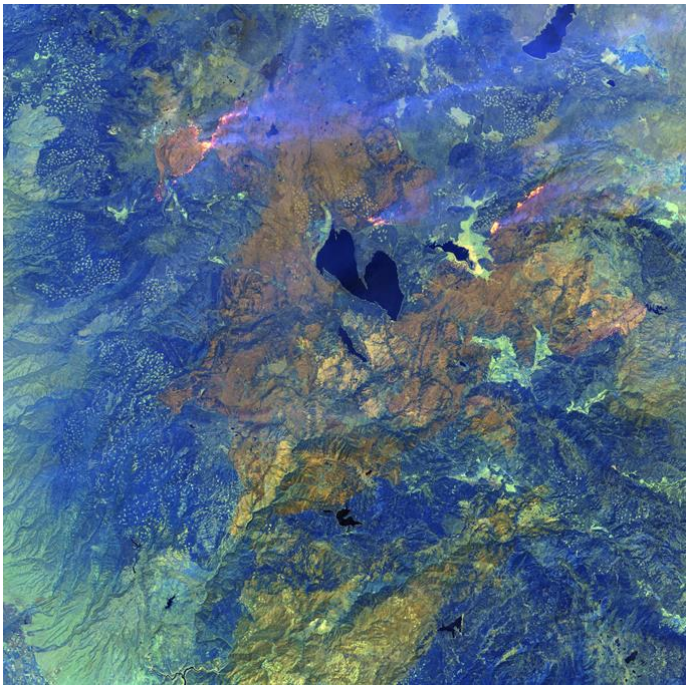


Figure 2.1 Composite of Sentinel-2 bands 12,11, and 8 of the study area

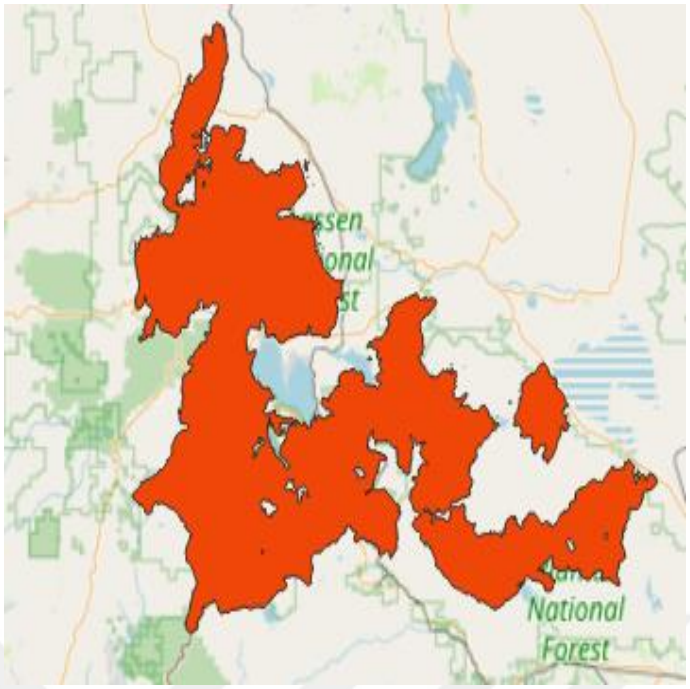


Figure 2.2 Representation of the study area where the red polygon marks the Dixie fire



Figure 2.3 RGB (B04,B03,B02) Sentinel-2 image of the study area

2.2 Dataset

A polygon was created on QGIS[48] that represents the fire area. According to this polygon, the Sentinel-2 images were acquired dates between 13/07/2021 and 25/11/2021 from Copernicus Open Access Hub [49]. Sentinel-1 and Sentinel-2

products were downloaded from Copernicus Open Access Hub to detect active fires for the proposed method in specific formats.

The Sentinel-1 mission is based on a constellation of identical C-band SAR satellites consisting of A and B units. Four imaging modes with varying resolution and coverage are supported by the 5.405 GHz SAR instrument: Interferometric Wide Swath Mode (IW), Extra Wide Swath Mode (EW), StripMap (SM), and Wave (WV). The repeat cycle is 12 days [50]. The revisit frequency of our study area in California is 12 days and the coverage frequency of our study area in California is 2-4 days as shown in Figure 2.4. The Sentinel-1 equipment can emit linear polarizations both in the horizontal (H) or vertical (V) directions. Data can be collected in single co-polarisation (HH or VV) or dual polarisation (HH+HV or VV+VH) for the SM, IW, and EW modes. Only single co-polarization data acquisition is available for WV mode (HH or VV only) [51]. Table 2.1 shows the Sentinel-1 IW mode characteristics.

Sentinel-1 imagery Level-1 Ground Range Multi-Look Detected (GRD) product in the IW mode with dual polarisation (VV+VH) was acquired from Copernicus Open Access Hub between 03/07/2021 and 01/11/2021. The image size is 10980×10980.

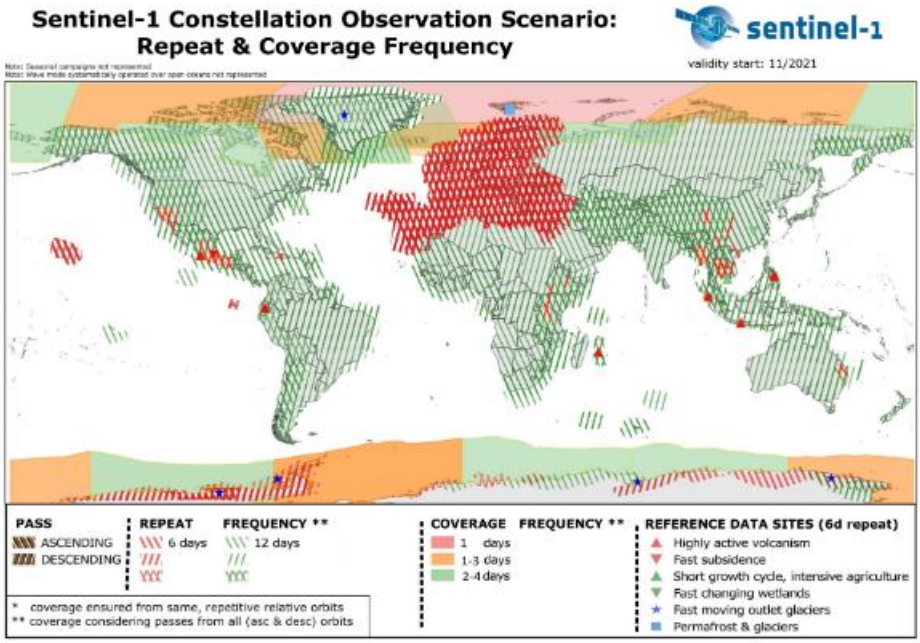


Figure 2.4 Sentinel-1 constellation revisit and coverage frequency (taken from [52])

Table 2.1 Sentinel-1 IW Mode Characteristics (adopted from [51])

Parameter	Value
Minimum Ground Swath Width	250 km
Incidence Angle Range	29.1°-46.0°
Number of Sub-swath	3
Azimuth Steering Angle	± 0.6°
Azimuth Resolution	20 m
Ground Range Resolution	5 m
Polarisation Options	Single (HH or VV) or Dual (HH+HV or VV+VH)

Sentinel Application Platform (SNAP) [53] was only used to see original SAR images. The goal of this study was to label fire on Sentinel-2 and Sentinel-1 SAR images.

Since active fire could not be seen and could not be labeled on SAR images, the task was only carried out with Sentinel-2 images.

The Sentinel-2 mission consists of two identical satellites with five revisit days at the equator.Sentinel-2 has 13 spectral bands ranging from visible (VIS) and near-infrared (NIR) to short-wave infrared (SWIR) with different spatial resolutions ranging from 10 to 60 m at ground level [54] as shown in Table 2.2. 42 fire, 1 before the fire, 4 after fire and Sentinel-2A and Sentinel-2B images were downloaded as zip files from Copernicus Open Access Hub between 18/07/2021 and 21/10/2021. Levels 1C and 2A products were used in this thesis.

Table 2.2 Spectral bands of Sentinel-2 (adopted from [55])

Band number	S2A		S2B		Spatial resolution (m)
	Central wavelength (nm)	Bandwidth (nm)	Central wavelength (nm)	Bandwidth (nm)	
1	442.7	20	442.2	20	60
2	492.7	65	492.3	65	10
3	559.8	35	558.9	35	10
4	664.6	30	664.9	31	10
5	704.1	14	703.8	15	20
6	740.5	14	739.1	13	20
7	782.8	19	779.7	19	20
8	832.8	105	832.9	104	10
8a	864.7	21	864	21	20
9	945.1	19	943.2	20	60
10	1373.5	29	1376.9	29	60
11	1613.7	90	1610.4	94	20
12	2202.4	174	2185.7	184	20

3
METHODOLOGY

3.1 Dataset Generation

As mentioned in section 2.2 Sentinel-2 imagery was used to generate the dataset. The process chain to generate and classify the wildfire patches is given in Figure 3.1.



Figure 3.1 Visualization of the processing chain to generate and classify the wildfire patches

3.1.1 Image Preparation

After a literature review and trying several band combinations and visual analysis, (B12-B11-B8) false-color composite was chosen as the most suitable band combination for detecting active fire. Observed band combinations and images created are represented in Fig.3.2. The image size is 10980×10980. All the downloaded images were interpreted visually. The dataset did not include images with artifacts, large clouds, or damage ataset. Images were created in two different files TIFF and JPEG. JPEG were used for visulization and the CNNs used the TIFF ones. The last step in the pre-processing chain was to cut out the labeled areas, those tiles has been saved in TIFF format. Generated band combinations of Sentinel-2 images to detect wildfires is shown in Fig.3.2.

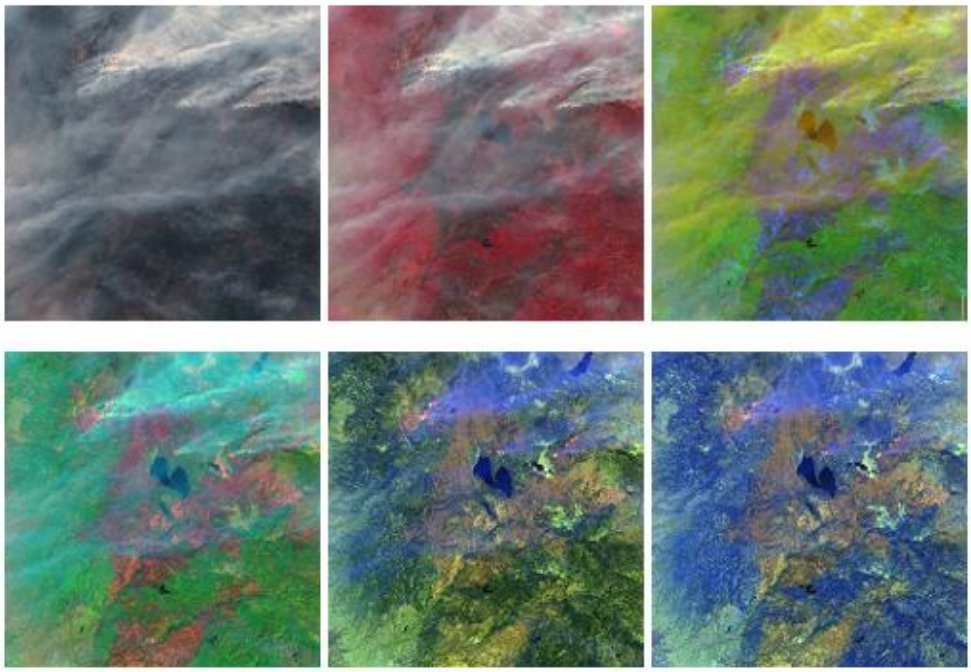


Figure 3.2 Generated Sentinel-2 band combinations to detect wildfire. From left to right: (B04,B03,B02), (B08,B04,B03), (B02,B08,B12), (B12,B08,B02), (B12,B11,B04), (B12,B11,B08)

The selected Sentinel-2 image band combination (B12,B11,B8) is shown in Figure 3.3.

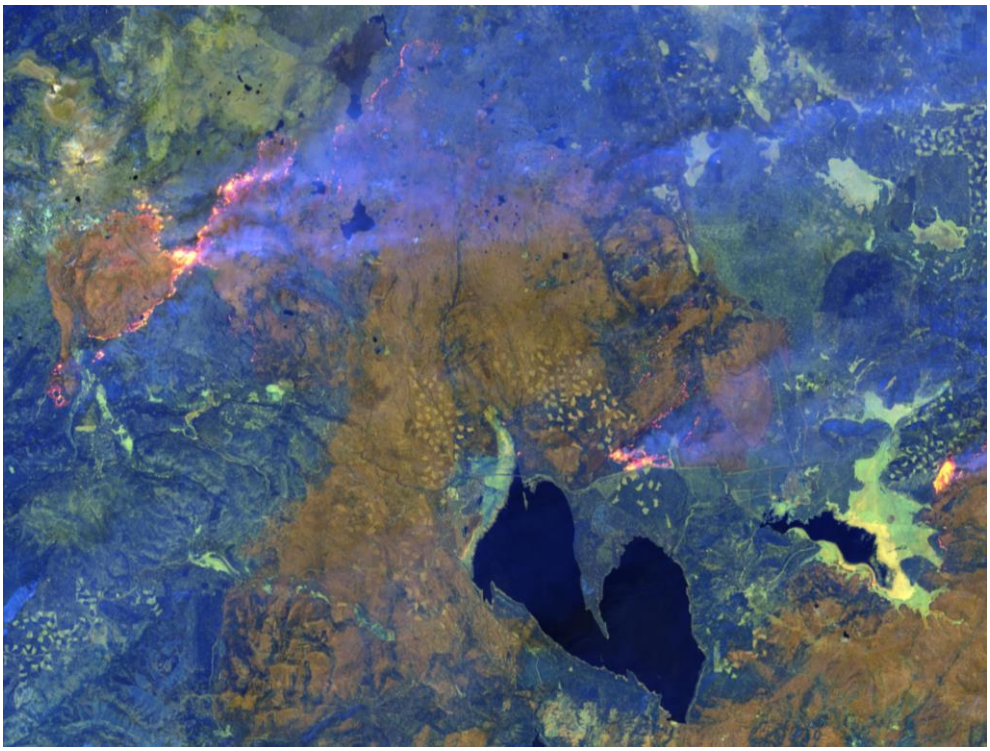


Figure 3.3 Captured wildfire from Sentinel- 2 image (B12,B11,B8)

3.1.2 Labeling

After having the TIFF images; three classes, fire, no-fire, and burnt area were chosen as target classes. First an empty shape file was created in QGIS and TIFF remote sensing images were manually labeled according to these three classes. Then pixels were chosen carefully not to overlap. Sentinel-2 image with these sample labels is shown in Fig.3.4. Lastly, all these points were inserted into the shape file and saved on the disk. Sentinel-1 in VV + VH and Sentinel-2 images generated during wildfire are shown in Figure 3.5. Sentinel-1 images couldn't be labeled since the active fire couldn't be detected by visual interpretation.

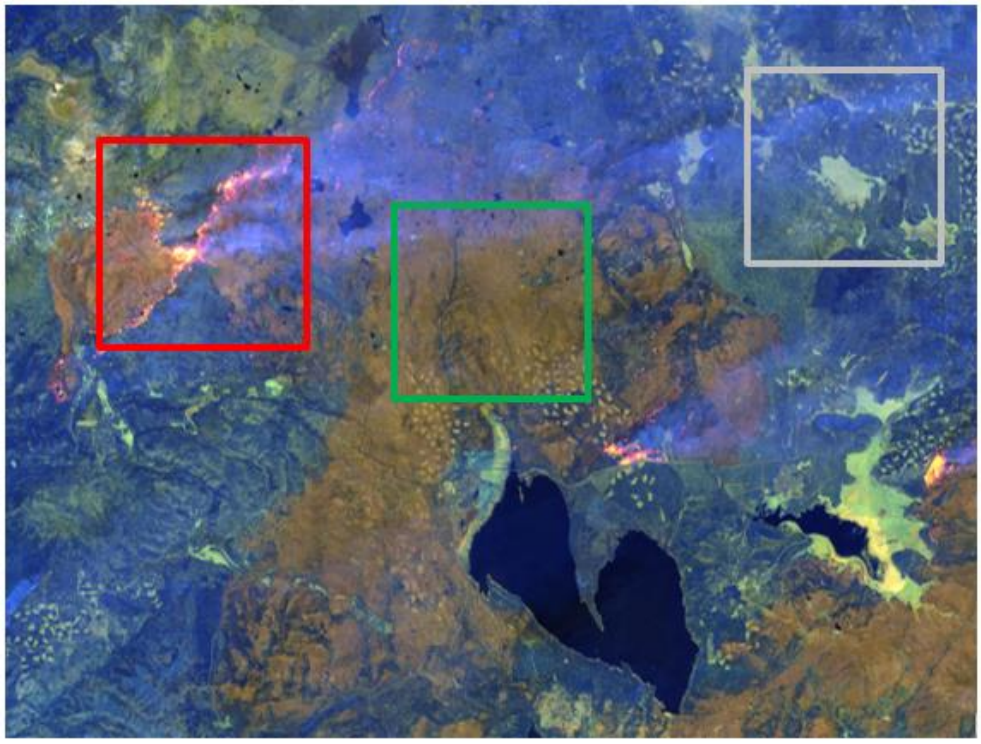


Figure 3.4 Sentinel-2 image with sample labels, where the red bounding box shows active fire, the green represents burnt area, and the gray box includes unaffected area

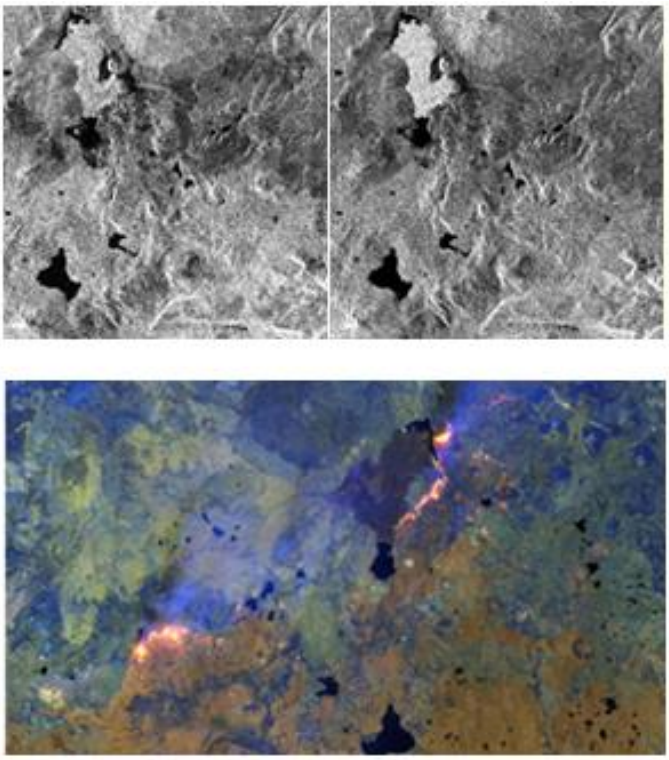


Figure 3.5 The study area during wildfire captured from Sentinel-1 (VV+VH) (above), and Sentinel-2 (below).

3.1.3 Patch Creation

After labeling, patches were created for each class with specific sizes of 128×128. Each coordinate was represented with a corresponding label. In order to generate the patches with the size of 128×128 pixels, each landmark which includes fire, no-fire, and, burnt area classes were considered as an origin. The patch creation stage was performed in Python 3.8. and train/test process of 10 different architectures were completed in Keras [56]. Fig.3.6 exemplifies the patches of the each class.



Figure 3.6 Patch samples for each class. From left to right: burnt area, active fire, no fire

Different patch sizes were also evaluated as shown in Figure 3.7.

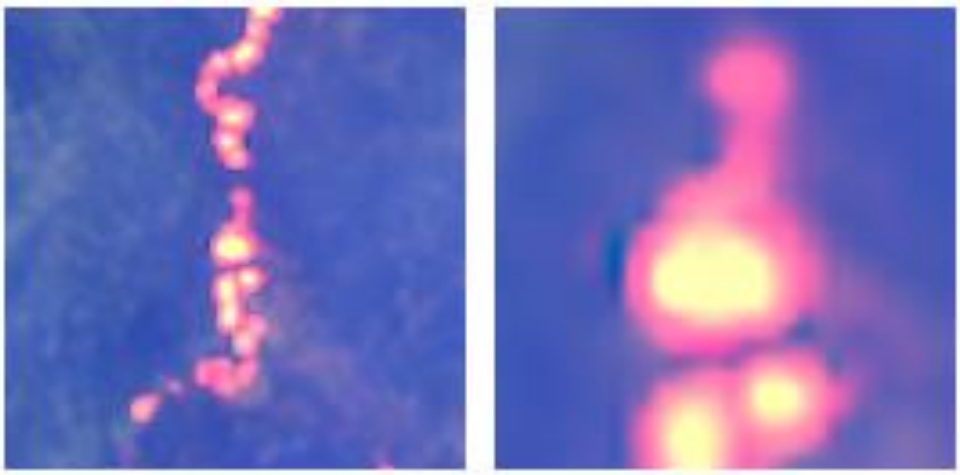


Figure 3.7 Fire patches in size:128x128 (left) and 32x32 (right)

3.1.4 Classification

The dataset consists of 3 categories; 668 fire, 377 no-fire, and 512 burnt area patches, in total there are 1557 patches in the dataset. In the classification part, this data was fed it into a network. In the network, each output was one of these label classes. Based on these results the CNNs were trained on different deep-learning models. The dataset was split into training and validation set.80% of the total images were used for training and 20% were used for validation. The training set had 1247 images and the validation set had 310 images. Each image was 1281×28 in shape, and had 3 channels. Adaptive moment estimation (Adam) optimization algorithm [57], a learning rate of 0.00 1,a batch size of 32, sparse categorical cross entropy as the loss function, and 20 epochs were used to train CNNs. As initial weights, weights learned on ImageNet as part of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [58] were used.

3.2 Deep Learning

Deep learning can be defined as a machine learning method that performs the learning process based on concept hierarchies by making decisions similar to humans, which is used in many research branches such as segmentation, classification, speech recognition, and natural language processing [59]. The biggest advantage of this method is that the computer can acquire the features or attributes which will be used for learning purposes on its own [60]. Based on the

type of experience that machine learning algorithms are permitted to acquire while learning, they can be roughly classed as supervised or unsupervised [59]. In supervised learning, data must contain labels to learn the output functions. In unsupervised learning, data don't have labels. These algorithms serve as feature learning, representation learning, or clustering techniques that conclude directly from the data itself and may be applied to data-driven analytics.

Convolutional neural networks handle data presented as numerous arrays, such as a color image made up of three 2D arrays representing pixel value for the RGB image. Numerous data formats take the shape of multiple arrays: 1D for signals and sequences, such as speech; 2D for images; and 3D for video [61]. Their capability can be adjusted by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). As a result, when compared to standard feedforward neural networks with similar-sized layers, CNNs have far fewer connections and parameters, making them easier to train [62]. Deep neural networks have multiple non-linear hidden layers, making them highly expressive models capable of learning extremely complex relationships between their inputs and outputs [63].

3.2.1. VGG

In 2015 [64] authors investigated a convolutional layer architecture that aims to increase the depth of a model by using 16-19 weight layers. To get this result the authors added layers to gradually increase the depth while keeping other parameters fixed. And they got a more accurate architecture.

D convolutional with 16 weight layers was used in this study. It consists of 13 3×3 convolutional layers, five max-pooling layers, three fully connected layers, and a soft-max layer. VGG architecture is shown in Table 3.1.

Table 3.1 VGG architecture (adopted from [64])

11	11	13	16	16	19
conv3 64	conv3 64 LRN	conv3 64 conv3 64	conv3 64 conv3 64	conv3 64 conv3 64	conv3 64 conv3 64
maxpool					
conv3 128	conv3 128	conv3 128 conv3 128	conv3 128 conv3 128	conv3 128 conv3 128	conv3 128 conv3 128
maxpool					
conv3 256 conv3 256	conv3 256 conv3 256	conv3 256 conv3 256	conv3 256 conv3 256 conv1 256	conv3 256 conv3 256 conv3 256	conv3 256 conv3 256 conv3 256 conv3 256
maxpool					
conv3 512 conv3 512	conv3 512 conv3 512	conv3 512 conv3 512	conv3 512 conv3 512 conv1 512	conv3 512 conv3 512 conv3 512	conv3 512 conv3 512 conv3 512 conv3 512
maxpool					
conv3 512 conv3 512	conv3 512 conv3 512	conv3 512 conv3 512	conv3 512 conv3 512 conv1-512	conv3 512 conv3 512 conv3-512	conv3 512 conv3 512 conv3-512 conv3-512
maxpool					
FC 4096					
FC 4096					
softmax					

3.2.2 ResNet 50 – ResNet101 – ResNet152

In 2015 [65] authors introduced a residual learning architecture to make it simpler to train deeper networks, and to overcome degradation which means in a deep residual learning architecture, accuracy gets saturated as network depth rises and subsequently rapidly declines. ResNet architecture is based on the VGG architecture but the ResNet uses a 34-layer plain architecture that is simpler than VGG and uses fewer filters. The architecture is then converted into a residual block by adding shortcut connections to this plain layer. Convolutional layers with a stride of 2 immediately execute down sampling. After that, a 1000-way fully-connected layer with softmax and a global average pooling layer was added.

ResNet50 is obtained by substituting this 3-layer bottleneck block for each of the ResNet34's 2-layer blocks. ResNets101 and ResNet152 layers are generated with more 3-layer blocks. Despite the higher depth, ResNet152 is less complex than VGG nets. ResNet architecture is shown in Table 3.2. A residual function including a building block for ResNet-34 and a bottleneck building block for ResNet50, ResNet101 and ResNet152 is shown in Fig.3.8.

Table 3.2 ResNet Architecture(adopted from [65])

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{pmatrix}3\times3,64\\3\times3,64\end{pmatrix}\times2$	$\begin{pmatrix}3\times3,64\\3\times3,64\end{pmatrix}\times3$	$\begin{pmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{pmatrix}\times3$	$\begin{pmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{pmatrix}\times3$	$\begin{pmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{pmatrix}\times3$
conv3_x	28×28	$\begin{pmatrix}3\times3,128\\3\times3,128\end{pmatrix}\times2$	$\begin{pmatrix}3\times3,128\\3\times3,128\end{pmatrix}\times4$	$\begin{pmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{pmatrix}\times4$	$\begin{pmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{pmatrix}\times4$	$\begin{pmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{pmatrix}\times8$
conv4_x	14×14	$\begin{pmatrix}3\times3,256\\3\times3,256\end{pmatrix}\times2$	$\begin{pmatrix}3\times3,256\\3\times3,256\end{pmatrix}\times6$	$\begin{pmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{pmatrix}\times6$	$\begin{pmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{pmatrix}\times23$	$\begin{pmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{pmatrix}\times36$
conv5_x	7×7	$\begin{pmatrix}3\times3,512\\3\times3,512\end{pmatrix}\times2$	$\begin{pmatrix}3\times3,512\\3\times3,512\end{pmatrix}\times3$	$\begin{pmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{pmatrix}\times3$	$\begin{pmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{pmatrix}\times3$	$\begin{pmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{pmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

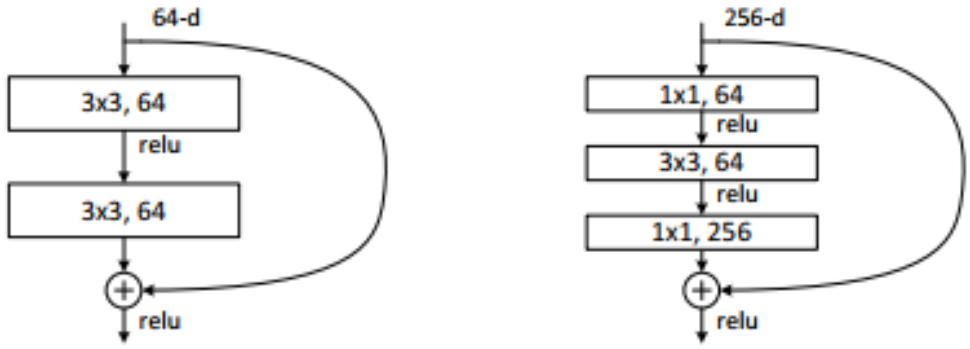


Figure 3.8 A residual function. Left: a building block (on 56×56 feature maps) for ResNet-34. Right: a bottleneck building block for ResNet50-ResNet101-ResNet152 (taken from [65])

3.2.3 Inceptionv3

In [66] Network in Network (NIN) was introduced . Unlike traditional CNNs the authors created micro neural networks with more complicated structures to separate data from the receptive field. The micro neural network was created using a multilayer perceptron. The feature maps were created by sliding the micro networks across the input and sent into the next layer. This novel structure is made up of mlpconv layers, which use multilayer perceptrons to convolve the input, and a global average pooling layer, which replaces the traditional CNN's fully connected layers. NIN is employed to strengthen neural networks' capacity for representation. The method could be considered as adding more 1×1 convolutional layers on the existing convolutional layers, usually followed by rectified linear activation [40].The network in network architecture is shown in Fig.3.9.

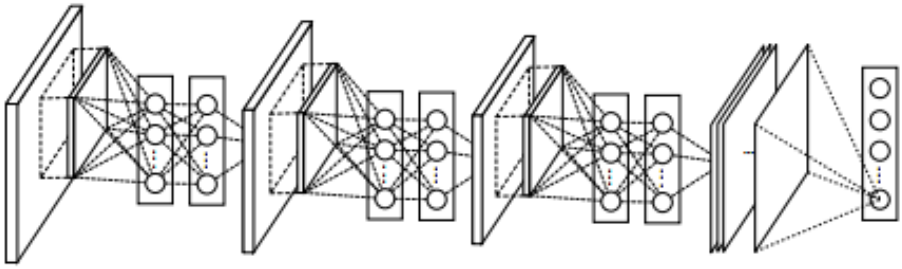


Figure 3.9 Network in network architecture (taken from [66])

In 2014 [67] took NIN as a baseline and introduced the Inception module that a region of the input image is represented by each unit from the previous layer, and these units are arranged into filter banks. The current iterations of the Inception architecture are limited to filters of sizes 1×1, 3×3, and 5×5.

Instead of choosing one, all filters can be used together with a subsequent pooling operation. Each operation is carried out in parallel. Then the results can be combined. But this causes a computational burden. Due to this reason, 1×1 convolutional is used before 3×3 and 5×5 convolutional to reduce the depth of the feature map.

In [68] authors extended their Inception architecture to Inceptionv3 in which the auxiliary classifiers serve as a regularizer. This is corroborated due to the network's primary classifier performance depends on the dropout layer. The term batch

norm-auxiliary describes the form of the auxiliary classifier in which not only the convolutions but also the fully connected layer are batch-normalized. In Fig.3.10 different inception modules are shown. In Fig.3.10 (a) inception modules that uses two 3×3 convolution instead of 5×5 convolution are shown. In Fig.3.10 (b) Inception modules after the factorization of the n×n convolutions are shown. n is chosen 7 for the 17×17 grid. In Fig.3.10 (c) inception modules with expanded filter bank outputs are shown [68].The architecture of Inceptionv3 is shown in Fig.3.11.

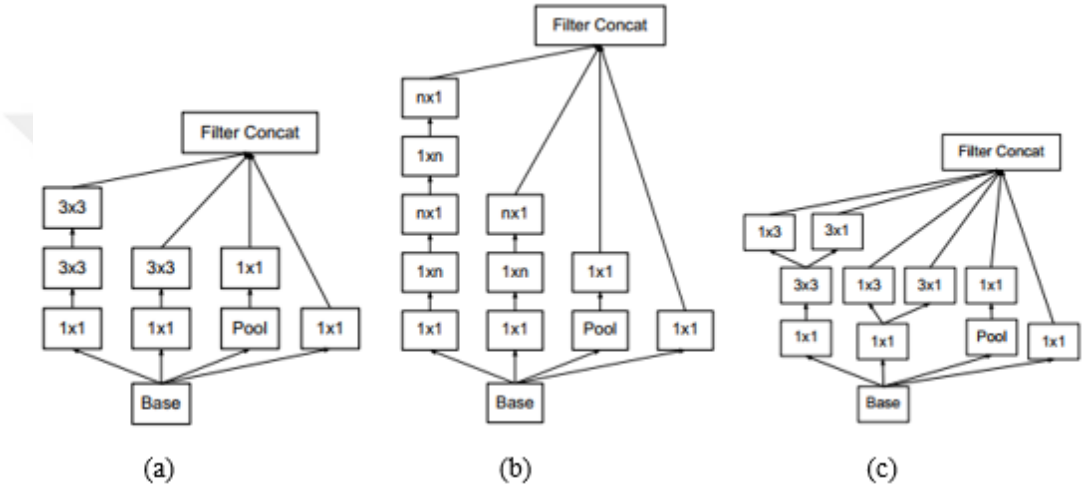


Figure 3.10 Inception modules. (a): 5×5 convolution is replaced by two 3×3 convolution, (b): after the factorization of the n × n convolutions, (c): with expanded the filter bank outputs (taken from [68])

Table 3.3 Inceptionv3 architecture (adopted from [68])

type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in Fig. 3.10 (a)	35×35×288
5×Inception	As in Fig. 3.10 (b)	17×17×768
2×Inception	As in Fig. 3.10 (c)	8×8×1280
pool	8×8	8×8×2048
linear	logits	1×1×2048
softmax	classifier	1×1×1000

3.2.4 Xception

In 2017 Xception was introduced by [44] which can be described as an extreme version of Inception. The spatial correlations of each output channel are individually mapped by Xception after mapping cross-channel correlations using a 1×1 convolution and achieved better results than Inception v3. In Xception architecture, standard Inception modules have been replaced with depth-wise separable convolutions. The architecture consists of 36 convolutional layers which are built into 14 modules with residual connections. The architecture of Xception is shown in Fig.3.11.

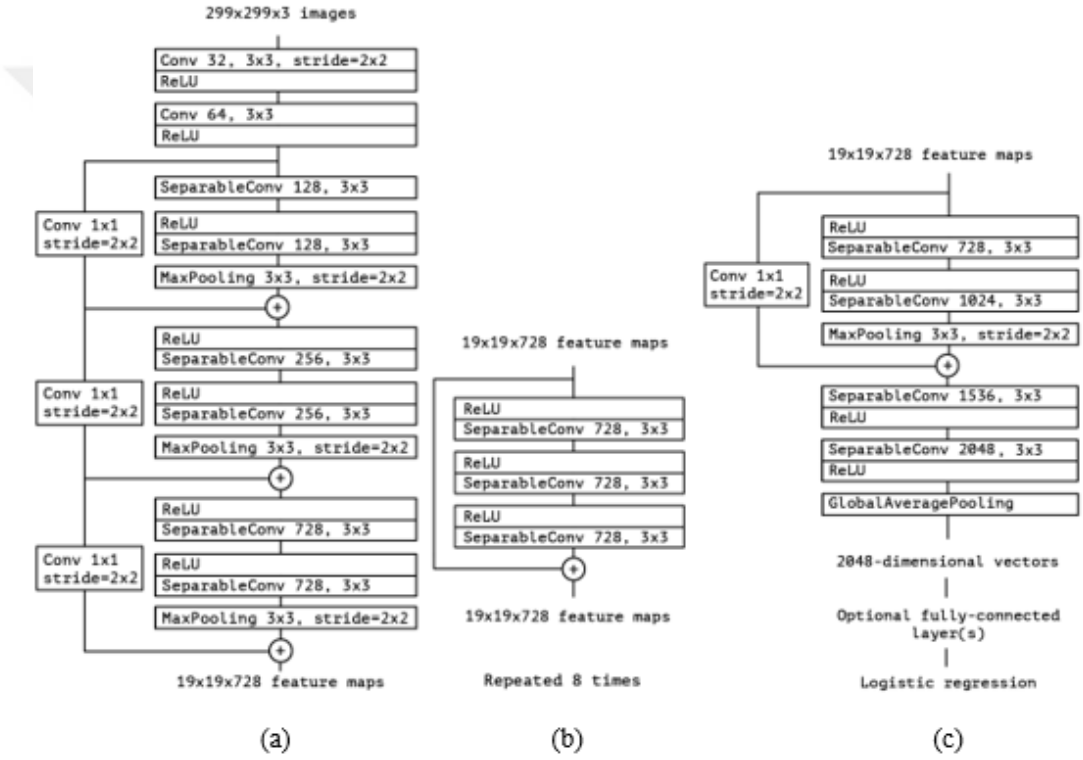


Figure 3.11 Xception architecture. (a):enrty flow, (b):middle flow, (c):exit flow (taken from [44])

3.2.5 MobileNet

In 2017 an architecture that is established on depth-wise separable convolutions which consist of a depth-wise convolution and a 1×1 convolution known as a pointwise convolution was proposed by [69]. This is the key distinction between a traditional CNN and MobileNets as shown in Fig. 3.12.

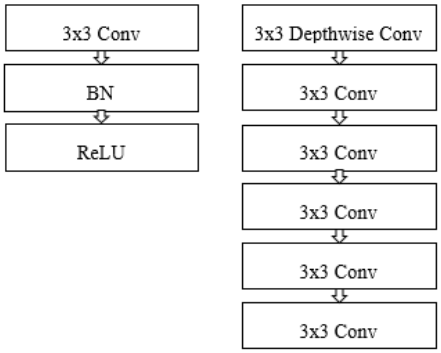


Figure 3.12 Left: Standard convolutional layer. Right: Depthwise Separable convolutions (adopted from [69])

Except for the final fully connected layer, which has no nonlinearity and feeds into a softmax layer for classification, all layers are followed by a batch norm and ReLU nonlinearity. The architecture of MobileNet is shown in Table 3.4.

Table 3.4 MobileNet Architecture adopted from [69]

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
conv dw/s1 conv/s1 × 5	3 × 3 × 512 dw 1 × 1 × 512 × 512	14 × 14 × 512 14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

3.2.6 DenseNet121-DenseNet169-DenseNet201

In [42] the authors proposed the Dense Convolutional Network (DenseNet) in which all layers are connected directly with one another with matching feature-map sizes to achieve the greatest amount of information flow across levels in the network.

Each layer receives extra inputs from all earlier layers and transmits its feature maps to all later layers to maintain the feed-forward character of the system. Even though the network design and training techniques of CNN differ, all CNN architectures generate short paths from previous layers to later layers. Unlike ResNet, DenseNet combines features by concatenating them rather than summarizing them before passing them into a layer. In addition to having superior parameter efficiency, DenseNets are easy to train and improve information flow and gradients throughout the network. The architecture consists of a transition layer between two adjacent dense blocks, 1×1 Conv, followed by 2×2 average pooling. In the dense block, the size of each feature map is the same which makes it simple to concatenate them. A global average pooling is used at the end of the final dense block, and a softmax classifier is then added. The architecture of DenseNet is shown in Table 3.5.

Table 3.5 DenseNet architecture adopted from [42]

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times6$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times6$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times6$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times6$
Transition Layer(1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times12$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times12$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times12$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times12$
Transition Layer(2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times24$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times32$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times48$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times64$
Transition Layer(3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times16$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times32$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times32$	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

3.3 Evaluation Metrics

The evaluation of the CNN models were made by using following metrics from [70]:

True positives (TP) - number of correct prediction (positive situation)
True negatives (TN) - number of correct predicition (negative situation)
False positives (FP) - number of incorrect prediction (negative reference , positive prediction)
False negatives (FN) - number of incorrect prediction (positive reference, positive prediction)

Precision is the classifier's capacity not to classify a negative sample as positive.Precision is defined as :

$$\frac{TP}{TP + FP} \tag{3.1}$$

Recall is the classifier's capacity to detect all the positive samples.Recall is defined as:

$$\frac{TP}{TP + FN} \tag{3.2}$$

F1 score can be defined as a harmonic mean of precision and recall, with the best value being 1 and the worst being 0. Precision and recall both contribute equally in terms of percentage to the F1 score.F1 is defined as:

$$2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.3}$$

3.4 Findings

The main difficulty with machine learning is that our model must deliver good results with unseen inputs. Generalization is the capacity to perform well on previously unseen inputs [59]. Obtaining good generalization error is the main objective of learning, which can be calculated by evaluating performance on a test set. If test error is too high, the first thing to assess is whether overfitting or

difficulties in optimizing the training criterion is the reason [71].If there is an excessive difference between the training error and the test error, the model over fits the dataset [59]. By gathering more training data, adding more or better regularization terms, multi-task training, or taking various function families into account, overfitting issues can be resolved [71].

In this thesis, a dataset was generated from Sentinel-2 images to detect active fire, no fire, and burnt areas.11 deep learning models were trained on this dataset. Evaluation metrics results are shown in Table 3.6. F1 scores in the table were produced from confusion matrixes. True predictions of each model for each class is shown in Table 3.7 It is seen that Xception got a very high value in terms of accuracy and the performance curve also matches with this result. Even though Inceptionv3 is considered successful in terms of accuracy and the performance curve, it tends to overfit. One can say that curves are going apart which is a sign of overfitting. It is clear that generalization gap is not too large but it could go in the direction specifically when there is a very high value. As for VGG16 and 19, results with regard to the accuracy and performance curves seem trustworthy. ResNet101, and ResNet152 faced underfitting problem which occurs when the model poorly performed on the training data. One possible explanation could be to this that residual blocks may not be suitable for classifying this dataset. DenseNet169, and DenseNet201 got an accuracy of 1.0. The reason behind this poor performance could be caused by the size of dataset. It could be said that a deep network such as DenseNet overfits when it is trained on a small dataset. On the other hand the performance curves indicate somewhat different results from accuracies. MobileNet has a high accuracy however it has a value of 1.0 in each class. Similarly, it has also overfitting problem like other deep networks DenseNets and ResNets.

From the Table and performance curves one can say that Xception is the best performing model followed by VGG19 and VGG16.

Table 3.6 Accuracy assessment of all results

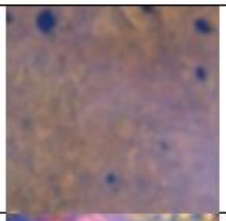
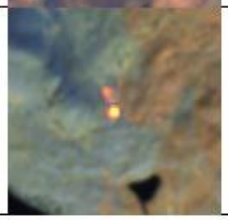
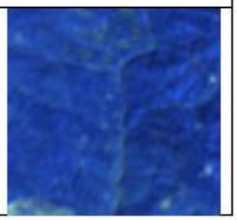
Model	Classes	Precision	Recall	F1	Accuracy
VGG16	Burnt area	0.96	0.94	0.95	0.948
	Active fire	0.97	0.95	0.96	
	No fire	0.90	0.95	0.92	
VGG19	Burnt area	0.93	0.97	0.95	0.951
	Active fire	0.97	0.98	0.97	
	No fire	0.96	0.88	0.92	
Xception	Burnt area	0.97	0.99	0.98	0.980
	Active fire	0.99	0.98	0.98	
	No fire	0.97	0.97	0.97	
ResNet152	Burnt area	0.00	0.00	0.00	0.429
	Active fire	0.43	1.00	0.60	
	No fire	0.00	0.00	0.00	
ResNet101	Burnt area	0.76	0.92	0.83	0.730
	Active fire	0.70	0.98	0.82	
	No fire	0.00	0.00	0.00	
MobileNet	Burnt area	1.00	0.99	1.00	0.996
	Active fire	1.00	1.00	1.00	
	No fire	0.99	1.00	0.99	
Inceptionv3	Burnt area	0.97	0.93	0.95	0.958
	Active fire	0.98	0.98	0.98	
	No fire	0.91	0.96	0.94	
DenseNet201	Burnt area	1.00	1.00	1.00	1.000
	Active fire	1.00	1.00	1.00	
	No fire	1.00	1.00	1.00	
DenseNet169	Burnt area	1.00	1.00	1.00	1.000
	Active fire	1.00	1.00	1.00	
	No fire	1.00	1.00	1.00	
DenseNet121	Burnt area	0.98	0.98	0.98	0.990
	Active fire	0.99	0.98	0.99	
	No fire	0.99	1.00	0.99	

Table 3.7 Confusion matrix results

	Number of True Predictions out of 310 Test Patches		
Model	Burnt area (102 patches)	Active fire (133 patches)	No fire (75 patches)
DenseNet121	100	131	75
DenseNet169	102	133	75
DenseNet201	102	133	75
InceptionV3	95	130	72
MobileNet	101	133	75
ResNet101	94	131	0
ResNet152	0	133	0
VGG16	96	127	71
VGG19	99	130	66
Xception	101	130	73

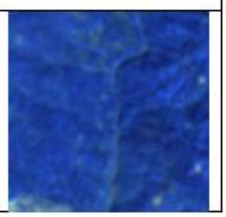
As mentioned before the results of DenseNet121 from confusion matrix and model curves has overfitting problem. It is seen that the model confused some of burnt area and active fire patches in Table 3.8.This can be explained that there are also burnt areas in active fire areas.

Table 3.8 DenseNet121 predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area			No prediction
Active fire			No prediction
No fire	No prediction	No prediction	

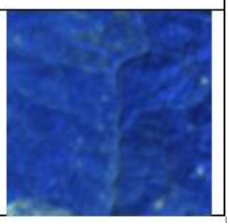
DenseNet169 predictions are represented in Table 3.9. It is seen from the table that result from accuracy and model curve match with this table, and the model overfitted the dataset.

Table 3.9 DenseNet169 predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area		No prediction	No prediction
Active fire	No prediction		No prediction
No fire	No prediction	No prediction	

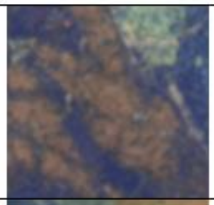
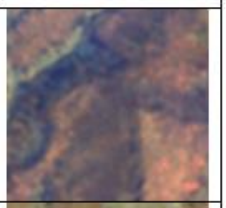
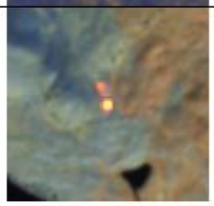
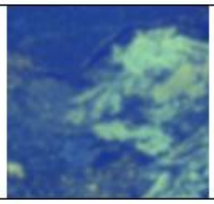
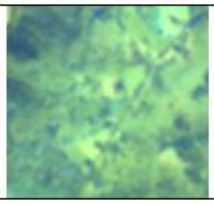
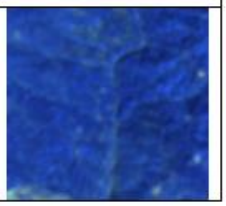
Similarly to DenseNet169, DenseNet201 didn't confuse the classes, overfitted the dataset as well. DenseNet 201 predicitions are given in Table 3.10.

Table 3.10 DenseNet201 predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area		No prediction	No prediction
Active fire	No prediction		No prediction
No fire	No prediction	No prediction	

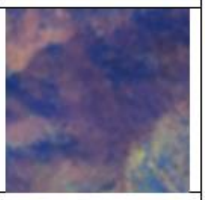
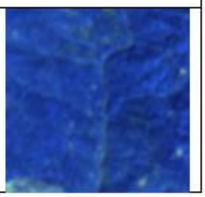
Inceptionv3 predictions are shown in Table 3.11. Except a few patches it didn't confuse the classes.

Table 3.11 Inceptionv3 predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area			
Active fire			
No fire			

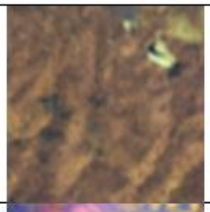
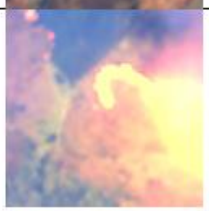
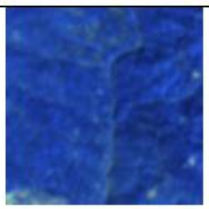
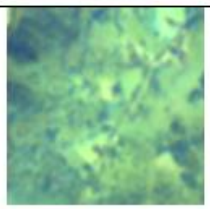
MobileNet predictions are shown in Table 3.12.The model predicted one burnt area patch as no fire patch.

Table 3.12 MobileNet predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area		No prediction	
Active fire	No prediction		No prediction
No fire	No prediction	No prediction	

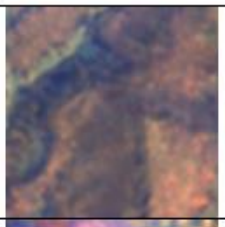
ResNet101 predictions are represented in Table 3.13. It is clear from the table that the model didn't detect no fire areas.

Table 3.13 ResNet101 predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area			No prediction
Active fire			No prediction
No fire			No prediction

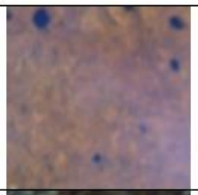
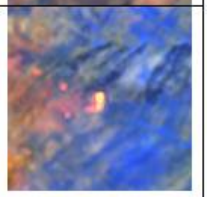
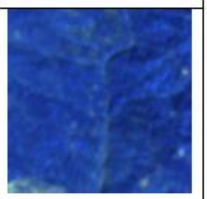
ResNet152 predicitions are given in Table 3.14.Similarly to ResNet101 the model couldn't detect all classes it only detected active fire patches.

Table 3.14 ResNet152 predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area	No prediction		No prediction
Active fire	No prediction		No prediction
No fire	No prediction		No prediction

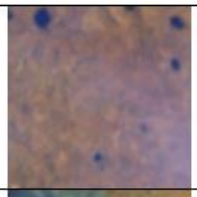
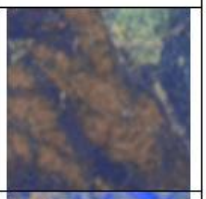
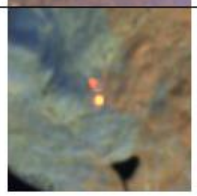
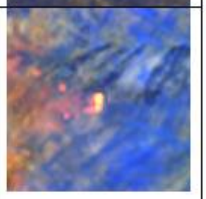
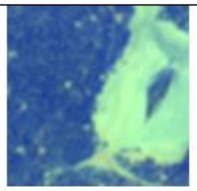
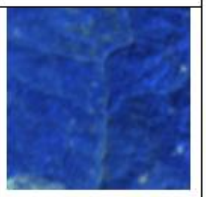
VGG16 predictions are shown in Table 3.15. The model performed well on the dataset, and detected most of the patches.

Table 3.15 VGG16 predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area			
Active fire			
No fire			

VGG19 predictions are given in Table 3.16. The model performed similarly to VGG16 and detected most of the patches.

Table 3.16 VGG19 predictions

Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area			
Active fire			
No fire			

Xception predictions are given in Table 3.17.The model generalized well.

Table 3.17 Xception predictions

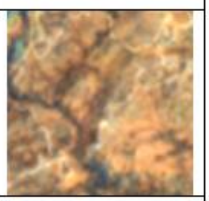
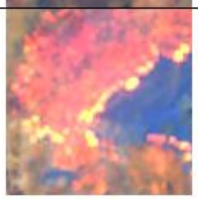
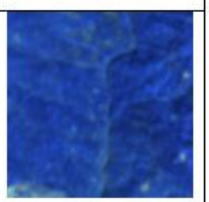
Reference/ Predicted	Burnt area	Active fire	No fire
Burnt area		No prediction	
Active fire			No prediction
No fire		No prediction	

Fig. 3.13 shows the deep learning model curves.

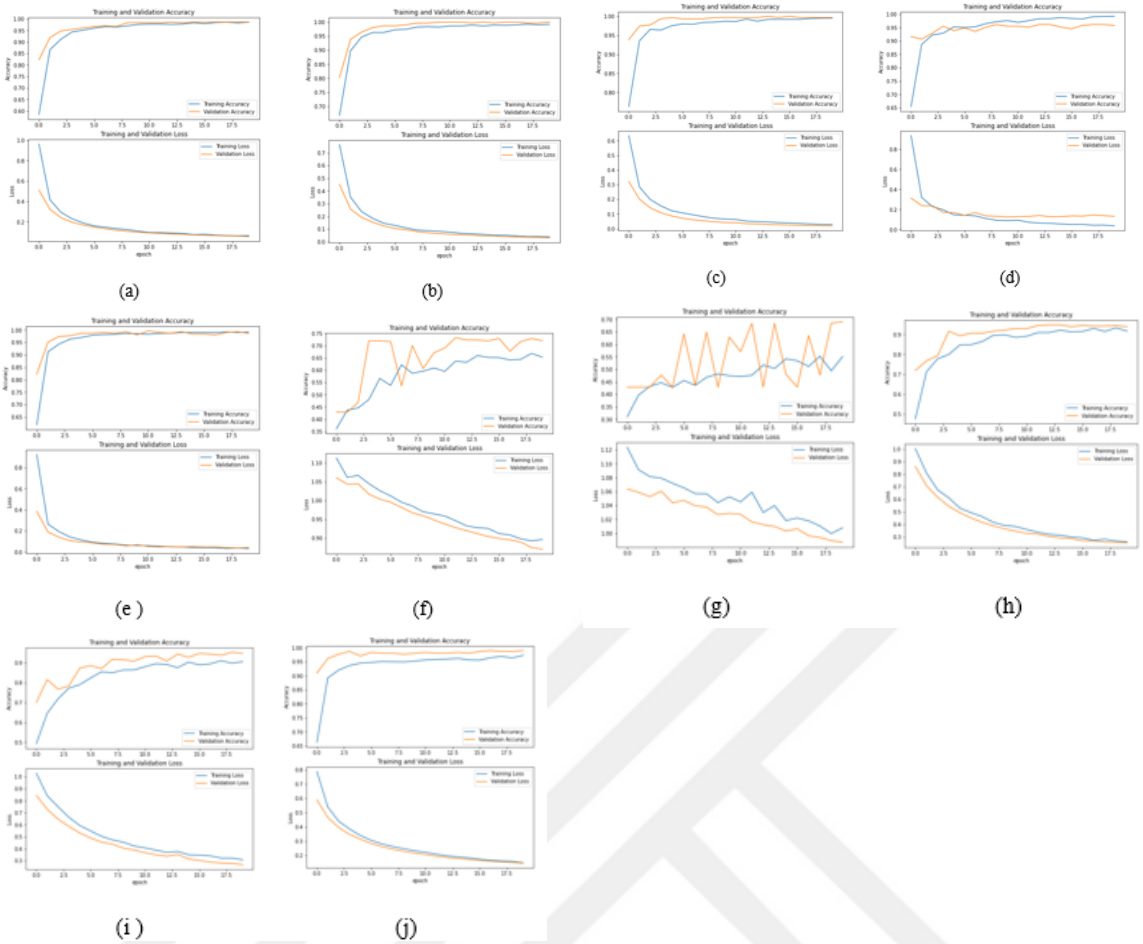


Figure 3.13 Model curves of each architecture.(a):DenseNet 121,(b):DenseNet169, (c):DenseNet201, (d):Inceptionv3, (e):MobileNet,(f) :ResNet101, (g):ResNet152, (h):VGG16, (i):VGG19, (j)Xception

4
RESULTS AND DISCUSSION

The objective of this thesis as stated in the introduction was to use deep neural networks which have different types of weight transfer mechanisms to classify active-fire, no-fire, and burnt forest areas from optical and SAR data. To this end, a manually labeled dataset was generated using Sentinel-2 images. VGG16, VGG19, Xception, ResNet101, ResNet152, MobileNet, Inceptionv3, DenseNet121 DenseNet169, and DenseNet201 deep learning architectures were used to detect fire, no-fire, and burnt areas. Among the 10 deep neural networks, Xception architecture was outperformed with an accuracy of 98%. It is thought that deep neural architectures are capable of classifying active fire, burnt area, and no-fire areas. Sentinel-1 images could not be used to detect active fires since they could not be manually labeled. The labeling process was quite good. As a result of this there are a few networks which can train very less data. Avoiding overlapping labels could be optimized. Labeling more data would also significantly lead to better results.

BIBLIOGRAPHY

[1] United Nations Environment Programme, “Spreading like Wildfire – The Rising Threat of Extraordinary Landscape Fires,” Nairobi, 2022. [Online]. Available: https://www.unep.org/resources/report/spreading-wildfire-rising-threat-extraordinary-landscape-fires.

[2] A. N. Ross, M. J. Wooster, H. Boesch, and R. Parker, “First satellite measurements of carbon dioxide and methane emission ratios in wildfire plumes,” *Geophys. Res. Lett.*, vol. 40, no. 15, pp. 4098–4102, 2013, doi: 10.1002/grl.50733.

[3] S. C. P. Coogan, F. N. Robinne, P. Jain, and M. D. Flannigan, “Scientists’ warning on wildfire — a canadian perspective,” *Can. J. For. Res.*, vol. 49, no. 9, pp. 1015–1023, 2019, doi: 10.1139/cjfr-2019-0094.

[4] D. Bowman *et al.*, “Wildfires: Australia needs national monitoring agency,” *Nature*, vol. 584, no. 7820, pp. 188–191, 2020, doi: 10.1038/d41586-020-02306-4.

[5] G. Canon, “What the numbers tells us about a catastrophic year of wildfires,” *The Guardian*, Dec. 25, 2021. [Online]. Available: https://www.theguardian.com/us-news/2021/dec/25/what-the-numbers-tells-us-about-a-catastrophic-year-of-wildfires.

[6] Q. Yuan *et al.*, “Deep learning in environmental remote sensing: Achievements and challenges,” *Remote Sens. Environ.*, vol. 241, no. February, p. 111716, 2020, doi: 10.1016/j.rse.2020.111716.

[7] P. Jain, S. C. P. Coogan, S. G. Subramanian, M. Crowley, S. Taylor, and M. D. Flannigan, “A review of machine learning applications in wildfire science and management,” *Environ. Rev.*, vol. 28, no. 4, pp. 478–505, 2020, doi: 10.1139/er-2020-0019.

[8] E. Chuvieco and A. Huete, *Fundamentals of satellite remote sensing*, 1st Editio. Boca Raton: CRC Press, 2009. doi: 10.1201/b18954.

[9] E. Chuvieco, I. Aguado, J. Salas, M. García, M. Yebra, and P. Oliva, “Satellite Remote Sensing Contributions to Wildland Fire Science and Management,” *Curr. For. Reports*, vol. 6, no. 2, pp. 81–96, 2020, doi: 10.1007/s40725-020-00116-5.

[10] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep learning in remote sensing applications: A meta-analysis and review,” *ISPRS J. Photogramm. Remote Sens.*, vol. 152, no. November 2018, pp. 166–177, 2019, doi: 10.1016/j.isprsjprs.2019.04.015.

[11] M. J. Sousa, A. Moutinho, and M. Almeida, “Wildfire detection using transfer learning on augmented datasets,” *Expert Syst. Appl.*, vol. 142, 2020, doi: 10.1016/j.eswa.2019.112975.

[12] Z. Hong *et al.*, “Active Fire Detection Using a Novel Convolutional Neural Network Based on Himawari-8 Satellite Images,” *Front. Environ. Sci.*, vol. 10, no. March, pp. 1–11, 2022, doi: 10.3389/fenvs.2022.794028.

[13] S. N. Hirsch, “Application of infrared scanners to forest fire detection,” 1971. doi: 19730007663.

[14] M. Matson and J. Dozier, “Identification of subresolution high temperature sources using a thermal IR sensor.,” *Photogramm. Eng. Remote Sensing*, vol. 47, no. 9, pp. 1311–1318, 1981.

[15] L. Giglio, J. Descloitres, C. O. Justice, and Y. J. Kaufman, “An enhanced contextual fire detection algorithm for MODIS,” *Remote Sens. Environ.*, vol. 87, no. 2–3, pp. 273–282, 2003, doi: 10.1016/S0034-4257(03)00184-6.

[16] Y. J. Kaufman *et al.*, “Potential global fire monitoring from EOS-MODIS,” *J. Geophys. Res. Atmos.*, vol. 103, no. D24, pp. 32215–32238, 1998, doi: 10.1029/98JD01644.

[17] L. Giglio, W. Schroeder, and C. O. Justice, “The collection 6 MODIS active fire detection algorithm and fire products,” *Remote Sens. Environ.*, vol. 178, pp. 31–41, 2016, doi: 10.1016/j.rse.2016.02.054.

[18] W. Schroeder, P. Oliva, L. Giglio, B. Quayle, E. Lorenz, and F. Morelli, “Active fire detection using Landsat-8/OLI data,” *Remote Sens. Environ.*, vol. 185, pp. 210–220, 2016, doi: 10.1016/j.rse.2015.08.032.

[19] S. W. Murphy, C. R. de Souza Filho, R. Wright, G. Sabatino, and R. Correa Pabon, “HOTMAP: Global hot target detection at moderate spatial resolution,” *Remote Sens. Environ.*, vol. 177, pp. 78–88, 2016, doi: 10.1016/j.rse.2016.02.027.

[20] S. S. Kumar and D. P. Roy, “Global operational land imager Landsat-8 reflectance-based active fire detection algorithm,” *Int. J. Digit. Earth*, vol. 11, no. 2, pp. 154–178, 2018, doi: 10.1080/17538947.2017.1391341.

[21] G. H. de Almeida Pereira, A. M. Fusioka, B. T. Nassu, and R. Minetto, “Active fire detection in Landsat-8 imagery: A large-scale dataset and a deep-learning study,” *ISPRS J. Photogramm. Remote Sens.*, vol. 178, pp. 171–186, 2021, doi: 10.1016/ j.isprsjprs.2021.06.002.

[22] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, no. Cvd, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[23] N. T. Toan, P. Thanh Cong, N. Q. Viet Hung, and J. Jo, “A deep learning approach for early wildfire detection from hyperspectral satellite images,” *2019 7th Int. Conf. Robot Intell. Technol. Appl. RiTA 2019*, pp. 38–45, 2019, doi: 10.1109/ RITAPP.2019.8932740.

[24] H. Huang *et al.*, “Separability analysis of Sentinel-2A Multi-Spectral Instrument (MSI) data for burned area discrimination,” *Remote Sens.*, vol. 8, no. 10, 2016, doi: 10.3390/rs8100873.

[25] G. Navarro, I. Caballero, G. Silva, P. C. Parra, Á. Vázquez, and R. Caldeira, “Evaluation of forest fire on Madeira Island using Sentinel-2A MSI imagery,” *Int. J. Appl. Earth Obs. Geoinf.*, vol. 58, pp. 97–106, 2017, doi: 10.1016/j.jag.2017.02.003.

[26] V. Mithal, G. Nayak, A. Khandelwal, V. Kumar, R. Nemani, and N. C. Oza, “Mapping burned areas in tropical forests using a novel machine learning framework,” *Remote Sens.*, vol. 10, no. 1, pp. 1–22, 2018, doi: 10.3390/rs10010069.

[27] E. Roteta, A. Bastarrika, M. Padilla, T. Storm, and E. Chuvieco, “Development of a Sentinel-2 burned area algorithm: Generation of a small fire database for sub-Saharan Africa,” *Remote Sens. Environ.*, vol. 222, no. November 2018, pp. 1–17, 2019, doi: 10.1016/j.rse.2018.12.011.

[28] M. Gargiulo, D. A. G. Dell’aglio, A. Iodice, D. Riccio, and G. Ruello, “A CNN-based Super-resolution Technique for Active Fire Detection on Sentinel-2 Data,” *Prog. Electromagn. Res. Symp.*, vol. 2019-June, no. July 2017, pp. 418–426, 2019, doi: 10.1109/PIERS-Spring46901.2019.9017857.

[29] Z. Langford, J. Kumar, and F. Hoffman, “Wildfire mapping in interior alaska using deep neural networks on imbalanced datasets,” *IEEE Int. Conf. Data Min. Work. ICDMW*, vol. 2018-Novem, pp. 770–778, 2019, doi: 10.1109/ICDMW.2018.00116.

[30] F. Filipponi, “Exploitation of sentinel-2 time series to map burned areas at the national level: A case study on the 2017 Italy wildfires,” *Remote Sens.*, vol. 11, no. 6, 2019, doi: 10.3390/rs11060622.

[31] A. Farasin, L. Colomba, and P. Garza, “Double-step U-Net: A deep learning-based approach for the estimation ofwildfire damage severity through sentinel-2 satellite data,” *Appl. Sci.*, vol. 10, no. 12, pp. 1–22, 2020, doi: 10.3390/app10124332.

[32] P. P. de Bem, O. A. de Carvalho Júnior, O. L. F. de Carvalho, R. A. T. Gomes, and R. F. Guimarães, “Performance analysis of deep convolutional autoencoders with different patch sizes for change detection from burnt areas,” *Remote Sens.*, vol. 12, no. 16, pp. 1–19, 2020, doi: 10.3390/RS12162576.

[33] Z. Zhang, Q. Liu, and Y. Wang, “Road Extraction by Deep Residual U-Net,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 749–753, 2018, doi: 10.1109/ LGRS.2018.2802944.

[34] L. Knopp, M. Wieland, M. Rättich, and S. Martinis, “A deep learning approach for burned area segmentation with Sentinel-2 data,” *Remote Sens.*, vol. 12, no. 15, 2020, doi: 10.3390/RS12152422.

[35] M. M. Pinto, R. Libonati, R. M. Trigo, I. F. Trigo, and C. C. DaCamara, “A deep learning approach for mapping and dating burned areas using temporal sequences of satellite images,” *ISPRS J. Photogramm. Remote Sens.*, vol. 160, no. July 2019, pp. 260–274, 2020, doi: 10.1016/j.isprsjprs. 2019.12.014.

[36] Y. Ban, P. Zhang, A. Nascetti, A. R. Bevington, and M. A. Wulder, “Near Real-Time Wildfire Progression Monitoring with Sentinel-1 SAR Time Series and Deep Learning,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–15, 2020, doi: 10.1038/s41598-019-56967-x.

[37] D. Stavrakoudis, T. Katagis, C. Minakou, and I. Z. Gitas, “Automated Burned Scar Mapping Using Sentinel-2 Imagery,” *J. Geogr. Inf. Syst.*, vol. 12, no. 03, pp. 221–240, 2020, doi: 10.4236/jgis.2020.123014.

[38] M. A. Belenguer-Plomer, M. A. Tanase, E. Chuvieco, and F. Bovolo, “CNN-based burned area mapping using radar and optical data,” *Remote Sens. Environ.*, vol. 260, p. 112468, 2021, doi: 10.1016/j.rse.2021.112468.

[39] X. Hu, Y. Ban, and A. Nascetti, “Sentinel-2 MSI data for active fire detection in major fire-prone biomes: A multi-criteria approach,” *Int. J. Appl. Earth Obs. Geoinf.*, vol. 101, p. 102347, 2021, doi: 10.1016/j.jag.2021.102347.

[40] D. Rashkovetsky, F. Mauracher, M. Langer, and M. Schmitt, “Wildfire Detection from Multisensor Satellite Imagery Using Deep Semantic Segmentation,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 7001–7016, 2021, doi: 10.1109/JSTARS.2021.3093625.

[41] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.

[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[43] R. Ghali, M. A. Akhloufi, and W. S. Mseddi, “Deep Learning and Transformer Approaches for UAV-Based Wildfire Detection and Segmentation,” *Sensors*, vol. 22, no. 5, pp. 1–18, 2022, doi: 10.3390/s22051977.

[44] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[45] California Department of Forestry and Fire Protection (CAL FIRE), “Dixie Fire Incident Report,” 2022. https://www.fire.ca.gov/incidents/2021/7/13/dixie-fire/.

[46] California Department of Forestry and Fire Protection (CAL FIRE), “Top 20 Largest California Wildfires,” 2022. [Online]. Available: https://www.fire.ca.gov/media/4jandlhh/top20_acres.pdf.

[47] K. Alexander, “Last year’s fire season in California set record for cost, Dixie Fire most expensive in U.S. history,” *San Francisco Chronicle*, 2022. [Online]. Available: https://www.sfchronicle.com/bayarea/article/Last-year-s-fire-season-in-California-set-17010110.php.

[48] QGIS Development Team.Open Source Foundation., “QGIS Geographic Information System.” 2009. [Online]. Available: http://qgis.org/.

[49] “Copernicus Open Access Hub.” https://scihub.copernicus.eu/.

[50] N. Yague-Martinez *et al.*, “Interferometric Processing of Sentinel-1 TOPS Data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 4, pp. 2220–2234, 2016, doi: 10.1109/TGRS.2015.2497902.

[51] M. Aulard-Macler, “Sentinel-1 Product Definition S1-RS-MDA-52-7440,” 2011.

[52] “Sentinel-1 Observation Scenario,” 2021. https://sentinel.esa.int/web/sentinel/missions/sentinel-1/observation-scenario.

[53] “SNAP - ESA Sentinel Application Platform.” http://step.esa.int.

[54] M. Drusch *et al.*, “Sentinel-2: ESA’s Optical High-Resolution Mission for GMES Operational Services,” *Remote Sens. Environ.*, vol. 120, pp. 25–36, 2012, doi: 10.1016/j.rse.2011.11.026.

[55] European Space Agency (ESA), “Sentinel-2 MSI Radiometric Resolutions.” https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric.

[56] F. Chollet, “Keras,” 2015. https://keras.io.

[57] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[58] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[59] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.

[60] J. Patterson and A. Gibson, *Deep learning: A practitioner’s approach*. O’Reilly Media, Inc., 2017.

[61] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[64] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[65] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[66] M. Lin, Q. Chen, and S. Yan, “Network in network,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, pp. 1–10, 2014.

[67] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[68] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[69] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017, [Online]. Available: http://arxiv.org/abs/ 1704.04861.

[70] P. Pedregosa, F; Varoquaux, G; Gramfort, A; Michel, V; Thirion, B; and Grisel, O. and Blondel, . and Prettenhofer *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, 2011. http://arxiv.org/abs/ 1201.0490.

[71] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 437–478, 2012, doi: 10.1007/978-3-642-35289-8_26.

PUBLICATIONS FROM THE THESIS

Conference Papers

[1] D. Denizoglu, F. Sanli, M. Werner, (2022), “The Role of Synthetic Aperture RADAR for Observing Wildfires”, *The Turkish National Photogrammetry and Remote Sensing Association Symposium*, Mersin.