İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ
2010
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

Deep Learning Based Smoke Detection for Foggy Environments

Submitted to the Graduate School of Natural and Applied Sciences in partial fulfillment of the requirements for the degree of

Master of Science

in Electrical and Electronics Engineering

by

Uğur Emre Yıldız

ORCID 0000-0003-1166-4542

June, 2021

This is to certify that we have read the thesis **Deep Learning Based Smoke Detection for Foggy Environments** submitted by **Uğur Emre Yıldız**, and it has been judged to be successful, in scope and in quality, at the defense exam and accepted by our jury as a MASTER'S THESIS.

APPROVED BY:

Advisor:	**Asst. Prof. Dr. Mehmet Erdal Özbek** İzmir Kâtip Çelebi University	……..
Committee Members:		
	Asst. Prof. Dr. Nalan Özkurt Yaşar University	……..
	Asst. Prof. Dr. Esra Aycan Beyazıt İzmir Kâtip Çelebi University	……..

Date of Defense: June 24, 2021

Declaration of Authorship

I, **Uğur Emre Yıldız**, declare that this thesis titled **Deep Learning Based Smoke Detection for Foggy Environments** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for the master's degree at this university.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. This thesis is entirely my own work, with the exception of such quotations.
- I have acknowledged all major sources of assistance.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signature:

_____

Date: 24.06.2021

_____

Deep Learning Based Smoke Detection for Foggy Environments

Abstract

In recent years, as the global warming threat manifests itself more, the number of studies on outdoor smoke detection is increased to prevent wildfires, and these studies are mostly focused on image-based detection of smoke. However, the major bottleneck of the detection of smoke with high accuracy is harsh weather conditions. Especially fog, due seems to smoke, prevents detecting with high accuracy.

In this thesis, a method is proposed for the detection of fog and smoke images. In this method, videos from various databases including footage with and without smoke are divided into their images. By changing the brightness values of images, foggy images are created artificially. The dataset containing foggy and smoky samples is classified using modern Convolutional Neural Network architectures in various color spaces. By comparing the performances of color spaces and deep learning architectures the best solutions in terms of memory usage and classification accuracy are determined. The proposed method as one of these solutions, is overtaken the literature studies.

Keywords: Smoke detection, deep learning, color spaces, convolutional neural networks, image processing

Sisli Ortamlar için Derin Öğrenme Tabanlı Duman Tespiti

Öz

Son yıllarda küresel ısınma tehdidinin kendisini daha çok göstermesiyle orman yangınlarını önlemek için dış ortam duman tespitiyle ilgili çalışmaların sayısı artmış ve bu çalışmalarda çoğunlukla dumanın görüntü tabanlı tespitine odaklanılmıştır. Lakin yüksek doğrulukla duman tespitinin en önemli darboğazı zorlu hava koşullarıdır. Özellikle sis, duman gibi gözüktüğü için yüksek doğrulukta tespit yapılmasını engeller.

Bu tezde, sis ve duman görüntülerinin tespiti için bir yöntem önerildi. Bu yöntemde, dumanlı ve dumansız çekimleri içeren çeşitli veri tabanlarından videolar görüntülerine ayrıldı. Görüntülerin parlaklık değerleri değiştirilerek yapay olarak sisli görüntüler oluşturuldu. Sisli ve dumanlı örnekleri içeren veri seti, çeşitli renk uzaylarında modern Konvolüsyonel Sinir Ağları mimarileri kullanılarak sınıflandırıldı. Renk uzayları ve derin öğrenme mimarilerinin performansları karşılaştırılarak bellek kullanımı ve sınıflandırma doğruluğu açısından en iyi çözümler belirlendi. Bu çözümlerden biri olan önerilen yöntem, literatür çalışmalarını geride bıraktı.

Anahtar Kelimeler: Duman tespiti, derin öğrenme, renk uzayları, evrişimli sinir ağları, görüntü işleme

To my family

Acknowledgment

I would like to thank my esteemed thesis advisor Asst. Prof. Dr. Mehmet Erdal Özbek, for his encouragement and guidance throughout my research.

I would also like to thank Research Assistant Özge Taylan Moral, who helped me a lot in solving my problems and was very important in the success of the thesis.

Finally, I would like to thank my dear family who have always been by my side, supported me and brought me today.

Table of Contents

Declaration of Authorship ii

Abstract iii

Öz iv

Acknowledgment vi

List of Figures x

List of Tables xii

List of Abbreviations xiii

List of Symbols xiv

1 Introduction 1

1.1 Motivation 2

1.2 Thesis Organization 2

2 Background 4

2.1 Color Spaces 5

2.2 Machine Learning 7

2.3 Convolutional Neural Network 10

2.3.1 CNN Layers 10

2.3.1.1 Convolution Layer 11

2.3.1.2 Pooling Layer 13

2.3.1.3 Batch Normalization 14

2.3.1.4 Dropout Layer 14

2.3.1.5 Connected Layer 15

2.3.2 CNN Architecture Models 16

2.3.2.1 VGG 16
2.3.2.2 Inception V3 17
2.3.2.3 Inception ResNetV2 18
2.3.2.4 Xception 19
2.3.2.4 Densely Connected Networks 20
2.3.2.5 MobileNetV2 21
2.4 Smoky and Normal Images Databases 23
2.4.1 Center for Wildfire Research Database 23
2.4.2 VisiFire Database 24
2.4.3 Firesense Database 24
2.4.4 MIVIA Smoke Database 24
2.4.5 Wildfire Smoke Detection Database 24
2.4.6 Other Online Video Database 24
2.4.7 Analysis of Database Footages 25
3 Methodology 29
3.1 Dataset Operations 29
3.1.1 Preparing Dataset 29
3.1.2 Adding Artificial Fog into Images 32
3.1.3 Color Space Transformations 37
3.1.3.1 Comparison of Color Spaces 39
3.1.4 Dataset Preprocessing 47
3.2 Model Modifications 48
3.3 Hardware Equipment and Software Environments 50
4 Results 51
5 Conclusion 58
References 60
Appendices 66

Appendix A Publications from the Thesis 67

Appendix B Front Cover 68

Curriculum Vitae 69

List of Figures

Figure 2.1 RGB color space and channels of an image 6
Figure 2.2 Artificial Intelligence sub-fields 7
Figure 2.3 Activation functions, (a) ReLU, (b) Sigmoid 9
Figure 2.4 General architecture of CNN 11
Figure 2.5 Convolution operation 12
Figure 2.6 Convolution on image, (a) Original image, (b) Selected features of the original image convoluted with 4 different filters 12
Figure 2.7 Maximum and average pooling with 2x2 filters and (2x2) stride 14
Figure 2.8 Connected layers, (a) Fully connected layer, (b) Dropout operation 15
Figure 2.9 Flattening operation 15
Figure 2.10 VGG model architectures, (a) VGG 16, (b) VGG 19 17
Figure 2.11 Inception V3 model architecture 18
Figure 2.12 Inception-ResNetV2 model architecture 19
Figure 2.13 Xception model architecture 20
Figure 2.14 Dense model architectures (a) DenseNet 169, (b) DenseNet 201 21
Figure 2.15 Exclusive ReLU6 activation function used in MobileNetV2 22
Figure 2.16 MobileNetV2 bottleneck blocks 22
Figure 2.17 MobileNetV2 model architecture 23
Figure 2.18 Different resolution smoky images of video, (a) Low, (b) High 25
Figure 2.19 Different colored smoke images, (a) Darkly, (b) Whitely 26
Figure 2.20 Smoky images located in different places, (a) Left, (b) Right 26
Figure 2.21 Smoke footage from different distances, (a) Short, (b) Long 27
Figure 2.22 Cloudy images, (a)Smoky, (b) Normal 27
Figure 2.23 Sunlight reflected images, (a) Smoky, (b) Normal 28
Figure 2.24 Normal footages, (a) Close, (b) Far 28

Figure 3.1 Black colored smoky images labeled as, (a) Smoky, (b) Foggy-smoky . 34
Figure 3.2 White colored smoky images labeled as, (a) Smoky, (b) Foggy-smoky. 35

Figure 3.3 Long distances smoky images labeled as, (a) Smoky, (b) Foggy-smoky 35
Figure 3.4 Selected normal images labeled as, (a, c, e) Normal, (b, d, e) Foggy 36
Figure 3.5 RGB color space, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled images 40
Figure 3.6 YUV color space, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled images 41
Figure 3.7 $L^* A^* B^*$ color space, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled images 42
Figure 3.8 HSV color space, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled images..... 43
Figure 3.9 HSV color space H-Channel, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled..... 44
Figure 3.10 HSV color space S-Channel, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled 45
Figure 3.11 HSV color space V-Channel, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled..... 46
Figure 3.12 Histogram of S-Channel Images, (a) Figure 3.10a, (b) Figure 3.10b, (c) Figure 3.10c, (d) Figure 3.10d..... 47
Figure 3.13 The graph of number of training, validation and testing images 48
Figure 3.14 Model with 4-labeled Softmax layer..... 49

Figure 4.1 Color space comparison after 5 sessions over all architectures 52
Figure 4.2 False predicted images with proposed model 54

List of Tables

Table 3.1 Dataset prepared by splitting videos into images 31
Table 3.2 Collected images 32
Table 3.3 Model training parameters 49

Table 4.1 Model highest accuracy (%) 51
Table 4.2 Average accuracy (%) of models 52
Table 4.3 Confusion matrix of the best MobileNetV2-HSV Score 53
Table 4.4 Evaluation with various metrics of the best MobileNetV2-HSV Score .. 54
Table 4.5 Model comparison based on the number of parameters, approximately elapsed training time, and disk usage 55
Table 4.6 Comparison between the literature and proposed methods 56

List of Abbreviations

ANN	Artificial Neural Network
CCTV	Closed Circuit Television
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DenseNet	Densely Connected Network
FPS	Frame per Second
GPU	Graphics Processing Unit
HSV	Hue Saturation Value
ILSVRC	The ImageNet Large Scale Visual Recognition Challenge
MATLAB	Matrix Laboratory
OpenCV	Open-Source Computer Vision
RAM	Random Access Memory
ReLU	Rectified Linear Units
ResNet	Residual Network
RGB	Red Green Blue
SGD	Stochastic Gradient Descent
VGG	Visual Geometry Group

List of Symbols

β	FPS [frame/s]
δ	Duration of video [s]
λ	Label coefficient of video splitting
ξ	Total frame of video

Chapter 1

Introduction

Wildfires are one of the most serious natural disasters nowadays, as they spread very rapidly. Therefore, wildfire detection has an important role in the safety of the environment and people. The most important stage in this detection is its beginning time. At this stage, the fire emits only visible smoke. For this reason, the detection of smoke also brings about interfering with the fires.

Due to the smoke rises upwards, it can be noticed even from far away. By means of this attribute of the smoke, sensors placed on the ceilings of confined spaces for detection. Nonetheless these sensors, remain incapable in open-air spaces due to the fact that presence of winds and flammable wood materials. This incapability, occurring especially in forest fires, paves the way for wasting of resources, extinction of many living species, global warming, and greenhouse gas emission. Therefore, the studies mostly focused on developing detectors for smoke in open-air areas.

By means of smart monitoring systems in major cities, it makes it possible to detect smoke with computer vision systems. However, these systems are not efficient for some environmental factors such as fog, dust, and clouds. One of the most challenging factors in detecting smoke in open-air areas is the presence of fog in the environment. Especially in recent years, largely populated cities have been affected by the formation of fog due to air pollution. In fact, fog and smoke can be detected separately from each other. While the smoke is usually concentrated in an area, the fog is more homogeneously spread around the environment. While smoke can be of various colors and densities depending on the type of flammable substance, fog is generally like a white tulle. Nevertheless, detecting smoke in foggy environments is a much more difficult problem than detecting both separately.

1.1 Motivation

In classical smoke detection studies, color space-based approaches are often used in determining the smoky area in the image. In these studies, the features required for smoke detection are usually extracted manually. Nowadays, the success of classical image processing-based smoke detection studies has been left behind with the execution of artificial intelligence studies in this field. In artificial intelligence-based approaches, features can be automatically extracted with a series of filtering methods using deep learning-based models. However, obtaining higher results in systems based on deep learning requires a large number of examples to be trained. Notwithstanding, due to lacking dataset much deep learning-based smoke detection models are trained only in footage from optimum weather conditions. On the contrary, models that are not trained for harsh weather conditions like the foggy environment will not classify the smoke correctly.

In this thesis, a method is proposed for the integration of the classical color space-based approach with the modern deep learning approach to detect smoke in the foggy environment. Choosing the right color space has been an important parameter to increase performance. It has been noticed that there is a lack of literature studies in the field of deep learning approaches in smoke detection in different color spaces for real-world conditions. It is also planned to overcome the lack of the dataset by creating artificial foggy images from smoky and normal images. After overcoming the lack of the dataset, classification using modern convolutional neural networks that sub-branch of deep learning is proposed. With the proposed method, it is aimed to detect smoke in the foggy environment with high accuracy.

1.2 Thesis Organization

Today, many successful methods have been proposed for image processing-based smoke detection systems. However, recent studies focus on artificial intelligence-based approaches that allow machines to make decisions like humans. For establishing a base for this thesis, studies in the literature for computer vision-based smoke detection systems, color spaces and deep learning background information, smoke detection databases are given in Chapter 2.

In Chapter 3, is proposed methodology of this thesis, for this, dataset images are collected from literature database video sources. This dataset images are increased by adding artificial fog. After artificial fog is added, the data set is transformed into various color spaces in order to better detect smoke from images. Training sessions are carried out with modern deep learning architectures using this data set transformed into color spaces. In addition to these, the hardware equipment and software environments where the training sessions carrying out are explained.

In Chapter 4, the results were obtained when the training sessions are finished. These results are schematized with various tables and graphics. Numerous modern deep learning models for smoke detection in foggy environments are trained in various color spaces. The models were compared with each other and the model with the highest accuracy and the lowest memory requirement is chosen as the proposed model. In addition, color spaces are compared with each other in terms of accuracy level. The results obtained are compared with other results in the literature.

In Chapter 5, the general evaluation of the thesis, its conclusion and achievements that will shed light on future studies are described.

Chapter 2

Background

Smoke, due to its structure, does not have a fixed distinctive shape, can occur in any environment where there is air, its color changes with the type of flammable substance and can be transparent with the background color, and also it can be unclear in cloudy or foggy environments. By using this color-oriented characteristic of the smoke, studies were carried out to determine the places where the smoky regions in footages. In these studies, color space in which color is represented as a numerical value was also discussed [1].

Artificial intelligence means machines that perform the ascriptitious tasks to human beings by imitating human intelligence. In this way, people are prevented for spending time with their work, and both labor and time are saved. In subfields of artificial intelligence such as machine learning, images are processed, and features are extracted from them. The model is trained using these features, and by means of this model, an image is classified with a label. Using machine learning, classification studies on whether there is smoke in the image were made in the problems of detecting smoke from the image. In these studies color space based features were obtained from smoke images manually, have been used for classification by machines [2]. By editing fog, light, and noise of frames to adapt the problem more related to real-world conditions, studies were also carried out the detection of smoke based on machine learning [3].

In recent years, the success achieved with deep learning methods in classification exceeds the success point arrived at machine learning. Recent studies in the literature were focused on the use of the deep learning-based smoke detection [4]. The performance of different segmentation approaches based on deep learning-based smoke detection was investigated [5]. There are studies in the literature that increase the classification performance with regard to simple models by combining various

model features used as inputs to the networks. [6]. Studies were conducted on the effect of the deep learning-based color space approach on smoke detection [7, 8]. Additionally, there are studies in which smoke images in the used dataset are also created artificially. The performance of synthetic and contaminated smoke images with deep learning-based approaches was studied [9-12]. In particular, generator and discriminator based deep learning models used to generate new data, were used in smoke detection problems [13]. The size of the smoke in the footages can be small or large. Smaller smokes are more difficult to detect from the image than large ones. Studies were made to detect small-sized smoke with deep learning-based models under normal and harsh weather conditions [14]. Studies in which deep learning and machine learning used together were also conducted. The features obtained by deep learning were tested with machine learning classifiers [15]. Unmanned aerial vehicles were originally produced for monitoring purposes and were subsequently used for smoke detection and fire extinguishing purposes. Smoky images obtained from unmanned aerial vehicles were trained in different color spaces with deep learning and machine learning-based approaches [16]. Deep learning models can be applied in embedded systems for real time applications. Hence, studies on real-time smoke detection model deployed embedded system based on deep learning were implemented [17]. All these studies show that there are many approaches in the literature about smoke detection and these approaches can work integrated with each other. In brief, deep learning-based studies are become more popular in recent years due they have been more successful.

2.1 Color Spaces

Color is the wavelength of light reflected from an object. While the light is reflected, it is partially absorbed, thus the colors are very diverse. By virtue of this diversity brought about the need to standardize the colors and display them. Color spaces are proposed to implement these standards in digital imaging with computer graphics. Color spaces are composed of channels. A color is formed by combining the values in these channels of space at a certain rate.

A lot of color spaces are proposed to apply color representation in various computer graphics applications [18]. The reason for the color space diversity is the perceptual

quality or technical characteristics of the color of the systems used to define the color. The perceptual quality of color defines how well the human perceives the color. The technical characteristic of the color defines the wavelength value of that color. By using these color spaces, the same color can be defined in different spaces. RGB is a three-dimensional cartesian coordinate system, consists of the of red, green and blue color channels as shown in Figure 2.1.

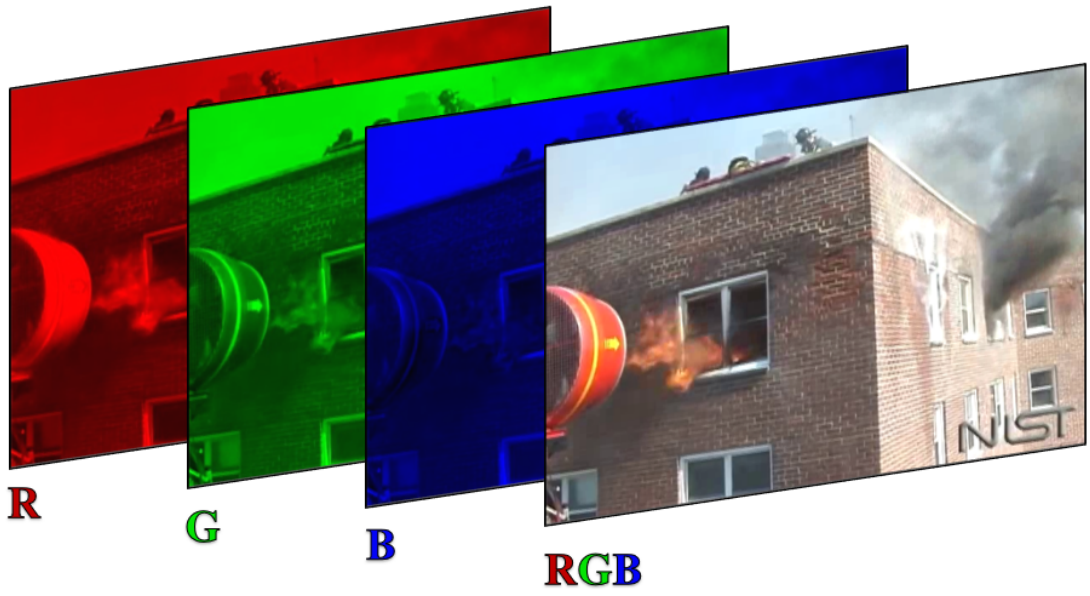


Figure 2.1: RGB color space and channels of an image [19]

YUV color space, channels consist of luminance, blue-based chrominance and red-based chrominance channels [20]. In the $L^*A^*B^*$ color space, L^* corresponds to the luminance value, A^* corresponds to the color on the red-green axis, and B^* to the color on the blue-yellow axis [21]. Similarly, HSV is another color space composed of hue, saturation and value color channels [22]. Choosing the appropriate color space to emphasize the desired area in an image can significantly improve performance. From this emphasized area, the features that better define the problem can be extracted. Color spaces can be transformed into each other using various functions. This transformation is done due to the fact that the transmission between various devices used different color spaces or emphasizing the desired areas of the image.

2.2 Machine Learning

Artificial intelligence-based systems show great progress with the increase of studies in this field. Especially in recent years, artificial intelligence supplies, better outcomes than even human experts [23]. Just as people benefit from their experience while doing a job, artificial intelligence also benefits from data. In this field it is aimed to find appropriate results to solve the problem by making use of statistical models and patterns among the data. Artificial intelligence hosts many sub-fields. These sub-fields are shown in Figure 2.2 and explained in the following headings.

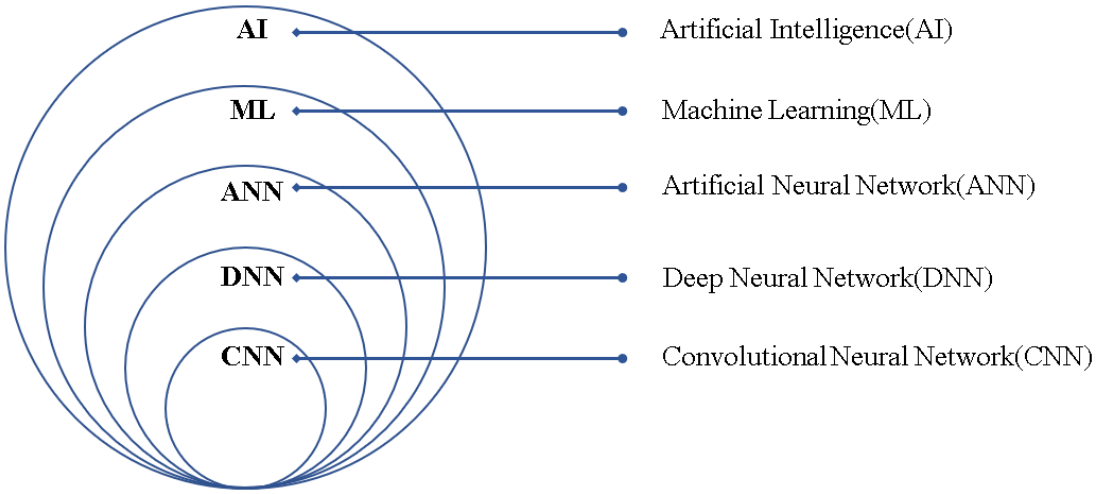


Figure 2.2: Artificial Intelligence sub-fields

Machine learning is an approach for the estimation of action by performing mathematical operations on data. This approach is based on a machine model that can learn like humans and can be used to solve complex problems.

Data is used in machine learning and splits in training, validation and test sets. The training set is used for model fitting, the validation set is used to tuning hyperparameters with unbiased evaluation, and the test set is used to calculate model performance. While learning like humans, the machine can create a function between the labelled input data and the labelled output data. This type of learning where both input and output are labelled is called supervised learning. Classification is one of the

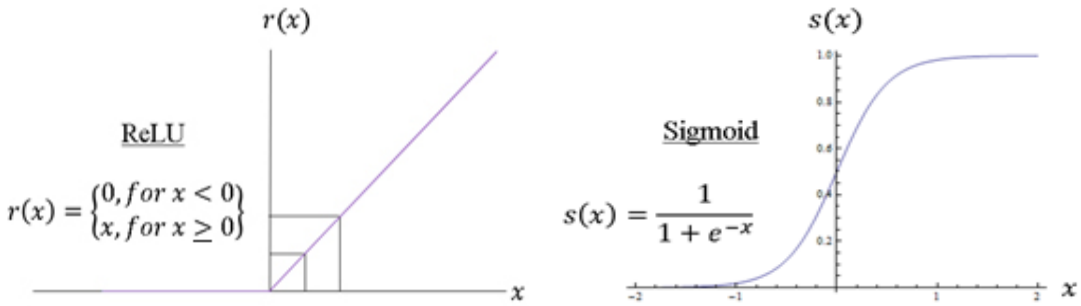
types used to predict categorical data in supervised learning. In the classification problem, features are used to extract meaningful values from data. In an image dataset, these features can be brightness, contrast, histogram, vertices, and edges. In machine learning, the failure to extracting the features suitable for the problem or selecting appropriate learning parameters results in overfitting or underfitting. In general, an overfitting model has redundant parameters, while on the contrary, an underfitting model has a lack/wrong parameters chosen to generalize the problem [24].

Classification is made on the test dataset with the weights obtained using train and validation datasets. Accuracy is used to determine the performance of the model as a result of the classification. On the other hand, the accuracy is not sufficient parameter to confirm the correctness of the model. According to the class distribution of the samples in the test dataset, it is necessary to clarify which prediction the model made for which class. The matrix that shows to what accurate the model classifies the test dataset consisting of samples of different classes is called a confusion matrix.

One of the sub-branches of machine learning used today is the artificial neural networks (ANN) that aim to learn by imitating the working mechanism of the human brain. The human brain actually works with the principle that many neurons connected to each other with the help of axons and dendrites interact with their synapses and take action. The single layer neural network model is called the perceptron, and perceptrons have interconnected each other just like neurons via weight parameters. The weight parameter and the bias independent variable are used by the neural network to produce the correct output and they are also updated until the iteration is over. This process, in which weight and bias are updated with forward and backwardly in ANN, is called forward and backward propagation. The number of samples in forward and backward propagation at the same time is named as batch size. The first derivative-based gradient descent algorithm is often preferred for propagation algorithm [25]. The Stochastic Gradient Descent (SGD) algorithm, that uses the cost function over the gradient of a sample instead of all samples, is often preferred among these algorithms.

In ANN, an iteration is called as epoch. At the end of the epoch, it is aimed to prevent false prediction and minimize the cost function. The activation function is applied to an input before it is transmitted to the output of ANN. Activation is a kind of threshold function that decides whether transmit to the output or not the value obtained from the

perceptron. If the activation function is not used, the function remains a one-degree linear polynomial. In this state, the activation function can not solve requiring multiple perceptrons complex real-world problems [26]. Rectified linear units (ReLU) and Sigmoid are generally preferred as the activation functions [27]. Softmax function, which is a kind of Sigmoid function, is used when classifying in multiple outputs. Figure 2.3 shows the mathematical model and thegraphs of these activation functions. In Figure 2.3, The ReLU function takes the value 0 for negative values and produces the same output between 0 and positive inputs. On the other hand, Sigmoid function produces between 0 and 1 as output.



(a) (b)

Figure 2.3: Activation functions, (a) ReLU, (b) Sigmoid

An ANN consisting of a single perceptron is insufficient to solve non-linear problems. Perceptrons can be connected to each other in various ways, just like neurons. In order to solve ANN problems such as exclusive-or perceptrons should be placed in successive layers [28]. Models consisting of more than one perceptron in many layers named as multi-layer perceptron are used for solving more complicated problems.

If there is more than one layer including perceptrons between the input layer and the output layer in an ANN, it is called a deep neural network. It is the presence of more than one hidden layer between the input and output layers that differences a deep neural network from an ANN. Training using multilayer ANN is called deep learning. Deep learning actually has been found in theory long before, it has recently become popular with the massive increase in processor capacity and the amount of data. By using deep learning, applications were developed for our daily life without human

intervention in areas such as translators, autonomous vehicles, disease diagnosis, object detection, face recognition, chatbots, search engines, marketing recommendations, elderly and disabled people assistant services, robotic technologies and the entertainment sector [29].

2.3 Convolutional Neural Network

Today, as the amount of visual data increases, the need to extract meaningful features from this data has emerged. With a convolutional neural network (CNN) model, accessing this meaningful information is not possible with a smaller number of perceptrons. At this point, CNN has been proposed to obtain the features by using visual data [30]. CNN is a kind of deep learning algorithm based on ANN and mostly used in computer vision, object detection, image processing and classification fields [31]. In CNN there is not used exact feature extraction. The model learns to do feature extraction properly via convolutional layers. The main reason why CNN has become popular in recent years is the development of computing resource hardware equipment, especially in graphical process units (GPU), and image processing fields related to image filters. With these developments over the years, CNN is used in image and video classification by using its many layers.

2.3.1 CNN Layers

The data to be used as input should be arranged with processing methods in order to ensure stability before beginning the convolution process. This arrangement may include the steps like the fixing of the image/video sizes. A classic CNN consists of several sequential convolutional layers and pooling layers. At the end of the convolutional and pooling layers, features are obtained. The achieved features are transmitted to the flattening and connected layers. At the last stage, the Softmax activation layer takes place, and the classification is made as a result of this layer. This general architecture of CNN is shown in Figure 2.4.

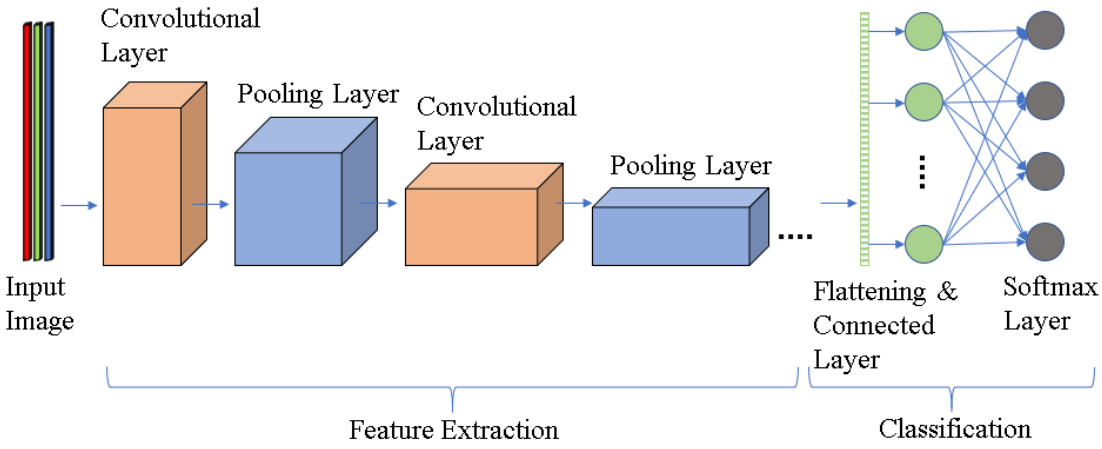


Figure 2.4: General architecture of CNN

2.3.1.1 Convolution Layer

Convolution is originally a mathematical operation, it is used to extract meaningful features from the image by applying filters. In the CNN algorithm, the convolution is the most important structure, and it is used to extract appropriate features to the problem from an input matrix (this matrix can be video or photo) according to the solution of the problem. While creating CNN architectures, arranged at the entrance layer is generally the convolution layer. In the convolution layer, various filters are applied to extract the low and high-level features in the image, such as detecting the edges of the image, removing the noise in the image. These filters can also be placed consecutively, depending on the type of features [32]. In the convolution process, the symmetry of the filter with respect to the x and y-axis is taken and applied to the two-dimensional input matrix. All values in the filter are multiplied by the corresponding element in the input matrix, and the sum of all multiplied values is assigned as the corresponding element of the output matrix. This operation is shown in Figure 2.5.

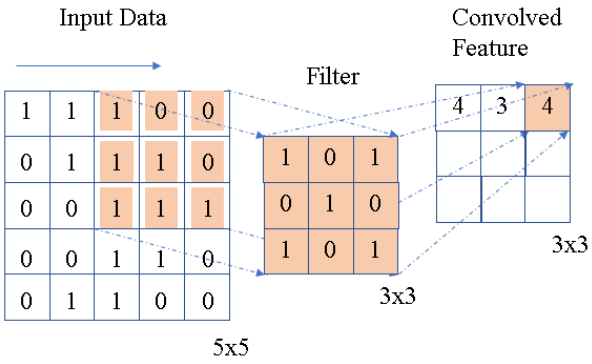
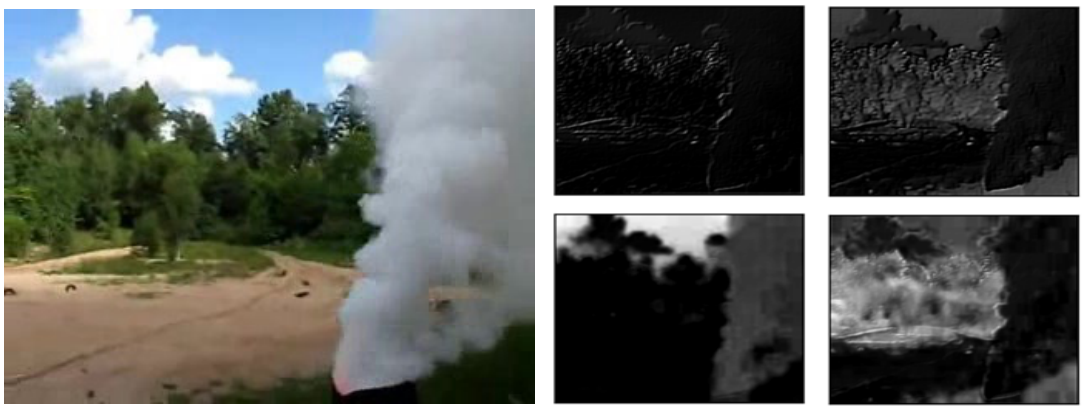


Figure 2.5: Convolution operation

The filters are operated on the all-input image with shifts over. A number of pixels can also be skipped while shifting over. This skipping is called as stride. The output matrix in other words output image or feature map, shrinks after the convolution operation according to the input matrix size. A sample image and convolved features of the image are shown in Figure 2.6.



(a) (b)

Figure 2.6: Convolution on an image, (a) Original image [19], (b) Selected features of the original image convoluted with 4 different filters

Pixels are added to the output matrix to prevent this shrinkage. This adding called as padding, zeros are added from all four sides of frame. Output matrix size m is found by:

$$m = \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \tag{2.1}$$

Here, p is the pixel size padded to input matrix, s is the stride number, the input matrix size is n and the filter matrix size is f, is applied to detect m via floor function.

At the end of the convolution layers there are usually ReLU activation functions. These functions, used to speed up learning in the network, transform the feature map from linear to the non-linear input function.

2.3.1.2 Pooling Layer

Following the convolution layer generally the pooling layer is used. The pooling layer significantly decreases the number of parameters in the network. As the matrix size decreases, the loss of matrix information increases, whereas the learning speed increases. Pooling layer paves the way for less calculation operations in the next layers due to this shrinkage of the matrix and prevents the system from overfitting. In the pooling layer, maximum and average filters are generally used for matrix shrinkage.

These filters can be in a fixed form specific to the architecture or selected according to the requirement of the problem. Pooling filters can be operated on the input image with strides just as in the convolutional filters. The size of the network can be further reduced with the preferred pooling layers along with the strides. Due to this feature, the pooling layer is also called the down-sampling layer. Figure 2.7 shows the application of average and maximum (2,2) pooling layers with (2,2) stride.

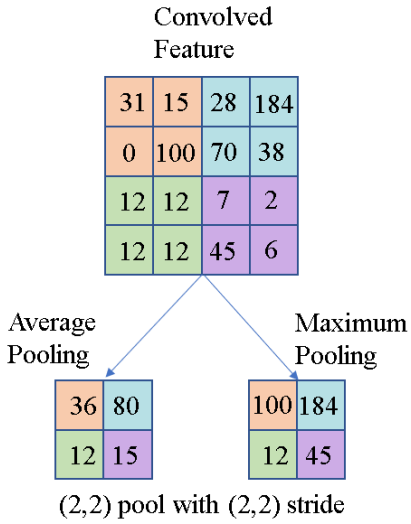


Figure 2.7: Maximum and average pooling with 2x2 filters and (2x2) stride

2.3.1.3 Batch Normalization

Batch normalization is used for scaling the layer values between 0 and 1. By normalizing the layers in this way, the performance of CNN increases. By means of this normalization process with batch normalization, the vanishing gradients that occurred during training in CNN are largely prevented. In terms of the machine learning evaluation, better performance can be achieved with less number of epochs by using batch normalization layers [33].

2.3.1.4 Dropout Layer

Dropout layer avoids overfitting by ignoring parameters with certain rate between 0 to 1. It is assumed that the removal of some connections within the network parameters will improve training performance. When training is carried out in networks with very large structures, this layer allows to randomly eliminate the connection between perceptron in certain rate. This layer is usually placed between connected layers [34]. Connected layer net and dropout operation to it are shown in Figure 2.8. By applying dropout operation, one of the 3 perceptrons in the left layer and half of the perceptrons in the middle layer are closed.

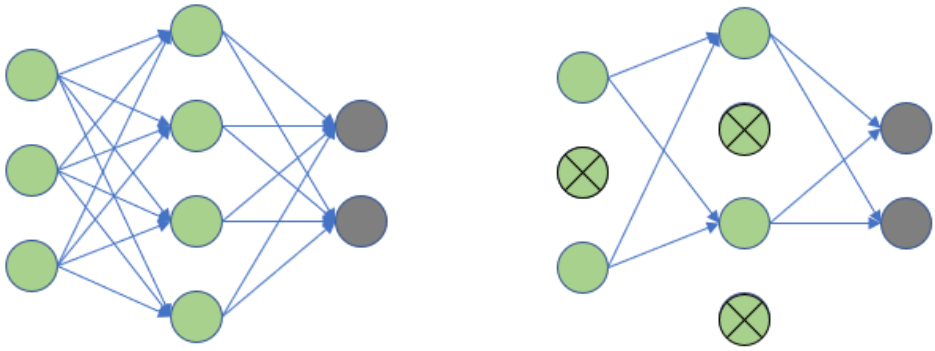


Figure 2.8: Connected layers, (a) Fully connected layer, (b) Dropout operation

2.3.1.5 Connected Layer

After the convolutional and pooling layers, flattening layer is placed in CNN architecture. Flattening layer is used in changing matrix shape before the connected layers. By means of the flattening layer, the input matrix of connected layers resizes to one-dimensional array. This operation can be seen in Figure 2.9.

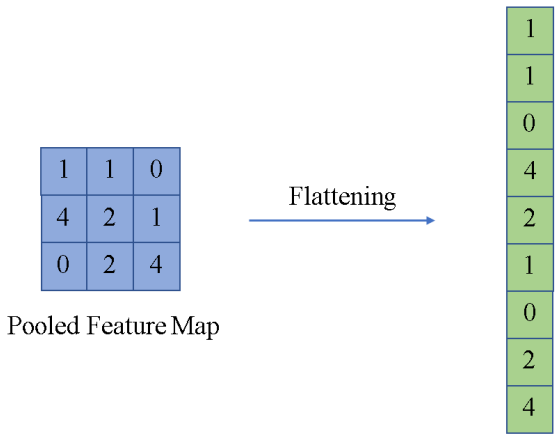


Figure 2.9: Flattening operation

The fully connected layer following the flattening layer, is operated for connecting all the inputs to all neurons. The operations performed in fully connected layers are similar to the calculations performed in the multilayer ANN. The architecture is completed with building these fully connected layers ending with activation function.

For classification problems, Softmax function is used to make a decision for the best option of multi output that has the highest probability.

2.3.2 CNN Architecture Models

In CNN, models compete with each other every year to obtain better results in visual object detection and classification. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is one of the annual computer vision competition [35]. Since 2012, the minimum error rate in this challenge has been taken by using CNN-based systems [36]. The ImageNet dataset has 14 million images with 1000 classes in RGB channels. Therefore, models that achieve high results in this challenge are known as models that have proven their success. In this way, models are frequently preferred in classification problems in different datasets. Below, models with proven success in the challenge and used in this thesis will be described.

2.3.2.1 VGG

Visual Geometry Group (VGG) model won 1st and 2nd place with its VGG 16 network on the classification and detection categories respectively in the 2014 ILSVRC challenge [37]. This model was accomplished 92,7% top-5 test accuracy on the ImageNet dataset. Generally, this model shows that using more filters of smaller matrix size involves fewer parameters than using a single large size filter. In this context, the same performance can be obtained with fewer parameters when small size convolution filters are used in succession. The difference that distinguishes this model from the previous literature models is that the pooling layer places after the double-triple (quadruple for VGG 19) mini-size convolutional layers [37]. VGG 16 and VGG 19 architectures of the models can be seen in Figure 2.10.

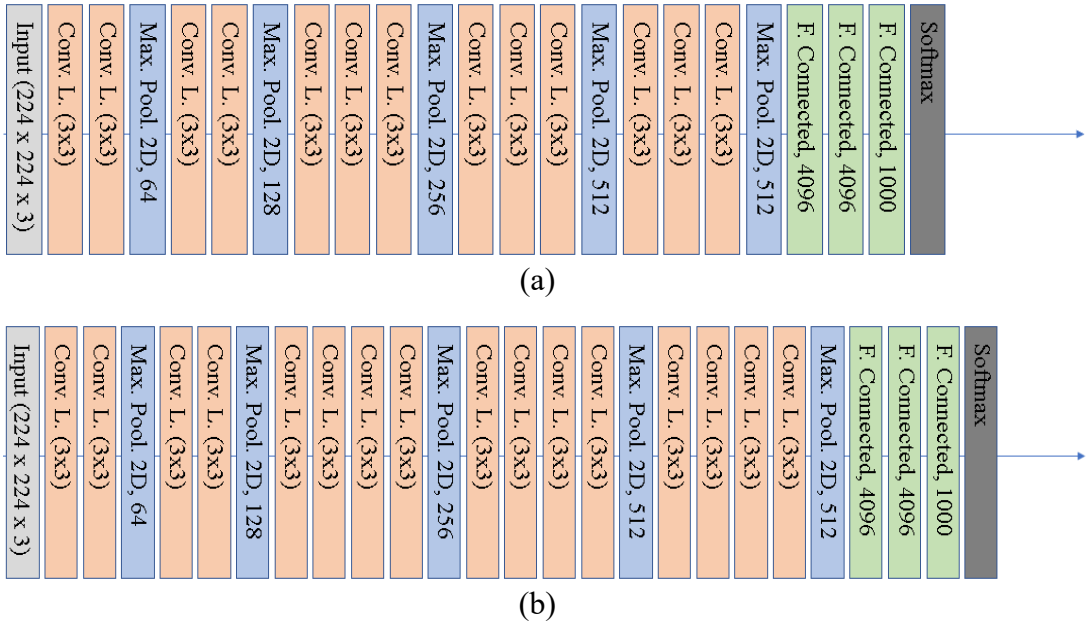


Figure 2.10: VGG model architectures, (a) VGG 16, (b) VGG 19

2.3.2.2 Inception V3

The Inception V3 Model is not a very deep but a broader focused CNN model. In an Inception model, many mini-network modules are built within the main network. Multiple convolution filters of different sizes are carried out together in the layer and the results obtained from these filters are concatenated. In the Inception V3 model, the number of filters and layers is increased, and also the convolutional filter size is reduced and enriched with batch normalization and fully connected layers. The Inception V3 model thus accomplished lower error rate success in the ImageNet dataset than the VGG model [38]. The architecture of this model is shown in Figure 2.11.

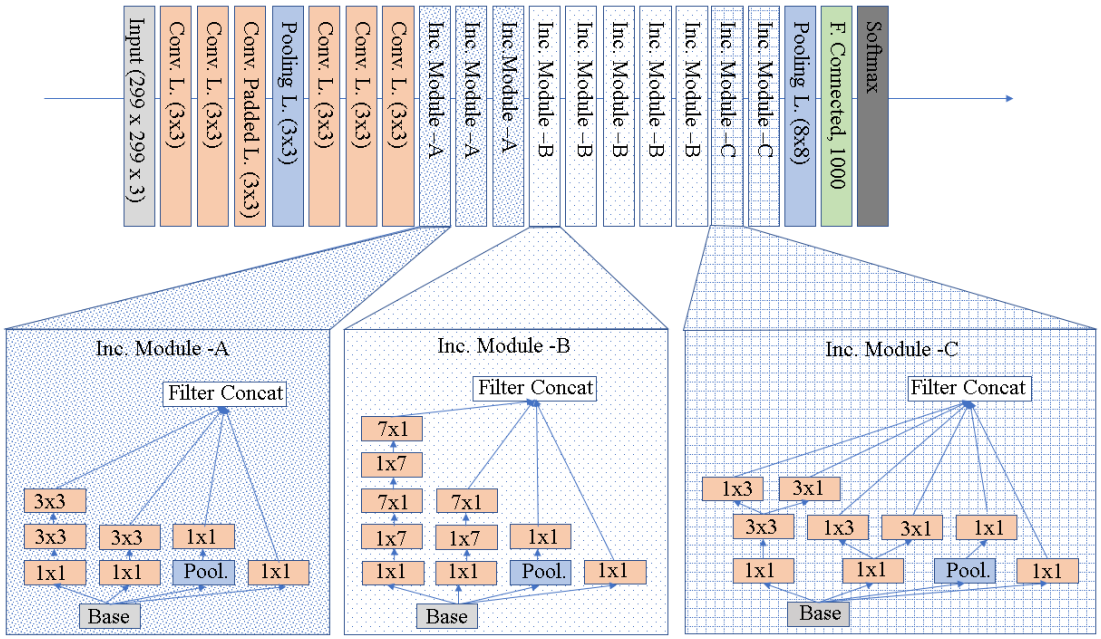


Figure 2.11: Inception V3 model architecture

2.3.2.3 Inception ResNetV2

As the number of CNN layers are increased (especially after about 20 layers), it occurs the problem of vanishing gradients. In this model, the problem is largely solved by assigning a shortcut path to each layer's input and output. By means of this method called Residual Networks (ResNet), the lowest error rate on the ImageNet dataset was reduced by using 152 layers [39]. The Inception-Resnet Model was created by combining the depth-based approach of the ResNet Model and the broad-based approach of the Inception Model. The Inception-ResNet Model includes the Inception-ResNetV1 and Inception-ResNetV2 models, whose general scheme is the same. Using the Inception-ResNetV2 model, a lower error rate was obtained on the ImageNet dataset than the InceptionV3 model [40]. Inception-ResNetV2 model architecture can be seen in Figure 2.12.

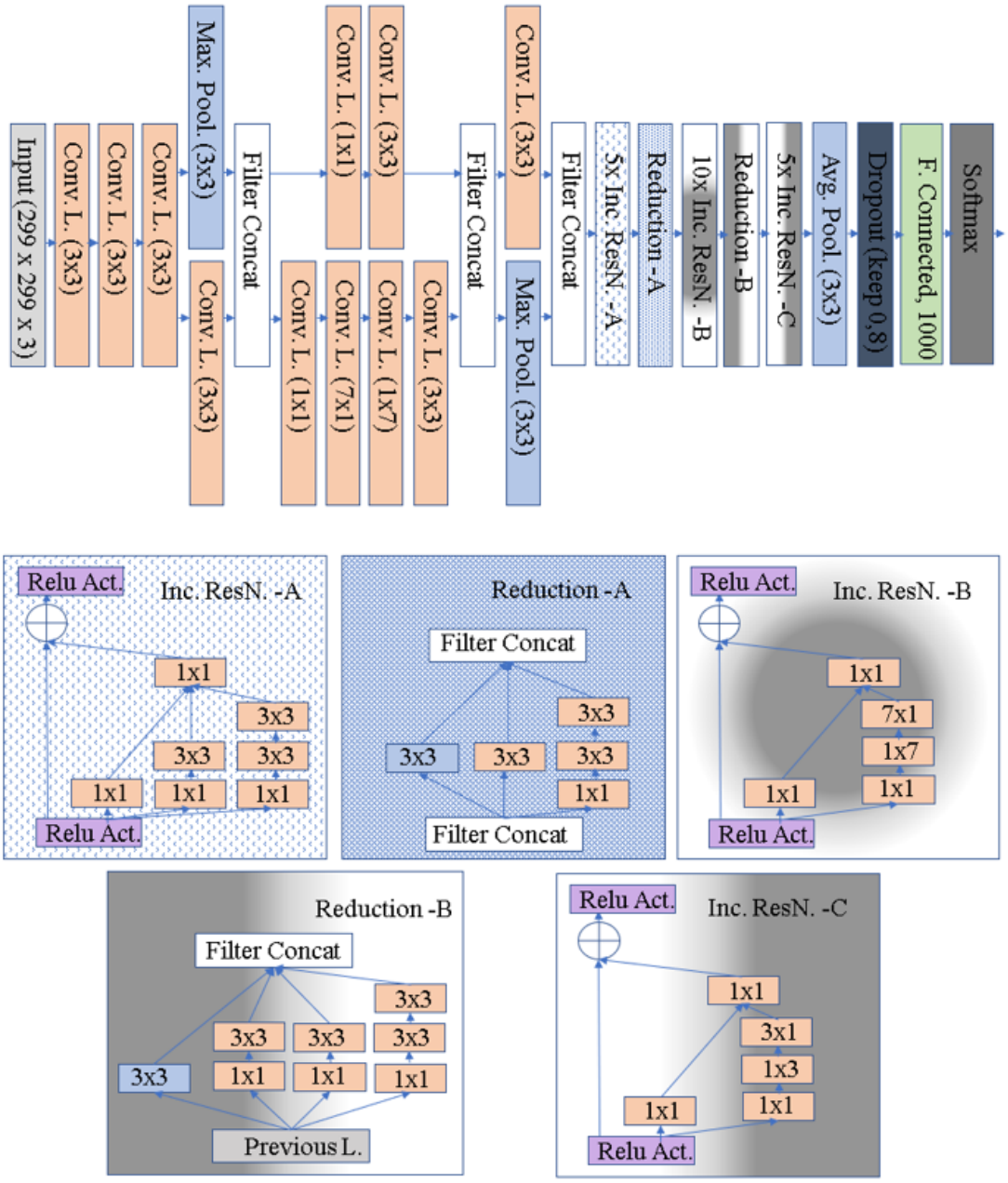


Figure 2.12: Inception-ResNetV2 model architecture

2.3.2.4 Xception

Xception is a model proposed to evolve the Inception V3 model result. The most important factor in achievements compared to the Inception based models is in the convolutional layer differences. Unlike the conventional convolution layers in the Inception model, depth-wise convolution and pointwise convolution processes are applied in this model. The number of multiplications and therefore the computing cost

in classical convolution processes is quite high. When depth-wise and pointwise convolution are used together named as separable convolution operation, the number of multiplications decreases considerably. Thus, the Xception model was obtained better performance on the ImageNet dataset with the same number of parameters as for the Inception V3 model, since the parameters were used effectively [41]. The Xception model architecture is shown in Figure 2.13.

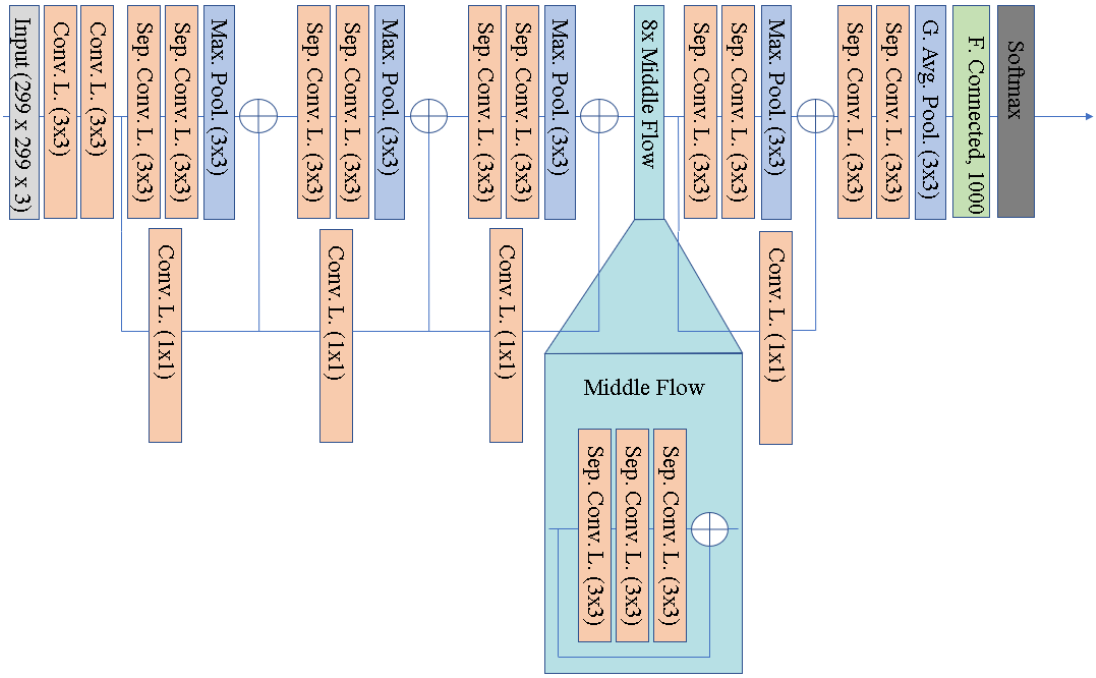


Figure 2.13: Xception model architecture

2.3.2.5 Densely Connected Networks

Studies in the field of CNN show that short-cuts between the entrance layers and the layers near to the output layers like in ResNet make the training more successful [42]. In the DenseNet approach, a difference from the ResNet approach, all layers in the Dense blocks transmit the features they extract to the next layer as input. Thus, while solving the problem of vanishing gradients occurring in deep CNN models, the features are reused. DenseNet model consists of dense blocks and between they include convolution and pooling layers called transition layers. By means of transition layers, feature maps are diversified, and their sizes are adjusted. The DenseNet

architecture consists of models such as DenseNet 169 and DenseNet 201 according to the number of convolution and pooling layers. The default memory space and the number of parameters vary between these two DenseNet models [43]. DenseNet 169 and DenseNet 201 architectures of the models can be seen in Figure 2.14.

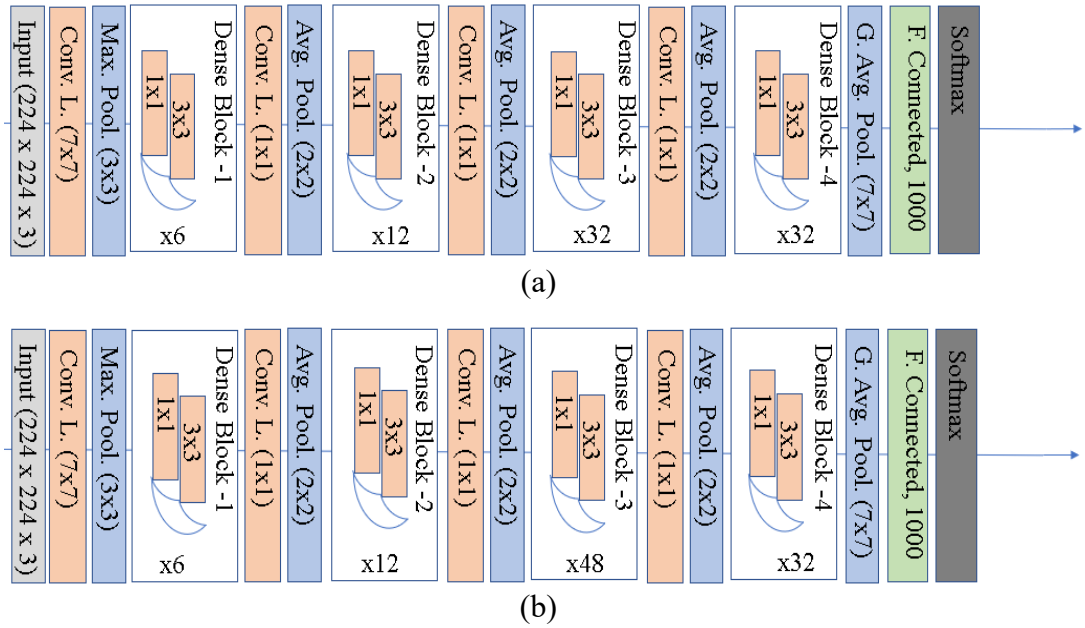


Figure 2.14: Dense model architectures (a) DenseNet 169, (b) DenseNet 201

2.3.2.5 MobileNetV2

In recent years, studies in the field of CNN have been directed to compose architectures with low parameters that will work effectively on mobile devices. MobileNetV2 architecture developed for this purpose consists of bottleneck residual blocks [44]. In bottleneck residual blocks, the amount of data flowing over the network is reduced with residual connection. In these blocks the ReLU6 function is used as the activation function, very similar to ReLU, but limits the activation to a maximum size of 6. The mathematical model of ReLU6 and its graphs are shown in Figure 2.15.

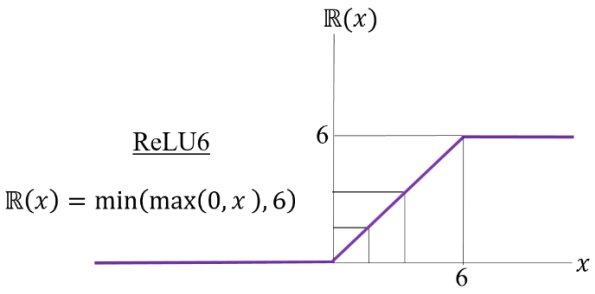


Figure 2.15: Exclusive ReLU6 activation function used in MobileNetV2

In these blocks, depth-wise and pointwise convolution processes are applied just like in the Xception architecture. However here there are two types of pointwise convolution processes as expansion and projection.

Expansion process is made on entrance of blocks and transmits data with a low number of dimensions (channels) into a tensor with a much higher number of dimensions. On the other hand, projection process has fewer output channels than input channels and places in exit of blocks. The factor in the expansion and projection operations is selected as 1 or 6. There is no pooling layer between the expansion and projection processes, instead stride value is selected as 2 to reduce the data size. Batch normalization is applied after every layer. Bottleneck blocks used in MobileNetV2 architecture are shown in Figure 2.16.

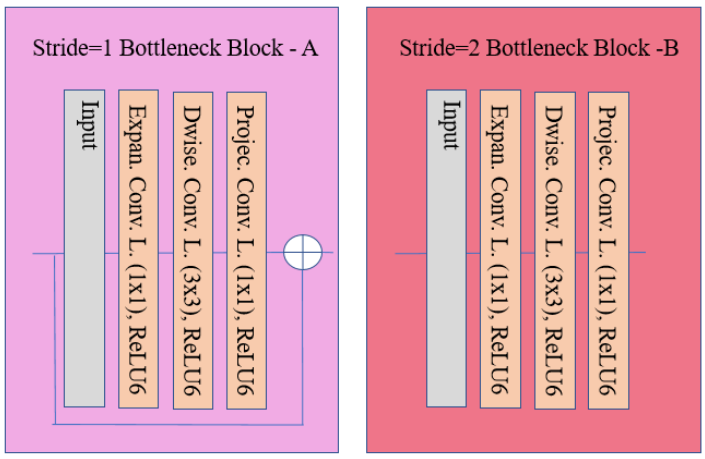


Figure 2.16: MobileNetV2 bottleneck blocks

By means of Figure 2.16, the MobileNetV2 architecture created using bottleneck blocks is shown in Figure 2.17.

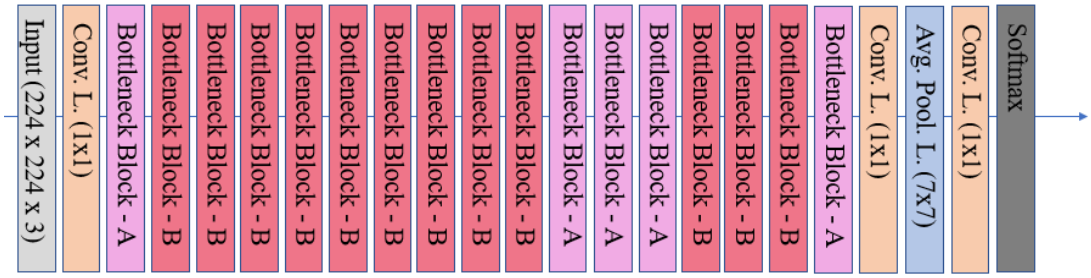


Figure 2.17: MobileNetV2 model architecture

2.4 Smoky and Normal Images Databases

Many studies were conducted in the literature for forest fire smoke detection implementations on computer vision-based systems. Some of these studies also keep the video sources they use as a database on the internet. By editing on databases created by academic institutions in various areas of the world were used as datasets. These databases, described below, are open source and obtained from various parts of the world using different smoke sources. Smoke generated places are mostly in outdoor environments. In addition, normal videos mostly consist of outdoor footage. The use of many databases has enabled the acquisition of footage in different environments.

2.4.1 Center for Wildfire Research Database

The smoke detection task is handled in this database image and video frame based. This database is collected from wildfire smoke sequences. In this database, the segmentation areas where the smoke area is detected for the whole image are also highlighted [45].

2.4.2 VisiFire Database

In this study, fire, smoke, forest smoke and normal video clips were used. In addition to detection with coding screen, Closed Circuit Television (CCTV) integrated graphical-user-interfaced software was developed [46].

2.4.3 Firesense Database

Firesense database consists of 13 smoky and 9 normal videos. This database is also subproject of the Projection of Cultural Heritage Areas from the Risk of Fire and Extreme Weather [19].

2.4.4 MIVIA Smoke Database

This database obtained in July 2012 and composed of 149 videos totally 35 hours recording. Some of these videos are include such as cloudy landscape without smoke, sun reflection and low luminance footage [47].

2.4.5 Wildfire Smoke Detection Database

This database of project is a part of the “Adaptive Systems for Environmental Monitoring” initiative. The database contains smoky and normal real and synthetic frame sequences [48].

2.4.6 Other Online Video Databases

There are many online video database websites on the Internet platform. Pond5 platform is New York based media company and includes stock music, video, and photography footage [49]. This platform is a kind of marketplace for commercial content producer comprises high-quality video with various tags. The other platform is HDNatureFootage that provides high-definition wildlife and nature stock video footage [50]. These platforms contain videos that do not contain smoke images. The reason for using HDNatureFootage and Pond5 is that normal video content is limited in academic databases.

2.4.7 Analysis of Database Footages

Since the videos in each database were recorded with different hardware products, they include videos of different resolutions. Utilizing videos of different resolutions allows the detection of smoke captured from different video sources. In this way, the trained model can classify the smoky images of video from different resolutions. Low- and high-resolution images of video can be examined from Figure 2.18.

(a) (b)

Figure 2.18: Different resolution smoky images of video, (a) Low [19], (b) High [48]

Having different colors of smoke in the videos accessed from databases is also an important parameter. The fact that the smoke in all images is the same color makes classification easier. Since the substance burning in real life cannot be chosen, different colored smokes will be emitted. Different colored smoke image of video samples in the databases can be seen in Figure 2.19.

(a) (b)

Figure 2.19: Different colored smoke images, (a) Darkly [46], (b) Whitely [19]

The different locations of the smoke area in the images can significantly improve training success. Searching for smoke in the same place in all images can be modelled as if smoke could only occur there. This results in wrong modelling. Examples of smoky images located in different places of video are shown in Figure 2.20.

(a) (b)

Figure 2.20: Smoky images located in different places, (a) Left [19], (b) Right [47]

The distance of the smoke in the video images from the footage area is also important in the detection of the smoke. In real life, it is impossible to have large numbers of equipment capable of footage in the forest with very short distances. For this reason, the smoke area in the images should not be too close to the area where the footage was taken. On the other hand, the detection of smoke from a very long distance also means

that a very long time passed since the fire started. In this situation the smoke already ascending overly and can be seen even from the shooting area. Actually, this situation shows us that it is late in classification. Therefore, in order to provide real-world conditions, both the smoke images are taken from a short and long distance takes place together. Images of video that smoke occurred in different distances are shown in Figure 2.21.

(a) (b)

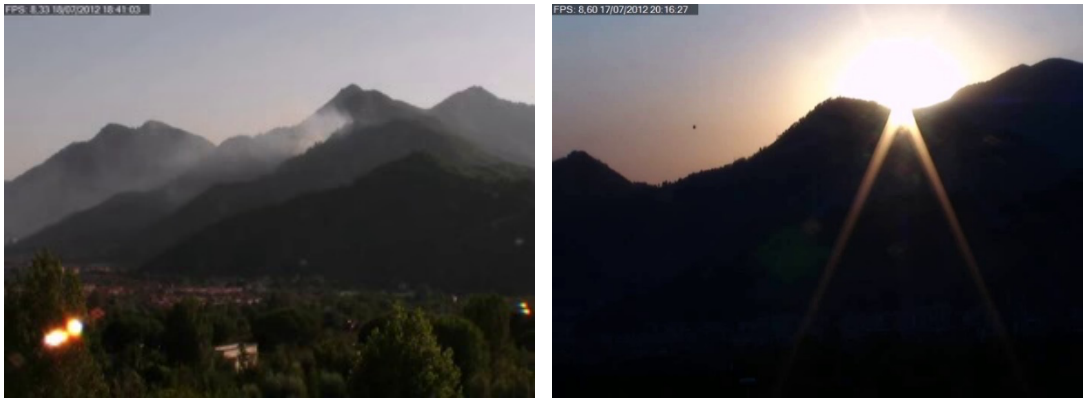
Figure 2.21: Smoke footage from different distances, (a) Short [19], (b) Long [45]

Due to smoke structure, it can be confused with clouds. This may result in a false alarm in the classification. Therefore, both images containing smoke in cloudy environments and the images containing cloudy environments without smoke include in the dataset. Selected images of video in cloudy environment are given in Figure 2.22.

(a) (b)

Figure 2.22: Cloudy images, (a) Smoky [48], (b) Normal [19]

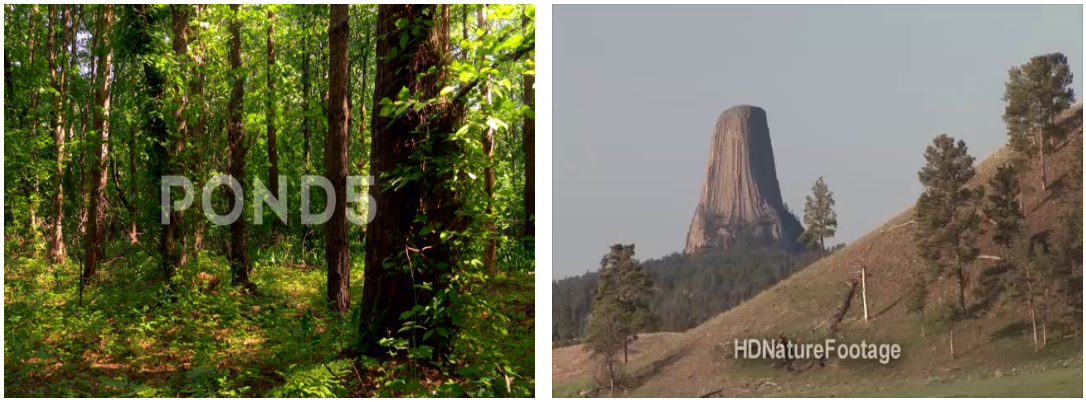
Sunlight may be reflected on the camera that capturing the footage using for smoke detection. The image is created by the camera due to the reflection of sunlight can be confused with the fire. Therefore, to make the correct classification by the model even under this condition, the dataset includes the shots containing such images. Sunlight reflected images in the dataset can be seen in Figure 2.23.



(a) (b)

Figure 2.23: Sunlight reflected images, (a) Smoky [47], (b) Normal [47]

Close and far shot footage without smoke are also included in the database as shown in Figure 2.24. Attention was paid to the fact that without smoke images are taken from the environment similar to those containing smoke.



(a) (b)

Figure 2.24: Normal footages, (a) Close [49], (b) Far [50]

Chapter 3

Methodology

There are many databases for smoke detection as mentioned in Chapter 2, but these databases do not supply the literature with foggy footage. Considering that deep learning-based studies need a lot of data to be able to model correctly, the first process in this Chapter was to create the dataset. Then, color spaces were used to make the perceptibility of smoke and fog more apparent in this dataset. After the dataset operations were completed, the stages of the deep learning model modifications align with the dataset were implemented. This chapter consists of three sections, in which dataset operations, model modifications, hardware equipment and software environmens.

3.1 Dataset Operations

In this section, the operations of the dataset are explained. These steps include the preparing dataset, the addition of artificial fog into the dataset, the color space transformations of the foggy images, and the pre-processing of dataset.

3.1.1 Preparing Dataset

In this preparation, firstly, the databases described in Chapter 2 have been accessed and downloaded. After obtaining videos from databases to detect smoke in various conditions, the stage of splitting the videos into images was implemented. Dataset using in this project is acquired with these images split from the videos.

While preparing dataset, a total of 188 videos consisting of 94 normal videos and 94 smoky videos were used. These videos consisted of different durations and different

number of frames per second (FPS). For this reason, the number of images obtained from videos is not equal. Homogenizing the dataset and using a close number of images from each video makes modelling more successful as it increases the possibility of training more diverse images. The number of images γ splitting from a video can be found by,

$$\xi = \delta \times \beta \tag{3.1}$$

$$\gamma = \frac{\xi}{\lambda_{1,2}} \tag{3.2}$$

Here, δ is the duration seconds of the video, β is the FPS, ξ is the total frame of the video, λ is the constant number assigned by each smoke labelled video for λ_1 and each normal labelled video for λ_2.

In the study, to obtain a close number of images with the studies in the literature λ_1 and λ_2 are chosen as 197 and 182 respectively [51]. After the Equation 3.2 can be detected the γ number. The first frame of each video is taken and saved, then the next frame is saved as an image, skipping with γ number of frames. This process is repeated until the last frame of the video. Thus, approximately close number of images are obtained from each video. This process is repeated for 188 videos and the saved images are divided into folders as “smoky” and “normal” according to their labels. The dataset created by splitting videos into images is shown in Table 3.1.

Table 3.1 Dataset prepared by splitting videos into images

No.	Video Name	Video Label	δ	β	Video Resolution	ξ	γ	Obtained Images
1	3-20…	Smoky	301	25	292*240	7525	38,19	198
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
106	3-20…	Normal	901	8,3	292*240	7508	41,25	183
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
154	test…	Smoky	109	4,69	320*240	511	2,59	257
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
167	test…	Normal	10	25	352*288	250	1	250
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
177	smok…	Smoky	240	50	1200*676	12000	60,91	76
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
188	V10…	Normal	17	29,97	640*360	510	2,79	256

As shown in Table 3.1, a total of 36110 images were acquired using these 188 smoky and normal labelled videos. These 36110 images were used as the dataset. The dataset obtained from databases is shown in Table 3.2.

Table 3.2 Collected images

Source	Video Address	Obtained Images
[45]	http://wildfire.fesb.hr	718
[46]	http://signal.ee.bilkent.edu.tr/VisiFire	1773
[19]	https://zenodo.org/record/836749#.XrFiavkzZEY	5255
[47]	https://mivia.unisa.it/datasets/video-analysis-datasets/smoke-detection-dataset/	24621
[48]	https://homes.di.unimi.it/genovese/wild/wildfire.htm	
[49]	https://www.pond5.com	3743
[50]	http://www.hdnaturefootage.net	
	Total	36110

3.1.2 Adding Artificial Fog into Images

The obtained 36110 images may contain sufficient information to detect smoke in normal weather conditions. However, in this situation, the dataset does not contain sufficient data to detect smoke in harsh weather conditions. The fog is an important parameter in the detection in outdoor environments due to confusion with smoke. Thus, the fog parameter was chosen to increase the alignment with real world condition of the smoke detection problem in this study. Fog can severely limit vision on long distances. In addition, the distinctiveness of colors also deteriorates in foggy environments. From this point, the fog significantly changes the brightness of the colors. Therefore, in the study, the fog was added artificially by changing the brightness value of all the images in the dataset.

In each channel of RGB color space, images have possible values ranging in 8-bit unsigned format from 0 to 255. As closer the pixel brightness value of the image to 255 in all channels, whiter image is obtained. Therefore, by increasing the brightness value of the image in 3 channels, it can be seen as if the image was taken from foggy environment footage.

Fog is added to the images as described below. In an image, assuming that Ω corresponds to highest brightness value of the pixels in each color channel, Ω can be found as follows.

$$\Omega = arg\ max\ (I_{R,G,B}(p)_{m\times n}) \tag{3.3}$$

Here, $I_{R,G,B}$ corresponds to an 8-bit 3-channel input image, p is the pixel, m and n are value of row and column respectively. Then, I_{whiten} is obtained by increasing the brightness value.

$$I_{whiten} = I_{R,G,B} + 100 \tag{3.4}$$

Where, I_{whiten} is an intermediate image formed by adding 100 brightness values to pixels of $I_{R,G,B}$. Again, ϕ corresponds to the the highest brightness value of the pixels in each color channel of I_{whiten} can be reached as follows.

$$\phi = arg\ max\ (I_{whiten}(p)_{m\times n}) \tag{3.5}$$

By multiplying all the pixels in I_{whiten} with the $\frac{\Omega}{\phi}$ value, the I_{foggy} output image is obtained. Afterwards, the result obtained is converted into an integer using the floor function.

$$I_{foggy} = \left\lfloor I_{whiten} \times \frac{\Omega}{\phi} \right\rfloor \tag{3.6}$$

After this multiplication, in order to limit I_{foggy} in 8-bit range thresholding operation is done.

$$I_{foggy}(p)_{m\times n} = \begin{Bmatrix} p, & for\ p \leq 255 \\ 255 & for\ p > 255 \end{Bmatrix} \quad (3.7)$$

Hence, an I_{foggy} 3-channel 8-bit image is obtained from an $I_{R,G,B}$ 3-channel 8-bit image.

The artificial fog was added to each of the 36110 images using in Equation 3.3 to 3.7. With the addition of artificial fog, the dataset consisting of 36110 images with 2 labels has been increased to 72220 images with 4 labels. Thus, "foggy-smoky" labelled images were created by adding fog to "smoky" labelled images, and "foggy" labelled images were also created by adding fog to "normal" labelled images. In this project, this dataset consisting of 72220 images and 4 labels was used in the deep learning stages.

Selected images and after applying adding fog of these can be seen from Figure 3.1, 3.2, 3,3, and 3.4. The fog in the images causes a loss of details generally. According to Figure 3.1(a) and 3.1(b), it has been found that the fog causes the dark-colored smoke to appear lighter in color.

(a) (b)

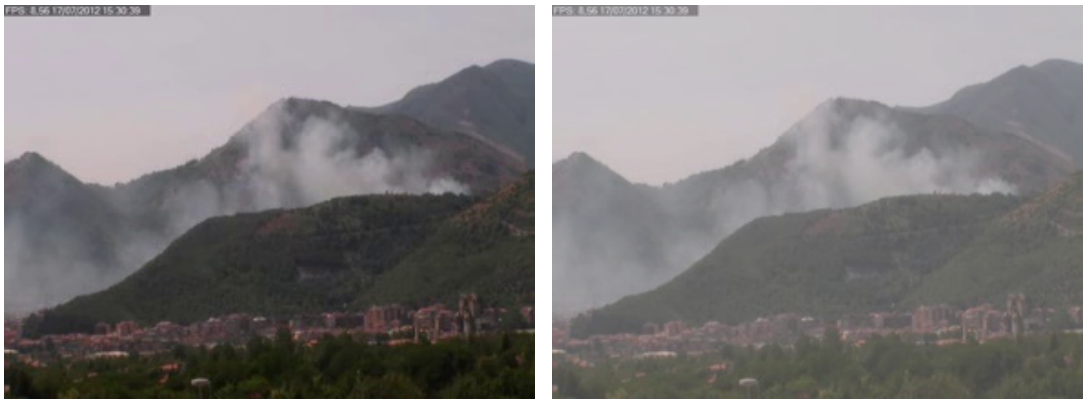
Figure 3.1: Black colored smoky images labeled as, (a) Smoky [46], (b) Foggy-smoky

Especially in Figure 3.2(a) and 3.2(b) areas with white colored smoke, the fog decreases the sharpness of the smoke.

(a) (b)

Figure 3.2: White colored smoky images labeled as, (a) Smoky [19], (b) Foggy-smoky

It has been noticed from Figure 3.3(a) and 3.3(b) that unless the smoke is very dense, it can be confused with fog, especially at long distances.

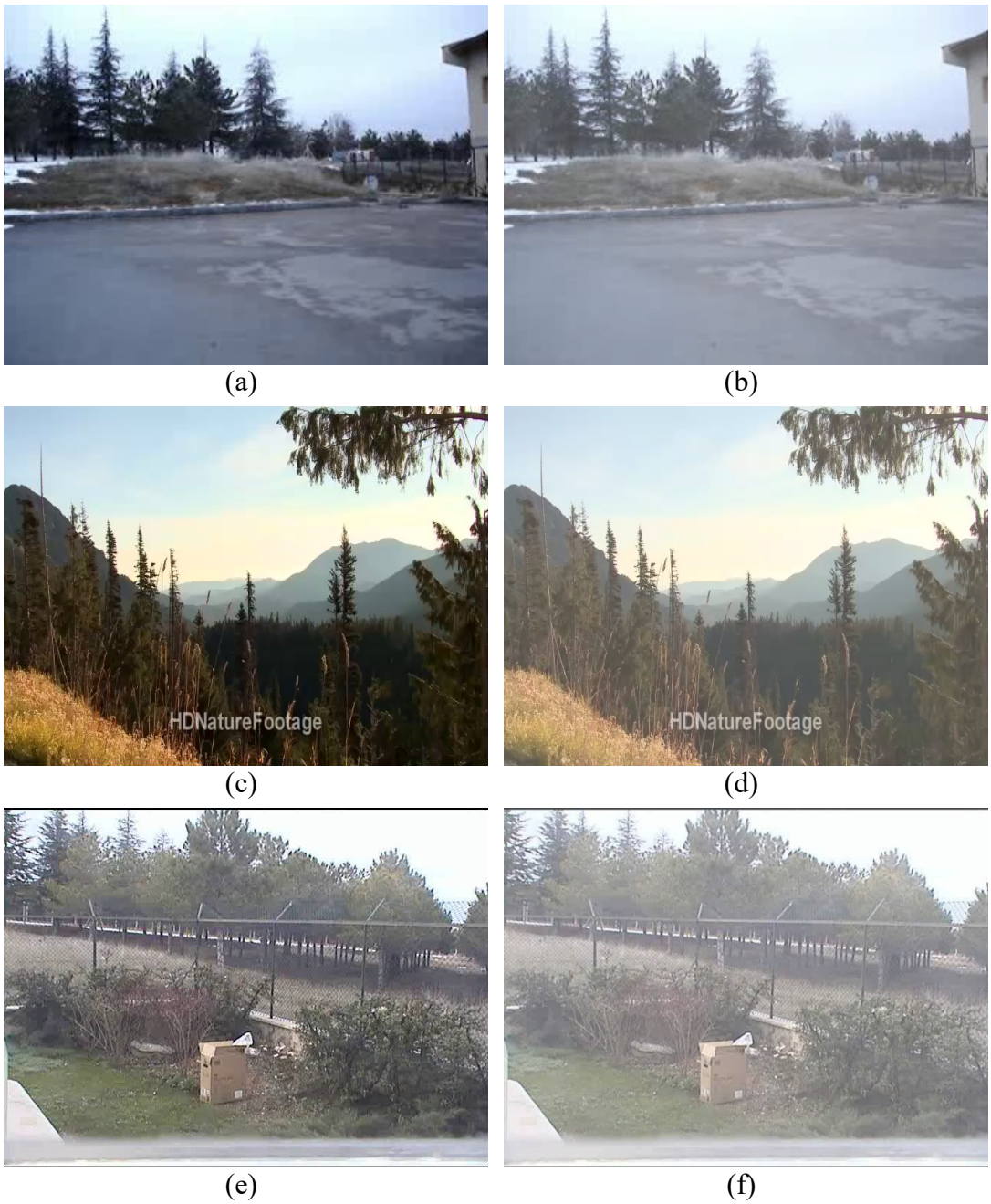


(a) (b)

Figure 3.3: Long distances smoky images labeled as, (a) Smoky [47], (b) Foggy-smoky

It has been noticed that also foggy labeled images can be confused with normal, and smoky labeled images in general. When fog is added to the images, the vividness of the colors has decreased. The dataset with the fog is added, increases the difficulty of

detection significantly. Thus, the possibility of confusing foggy labeled images and smoke labeled images with each other has increased considerably. Selected sample images can be seen in Figure 3.4.



(a) (b) (c) (d) (e) (f)

Figure 3.4: Selected normal images labeled as, (a [46],c [50],e [19]) Normal, (b,d,e) Foggy

The addition of fog is achieved by changing the brightness value of the image. To differentiate smoke from fog, research has been done on a parameter that is less affected by the fog brightness. It has been suggested that fog and smoke can be detected more easily by visualizing in different color spaces instead of RGB color space.

3.1.3 Color Space Transformations

When an image is read in digital media, it is mostly expressed in RGB color space. Although this color space is popular because it can easily distinguish 3 channels from each other, it does not always produce the most appropriate result for visualization. Depending on the visualization, different color spaces can be proposed.

Color spaces and their transformations can be various. This variety is mostly determined by concerning bit-range of the output matrix. In this thesis, both the input image and the output image are selected in 8-bit range. Therefore, at the end of the applied color space transformation, thresholding processes have been carried out to ensure that the output image is within this 8-bit range.

Images in the default RGB color space is transformed into YUV, L*A*B*, and HSV color spaces in order to distinguish better fog and smoke. It is aimed to extract the features that will make detection more successful with deep learning architectures by transforming the color space. Although there are many color space conversion methods that have been made, one of the most frequently used transformation methods has been chosen in this thesis.

The OpenCV library [52] mostly preferred in the literature was used in color space transformation and thresholding operations [20]. The operations in this library are based on previous studies in the literature [53, 54]. The formulas for transformation from RGB to YUV color space in this thesis is shown below.

$$Y = 0{,}299 \times R + 0{,}587 \times G + 0{,}114 \times B$$

$$U = (R - Y) \times 0{,}713 + 128 \tag{3.7}$$

$$v = (B - Y) \times 0{,}564 + 128$$

Here Y, U, and v correspond to each channel of the YUV 3-channel output image. 128 value is added so that the output image is in the 8-bit range like in RGB.

The formulas for transformation from RGB to L*A*B* color space is shown below.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0{,}412453 & 0{,}357580 & 0{,}180423 \\ 0{,}212671 & 0{,}715160 & 0{,}072169 \\ 0{,}019334 & 0{,}119193 & 0{,}950227 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$X \leftarrow X \div X_n, \text{ where } X_n = 0{,}950456$$

$$Z \leftarrow Z \div Z_n, \text{ where } Z_n = 1{,}088754$$

$$f(t) = \begin{cases} \sqrt[3]{t} & for\ t > 0{,}008856 \\ 7{,}787t + 0{,}137931 & for\ t \leq 0{,}008856 \end{cases}$$

$$L^* = \begin{cases} 116 \times \sqrt[3]{Y} - 16 & for\ Y > 0{,}008856 \\ 903{,}3 \times Y & for\ Y \leq 0{,}008856 \end{cases} \quad (3.8)$$

$$A^* = 500\big(f(X) - f(Y)\big) + 128$$

$$B^* = 200\big(f(Y) - f(Z)\big) + 128$$

In this OpenCV based transformation, temporary variables X, Y, Z, and $f(t)$ function are used. Here L*, A*, and B* correspond to each channel of the L*A*B* 3-channel output image.

Here, the output range is between $0 \leq L^* \leq 100$, $-127 \leq A^* \leq 127$, $-127 \leq B^* \leq 127$. As can be seen, the values found here are not in the unsigned 8-bit range. Therefore, to output image fit in the 8-bit range, each channel of the L*A*B* is calculated as:

$L^* = L^* \times 255 \div 100$, $A^* = A^* + 128$, $B^* = B^* + 128$.

Hence, it is completed the conversion from RGB to 3-channel 8-bit L*A*B* image.

On the other hand, the formulas for transformation from RGB to HSV color space is considered below.

$$V = \max(R, G, B)$$

$$S = \begin{cases} \dfrac{V - min(R,G,B)}{V} & , \quad if\ V \neq 0 \\ 0 & , \quad otherwise \end{cases}$$

$$H = \begin{cases} \dfrac{60(G-B)}{\left(V - \min(R,G,B)\right)} & , \quad if\ V = R \\ \dfrac{120 + 60(B-R)}{(V - \min(R,G,B))} & , \quad if\ V = G \\ \dfrac{240 + 60(R-G)}{(V - \min(R,G,B))} & , \quad if\ V = B \end{cases} \tag{3.9}$$

Here H, S, and V correspond to each channel of the HSV 3-channel output image.

In this point, where if $H < 0$ then $H = H + 360$.

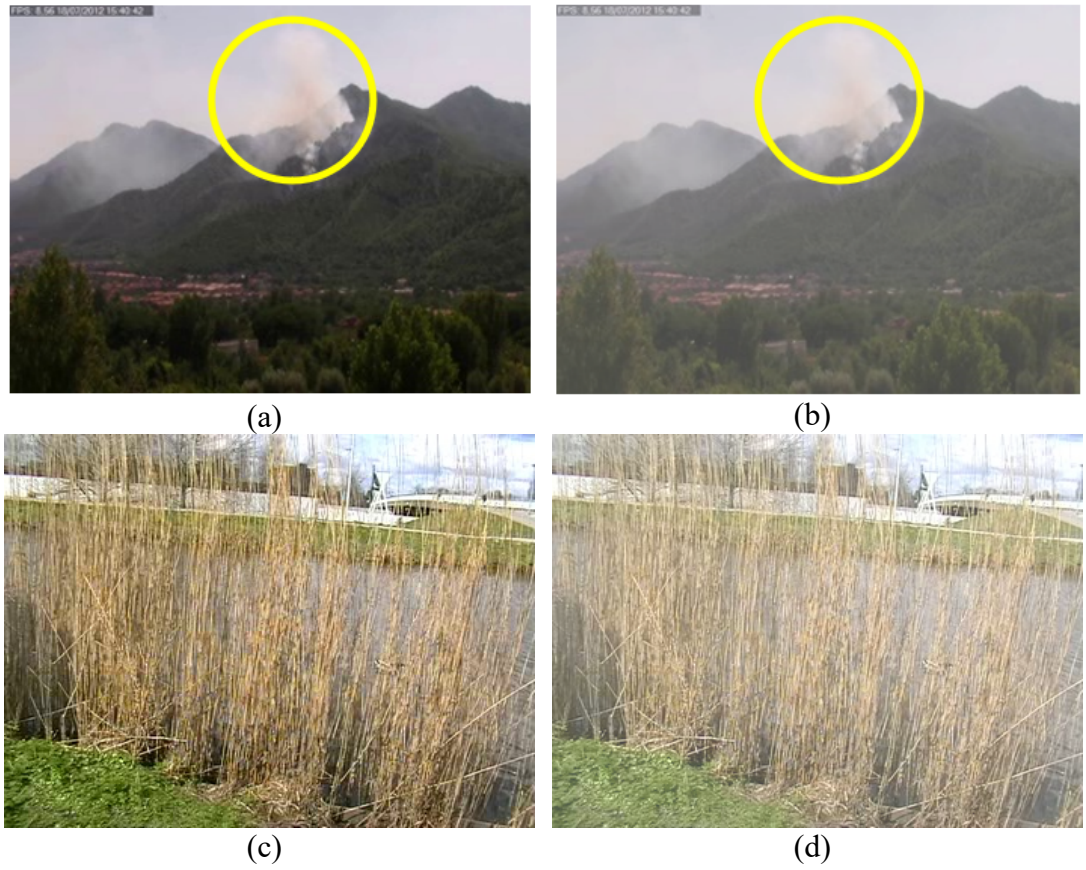
On output are $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$. As can be seen, the values range is not fit in 8-bit range. To fix this:

$V = V \times 255, S = S \times 255, H = H \div 2.$

Hence, it is completed the conversion from RGB to 3-channel 8-bit HSV image.

3.1.3.1 Comparison of Color Spaces

After the thresholding processes with the transformation from RGB to three other color spaces as described, the equivalents of the same images in different color spaces were examined. For this examination, sample images taken from images in RGB color space are presented in Figure 3.5. Smoke areas in images are marked with yellow to make the difference between images of different color spaces more clearly visible.



(a) (b)

(c) (d)

Figure 3.5: RGB color space, (a) Smoky labeled [47], (b) Foggy-smoky labeled, (c) Normal labeled [19], (d) Foggy labeled images

The representations of YUV, $L^*A^*B^*$, and HSV color spaces in RGB color space are shown in Figure 3.6, 3.7, 3.8 respectively.

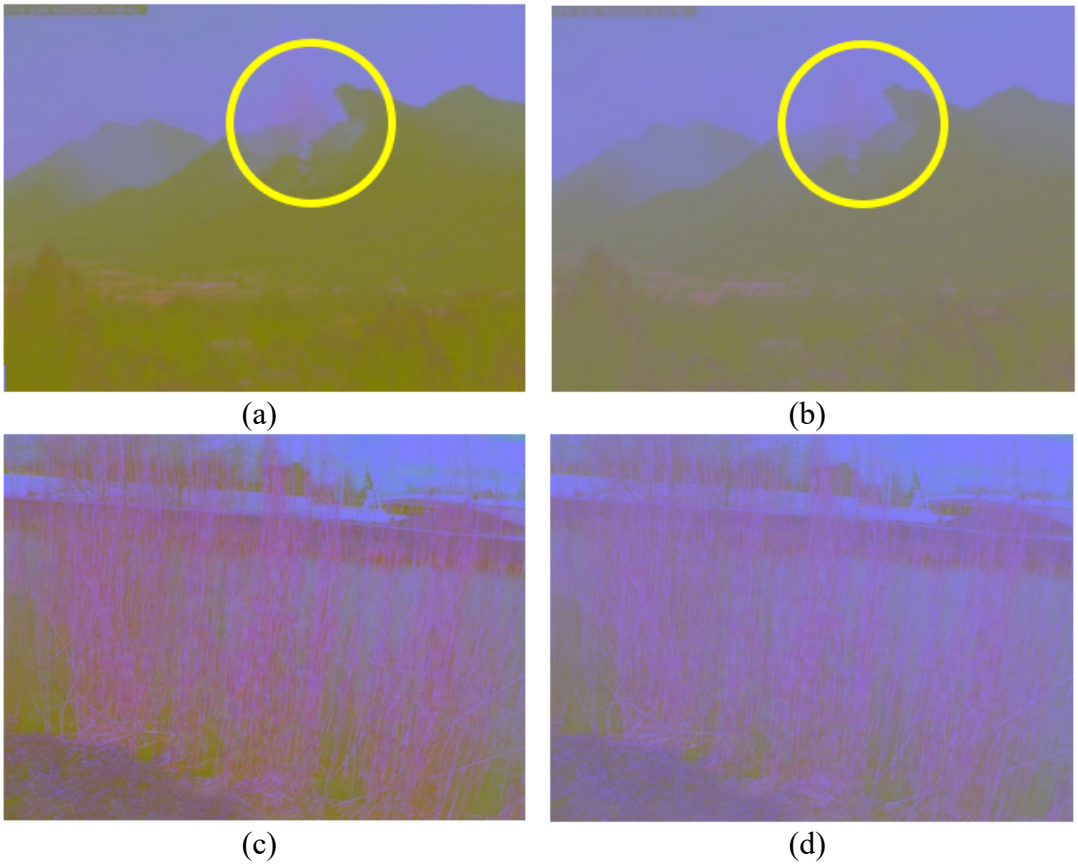


Figure 3.6: YUV color space, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled images

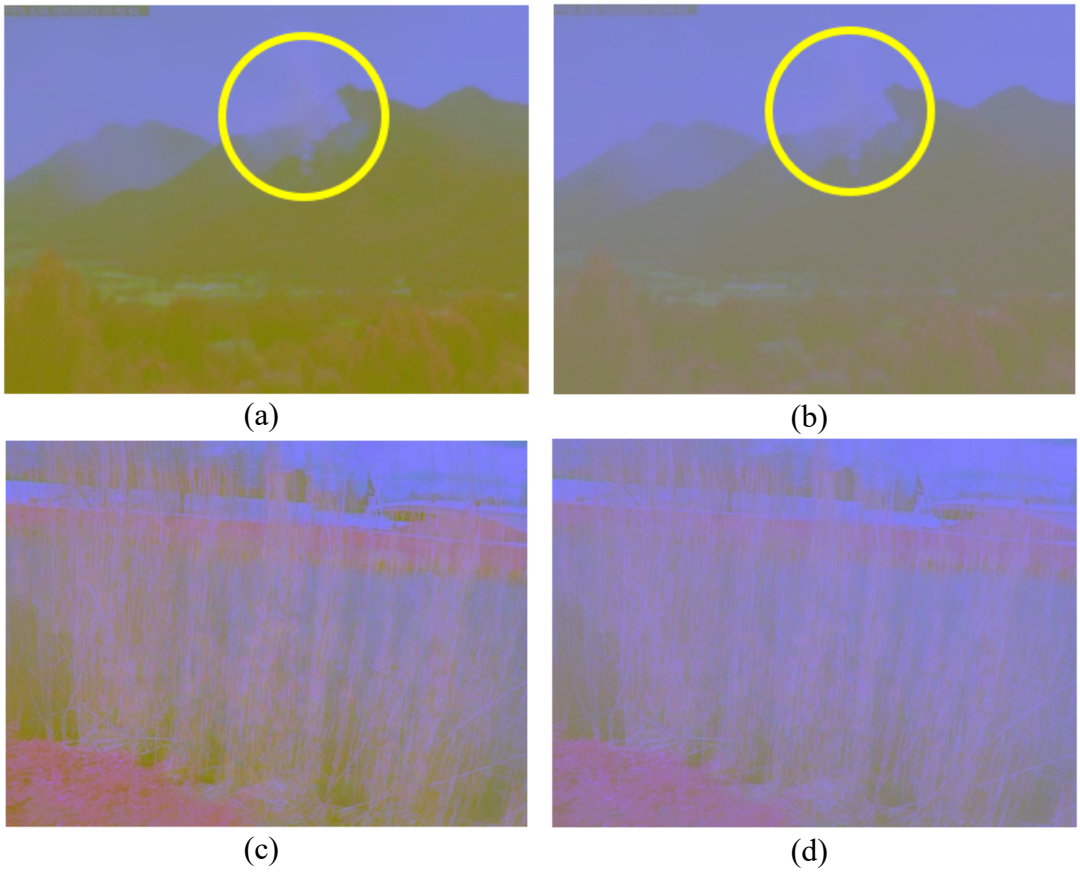


Figure 3.7: $L^*A^*B^*$ color space, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled images

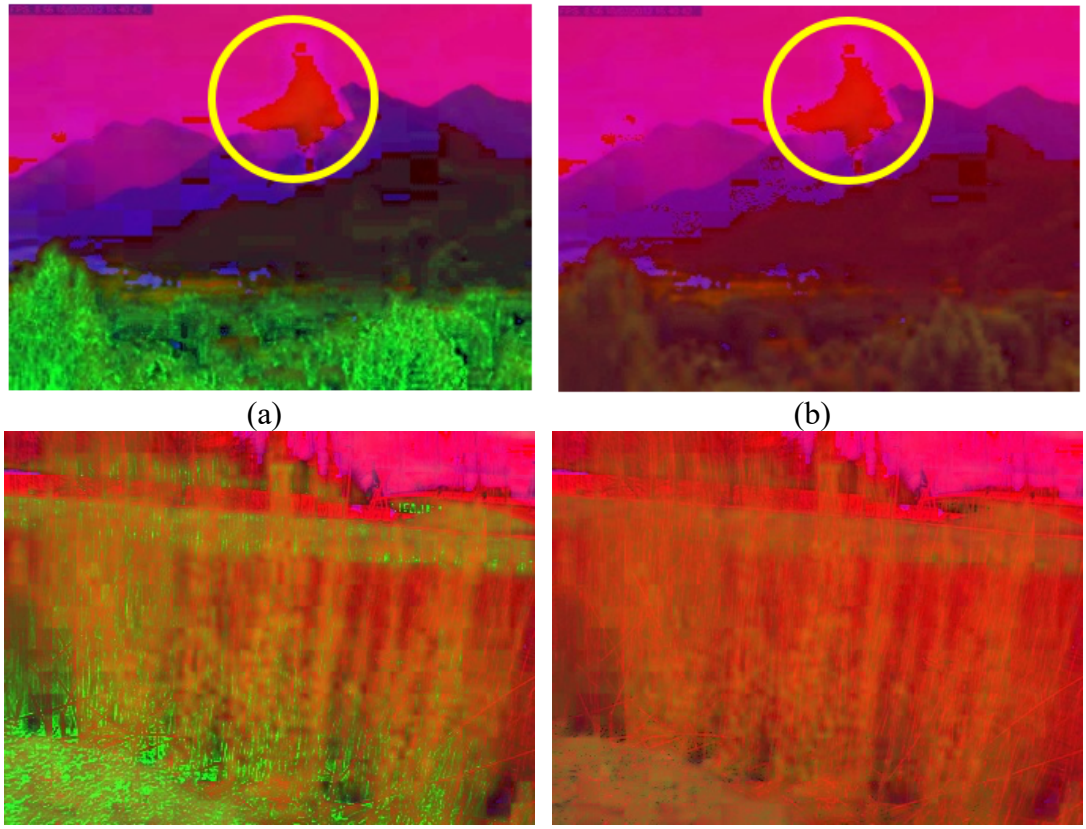


Figure 3.8: HSV color space, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled images

When the images given in Figure 3.6 to 3.8 were examined, it was highlighted that the smoky area marked with the yellow circle and the fog detection entire image were in the clearest HSV color space. Therefore, the images of the HSV color space in each color channel were examined separately in Figure 3.9 to 3.11.

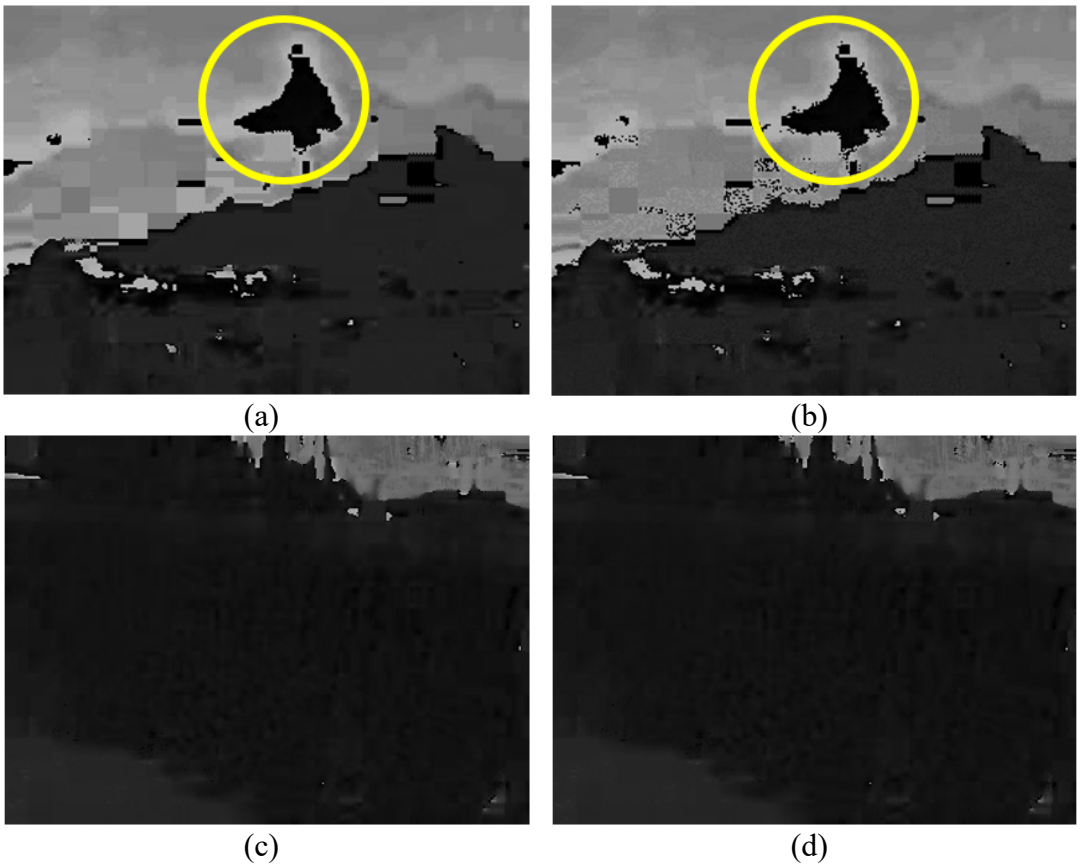
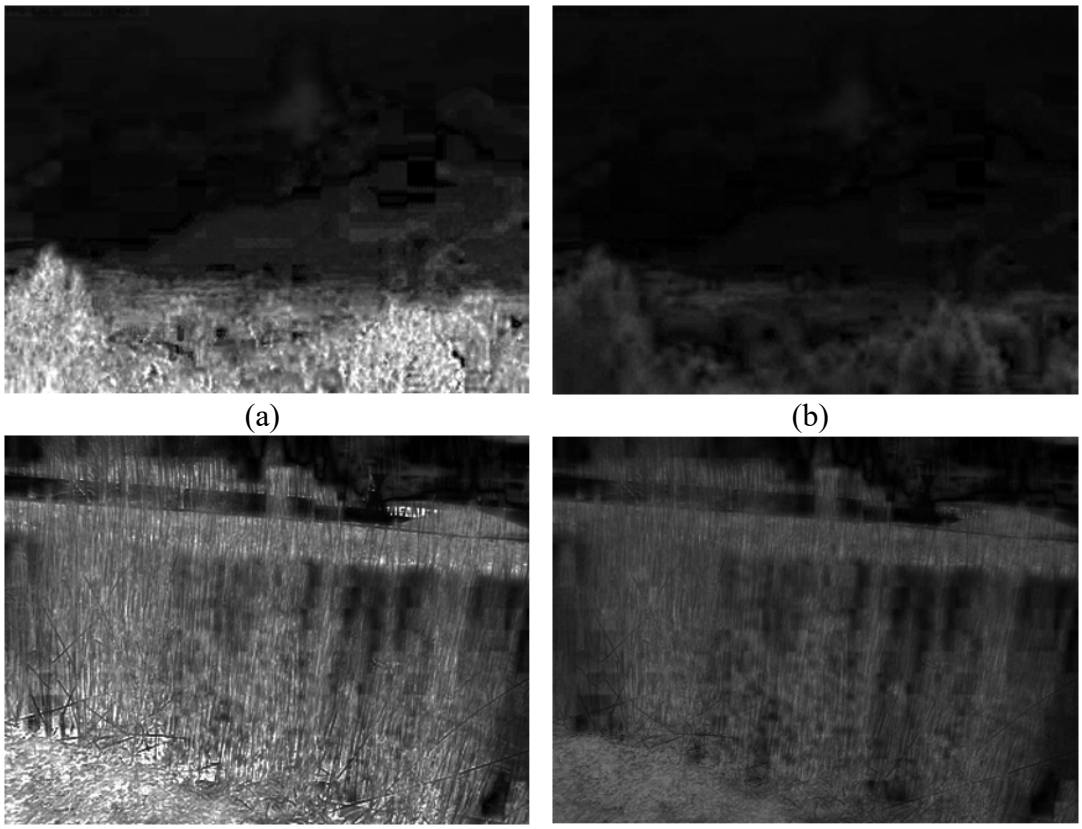


Figure 3.9: HSV color space H-Channel, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled



(a) (b)

(c) (d)

Figure 3.10: HSV color space S-Channel, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled

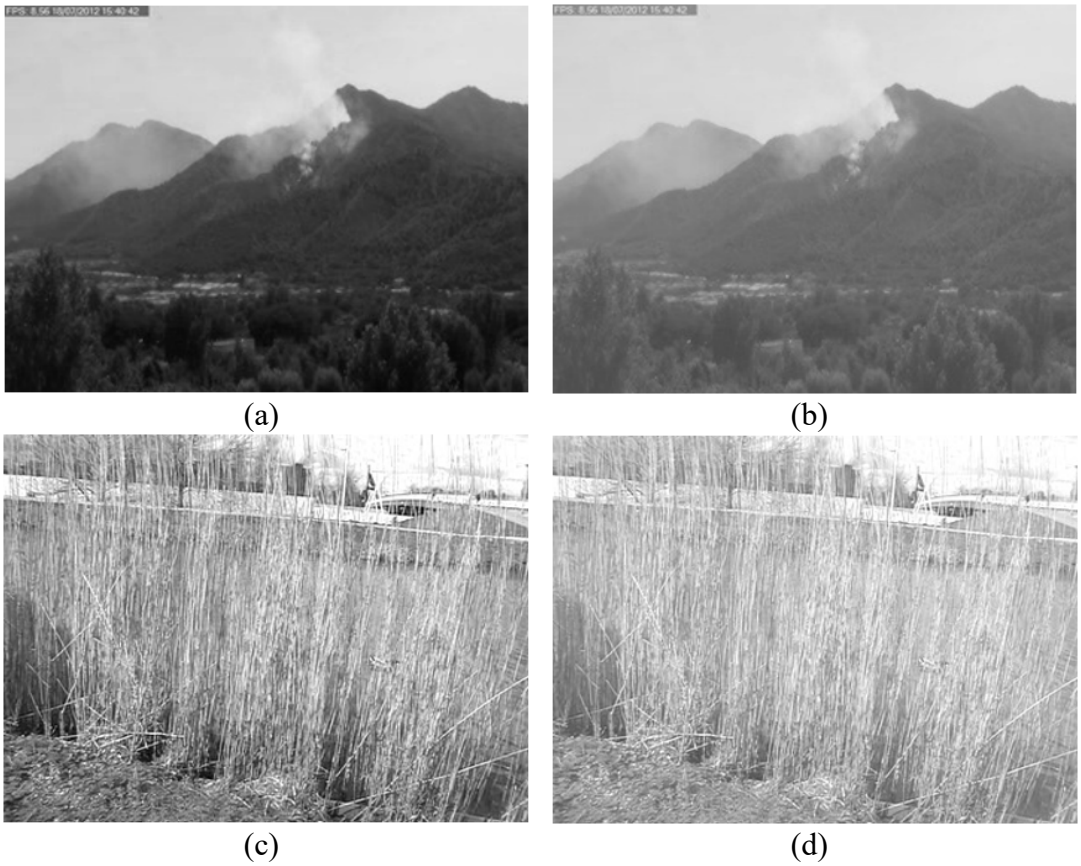


Figure 3.11: HSV color space V-Channel, (a) Smoky labeled, (b) Foggy-smoky labeled, (c) Normal labeled, (d) Foggy labeled

As can be seen from Figures 3.9 to 3.11, it was determined that the H color channel is more successful in detecting the smoke area, while the S and V channels are more successful in detecting the foggy images.

Especially in the S color channel, smoky and normal labeled images appear much brighter. This brightness affects the overall histogram of the image. The histogram of the images in Figure 3.10 are shown in Figure 3.12.

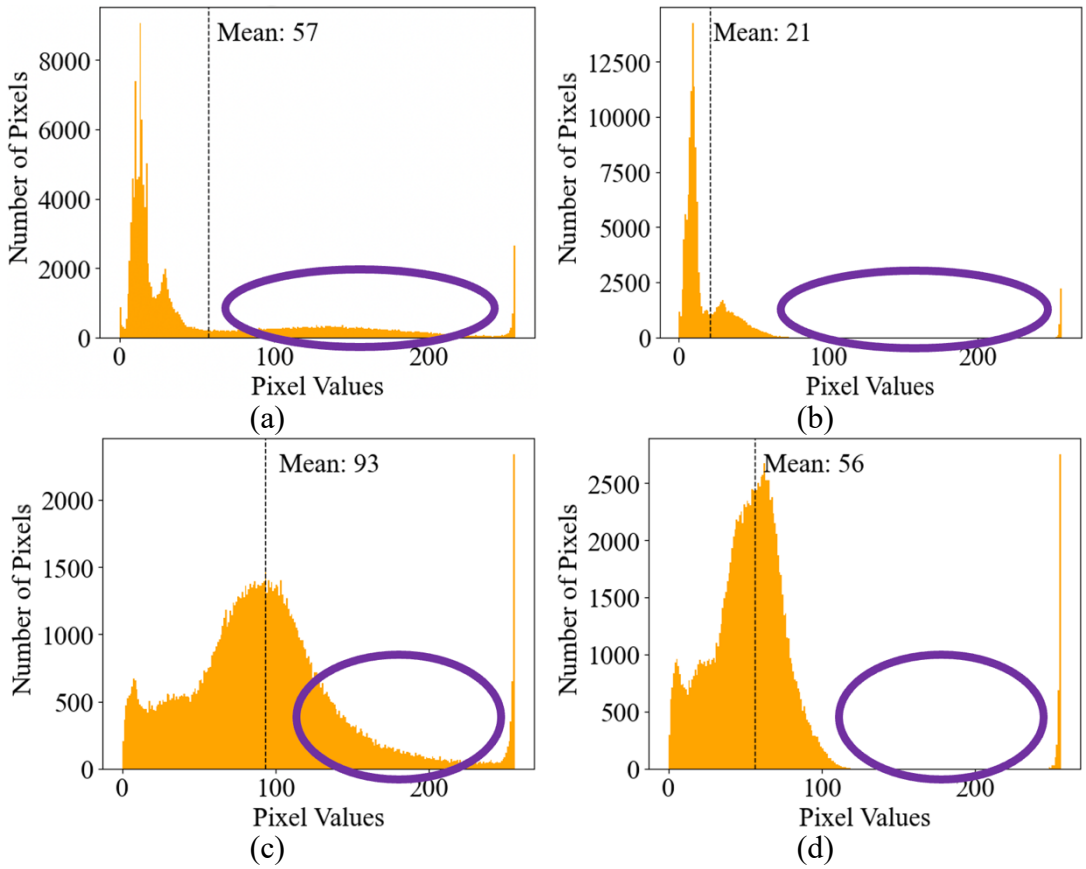


Figure 3.12: Histogram of S-Channel Images, (a) Figure 3.10a, (b) Figure 3.10b, (c) Figure 3.10c, (d) Figure 3.10d

Adding fog to the image reduces the average histogram value of the S-Channel of the image by 30 to 40 as can be seen from Figure 3.12. In addition, the values between the mean value and maximum values in the histogram graph, marked with purple, vanished considerably. It facilitates the detection of fog in images by means of the S-channel of the HSV color space.

3.1.4 Dataset Preprocessing

In this thesis, to fairly comparison, previously used data split rates in the literature are used [55]. 20% of the whole dataset is reserved for training, 30% for validation, and 50% for testing. Images representing numbers of training, validation, and testing data visualized in Figure 3.13.

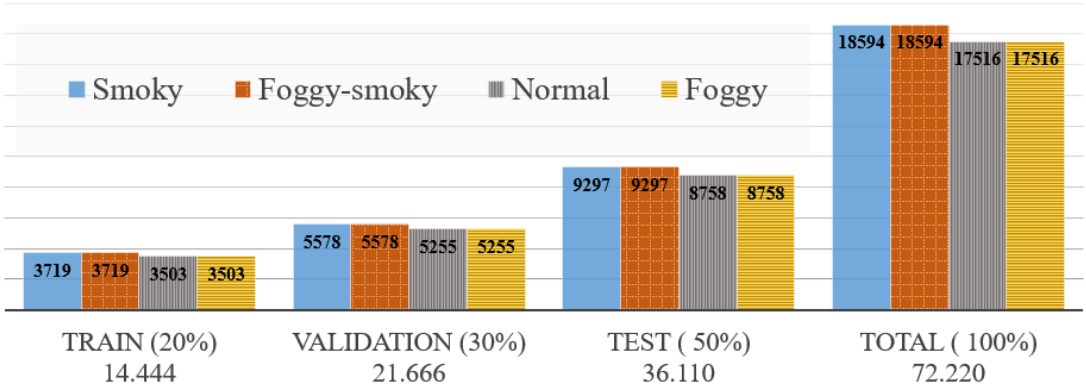


Figure 3.13: The graph of number of training, validation and testing images

The other pre-training process on dataset, training and validation images are resized in alignment with the model entrance. In this way, all these images with different sizes in their original form are brought to a specific width and height value. These width and height values are selected as 224x224 or 299x299 depending on the type of model used. In addition, one hot encoding method has been used in order to make a classification of categorically labeled images.

3.2 Model Modifications

After the dataset operations, model modifications are carried out. Model modifications have been implemented for architectures VGG 16, VGG 19, Inception V3, Inception-ResNetV2, Xception, DenseNet169, DenseNet201, and MobileNetV2. Originally, these architectures were built to classify the 1000-labeled Imagenet dataset. In this thesis, instead of the Softmax layer used to classify 1000-labeled Softmax layers in the output of these architectures, 4-labeled Softmax layers are used for the classification images with labels of Smoky, Foggy-smoky, Normal and Foggy. The revision in the Softmax layer of the architectural model is shown in Figure 3.14.

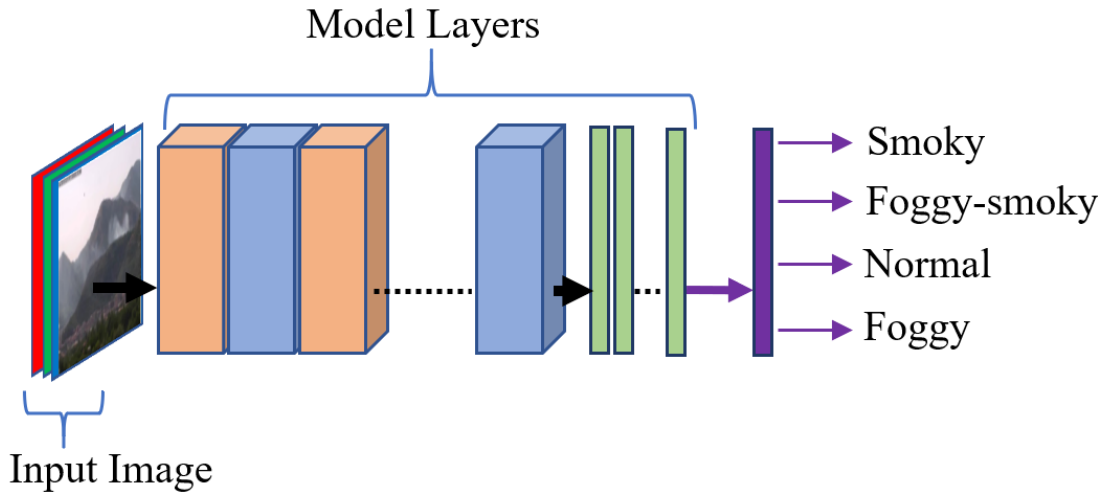


Figure 3.14: Model with 4-labeled Softmax layer

While training on the architectural model, the SGD algorithm was used to update the parameters iteratively by minimizing the cost function. In SGD, the training is faster because random batches in the dataset are used instead of the whole dataset. By means of a fixed learning rate, the minimum cost function can be reached with random batches. In this thesis, while in training section for all architectural models, SGD algorithm learning rate value is chosen as 0,001. These parameters is selected same values with literature used similar dataset [55]. Values of all training parameters used in the model are given in Table 3.3. In Table 3.3, the validation steps parameter is the number of batch iterations before the end of the epoch on validation data.

Table 3.3: Model training parameters

Parameters	Value
Initial Weight	None
Optimizer	SGD
Learning Rate	0,001
Batch Size	16
Epoch	30
Validation Steps	1

While training on each of the architectural model, the weights of the highest results according to the validation accuracy are saved. Then, these weights are loaded, and the results are compared on the test dataset.

3.3 Hardware Equipment and Software Environments

In this thesis, the training of the models is carried out on a computer with an Intel Core (TM) i7-10750H (CPU) processor, 16GB RAM capacity, Windows 10 operating system on NVIDIA GeForce RTX 2070 with Max-Q Design GPU. While the image processing toolbox in the 2020b version of the MATLAB program [56] is used in the process of adding fog to the images, all other operations related to the model and dataset are performed in the 3.7 version of the Python program [57]. Keras interface of the 2.1.0 version of the TensorFlow [58] library is used in the processes related to model modification and deep learning in the Python program. NumPy [59] and OpenCV [52] libraries are used in the operations related to the dataset in the Python program.

Chapter 4

Results

In this chapter, the findings obtained in the thesis will be explained. The eight state-of-the-art architectures described were trained in four color spaces. These training sessions were repeated five times in order to verify the performance of the results. The highest results achieved at the end of a total of 160 training sessions are listed in Table 4.1.

Table 4.1: Model highest accuracy (%)

Architecture	Color Spaces			
	RGB	YUV	$L^*A^*B^*$	HSV
VGG 16	97,82	97,64	97,75	98,71
VGG 19	96,55	98,45	96,83	97,95
InceptionV3	97,08	98,78	97,95	98,85
Inception-ResNetV2	96,43	98,32	98,01	98,38
Xception	98,17	98,61	98,47	98,48
DenseNet 169	97,72	98,37	98,50	98,90
DenseNet 201	98,21	97,07	98,65	97,92
MobileNetV2	**98,64**	**98,98**	**99,11**	**99,44**

After 5 sessions, calculated the arithmetic average of the results and the performance rates were compared in terms of color space with box plot shown in Figure 4.1.

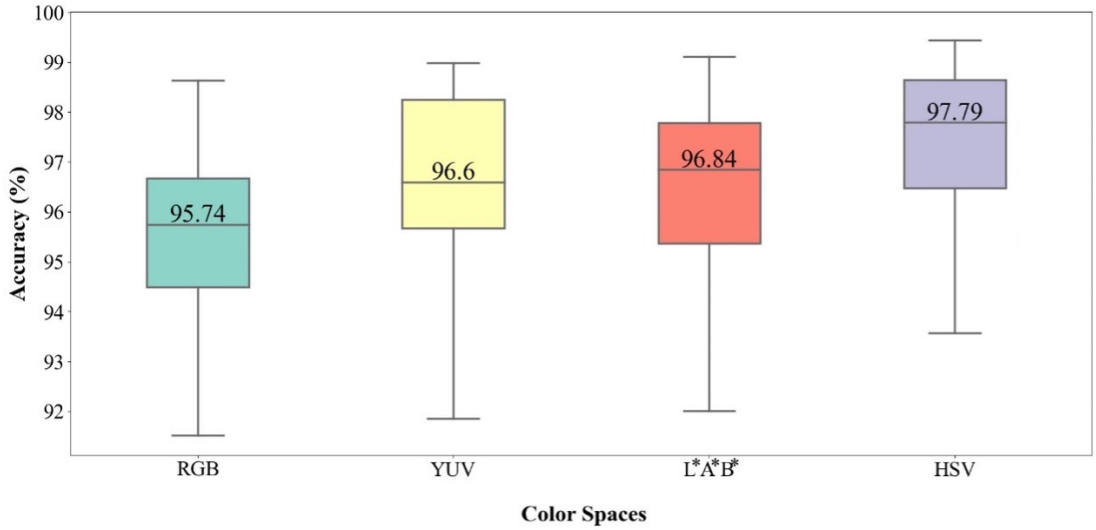


Figure 4.1: Color space comparison after 5 sessions over all architectures

It has been determined that the HSV color space has higher accuracy result than other color spaces in terms of median, maximum and minimum values. On the other hand, RGB color space has the lowest percentage of accuracy among the compared color spaces.

Accuracy percentages between color spaces and architectures at the end of 5 sessions are given in Table 4.2.

Table 4.2: Average accuracy (%) of models

Architecture	Color Spaces			
	RGB	YUV	L*A*B*	HSV
VGG 16	94,99	94,47	96,45	**97,50**
VGG 19	93,83	**96,36**	94,43	96,08
InceptionV3	95,46	97,50	95,20	**97,73**
Inception-ResNetV2	95,15	95,79	94,99	**97,62**
Xception	96,44	96,72	**97,24**	97,19
DenseNet 169	95,23	96,97	97,56	**98,14**
DenseNet 201	95,27	94,18	**97,13**	95,21
MobileNetV2	96,27	98,19	97,45	**98,95**

The highest average accuracy percentage was obtained in HSV color space using the MobileNetV2 based architecture. MobileNetV2 architecture in HSV color space is the method that we propose in this thesis, due it gives the best result at the end of both the 5 sessions table and the highest performance table. RGB has the poorest performance with no highest score in any of the architectures. The success of the method that we propose can be examined deeply from the confusion matrix in Table 4.3.

Table 4.3: Confusion matrix of the best MobileNetV2-HSV Score

Predicted Label	True Label			
	Smoky	Foggy-smoky	Normal	Foggy
Smoky	9234	15	50	0
Foggy-smoky	48	9236	0	22
Normal	9	0	8704	0
Foggy	6	46	4	8758

It is noticed that from the confusion matrix, the model makes of the most mistakes in classifying smoky true labeled images. Only 50 normal labeled images were predicted as smoke labeled according to the confusion matrix. In addition, 46 images labeled with foggy-smoky were predicted as foggy.

There are other performance metrics that are used to measure model performance in align with the confusion matrix in Table 4.3. Precision is expressed by the ratio of true positive samples to all correct values. Recall is defined as the ratio of true positive values to true positive and false negative values. The F1-Score is calculated by the harmonic average of the Precision and Recall values and takes a value between 0 and 1. The values are closer to 1, the more successful it is. In Table 4.4, the most successful performance of the MobileNetV2 model in the HSV color space, in accordance with the result of Precision, Recall, and F1-Score metrics can be examined. Over 99% success rate was achieved in all labeled images given to the classifier.

Table 4.4: Evaluation with various metrics of the best MobileNetV2-HSV Score

True Label	Precision	Recall	F1 - Score
Smoky	99,30	99,32	99,31
Foggy-smoky	99,25	99,34	99,30
Normal	99,90	99,38	99,64
Foggy	99,36	99,75	99,56

Selected false predicted images made by the proposed HSV color space based MobileNetV2 model in line with the Table 4.3 are shown in Figure 4.2.

(a)
Predicted: Foggy-smoky
Real: Foggy

(b)
Predicted: Foggy-smoky
Real: Smoky [19]

(c)
Predicted: Foggy-smoky
Real: Smoky [46]

(d)
Predicted: Smoky
Real: Foggy-smoky

Figure 4.2: False predicted images with the proposed model

As can be seen from Figure 4.2(a), the proposed model did not perform well on prediction in foggy low-light images. In Figures 4.2(b) and 4.2(c), where the smoke is formed right in front of the footage, the image is perceived by the proposed model as if it was an image taken in a foggy environment. Similarly, in Figure 4.2 (d), it is confused with smoky labeled with foggy-smoky because smoke is very close. It is hard to decide their label by seeing them also. If the proposed model is retrained using these false predicted images, these erroneous results may not be obtained. Furthermore, more successful results may be obtained by increasing the ratio of training and validation sets in the dataset.

Disk usage of the models is another parameter to be compared. The disk usage is an important parameter of the model when integrating into embedded systems and classifying on low-capacity systems. The training elapsed time is an important parameter in the model training phase, and its short duration is an indication that the model has successfully architecture. Here it consists of the size of the model, the weights, and the related parameters to be used in the classification.

Table 4.5: Model comparison based on the number of parameters, approximately elapsed training time, and disk usage

Architecture	Number of total parameter (mill.)	Approximately elapsed training time (sec.)	Size (MB)
VGG 16	134,276	5890	524,60
VGG 19	139,586	6257	545,35
InceptionV3	21,81	6268	86,03
Inception-ResNetV2	54,342	14474	214,27
Xception	20,869	14004	81,90
DenseNet 169	12,649	5408	50,96
DenseNet 201	18,329	8435	73,44
MobileNetV2	2,263	3210	9,29

Furthermore, the MobileNetV2-based model has the shortest training time using the least number of parameters. Accordingly, the lowest disk usage was realized in MobileNetV2 based model. Among the other architectures chosen, its size deploys approximately 18% of its closest competitor DenseNet 169. From these results, it is seen that the MobileNetV2 model gives more successful results than the other models used in the thesis.

In this thesis the proposed model has been compared with the studies that have a similar dataset. In align with the test dataset image quantity, the accuracy percentage on the test dataset of the models and the disk usage capacity of them were selected as the comparison criteria. This comparison is shown in Table 4.6.

Table 4.6: Comparison between the literature and the proposed method

	Test Dataset Image Quantity	Architecture	Color Space	Accuracy (%)	Size (MB)
[51]	36006	VGG 16	RGB	97,72	930,00
[60]	36110	DenseNet 169	YUV	97,80	50,96
[61]	36006	MobileNetV2	RGB	98,17	13,23
Proposed	36110	MobileNetV2	HSV	99,44	9,29

The VGG 16 architecture in RGB color space method by [51] achieved an accuracy of 97.72%. When this method was repeated in this thesis, the accuracy of [51] was within the 5-session range, but not equal exactly the same. The reasons for these might be, splitting the datasets at different rates from the videos, preferring different artificial fog algorithms, choosing different starting weights in training sessions, splitting the train test and validation datasets in different ways.

In this thesis, the MobileNetV2-based model in the HSV color space is proposed and it has achieved the highest accuracy result among the studies using the same dataset in the literature. Moreover, the proposed model has the lowest memory requirement

among the studies in Table 4.6. Since the model parameters of [61] were recorded in an environment different from the programming environment used in this thesis, their model size may have been higher.

Chapter 5

Conclusion

The necessity of smoke detection in the outdoor environment becomes more important with the integration of image processing areas and monitoring systems together. This detection is essential for the reliability and sustainability of the system, especially in harsh weather conditions. While there are various image processing methods for the detection of smoke, these methods are not feasible in harsh weather conditions such as fog.

In this thesis, a color space-based deep learning model is proposed for smoke detection for foggy environments. Firstly, videos were downloaded from databases about smoke detection in the literature. Images were extracted from these downloaded videos. Then, new foggy images were obtained by changing the brightness values of these images. With the addition of new foggy images, the dataset with 4 labels has been created. This dataset composed of RGB images were transformed into YUV, HSV, and $L^*A^*B^*$ color spaces searching for better detection. Eight of the state-of-the-art CNN models were trained and tested on the dataset that is transformed into color spaces. At the end of the training sessions, the comparison was made, and the most successful color space and model were determined. MobileNetV2-based model in the HSV color space is selected as proposed model in this thesis and was compared with other studies in the literature in terms of memory usage and accuracy results. At the end of the comparison, the proposed model showed better classification performance than other results in the literature. Moreover, most of the images that the proposed model misclassified consisted of close-up images, perhaps where fog detection was not necessary. This superior classification success of the proposed model was also seen when the model is evaluated in terms of memory requirement. The memory requirement of the proposed model is the lowest among the studies in the literature.

This situation showed that deep learning models that have already achieved saturation in performance success, can be used for wildfire monitoring applications with much lower memory requirements. The proposed model paved the way for detecting smoke in foggy environments in embedded systems integrated with cameras. It supports the automatic operation of wildfire monitoring systems. In addition, the model can be enhanced to show the segmentation of the smoke area in the image. The model can also detect fire by training with images containing fire pictures. The development of the proposed smoke detection model can significantly enhance the fire safety of forested areas in cities. It can be used to pinpoint the area where the fire started. Decreasing the number of forest fires could trigger the repopulation of many species facing extinction. Restoring the natural life balance leads to a more livable world. It can be an important milestone in the fight against the greenhouse gas effect related to carbon emission and the resulting global warming. By reducing the high budget appropriations allocated to prevent forest fires, not only ecological and environmental harmfulness but can also alleviate the burden on states economically.

References

[1] Çelik T, Özkaramanlı H, Demirel H. Fire and smoke detection without sensors: Image processing based approach. 15th European Signal Processing Conference 2007. 1794-1798.

[2] Prema CE, Vinsley S, Suresh S. Multi feature analysis of smoke in YUV color space for early forest fire detection. Fire Technology 2016; 52(5): 1319-1342.

[3] Genovese A, Labati RD, Piuri V, Scotti F. Wildfire smoke detection using computational intelligence techniques. 2011 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proceedings: IEEE; 2011. 1-6.

[4] Yin Z, Wan B, Yuan F, Xia X, Shi J. A deep normalization and convolutional neural network for image smoke detection. IEEE 2017; 5: 18429-18438.

[5] Xu G, Zhang Y, Zhang Q, Lin G, Wang Z, Jia Y, et al. Video smoke detection based on deep saliency network. Fire Safety Journal 2019; 105: 277-285.

[6] Gu K, Xia Z, Qiao J, Lin W. Deep dual-channel neural network for image-based smoke detection. IEEE Transactions on Multimedia 2019; 22(2): 311-323.

[7] Li T, Zhao E, Zhang J, Hu C. Detection of wildfire smoke images based on a densely dilated convolutional network. Electronics 2019; 8(10): 1131.

[8] Liu T, Cheng J, Du X, Luo X, Zhang L, Cheng B, et al. Video smoke detection method based on change-cumulative image and fusion deep network. Sensors 2019; 19(23): 5060.

[9] Yuan F, Zhang L, Xia X, Wan B, Huang Q, Li X. Deep smoke segmentation. Neurocomputing 2019; 357: 248-260.

[10] Zhang Q-x, Lin G-h, Zhang Y-m, Xu G, Wang J-j. Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. Procedia Engineering 2018; 211: 441-446.

[11] Pan H, Badawi D, Zhang X, Cetin AE. Additive neural network for forest fire detection. Signal, Image and Video Processing 2019: 1-8.

[12] Xu G, Zhang Y, Zhang Q, Lin G, Wang J. Deep domain adaptation based video smoke detection using synthetic smoke images. Fire Safety Journal 2017; 93: 53-59.

[13] Aslan S, Güdükbay U, Töreyin BU, Çetin AE. Early wildfire smoke detection based on motion-based geometric image transformation and deep convolutional generative adversarial networks. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): IEEE; 2019. 8315-8319.

[14] He L, Gong X, Zhang S, Wang L, Li F. Efficient attention based deep fusion CNN for smoke detection in fog environment. Neurocomputing 2021; 434: 224-238.

[15] Liu T, Cheng J, Yuan Z, Hua H, Zhao K. Video smoke detection with block DNCNN and visual change image. KSII Transactions on Internet & Information Systems 2020; 14(9).

[16] Hossain FA, Zhang YM, Tonima MA. Forest fire flame and smoke detection from UAV-captured images using fire-specific color features and multi-color space local binary pattern. Journal of Unmanned Vehicle Systems 2020; 8(4): 285-309.

[17] Jadon A, Omama M, Varshney A, Ansari MS, Sharma R. FireNet: a specialized lightweight fire & smoke detection model for real-time IoT applications 2019.

[18] Joblove GH, Greenberg D. Color spaces for computer graphics. Proceedings of the 5th annual conference on Computer Graphics and Interactive Techniques 1978. 20-25.

[19] Dimitropoulos K, Barmpoutis P, Grammalidis N. Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. IEEE Transactions on Circuits and Systems for Video Technology 2014; 25(2): 339-351.

[20] Podpora M, Korbas GP, Kawala-Janik A. YUV vs RGB-Choosing a Color Space for Human-Machine Interaction. FedCSIS (Position Papers) 2014. 29-34.

[21] Hunter RS. Photoelectric color difference meter. Josa 1958; 48(12): 985-995.

[22] Smith AR. Color gamut transform pairs. ACM Siggraph Computer Graphics 1978; 12(3): 12-19.

[23] Haefner N, Wincent J, Parida V, Gassmann O. Artificial intelligence and innovation management: A review, framework, and research agenda. Technological Forecasting and Social Change 2021; 162: 120392.

[24] Hawkins DM. The problem of overfitting. Journal of Chemical Information and Computer Sciences 2004; 44(1): 1-12.

[25] Bottou L. Stochastic gradient learning in neural networks. Proceedings of Neuro-Nimes 1991; 91(8): 12.

[26] Leshno M, Lin VY, Pinkus A, Schocken S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Networks 1993; 6(6): 861-867.

[27] Agarap AF. Deep learning using rectified linear units (relu) 2018.

[28] Belue LM, Bauer Jr KW. Determining input features for multilayer perceptrons. Neurocomputing 1995; 7(2): 111-121.

[29] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015; 521(7553): 436-444.

[30] LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, et al. Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems 1989; 2: 396-404.

[31] Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. International Conference on Engineering and Technology (ICET): IEEE; 2017. 1-6.

[32] Wu J. Introduction to convolutional neural networks. National Key Lab for Novel Software Technology Nanjing University China 2017; 5: 23.

[33] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift 2015.

[34] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 2014; 15(1): 1929-1958.

[35] Deng J, Dong W, Socher R, Li L, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition: IEEE; 2009. 248-255.

[36] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Communications of the ACM 2017; 60(6): 84-90.

[37] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition 2014.

[38] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016. 2818-2826.

[39] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016. 770-778.

[40] Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, inception-resnet and the impact of residual connections on learning 2016.

[41] Chollet F. Xception: Deep learning with depthwise separable convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017. 1251-1258.

[42] Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ. Deep networks with stochastic depth. European Conference on Computer Vision: Springer; 2016. 646-661.

[43] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017. 4700-4708.

[44] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018.

[45] Bugarić M, Jakovčević T, Stipaničev D. Adaptive estimation of visual smoke detection parameters based on spatial data and fire risk index. Computer Vision and Image Understanding 2014; 118: 184-196.

[46] Töreyin BU, Dedeoğlu Y, Cetin AE. Wavelet based real-time smoke detection in video. 13th European Signal Processing Conference: IEEE; 2005. 1-4.

[47] Foggia P, Saggese A, Vento M. Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. IEEE Transaction on circuits and systems for video technology. 2015; 25(9): 1545-1556.

[48] Labati RD, Genovese A, Piuri V, Scotti F. Wildfire smoke detection using computational intelligence techniques enhanced with synthetic smoke plume generation. IEEE Transactions on Systems, Man, and Cybernetics: Systems 2013; 43(4): 1003-1012.

[49] Pond5 [Internet]. 2021 [Available in 30.05.2021]. https://www.pond5.com/.

[50] PronghornProductions [Internet]. 2021 [Available in 30.05.2021]. http://www.hdnaturefootage.net/.

[51] Khan S, Muhammad K, Mumtaz S, Baik SW, de Albuquerque VHC. Energy-efficient deep CNN for smoke detection in foggy IoT environment. IEEE Internet of Things Journal 2019; 6(6): 9237-9245.

[52] Bradski G, Kaehler A. Learning OpenCV: Computer vision with the OpenCV library: O'Reilly Media, Inc.; 2008.

[53] Poynton C. Frequently asked questions about color. Retrieved June 1997; 19: 2004.

[54] Ford A, Roberts A. Colour space conversions. Westminster University, London 1998; 1998: 1-31.

[55] Muhammad K, Ahmad J, Lv Z, Bellavista P, Yang P, Baik SW. Efficient deep CNN-based fire detection and localization in video surveillance applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems 2018; 49(7): 1419-1434.

[56] Gonzalez RC, Woods RE, Eddins SL. Digital Image Processing Using MATLAB: Pearson Education India; 2004.

[57] Oliphant TE. Python for scientific computing. Computing in Science & Engineering 2007; 9(3): 10-20.

[58] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. 12th Usenix symposium on operating systems design and implementation (OSDI) 2016. 265-283.

[59] Van Der Walt S, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. Computing in Science & Engineering 2011; 13(2): 22-30.

[60] Yıldız UE, Özbek ME. Deep learning based smoke detection for foggy environments. 12th International Conference on Electrical and Electronics Engineering (ELECO): IEEE; 2020. 237-240.

[61] Muhammad K, Khan S, Palade V, Mehmood I, De Albuquerque VHC. Edge intelligence-assisted smoke detection in foggy surveillance environments. IEEE Transactions on Industrial Informatics 2019; 16(2): 1067-1075.

Appendices

Appendix A

Publications from the Thesis

Conference Papers

1. Yıldız UE, Özbek ME. Deep Learning Based Smoke Detection for Foggy Environments. 2020 12th International Conference on Electrical and Electronics Engineering (ELECO): IEEE; 2020. p. 237-240

Curriculum Vitae

Name Surname : Uğur Emre Yıldız

Education:

2012–2017	Kırklareli Univ. Dept. of Electrical Electronics Eng. Bachelor
2016–2020	Eskişehir Anadolu Univ. Business Administration Bachelor
2018–	İzmir Kâtip Çelebi Univ. Dept. of Electrical Electronics Eng. M.Sc.

Work Experience:

2018–2019	İzmir Kâtip Çelebi Univ. Full-Time Researcher Fellow
2019–	PTT A.Ş. Directorate of Construction Affairs Electrical Eng.

Publications:

1. Yıldız UE, Kılıç V. Detection of Melanoma with Multiple Machine Learning Classifiers in Dermoscopy Images. In 2019 Medical Technologies Congress (TIPTEKNO). IEEE; 2019. p. 1-4
2. Yıldız UE, Özbek ME. Deep Learning Based Smoke Detection for Foggy Environments. 2020 12th International Conference on Electrical and Electronics Engineering (ELECO): IEEE; 2020. p. 237-240