FOREST FIRE DETECTION VIA CAMERAS MOUNTED ON UNMANNED AERIAL VEHICLES

İNSANSIZ HAVA ARAÇLARINDA BULUNAN KAMERALAR KULLANILARAK ORMAN YANGINLARININ TESPİT EDİLMESİ

RAHMİ ARDA ARAL

ASST. PROF. DR. CEMIL ZALLUHOĞLU
Supervisor
PROF. DR. EBRU AKÇAPINAR SEZER
2nd Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Computer Engineering

June 2023

ABSTRACT

FOREST FIRE DETECTION VIA CAMERAS MOUNTED ON UNMANNED AERIAL VEHICLES

Rahmi Arda Aral
Master of Science, Computer Engineering
Supervisor: Asst. Prof. Dr. Cemil ZALLUHOĞLU
2nd Supervisor: Prof. Dr. Ebru AKÇAPINAR SEZER
June 2023, 66 pages

Unmanned aerial vehicles(UAVs) are invaluable technologies thanks to their remote control and monitoring capabilities. Operational forces and firefighters use UAVs in wildfire detection missions. Due to their high pattern recognition capabilities, Convolutional Neural Networks (CNNs) are one of the most prominent deep learning architectures, making them proper for the task of forest fire recognition using UAVs. Deep convolutional neural networks can perform effectively on hardware with high processing capability. While these networks can be easily operated in unmanned aerial vehicles managed from ground control stations with GPU-supported hardware, lightweight, small-sized, and efficient networks are required to execute on a typical UAV's limited computational resources. To overcome these impediments, this thesis presents comprehensive research for performing forest fire detection tasks using UAV vision data with deep and lightweight convolutional neural networks.

In this thesis, experiments have been carried out using well-known convolutional neural network architectures to achieve the most successful approach. We also performed transfer learning on several further models. In addition, convolutional neural network architectures have been modified by adding an attention mechanism to develop models with high accuracy.

Among the experimented models, the attention-based EfficientNetB0 backboned model emerged as the most successful architecture.

With the test accuracy of 92.02% in the FLAME dataset and the test accuracy of 99.76% in the infrared dataset, the addition of a layer of attention in detecting forest fire strongly reinforces the efficiency of the EfficientNetB0 based model.

Keywords: Forest Fires, UAV Imagery, Deep Learning, Fire Detection, Wildfire Classification, Attention, Transfer Learning

ÖZET

İNSANSIZ HAVA ARAÇLARINDA BULUNAN KAMERALAR KULLANILARAK ORMAN YANGINLARININ TESPİT EDİLMESİ

Rahmi Arda Aral

Yüksek Lisans, Bilgisayar Mühendisliği

Danışman: Asst. Prof. Dr. Cemil ZALLUHOĞLU

Eş Danışman: Prof. Dr. Ebru AKÇAPINAR SEZER

Haziran 2023, 66 sayfa

İnsansız hava araçları (İHA'lar), uzaktan kontrol ve izleme yetenekleri sayesinde paha biçilmez teknolojilerdir. Operasyonel kuvvetler ve itfaiyeciler, orman yangını tespit görevlerinde İHA'ları kullanır. En popüler derin öğrenme mimarilerinden biri olan evrişimli sinir ağları (CNN), yüksek örüntü tanıma yetenekleri nedeniyle insansız hava araçlarında bulunan kameralar ile elde edilen görüntüler aracılığıyla orman yangını tespitinde kullanılabilir. Derin evrişimli sinir ağları, yüksek işlem kabiliyetine sahip donanımlarda etkin bir şekilde çalışabilir. Bu ağlar, GPU destekli donanıma sahip yer kontrol istasyonlarından yönetilen insansız hava araçlarında kolaylıkla çalıştırılabilirken, tipik bir İHA'nın sınırlı hesaplama kaynakları üzerinde çalıştırılabilmesi için hafif, küçük boyutlu ve verimli ağlar gerekir. Bu engellerin üstesinden gelmek için bu tez, derin ve hafif evrişimli sinir ağları ile İHA görüş verilerini kullanarak orman yangını algılama görevlerini gerçekleştirmeye yönelik kapsamlı bir araştırma sunmaktadır.

Bu tezde, en başarılı yaklaşımı elde etmek için iyi bilinen evrişimli sinir ağı mimarileri kullanılarak deneyler yapılmıştır. Ayrıca, belirlenen modellerde transfer öğrenimi gerçekleştirilmiştir. Ek olarak, evrişimli sinir ağı mimarileri, yüksek doğrulukta modeller geliştirmek için bir dikkat mekanizması eklenerek modifiye edilmiştir.

Denenen modeller arasında dikkat tabanlı EfficientNetB0 omurgalı model, orman yangını algılama konusunda en başarılı mimari olarak ortaya çıkmıştır.

FLAME veri setinde $92,02\%$ test doğruluğu ve kızılötesi veri setinde $99,76\%$ test doğruluğu ile, orman yangınını algılamada bir dikkat katmanının eklenmesi, EfficientNetB0 tabanlı modelin verimliliğini güçlü bir şekilde kanıtlamıştır.

Keywords: Orman Yangını, İHA'lar tarafından çekilen görüntüler, Derin Öğrenme, Yangın Tespiti, Orman Yangını Sınıflandırması, Dikkat katmanı, Transfer Öğrenme

ACKNOWLEDGEMENTS

First and foremost, I would like to thank to my supervisors Prof. Dr. Ebru AKÇAPINAR SEZER and Asst. Prof. Dr. Cemil ZALLUHOĞLU for their guidance, support, encouragements and toleration throughout the preparation of this thesis.

Furthermore, I would like to thank my thesis committee members, Assoc. Prof. Dr. Ahmet Burak CAN and Assoc. Prof. Dr. Mehmet Serdar GÜZEL for reviewing my thesis and their valuable comments.

I would like to express my heartfelt gratitude to my parents, Nükhet and İbrahim ARAL, for their unwavering support and encouragement throughout my educational journey. Their constant belief in me, unwavering trust, and unconditional love have been the driving force behind my accomplishments. I am truly grateful for their sacrifices, guidance, and endless encouragement. Words cannot fully convey the depth of my appreciation for everything they have done for me. Thank you, Nükhet and İbrahim ARAL, for being my pillars of strength and for always standing by my side.

I would like to express my sincere gratitude to Assoc. Prof. Dr. Murat HACIÖMEROĞLU, who has been an invaluable person in my life and provided unwavering support. Thank you, Murat HACIÖMEROĞLU, for your remarkable contributions and belief in my potential.

I would like to thank Bedri Ege ÖZTÜRK for for always being by my side and supporting me. I am deeply thankful for him unwavering friendship and support.

Finally, I would also like to thank Turkish Aerospace Industries (TAI) UAV Systems Software Engineering team for their contributions.

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

ANN	:	**A**rtificial **N**eural **N**etwork
CIFAR-100	:	**C**anadian **I**nstitue **F**or **A**dvanced **R**esearch-**1**00 **C**lasses
CNN	:	**C**onvolutional **N**eural **N**etwork
CPU	:	**C**entral **P**rocessing **U**nit
Conv2D	:	**2 D**imensional **C**onvolution
DepthwiseConv2D	:	**D**epthwise **2 D**imensional **C**onvolution
EO/IR	:	**E**lectro **O**ptic **I**nfrared
FLAME	:	**F**ire **L**uminosity **A**irborne-based **M**achine **L**earning **E**valuation
FPS	:	**F**rame **P**er **S**econd
F1	:	**F**1 **S**core
GPU	:	**G**raphics **P**rocessing **U**nits
MALE	:	**M**edium **A**ltitude **L**ong **E**ndurance
MBConv	:	**M**obile **I**nverted **B**ottleneck **C**onvolution
RGB	:	**R**ed **G**reen **B**lue
UAV	:	**U**nmanned **A**erial **V**ehicle
USA	:	**U**nited **S**tates **O**f **A**merica

1. INTRODUCTION

Forest fires are among the most devastating disasters for the environment. They not only threaten the lives of people and other critters but also pose a threat to the ecological balance. Wildfires cause proliferation in the number of other natural disasters such as erosion, flooding, and air pollution besides [1]. Due to climate change and global warming, forest fire seasons have extended, and the impact areas of wildfires have expanded [2–4]. They have occurred more frequently in various parts of the world like the USA, Australia, and Eastern and South-Eastern Europe [5, 7, 8]. In addition, wildfires have started to occur even in regions where forest fires have never happened in the past [2]. Operational forces and firefighters leverage technology to avert devastating damage from wildfires. Fire detection systems occupy an important place among these technologies. Smoke and heat sensors are technological devices used in detection of forest fires. However, using the sensors unaidedly is insufficient because of the limitations, such as limited area coverage and delayed response [9, 10]. Satellite images are also commonly used for fire detection and monitoring. Nevertheless, the detection of wildfires in early state continues to be a challenging task primarily due to the extended scan time and limited resolution [11, 12].

Unmanned aerial vehicles (UAVs) are valuable technologies with remote mission and autonomous flight capabilities [13]. They are also highly suitable in the field of wildfire detection, by dint of the cameras equipped with them [14]. Furthermore, UAVs are able to operate effectively day and night, thanks to high-resolution cameras mounted as payloads[15]. Progress in computer vision methods has made it probable to recognize forest fires utilizing aerial images captured by UAV cameras [14, 16, 17]. CNNs, one of the most effectual deep learning architectures, have been utilized for detecting wildfires using aerial images [14]. Using the images obtained with UAV cameras, forest fires can be detected with the benefit of CNNs, which are effectual deep learning architectures that give highly accurate results [18–20]. However, this is a real challenge as unmanned aerial vehicles can fly at distinct altitudes and fire has a variety of colors and textures.

Heat-sensitive infrared cameras are particularly effective at detecting forest fires in the early stages, where there is no visible smoke or flame [21, 22]. They provide a clear field of view for UAVs observing day and night, even at high altitudes, even in rural areas with little or no illumination. UAVs also perform forest fire detection tasks via these high-resolution cameras [21, 22]. This task is usually accomplished through human observation at ground control stations.

In the thesis. we suggest deep learning approaches to enhance forest fire detection task, traditionally carried out via human observation. In the first part of this study, infrared real forest fire images obtained from the MALE (medium altitude long endurance) class Turkish Aerospace Industries AKSUNGUR UAV EO/IR cameras were used as the dataset [6]. We conducted experimentations utilizing state-of-the-art CNNs to achieve optimal results. ResNet101 [24], ResNet50 [24], VGG16 [26], Densenet121 [27],EfficientNetB1 [25] and EfficientNetB0 [25] based architectures experimented on the infrared dataset.

The task of fire detection with UAVs is carried out not only with EO/IR cameras but also with RGB cameras. For this reason, experiments were carried out using convolutional neural networks for forest fire recognition with UAVs equipped with RGB cameras.

We performed second part of our work using the FLAME dataset UAV captured RGB images [23]. We carried out experiments with deep and lightweight convolutional neural networks. In the experiments, EfficientNetB0 [25], EfficientNetB2 [25], EfficientNetB4 [25], XCeption [28], ResNet50 [24], VGG16 [26], NASNet Mobile [32], MobileNetV3Small [31], and MobileNetV2 [30] networks were used as feature extractors for the architectures we determined.

In the first part of the FLAME Dataset experimentations, we performed experiments using the architectures. Then, we conducted experiments using ImageNet weights to improve the wildfire recognition capability of the experimented networks. In the last part, we modified the networks we used as feature extractors by adding the channel-based attention layer to achieve better results.

Among the experiments, for the FLAME dataset, test accuracy of 92.02% and test accuracy of 99.76% for the private infrared dataset strongly support the efficiency of the attention layer added EfficientNetB0 model in detecting wildfire from the image samples presented.

In this thesis, a comprehensive study was carried out using CNNs on images obtained from infrared and RGB UAV cameras. We also present a lightweight and effective model that can accurately detect forest fires using infrared and RGB images.

1.1. Contributions

This thesis addresses these constraints by presenting effective approaches. The primary contributions of the thesis can be digested as follows:

- We propose a lightweight and efficient neural network for intelligent wildfire detection with UAV-collected RGB and infrared images.
- We also suggest a comprehensive study of deep learning approaches and methods with comparative results.
- Unlike most of the previous works, we made experiments with both RGB and infrared dataset.
- For the first time, we made experimantations with real infrared forest fire images collected by UAV.
- Our results show that in experiments on two different datasets, the proposed approach was performed brilliantly.

1.2. Organization

The organization of the thesis is as follows:

- Chapter 1 illustrates a piece of brief information about forest fires and their effects, technologies used for forest fire detection, the role of UAVs in forest fire detection, Convolutional Neural Networks and fire detection, and information about our motivation to work.
- Chapter 2 provides background information about artificial neural networks, convolutional neural networks, training strategies and methods we used in experimentations.
- Chapter 3 furnishes an synopsis of the related works on wildfire detection.
- Chapter 4 provides the detailed information about utilized datasets and evaluation metrics.
- Chapter 5 details the different techniques and approaches we operated for two different datasets.
- Chapter 6 illustrates the experimental setup and analyzes the outcomes of the utilized approaches and architectures.
- Chapter 7 expresses the summary of the thesis and possible future works.

2. BACKGROUND OVERVIEW

This chapter provides a brief overview of Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and the CNN architectures that were experimented with. In addition, we will give information about our training strategies and the datasets we used. Lastly, we will explain the evaluation metrics that we utilized.

2.1. Artificial Neural Networks

Artificial neural networks mimic the method by which the biological neurons signal to one another. ANNs consist of connected node layers (neurons), which include an input layer, one or more hidden layers, and an output layer. The neurons are linked to other neurons. They maintain an associated weight and threshold. Figure 2.1 provides an example of ANN.

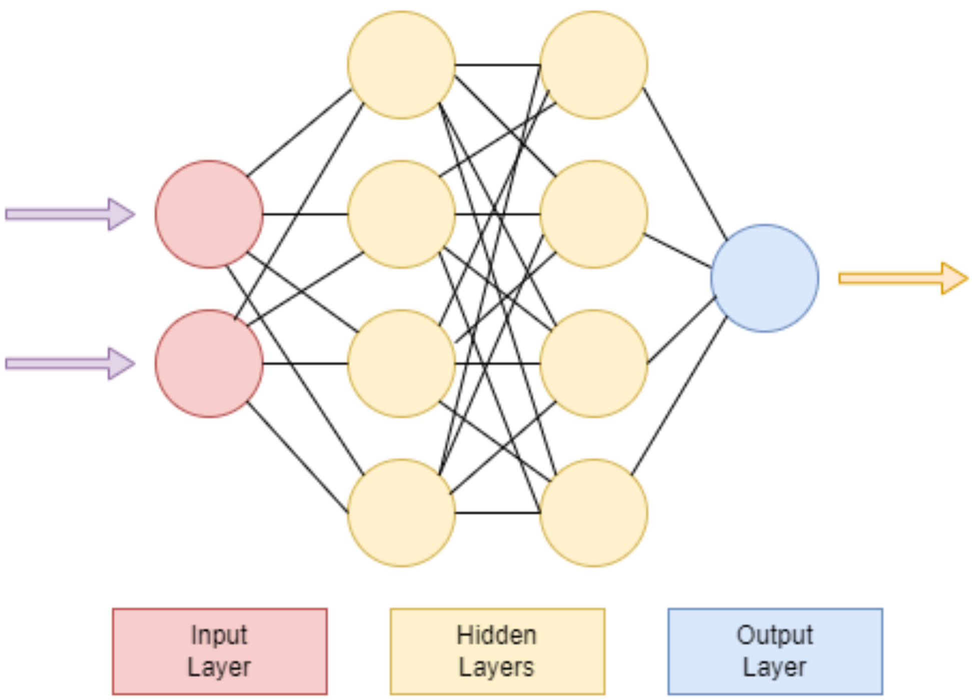

Figure 2.1 Sample Architecture of Artificial Neural Network

2.2. Convolutional Neural Networks

CNNs are specifically effective in locating patterns in images to identify objects. CNNs are differentiated from other artificial neural networks by their superior performance in the field of recognition of image, speech, or audio signals.

CNNs are composed of layers. The layers are named convolutional, non-linearity, fully connected, and pooling layers. With each layer, the CNN increases in its complexity. Earlier layers of CNN focus on straightforward features. As the image data advances via the layers of the network, it begins to identify details of the data until it eventually recognizes the planned object. For instance, in the realm of image classification, the initial layers of a CNN might detect basic features such as edges or colors. As the network goes deeper into ensuing layers, it can distinguish more complex patterns. Finally, in the later layers, CNN can recognize complicated features that contribute to the overall understanding and classification of the data. The most crucial inference concerning problems solved by CNN is that they should have spatially independent features. For example, in an object detection application, when we prepare the dataset there is no essential to pay attention to where the objects are in the images. The only point that needs to be concerned is to detect objects regardless of their location in the images.

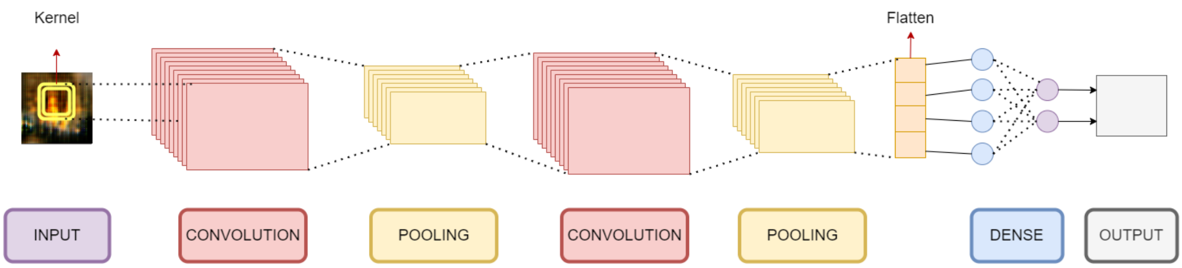

Figure 2.2 Sample Architecture of Convolutional Neural Network

The convolution layer derives its name from the mathematical operation it fulfills called convolution. The layer conducts a linear operation between matrices, allowing the network to extract significant features.

A CNN consists of numerous layers of neurons, where each layer executes a nonlinear operation on the linear transformation of the outputs from the preceding layer. The major layers in CNNs are the convolutional pooling and convolutional layers. Convolutional layers have trainable weights that are learned during the training process, enabling the network to capture appropriate features. On the other hand, pooling layers apply a specified function to transform the activation within a localized part of the image. In Figure 2.2 a sample CNN is illustrated.

2.2.1. Convolutional Layer

The convolution layers aim to detect the actuality of a set of features in images given as input data. It includes a group of filters, known as kernels, the parameters of which are learned throughout the training phase. Generally, the filter size is smaller than the image. The dimension is typically 3 × 3, but can sometimes be of different sizes, such as 5 × 5 or 7 × 7. The kernel slides over the entire input image and evaluates the dot product between the value of the input image and the weights of the kernel filters, resulting in the generation of the activation map. Thus, the network learns a visual feature.

The outputs of the convolution process, which is a linear operation, are then passed a non-linear activation function. This strategy is an integral part of the convolutional layer. The rectified linear unit (ReLU) is a widely adopted nonlinear activation function in contemporary applications. It can be mathematically represented as $f(x) = max(0, x)$, where it selectively passes positive values while filtering out negative values. ReLU smoothly computes the output value as the maximum between 0 and the input value, allowing for effective nonlinear transformations in the network. The reason for using ReLU as the activation function is converging more adequately for gradient reduction.

To overcome this issue, a procedure called padding is employed. With padding, numbers are added to the rows and columns on both sides of the input. This number is usually zero. In this way, the core of the kernel maintains its in-plane size during the convolution process.

The padding helps preserve the desired size and spatial information of the input throughout the convolution operation.

Stride directs to the spatial displacement between two consecutive kernel positions during the convolution. The commonly used stride value is 1.

2.2.2. Pooling Layer

Pooling layers virtually diminish the number of parameters to be learned within the network and minimize computational needs. In this way, the dimension of feature maps is downsampled. Unlike convolutional layers, pooling layers do not have learnable parameters. Instead, they use parameters such as filter size, stride, and padding to control the pooling process. By aggregating information within local areas, pooling layers extract important features while reducing the spatial dimensions, leading to more efficient presentations. Global average and max pooling are the most commonly utilized pooling techniques.

2.2.3. Fully Connected Layers

An array with one dimensional comes from the last pooling or convolutional layers. This array illustrates flattened features. Afterward, These features are linked to fully connected layers. The last fully connected layer, or dense layer, has output nodes equal to the number of classes.

2.2.4. Last Layer Activation Function

The selection of an activation function for the dense layers' last part should be based on the specific characteristics of the task at hand. For binary classification, the Sigmoid activation function is commonly chosen to classify binary classes. The Softmax activation function, on the other hand, is often preferred for performing classification with multiple classes.

2.3. CNN Architectures

We experimented on state-of-art architectures such as ResNet [24], EfficientNet [25], VGG16 [26], DenseNet121 [27], XCeption [28], MNASNet [32], InceptionV3 [29], MobileNetV2 [30],and MobileNetV3Small [31] as feature extractor.

2.3.1. ResNet

ResNet (Residual Network) is an Artificial Neural Network architecture designed to solve the vanishing gradient problem using residual blocks. ResNet uses the skip connections technique, which connects the activations of one layer to other layers, bypassing some layers in between. ResNet networks consist of stacked residual blocks.

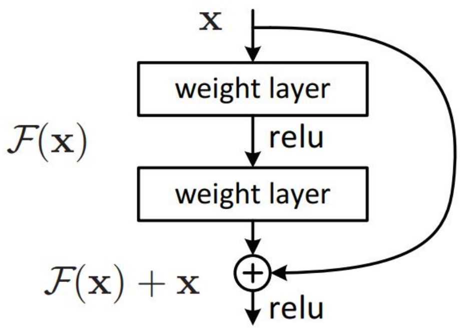

Figure 2.3 Residual Network [24]

ResNet50 has 50 of these layers, and ResNet101 has 101. With this Network, even networks with many layers can be trained without raising the training error rate[24]. Figure 2.3 shows the graphical form of Residual Network architecture. In Figure 2.3, X represents the output value that comes from the neuron from the previous layer.

2.3.2. EfficientNet

EfficientNet uniformly scales all model dimensions with a method anointed a coefficient compound. This technique provides EfficientNet models with more effective and accurate results than previously published CNN models [25].

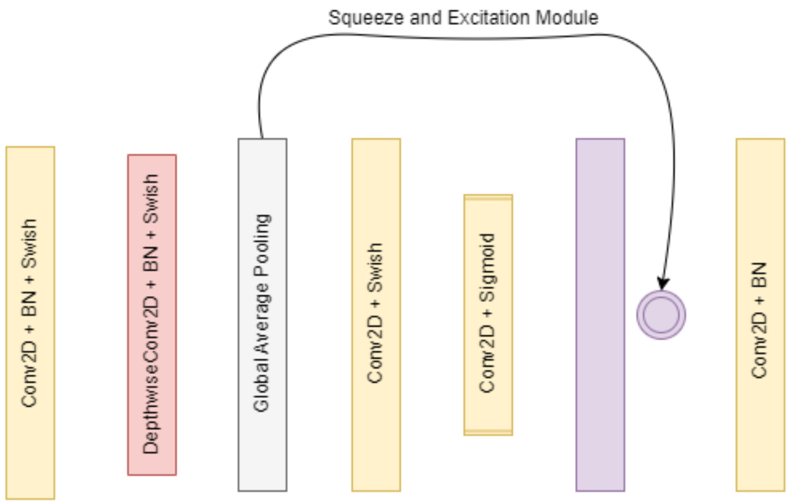

Figure 2.4 Inverted Bottleneck Convolution 1 (MBConv1) in EfficientNet

The EfficientNet family uses mobile inverted bottleneck convolution (MB Conv) as its main building block. MB Conv block contains depthwise convolution and squeeze and excitation unit. The depthwise convolution operation is efficient and requires fewer resources than a standard convolution operation. Squeeze and excitation unit brings dynamic recalibration feature to the network. In EfficientNet models, two different MBConv types are used : MBConv 1 and MBConv6. MBConv6 also has an inverted residual connection, unlike MBConv1.

MBConv1 and MBConv6 layers are demonstrated in Figure 2.4 and Figure 2.5. Conv2D stands for 2-dimensional Convolution, BN stands for Batch Normalization, and DepthwiseConv2D stands for 2-dimensional depthwise convolution. EfficientNet models utilize the inverted residual blocks introduced with MobileNetV2 [30]. However, unlike MobileNetV2, the Swish activation function is used in these modules. EfficientNet family achieves high accuracy on CIFAR-100, Flowers, and other well-known datasets [25].

2.3.3. VGG16

VGG architecture was developed to comprehend how to increase the depth of networks. It uses a 3x3 kernel in all layers to evade too many parameter sizes and make the network

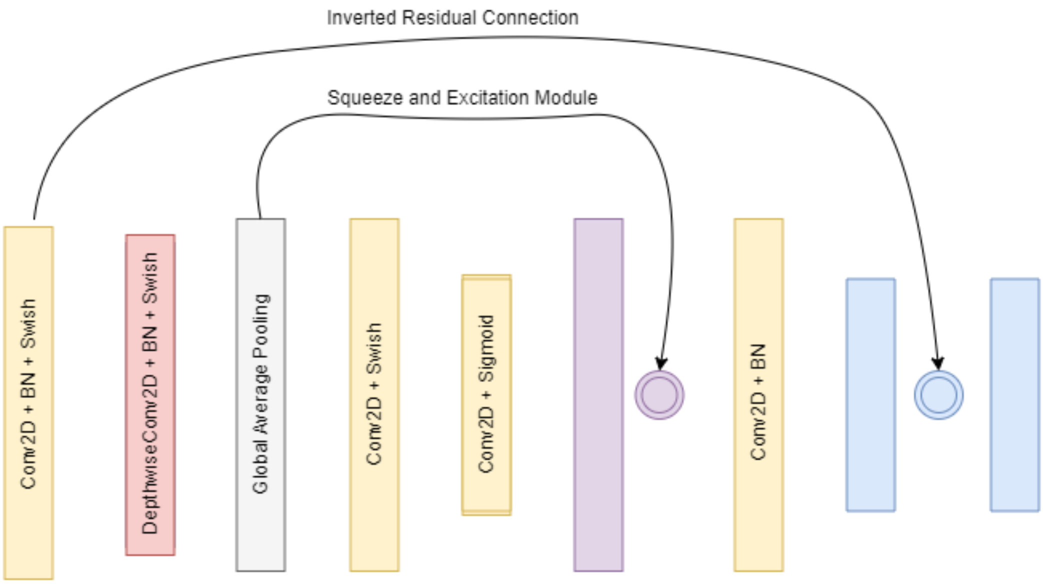

Figure 2.5 Inverted Bottleneck Convolution 6 (MBConv6) in EfficientNet

deeper[26]. VGG is a straightforward network architecture. VGG16 is specifically composed of 16 layers, which consist of 3 dense and 13 convolutional layers.

2.3.4. DenseNet

DenseNet was designed to improve the decreased accuracy induced by the vanishing gradient in neural networks. DenseNet sets dense connections between each layer and all preceding layers, resulting in a rich and various feature map set. The network is composed of multiple Transition layers and Dense Blocks in sequence. Dense blocks in DenseNet are mainly composed of a 1x1 and 3x3 convolution kernel. The transition layer in DenseNet is composed of a 1x1 convolution and pooling layer. DenseNet manages the issue of feature shrinkage by lessening the input vector dimension while possessing rich feature representations. This reduction in dimensionality not only decreases the number of parameters in the network but enhances feature propagation and enables feature reuse [27].

2.3.5. XCeption

Xception is a type of CNN architecture that maintains Deeply Separable Convolutions. It is a variation of the Inception network. The primary distinction between Inception and Xception lies in their convolutional approach: Xception utilizes depth-wise separable convolutions [28].

2.3.6. InceptionV3

InceptionV3 network incorporates several beneficial features. InceptionV3 architecture comprises three main components: the convolutional block, the improved Inception module, and the classifier. The convolutional layer is used as the feature extractor. The Improved Inception layer executes multi-scale convolutions in parallel and combines the convolution results of each branch [29].

2.3.7. MobileNetV2

MobileNetV2 is designed for good performance on mobile devices. The architecture uses inverted residual blocks. Unlike traditional Residual Blocks, Inverted Residual Blocks have a narrow, wide, narrow approach rely on the amount of channels. In these blocks, the parameter size is reduced by first applying 1x1 convolution and then 3x3 depthwise convolution. Then, 1x1 convolution is additionally applied to add inputs and outputs. These blocks consist of shortcut connections between the thin bottleneck layers. The architecture design contains MBConv blocks [30]. These blocks use the hard swish activation function and include squeeze-and-excitation modules. The comparison of inverted and traditional inverted residual block is showed in Figure 2.6.

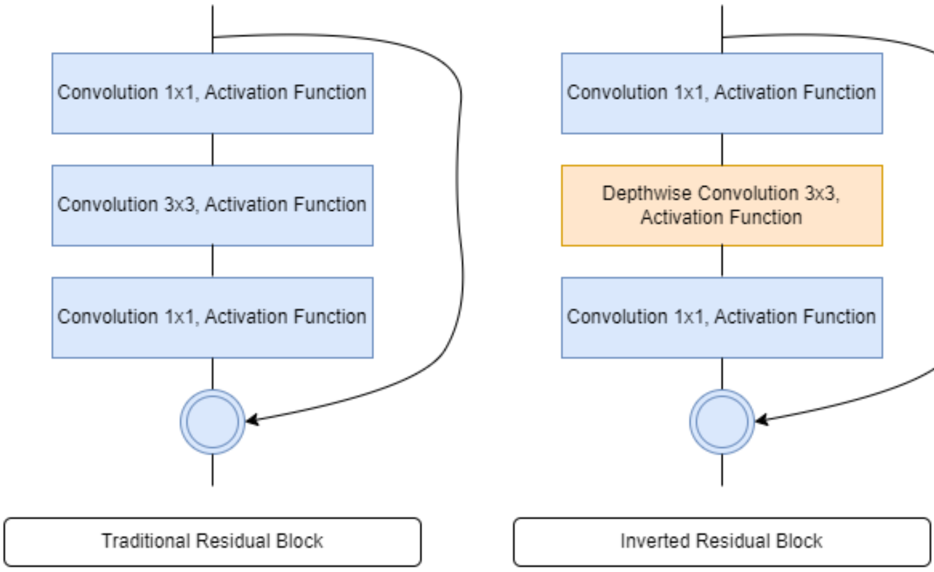

Figure 2.6 Comparision of Residual Block and Inverted Residual Block

2.3.8. MobileNetV3-Small

MobileNetV3-Small is among the MobileNetV3 architectures along with MobileNetV3 Large [31]. It is developed to work effectively on mobile device CPUs. Its architecture design contains MBConv blocks. These blocks use hard swish activation function and include squeeze-and-excitation modules [31].

2.3.9. NASNet Mobile

MNASNet is a CNN architecture optimized for mobile devices and strikes a good balance between accuracy and speed [32]. It uses the inverted residual blocks introduced with MobileNetV2 [30, 32].

3. RELATED WORK

Many researchers to date have handled forest fire detection rely on computer vision approaches. Researchers aimed to develop the most effective fire detection system using various methods. Previous forest fire detection studies relied on hand-crafted image processing techniques based on strong discriminators of fire, such as motion and color cues [38–45]. Among these studies, similar to this work, Dios et al. used infrared and RGB images collected by UAV cameras to detect fire pixels [39].

Contrary to hand-crafted image processing techniques, deep learning strategies can learn complex features [46]. Recent studies have shown that CNNs, which are deep learning architectures that provide accurate results, have higher accuracy than conventional methods in image and video recognition applications [47]. For this reason, most researchers have worked in the field of fire classification tasks using CNNs to achieve higher accuracy. Kim et al. proposed an eight-layer CNN for fire classification and used large forest fire aerial images[48]. In this work [49], researchers experimented on well-known CNN architectures like VGG13, AlexNet and GoogLeNet. They utilized a dataset contains UAV-captured images [49]. However, the auhors did not utilize any benchmark fire dataset. Dunnings et al. suggested an InceptionV1 model optimized for fire classification [50]. In this work, support vector machine and CNN classifiers were compared using a data set consisting of infrared images [51]. As a result of this comparison, the CNN classifier achieved higher accuracy [51]. Chen et al. [52] performed experiments on CNNs with a private and limited dataset captured from UAV cameras [52]. Zhao et al.[53] compared the Fire$n$_Net network they designed in the fire classification experiments with models such as Kim et al.'s proposed model[52], AlexNet, and an eight-layer CNN. Muhammad et al. provided a sophisticated comparison with CNNs using benchmark datasets and proposed a lightweight modified MobileNet-based model for fire detection tasks [54]. In another study, we found [55], the authors presented a lightweight CNN called FireNet for IoT applications [55]. Samarth et al [56] worked on CNN classifiers, suggested an InceptionV4 model modified for fire detection, and outperformed FireNet [55] and InceptionV1OnFire [50]. Yuanbin et al. modified a

CNN and applied the network to perform forest fire detection tasks, but they did not test the model with any benchmark dataset [57]. Demirtas utilized object detection models to smoke detection for detecting wildfires at an early state [58]. Dua et al. used CNNs with a transfer learning method on their private dataset [59]. Arteaga et al. made experiments on deep CNNs such as ResNet50, ResNet101, ResNet34, VGG13, and VGG16 for wildfire detection and applied transfer learning to the models [60]. Park et al. suggested a DenseNet-based model, experimented on the dataset for fire classification, and compared the model with ResNet50 and VGG16 architectures(25). They also applied data augmentation with fire images generated by CycleGAN [61]. Rahul et al. conducted transfer learning experiments on ResNet50, VGG16,and DenseNet121 models with the SGD optimizer [62].

Shamsoshoara et al.[23] offered an open access dataset called the FLAME dataset. The authors also performed classification and segmentation experiments on this dataset and proposed an Xception network for the fire classification task [23]. After the Flame dataset was published, researchers used this dataset and made experiments to perform forest fire detection task. Treneska et al. [63] performed transfer learning experiments with the FLAME Dataset using well-known CNNs such as ResNet50, VGG16, VGG19, InceptionV3, and Xception. Ghali et al.[64] conducted extensive research comparing well-known deep learning models with their suggested network. Zhang et al.[65] made experiments with ResNet50, Inception and VGG based models using FLAME dataset. Some researchers also performed segmentation experiments using the FLAME dataset. Wang et al. used UNet model with ResNet50 backbone and DeepLab3 model to perform forest fire segmentation [66]. Wang et al. proposed a semi-supervised learning model [67]. Guan et al.[68] suggested a RCNN based method.

4. UTILIZED DATASETS & EVALUATION METRICS

We determined different approaches and methods based on dataset characteristics. In this section, two different datasets we experimented on are defined. Also, the evaluation metrics that are used are explained.

4.1. Dataset

This section describes two different datasets on which we experimented. Sample images from datasets are given below.

4.1.1. Infrared Dataset

The infrared dataset was assembled using approximately 10 hours of forest fire video data gathered with 1920x1080 resolution. The video data contains 55 frames per second. The data were analyzed with data engineering methods to create a data set suitable for deep learning experiments. Firstly, the parts with and without fire in the video data were separated. Frames were extracted from the split videos, one frame every 25 frames. Some of the images we observed contained too many repetitions in the extracted frames and were eliminated. As a result, we created an infrared wildfire classification dataset has 18,887 images labeled as fire and no-fire. The dataset used in this study consists of a total of 14,556 fire images and 4,331 no-fire training image data. Fifteen percent of the dataset was used for validation, twenty percent was utilized for testing, and the remaining portion was used as training data. This dataset is used as private due to data privacy principles. Sample images in the dataset are demonstrated in figure 4.1. For data privacy reasons, The black areas in the picture were added to cover the overlay parts of the UAV-collected images.

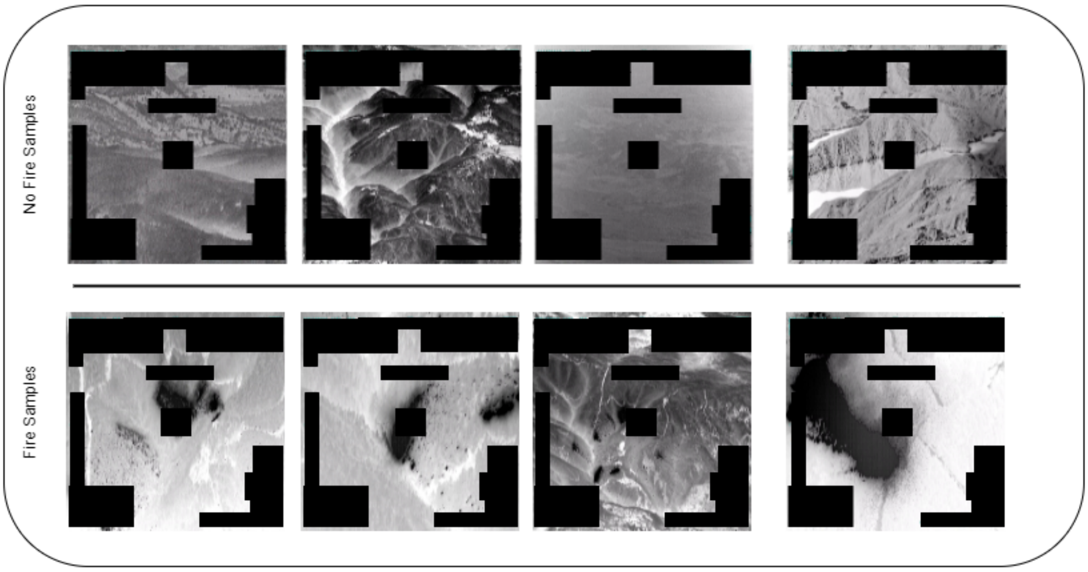

Figure 4.1 Infrared Dataset Samples

4.1.2. RGB Dataset

Among the Open Access datasets consisting of UAV collected images, the FLAME dataset is a very suitable dataset for our problem, as it is a large dataset and contains images from different points of view and various zoom levels. For these reasons, we conducted forest fire detection experiments using RGB images with the FLAME dataset. All samples in the dataset are images captured by multiple UAV cameras in Northern Arizona, USA. This dataset consists of two parts, classification and segmentation. The wildfire classification dataset we concentrate on comprises 39,375 training and 8,617 test RGB images labeled as Fire and No Fire. This dataset consists of 25,018 fire and 14,357 no fire images. Figure 4.2 representative input data in the FLAME dataset [23].

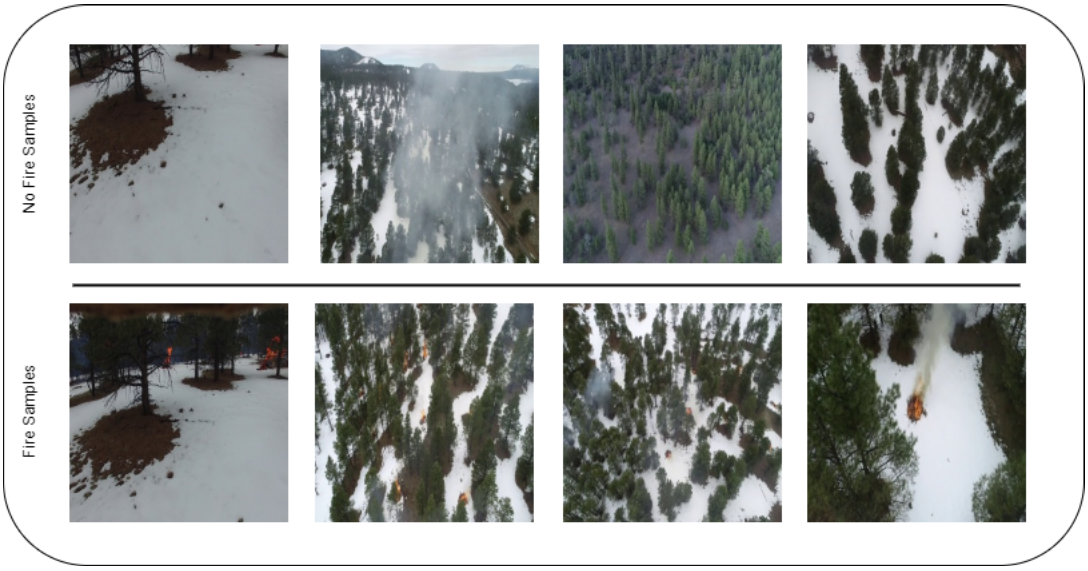

Figure 4.2 The FLAME Dataset Samples

4.2. Evaluation Metrics

In this thesis, the evaluation of results is conducted rely on several metrics: F1 Score, recall, accuracy,and precision. The metrics are calculated as shown in Eq. (4), (3), (2), and (1). All values are shown as weighted and macro average. Macro Average is calculated as the arithmetic mean of individual per-class scores, where the value (C) illustrates the total number of classes. The weighted average is an average resulting from the multiplication of each element by a factor mirroring its importance. Weighted average calculation is shown in Eq. (5), macro average calculation is shown in Eq. (6).

$$F1Score = 2 \times \frac{Recall * Precision}{Recall + Precision} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$WeightedAverage = \sum_{i=1}^{n} w_i x_i \tag{5}$$

$$MacroAverage = \frac{1}{C} \sum_{i=1}^{C} F1_i \tag{6}$$

5. PROPOSED METHOD

In this thesis, experiments were performed with using two different datasets: the FLAME and infrared datasets. In order to acquire highly accurate deep learning models, we determined different strategies based on dataset characteristics. In this section, different approaches determined for RGB and infrared datasets will be described.

5.1. Used Approaches

This section describes the approaches applied when developing fire detection models.

5.1.1. Inbalance Dataset Problem and Setting Class Weights

Classification with an imbalanced dataset is a challenge when experimenting with Deep Learning models [33]. If there is a class imbalance in the training dataset, networks usually exhibit a tendency to over classify the majority class due to its higher prior probability. In binary classification, class weights could be defined by calculating the commonness of the positive and negative ones and then inverting it. In this way, more weight is given to a class with fewer instances. That means assigning a higher value to the loss function to these samples. In this work, we enforced the setting class-weight procedure during training to deal with the imbalanced dataset issue in our experiments using the infrared dataset. The formula used to calculate the class weight is given in Eq. 7 and Eq. 8.

$$Weight_{Fire} = \frac{1}{Fire}\frac{Total}{2} \tag{7}$$

$$Weight_{NoFire} = \frac{1}{NoFire}\frac{Total}{2} \tag{8}$$

5.1.2. Transfer Learning

Transfer learning is a method used in machine learning approaches. It refers to the ability of machine learning techniques to leverage knowledge earned from solving one problem and apply it to another, thereby benefiting from the previously obtained information [34]. With transfer learning, models that perform better and learn faster with fewer data are obtained using previous knowledge. Figure 5.1 shows how transfer learning works. It aims to create a new model and increase performance by using the weights of a model that was previously trained with a dataset.

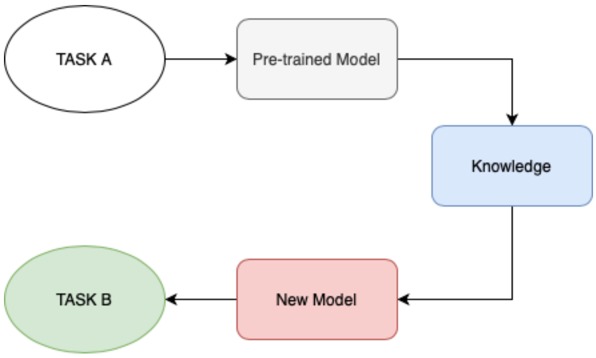

Figure 5.1 Principle of the Transfer Learning

5.1.3. Attention

The Attention method was developed to deal with the computational costs of deep CNNs [35]. The primary purpose of this technique is to focus on the significant parts of the information. It pays more attention to different regions of the input image while processing the data and employs acquired features of separate parts of the network as weights. Thus, the model concentrates on the relevant sections of the input data [36]. Channel attention is one of the attention applied for image classification [37]. It recalibrates the channel feature maps, allocates more weights to essential feature channels, and restrains unessential ones. In this study, we applied the channel attention method, which is one of the attention methods that gives very successful results in image classification.

5.2. Utilized Architectures

This section describes the neural network architectures used in Infrared and RGB image-based fire detection experiments.

5.2.1. General Architecture of ImegeNet Weighted Models and Trained From Scratch Models

In the fire detection experiments, we employ a two-part architecture comprising a classifier a feature extractor. In the FLAME dataset experiments, we experimented on ResNet50 [24], EfficientNetB0 [25], EfficientNetB2 [25], EfficientNetB4 [25], VGG16 [26], XCeption [28], NasNet Mobile [32], MobileNetV2 [30],and MobileNetV3Small [31] deep neural networks as feature extractors. In infrared dataset experiments, we used ResNet101 [24], ResNet50 [24], VGG16 [26], EfficientNetB1 [25] and DenseNet121 [27] networks as feature extractor. For the classifier part of our method, the classifier parts of the networks were replaced with a global average pooling layer. Then the layer's size of the networks was expanded to 256x256x3. A 0.5-rated dropout layer was added to avoid overfitting. Eventually, a dense layer with the Sigmoid function was applied. In the architecture we designed, we first resized all layers of a state-of-art neural network according to 256x256x3 resolution. Then we deleted the last softmax layer of the networks we used. Then we added a global average pooling layer. The global average pooling layer was followed by a dropout and sigmoid function. These features distinguish our architecture from other architectures such as classical ResNet and EfficientNet.

In the FLAME dataset experiments, we also accomplished transfer-learning experiments. The experiments were performed by keeping the ImageNet weights of the models used in the feature extractor section.

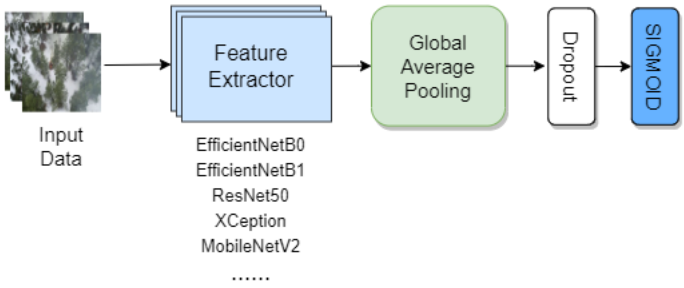

Figure 5.2 General Architecture of CNN Base Models

5.2.2. General Architecture of Attention Based Models

In order to improve the accuracy of the models for RGB UAV imagery-based forest fire detection, We conducted experiments by adding an attention layer using ResNet50 [24], EfficientNetB0 [25], EfficientNetB2 [25], EfficientNetB4 [25], VGG16 [26], XCeption [28], NasNet Mobile [32], MobileNetV2 [30], and MobileNetV3Small [31] models as backbones. For the attention-based network experiments, a batch normalization layer was added to the output layer of the network used as the backbone. The batch normalization layer was followed by an attention mechanism and global average pooling layers. Channel based attention layer contains 32 filtered convolutional layers, 16 filtered convolutional layers, 1 filtered convolutional layer, and 1280 filtered convolutional layers. After the convolutional layers, a global average pooling layer is added to perform element-wise multiplication between the output of the 1280 filtered convolutional layer and the output of batch normalization, resulting in the same shape. Another global average pooling layer was added to apply 2-dimensional global average pooling directly to the output of 1280 layered convolution layer resulting in a tensor of shape. A rescale function was added to apply a custom lambda function that rescales and concatenates the outputs of Global Average pooling layers. This operation introduces gating or attention mechanisms by selectively emphasizing or suppressing certain channels based on the information provided by BatchNormalization.

By multiplying the feature maps with the normalized activations from Batch Normalization, the network can potentially enhance or attenuate the importance of specific channels or features during the forward propagation process. This can be seen as a form of channel-wise attention, where the network learns to assign varying weights to different channels to concentrate on the most pertinent information for the task at hand. The channel based attention layer formula can be seen in Eq. 9.

$$attention\ output = conv\ output \times bn\ output \tag{9}$$

Afterward, a fully-connected layer with the ELU activation function was added after the global average pooling layers. Global average pooling layers fed a rescale function. The purpose of this function is to account for missing values from the attention model. Finally, the model was completed with a dropout layer followed by a dense layer with the Sigmoid activation function. The detailed model architecture is presented in Figure 5.3.

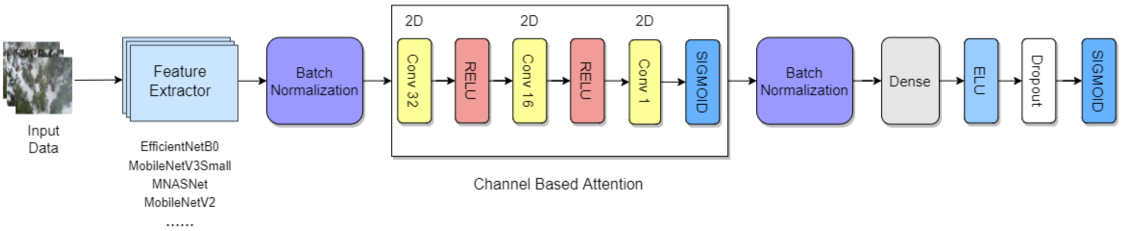

Figure 5.3 General Architecture of Attention Based Models

6. EXPERIMENTAL RESULTS

Experiments in forest fire detection studies with UAV vision data using CNNs are divided into two main parts: the RGB dataset and infrared dataset experiments. In RGB UAV imagery experiments, the FLAME dataset is used as the dataset. RGB dataset experiments are divided into three sub-parts: classification without transfer learning, classification with transfer learning, and attention-based model classification. The specified architectures used as feature extractor are VGG16, ResNet50[24], XCeption, EfficientNetB0[25], EfficientNetB2 [25], EfficientNetB3 [25],MobileNetV3 Small [31], MobileNetV2 [30], and NasNetMobile [32].

In infrared dataset experiments, only classification from scratch experiments was conducted. In the experiments, ResNet101 [24], ResNet50 [24], VGG16 [26], EfficientNetB1 [25] and DenseNet121 [27] networks were used as feature extractor. In addition, attention-based EfficientNetB0 model was experimented on the dataset.

These architectures are chosen based on their accuracy in the ImageNet dataset, model sizes, and the number of trainable parameters.

6.1. Experimental Setup

For the purpose of detecting wildfires using images collected from UAVs, we utilized TensorFlow-GPU version 2.7.0 and Python 3.7.0. The hardware environment employed for conducting the experiments was an NVIDIA GeForce RTX 3060 Laptop GPU.

6.2. Hyperparameters

For the training phase of the experiments, the initial hyper-parameters, optimizer, and loss function were kept alike. In our experiments, we employed input image data with a resolution of 256×256×3 and utilized Adam as the optimizer. The learning rate was set to 0.0001. During the training process, the early stop callback method based on validation accuracy was used to avoid overfitting. The hyperparameters we determined in our experiments are shown in Table 6.1.

Table 6.1 Hyperparameters

Hyperparameter	Hyperparameter Space
Input Shape	256×256×3
Learning Rate	0.0001
Optimizer	Adam
Loss	Binary Crossentropy
Last Layer Activation Function	Sigmoid

6.3. FLAME Dataset Experimental Results

6.3.1. Train From Scratch Experimental Results

We trained classification models with the FLAME dataset consisting of UAV imagery. Hyperparameters we determined in the experiments are shown in Table 6.1. Using the architecture illustrated in Figure 5.2, we experimented with state of art neural networks as feature extractors.

We carried out experiments without using any pre-trained weights. As a result of these experiments, the determined models and the test accuracy values obtained by these models are shown in Table 6.2. These experiments demonstrated that the model using the VGG16 architecture as a feature extractor acquired the highest test accuracy with 65.19%. However, the test results of the models that were experimented with were not at a sufficient level.

Table 6.2 Flame Dataset Experiments Training From Scratch

Feature Extractor	Test Accuracy %
VGG16	**65.19**
ResNet50	63.58
Xception	59.55
EfficientNetB4	58.97
EfficientNetB0	57.64
MobileNetV3-Small	54.19
MobileNetV2	52.86
NasNetMobile	52.04
EfficientNetB2	51.03

Since the Flame dataset is imbalanced and has repetitive samples, developing models with a high accuracy rate is a real challenge by avoiding overfitting. That's why we worked on various approaches and improvements.

6.3.2. Experimental Results With Transfer Learning

The fact that the FLAME dataset is imbalanced, and it has higher samples in fire class caused the models we trained to classify non-fire images as fire. We applied transfer learning to overwhelm this encumbrance and trained the models using the ImageNet weights. In transfer learning experiments, we used the same network architecture, feature extractors, and hyperparameters as we mentioned in the from-scratch experiments section. The hyperparameters used in transfer learning experiments are shown in Table 6.1. The network architecture used in transfer learning experiments is shown in Figure 5.2. Comparative results between experimentations are shown in Table 6.3.

Table 6.3 Experimental Results with Transfer Learning

Model	Accuracy %	Precision %	Recall %	F1 Score %
EfficentNetB4	**90.80**	**91.04**	**90.8**	**90.85**
XCeption	89.78	90.8	89.78	89.87
NasNetMobile	83.67	83.61	83.67	83.62
MobilenetV3Small	82.34	84.76	82.34	82.5
EfficentNetB0	82.19	84.43	82.19	82.36
ResNet50	81.06	81.38	81.06	80.57
EfficentNetB2	79.88	86.33	79.88	79.86
MobilenetV2	74.26	82.61	74.26	73.99
VGG16	68.23	68.59	68.23	65.63

Approximated with the results, we observed that all models show increased accuracy after applying transfer learning. In transfer learning experiments, the EfficientNetB4-based model has proved its effectiveness by achieving 90.80% test accuracy and 90.85% F1 score.

The confusion matrix obtained of the EfficientNetB4 based model is shown in Figure 6.1. Class-based recall, precision, and F1 score results of the model are shown in Table 6.4.

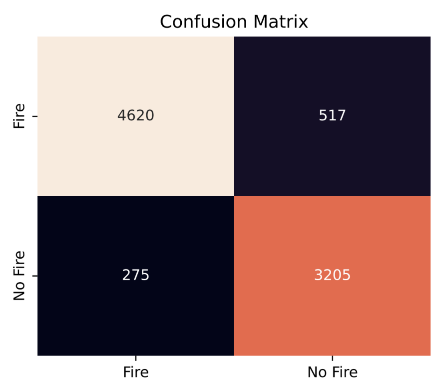

Figure 6.1 EfficientNetB4 Fine Tuned Model Confusion Matrix

Table 6.4 Detailed Experimental Results of the Model Using EfficientNetB4 as Feature Extractor

EfficentNetB4	Precision %	Recall %	F1 Score %
Fire	94.38	89.93	92.10
No Fire	86.10	92.09	89.00
Macro avg	90.24	91.01	90.55
Weighted avg	91.04	90.80	90.85
Accuracy %	90.80		

Considering the F1 score parameters, the architecture in which EfficientNetB4 is the feature extractor has been more successful in accurately predicting images containing the fire.

6.3.3. Experimental Results Based on Attention Based Models

Adding the attention layer is a popular approach that remarkably improves the recognition performance of CNNs. We applied this technique for forest fire detection with UAV vision data. For this purpose, we implemented the channel-based attention layer on determined CNN architectures and conducted experiments. Unlike from scratch and transfer learning experiments, we conducted attention-based experiments through the proposed architecture shown in Figure 5.3. The hyperparameters used attention based experiments are demonstrated in Table 6.1.

Table 6.5 Experimental Results of Attention Based Method

Arcitecture	Accuracy %	Precision %	Recall %	F1 Score %
EfficentNetB0 Backbone	**92.02**	**92.66**	**92.02**	**92.08**
NasNet Mobile Backbone	87.11	87.7	87.11	87.2
ResNet50 Backbone	86.61	88.50	86.61	86.74
XCeption Backbone	85.51	86.02	85.51	85.61
EfficentNetB4 Backbone	85.05	87.59	85.05	85.18
EfficentNetB2 Backbone	82.75	82.70	82.75	82.57
MobileNetV2 Backbone	81.95	84.11	81.95	81.81
MobileNetV3Small Backbone	81.15	81.41	81.15	81.23
VGG16 Backbone	75.59	75.70	75.59	74.80

In the light of these experiments, the attention-based model with the EfficientNetB0 backbone has achieved the highest test accuracy with 92.02% and the highest F1 score with 92.08%. Thus, all state of art models outperformed. The confusion matrix obtained as a result of the test of the model is shown in Figure 6.2.

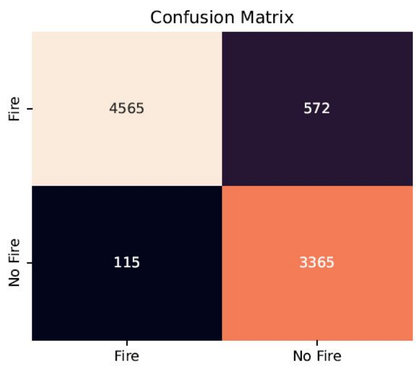

Figure 6.2 Confusion Matrix of Attention Based Model with EfficientB0 Backbone

Table 6.6 Detailed Experimental Results of Attention Based EfficientNetB0 Model

EfficentNetB0 Attention	Precision %	Recall %	F1 Score %
Fire	97.54	88.86	93.00
No Fire	85.47	96.69	90.73
Macro avg	91.50	92.78	91.86
Weighted avg	92.66	92.02	92.08
Accuracy %	92.02		

Class-based test scores of the experiments conducted with the attention-based network with the EfficientNetB0 backbone are illustrated in Table 6.6. Based on Table 6.6, the architecture predicted images of fire and non-fire with high accuracy, but it was more successful at correctly predicting fire images. Figure 6.3 demonstrates the model's class Gradient-weighted Class Activation Mapping (Grad-CAM) results in some images from dataset[69]. It can be seen that the model focused on the regions containing fire in the images sampled.

Figure 6.3 Attention Heatmap of Attention Based Model with EfficientB0 Backbone

6.3.4. Comparison with The State-of-Art Studies On The Flame Dataset

Many researchers have carried out experiments using the FLAME dataset. Table 6.7 demonstrates results of classifications experiments of papers used the FLAME dataset. Among the experiments using the FLAME dataset, Treneska et al.'s [63] Fine Tuned ResNet50 model acquired the highest test accuracy score.

Table 6.7 Classification Experiments on the Flame Dataset So Far

Paper	Classification Algorithm	Test Accuracy %
Shamsoshoara et al. [23]	Modified Xception	76.23
Ghali et al. [64]	Proposed Model	**85.12**
	InceptionV3	80.88
	DenseNet169	80.62
	Xception	78.41
	EfficientNet-B5	75.82
	EfficientNet-B4	69.93
	EfficientNet-B2	66.04
	EfficientNet-B3	65.81
	MobileNetV3-Large	65.10
	MobileNetV3-Small	51.64
Treneska et al. [63]	ResNet50	66.11
	ResNet50 (Fine Tuned)	**88.01**
	InceptionV3	67.48
	Inception V3 (Fine Tuned)	87.21
	VGG19	69.44
	VGG19 (Fine Tuned)	83.43
	Xception	71.93
	Xception (Fine Tuned)	81.3
	VGG16	77.89
	VGG16 (Fine Tuned)	80.76

In the experiments conducted in this study, the most successful models, as determined by test accuracy, precision, recall, and F1 score metrics, are the Attention-Based EfficientNetB0, fine-tuned EfficientNetB4, fine-tuned XCeption, and Attention-Based NasNet Mobile

The comparisons of the models with the highest accuracy from the experiments conducted with the FLAME dataset are presented in Table 6.8.

Table 6.8 Flame Dataset Experiments Comparison

Paper	Model	Accuracy %
Shamsoshoara et al. [23]	Modified XCeption	76.23
Ghali et al. [64]	Proposed Model	85.12
Treneska et al. [63]	ResNet50 (Fine Tuned)	88.01
Our Study	EfficentNetB4 (Fine Tuned)	90.80
	NasNet Mobile Based Attention Model	87.11
	EfficentNetB0 Based Attention Model	**92.02**

Table 6.8 demonstrates that Fine tuned XCeption, Fine Tuned EfficientNetB4, and attention-based EfficientNetB0 models have obtained the highest test accuracy values among the experiments. In the Flame dataset pioneer paper [23], the authors achieved 76% test accuracy using a modified Xception model. In another paper using the FLAME dataset, Ghali et al.'s method implemented DenseNet205 and EfficientNetB5 models as backbones and achieved 84.77% test accuracy [64]. In another study by Treneska et al., the model using the fine-tuned ResNet50 as a backbone acquired 88.01% test accuracy with the FLAME Dataset [63]. In our proposed method, the model using EfficientNetB4 as a feature extractor achieved 90.80% test accuracy, the model using XCeption as a feature extractor achieved test accuracy values of 89.78%, and the EficientNetB0 attention based network achieved 92.02% test accuracy ,attaining the highest accuracy among the experiments carried out so far. In addition, EficientNetB0 attention based model surpassed the method using the EfficientNetB4 and XCeption architectures pre-trained with ImageNet weights as a feature

extractor and achieved the highest performance ever among experiments with the FLAME dataset.

Table 6.9 Comparison of the Model and Number Of Trainable Parameters

Model	Number of Trainable Parameters
ResNet50	23,536,641
Xception	20,809,001
EfficientNetB4	17,550,409
NasNet Mobile Based Attention Model	4,404,884
EfficentNetB0 Based Attention Model	**4,215,742**

Table 6.9 shows that the trainable parameter numbers of the attention-based EfficientNetB0 backbone architecture with the best performed model in conducted experiments. Table 6.9 illustrates that the EfficientNetB0 backboned model has the lowest number of trainable parameters among the architectures that achieve the most successful test accuracy values. In this way, it was proven to be a lightweight and highly accurate model.

6.4. Infrared Dataset Experiments Results

We experimented with the strategies shown in Figure 5.2, using well-known deep CNN architectures as feature extractors for wildfire detection on infrared black hot images. The hyperparameters chosen in these experimentations are demonstrated in Table 6.1. Since the infrared black hot dataset is imbalanced, class weight was imposed with the formula specified in the methods section during training. Thanks to the strategies we determined for forest fire classification with heat-sensitive infrared images, the methods we experimented with achieved superior performance. In experiments with Deep Neural networks, it was observed that deeper networks provided better results among the models we used as feature extractors.

Table 6.10 Infrared Dataset Experiments

Model	Accuracy %	Precision %	Recall %	F1 Score %
Attention Based EfficientNetB0	**99.76**	**99.52**	**99.80**	**99.66**
ResNet101	99.20	99.11	98.63	98.87
ResNet50	98.04	98.05	98.04	98.04
VGG16	98.56	98.57	98.57	98.56
EfficientNetB1	98.54	98.53	98.54	98.53
DenseNet121	97.24	97.35	97.24	97.27

The comparative test performance of the experimented models is shown in Table 6.10. The test scores obtained by the trained models are close. In classification from scratch experiments, The ResNet101 model attained the highest performance with 99.20% test accuracy and 98.87% F1 score. Class-based test performance values of the ResNet 101 model are shown in Table 6.11. Based on the test accuracy, precision, recall, and F1 score parameters, it can be observed that the model using the ResNet101 as the feature extractor can detect fire and no-fire images with high accuracy. The confusion matrix obtained as a result of the test of the model is shown in Figure 6.4.

Table 6.11 Experimental results of the model using ResNet101 as Feature Extractor with Infrared Dataset

ResNet101	Precision %	Recall %	F1 Score %
Fire	99.28	99.69	99.48
No Fire	98.94	97.57	98.25
Macro Avg	99.11	98.63	98.87
Weighted Avg	99.20	99.20	99.20
Accuracy	99.20		

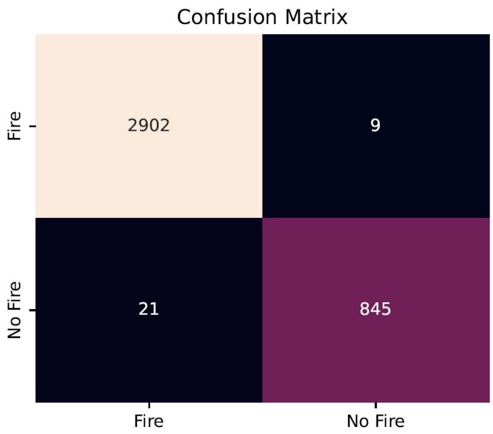

Figure 6.4 Model Using ResNet101 as Feature Extractor Confusion Matrix with Infrared Dataset

The Attention-based model with EfficientNetB0 backbone, which we developed in RGB image based wildfire classification experiments, was also tested with the infrared blackhot dataset. Our experiments have shown that this model surpasses other models with 99.76% test accuracy and 99.66% F1 score.

The confusion matrix obtained as a result of the test of the attention based model is shown in Figure 6.5. The confusion matrix indicates that the model showed high performance in classifying fire no fire images.

Table 6.12 Attention Based Model with EfficientB0 Backbone Infrared Experiments

EfficientB0 Backbone Attention	Precision %	Recall %	F1 Score %
Fire	99.96	99.72	99.84
No Fire	99.08	99.88	99.48
Macro Avg	99.52	99.80	99.66
Weighted Avg	99.76	99.76	99.76
Accuracy %	99.76		

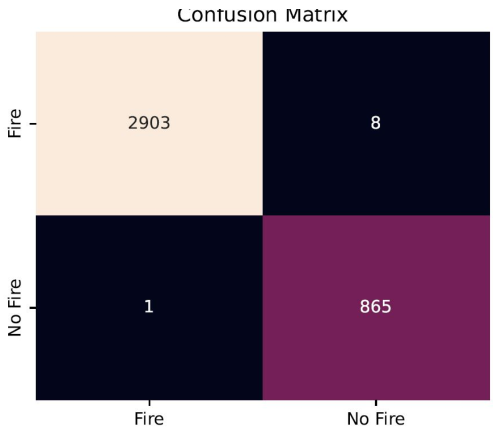

Figure 6.5 Attention Based Model with EfficientB0 Backbone Confusion Matrix with Infrared Dataset

Figure 6.6 demonstrates the model's class Gradient-Weighted Class Activation Mapping (Grad-CAM) results in infrared images [69]. It is observed that the Attention based model successfully focuses on the sections containing fire. The dark areas in the picture have been added for data privacy reasons.

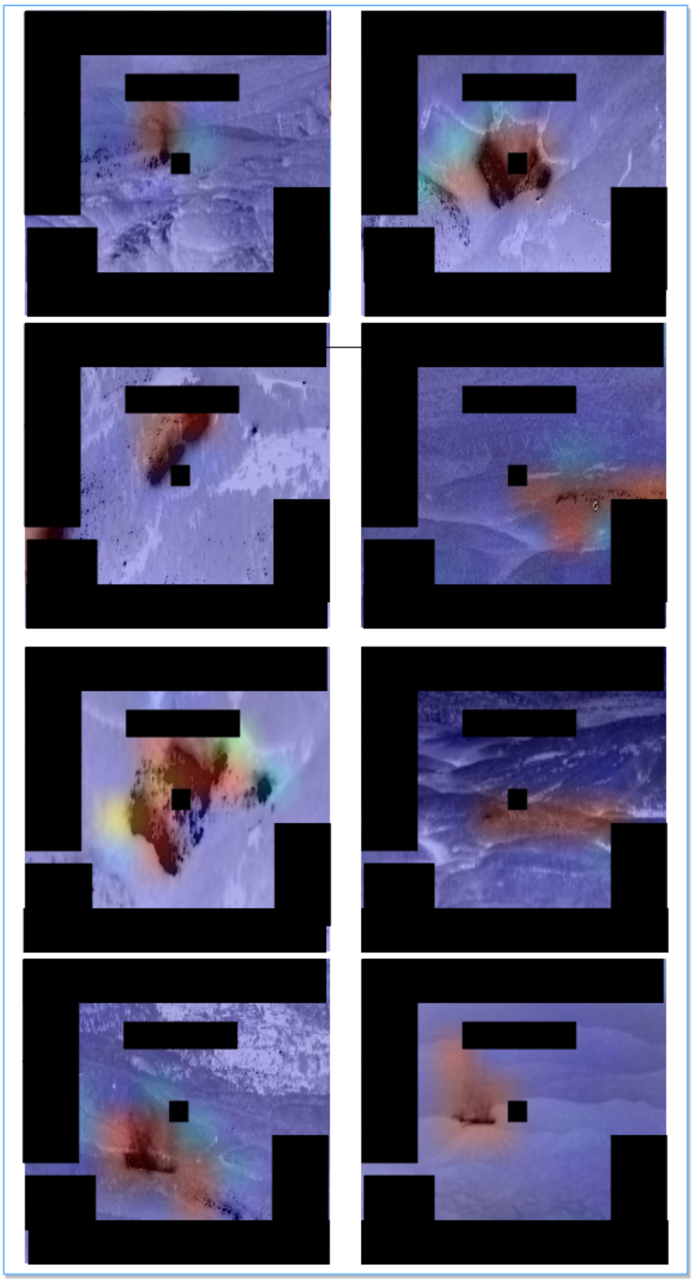

Figure 6.6 Attention Based Model with EfficientB0 Backbone Attention Heatmap with Infrared Dataset

6.4.1. K-FOLD Cross Validation

In order to test the exactness of the scores we obtained in the infrared blackhot dataset , we applied the 5 Fold Cross validation experiments mentioned in the Experiments section.

Table 6.13 Infrared Dataset K-FOLD Cross Validation Experimental Results

Model	Accuracy %	Precision %	Recall %	F1 Score %
Attention Based EfficientNetB0	**99.58**	**99.54**	**99.58**	**99.51**
EfficientNetB1	98.96	98.91	98.88	98.86
ResNet50	98.81	98.82	98.80	98.81
ResNet101	98.62	98.64	98.62	98.62
VGG16	98.14	98.14	98.14	98.02
DenseNet121	97.79	97.98	97.95	97.94

As a result of these experiments, according to the several observations, the average results of the k-fold experiment showed parallelism with the results acquired by the models experimented on the main dataset. The EfficientNetB0 backbone attention based model performed best on average. Table 6.13 indicates the average values of each parameter based on 5-Folds in K-Fold cross validation experiments.

7. CONCLUSION

This thesis addresses the problem of forest fire classification on UAV vision data using convolutional neural networks. Our experiments involved the use of two distinct datasets, which were obtained by capturing images from UAV cameras. The first dataset consists of real-world infrared black hot images, while the second dataset comprises RGB images. With the strategies and approaches we determined, we have developed models that can detect forest fires with high accuracy in both datasets. We evaluated state-of-the-art CNNs as feature extractors in the methods we applied. Applying transfer learning in experiments with RGB datasets and setting class weights during training in experiments with infrared datasets enabled models with high accuracy to be discovered. The experiments have shown that the developed models have surpassed all previous approaches and had the highest accuracy with two different datasets. Most deep CNN-based techniques have the major drawback of being computationally hungry for real-time wildfire recognition. The issue is exacerbated for UAVs performing missions with devices that have very low onboard computational power. To overcome this problem, we made some research efforts in that direction. Therefore, this thesis presents a lightweight and high-performance architecture for detecting and classifying forest fires. An efficient method, combining small CNN and channel attention, was developed to detect wildfires from UAV imagery. Of the several lightweight models used as feature extractors explored for this problem, the EfficientNetB0 proved itself to be the most efficient alternative. The results acquired on two different forest fire datasets with our attention-based architecture have been superior to all the current techniques in terms of most evaluation metrics like accuracy, f-measure, recall, and precision.

Detection of forest fires is a demanding issue due to the images taken from UAV cameras being taken from different altitudes, and the fire does not have a specific pattern. In addition, the imbalanced datasets we used during the experiments were an additional challenge for us. In future work, we plan to experiment with different problems, such as segmentation, and increase the capability of our models with more real-world UAV datasets collected.

REFERENCES

[1] Bladon, Kevin D., et al. ”Wildfire and the future of water supply.” (2014): 8936-8943.

[2] Artés, Tomàs, et al. ”A global wildfire dataset for the analysis of fire regimes and fire behaviour.” Scientific data 6.1 (2019): 1-11.

[3] Jolly, W. Matt, et al. ”Climate-induced variations in global wildfire danger from 1979 to 2013.” Nature communications 6.1 (2015): 1-11.

[4] Park Williams, A., et al. ”Temperature as a potent driver of regional forest drought stress and tree mortality.” Nature climate change 3.3 (2013): 292-297.

[5] NIFC, Available online: ”https://www.nifc.gov/fireInfo/fireInfon_{-}statistics.html” (accessed on 08 August 2022).

[6] Turkish Aerospace Industries AKSUNGUR UAS. Available online: ”https://www.tusas.com/en/products/uav/high-payload-capacity-uas/aksungur” (accessed on 08 August 2022).

[7] Climate Council. This is Not Normal: Climate Change and Escalating Bushfire Risk. Climate Council Briefing Paper. 12 November 2019. Available online: https://www.climatecouncil.org.au/wp-content/uploads/2019/11/bushfire-briefing-papern_{-}18-november.pdf (accessed on 08 August 2022). .

[8] Tedim, Fantina, et al. ”Forest Fire Causes and Motivations in the Southern and South-Eastern Europe through Experts’ Perception and Applications to Current Policies.” Forests 13.4 (2022): 562.

[9] Gaur, Anshul, et al. ”Fire sensing technologies: A review.” IEEE Sensors Journal 19.9 (2019): 3191-3202.

[10] Artim, Nick. ”An introduction to fire detection, alarm, and automatic fire sprinklers.” Northeast Document Conservation Center (1999).

[11] Gaur, Anshul, et al. Li, Zhanqing, Serge Nadon, and Josef Cihlar. ”Satellite-based detection of Canadian boreal forest fires: Development and application of the algorithm.” International Journal of Remote Sensing 21.16 (2000): 3057-3069.

[12] Friedlingstein, Pierre, et al. ”Global carbon budget 2020.” Earth System Science Data 12.4 (2020): 3269-3340.

[13] Cazzato, Dario, et al. ”A survey of computer vision methods for 2d object detection from unmanned aerial vehicles.” Journal of Imaging 6.8 (2020): 78.

[14] Ejaz, Waleed, et al. ”Unmanned aerial vehicles enabled IoT platform for disaster management.” Energies 12.14 (2019): 2706.

[15] Akhloufi, Moulay A., Andy Couturier, and Nicolás A. Castro. ”Unmanned aerial vehicles for wildland fires: Sensing, perception, cooperation and assistance.” Drones 5.1 (2021): 15.

[16] Zhang, Qingjie, et al. ”Deep convolutional neural networks for forest fire detection.” 2016 International Forum on Management, Education and Information Technology Application. Atlantis Press, 2016.

[17] Jain, Piyush, et al. ”A review of machine learning applications in wildfire science and management.” Environmental Reviews 28.4 (2020): 478-505.

[18] Maggiori, Emmanuel, et al. ”Convolutional neural networks for large-scale remote-sensing image classification.” IEEE Transactions on geoscience and remote sensing 55.2 (2016): 645-657.

[19] Cheng, Gong, et al. ”When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs.” IEEE transactions on geoscience and remote sensing 56.5 (2018): 2811-2821.

[20] Hohman, Fred, et al. ”Visual analytics in deep learning: An interrogative survey for the next frontiers.” IEEE transactions on visualization and computer graphics 25.8 (2018): 2674-2693.

[21] Samad, Abd Manan, et al. ”The potential of Unmanned Aerial Vehicle (UAV) for civilian and mapping application.” 2013 IEEE 3rd International Conference on System Engineering and Technology. IEEE, 2013.

[22] Yuan, Chi, Youmin Zhang, and Zhixiang Liu. ”A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques.” Canadian journal of forest research 45.7 (2015): 783-792.

[23] Shamsoshoara, Alireza, et al. ”Aerial imagery pile burn detection using deep learning: The FLAME dataset.” Computer Networks 193 (2021): 108001..

[24] He, Kaiming, et al. ”Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[25] Tan, Mingxing, and Quoc Le. ”Efficientnet: Rethinking model scaling for convolutional neural networks.” International conference on machine learning. PMLR, 2019.

[26] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

[27] Huang, Gao, et al. ”Densely connected convolutional networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[28] Chollet, François. ”Xception: Deep learning with depthwise separable convolutions.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[29] Szegedy, Christian, et al. ”Rethinking the inception architecture for computer vision.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[30] Sandler, Mark, et al. ”Mobilenetv2: Inverted residuals and linear bottlenecks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[31] Howard, Andrew, et al. ”Searching for mobilenetv3.” Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[32] Zoph, Barret, et al. ”Learning transferable architectures for scalable image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[33] Johnson, Justin M., and Taghi M. Khoshgoftaar. ”Survey on deep learning with class imbalance.” Journal of Big Data 6.1 (2019): 1-54.

[34] Zhuang, Fuzhen, et al. ”A comprehensive survey on transfer learning.” Proceedings of the IEEE 109.1 (2020): 43-76.

[35] Vaswani, Ashish, et al. ”Attention is all you need.” Advances in neural information processing systems 30 (2017).

[36] Hu, Jie, Li Shen, and Gang Sun. ”Squeeze-and-excitation networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[37] Woo, Sanghyun, et al. ”Cbam: Convolutional block attention module.” Proceedings of the European conference on computer vision (ECCV). 2018.

[38] Vicente, Jerome, and Philippe Guillemant. ”An image processing technique for automatically detecting forest fire.” International Journal of Thermal Sciences 41.12 (2002): 1113-1120.

[39] Martínez-de Dios, J. Ramiro, Luis Merino, and Aníbal Ollero. ”Fire detection using autonomous aerial vehicles with infrared and visual cameras.” IFAC Proceedings Volumes 38.1 (2005): 660-665.

[40] Li, Ying, et al. ”A hybrid contextual approach to wildland fire detection using multispectral imagery.” IEEE Transactions on Geoscience and Remote Sensing 43.9 (2005): 2115-2126.

[41] Moragues, J., I. Bosch, and L. Vergaa. ”Infrared image processing and its application to forest fire surveillance.” Conference Paper. October. 2007.

[42] Efficient Forest Fire Detection System: A Spatial Data Mining and Image Processing Based Approach

[43] Günay, Osman, et al. ”Video based wildfire detection at night.” Fire Safety Journal 44.6 (2009): 860-868.

[44] Angayarkkani, K., and N. Radhakrishnan. ”An intelligent system for effective forest fire detection using spatial data.” arXiv preprint arXiv:1002.2199 (2010).

[45] Vipin, V. ”Image processing based forest fire detection.” International Journal of Emerging Technology and Advanced Engineering 2.2 (2012): 87-95.

[46] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems 25 (2012).

[47] Schmidhuber, Jürgen. ”Deep learning in neural networks: An overview.” Neural networks 61 (2015): 85-117.

[48] Kim, Seonghyun, et al. ”Forest fire monitoring system based on aerial image.” 2016 3rd international conference on information and communication technologies for disaster management (ICT-DM). IEEE, 2016.

[49] Lee, Wonjae, et al. ”Deep neural networks for wild fire detection with unmanned aerial vehicle.” 2017 IEEE international conference on consumer electronics (ICCE). IEEE, 2017.

[50] Dunnings, Andrew J., and Toby P. Breckon. ”Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection.” 2018 25th IEEE international conference on image processing (ICIP). IEEE, 2018.

[51] Wang, Kewei, et al. ”Fire detection in infrared video surveillance based on convolutional neural network and SVM.” 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP). IEEE, 2018.

[52] Chen, Yanhong, et al. ”A UAV-based forest fire detection algorithm using convolutional neural network.” 2018 37th Chinese Control Conference (CCC). IEEE, 2018.

[53] Zhao, Yi, et al. ”Saliency detection and deep learning-based wildfire identification in UAV imagery.” Sensors 18.3 (2018): 712.

[54] Muhammad, Khan, et al. ”Efficient fire detection for uncertain surveillance environment.” IEEE Transactions on Industrial Informatics 15.5 (2019): 3113-3122.

[55] Jadon, Arpit, et al. ”FireNet: a specialized lightweight fire & smoke detection model for real-time IoT applications.” arXiv preprint arXiv:1905.11922 (2019).

[56] CA, Ganesh Samarth, Neelanjan Bhowmik, and Toby P. Breckon. ”Experimental exploration of compact convolutional neural network architectures for non-temporal real-time fire detection.” 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, 2019.

[57] Wang, Yuanbin, Langfei Dang, and Jieying Ren. ”Forest fire image recognition based on convolutional neural network.” Journal of Algorithms & Computational Technology 13 (2019): 1748302619887689.

[58] Demirtas A. ” Döner kanatlı insansız hava araçları ile görsel verilere dayalı erken yangın algılama sistemi.” Graduate School of Science and Engineering, Ankara, Ankara University, 2019.

[59] Dua, Mohit, et al. ”An improved approach for fire detection using deep learning models.” 2020 International Conference on Industry 4.0 Technology (I4Tech). IEEE, 2020.

[60] Arteaga, Byron, Mauricio Diaz, and Mario Jojoa. ”Deep Learning Applied to Forest Fire Detection.” 2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). IEEE, 2020.

[61] Park, Minsoo, et al. ”Wildfire-detection method using DenseNet and CycleGAN data augmentation-based remote camera imagery.” Remote Sensing 12.22 (2020): 3715.

[62] Rahul, Medi, et al. ”Early detection of forest fire using deep learning.” 2020 IEEE REGION 10 CONFERENCE (TENCON). IEEE, 2020.

[63] Treneska, Sandra, and Biljana Risteska Stojkoska. ”Wildfire detection from UAV collected images using transfer learning.” Proceedings of the 18th International Conference on Informatics and Information Technologies, Skopje, North Macedonia. 2021.

[64] Ghali, Rafik, Moulay A. Akhloufi, and Wided Souidene Mseddi. ”Deep learning and transformer approaches for UAV-based wildfire detection and segmentation.” Sensors 22.5 (2022): 1977.

[65] Zhang, Lin, et al. ”A Forest Fire Recognition Method Using UAV Images Based on Transfer Learning.” Forests 13.7 (2022): 975.

[66] Wang, Ziqi, Tao Peng, and Zhaoyou Lu. ”Comparative research on forest fire image segmentation algorithms based on fully convolutional neural networks.” Forests 13.7 (2022): 1133.

[67] Wang, Junling, et al. ”Semi-Supervised Learning for Forest Fire Segmentation Using UAV Imagery.” Forests 13.10 (2022): 1573.

[68] Guan, Zhihao, et al. ”Forest fire segmentation from Aerial Imagery data Using an improved instance segmentation model.” Remote Sensing 14.13 (2022): 3159.

[69] Selvaraju, Ramprasaath R., et al. ”Grad-cam: Visual explanations from deep networks via gradient-based localization.” Proceedings of the IEEE international conference on computer vision. 2017.