Identifying Bushfire Vulnerable Areas in Australia Through Satellite Imagery and Convolutional Neural Networks

by

Zicheng Mu

This thesis is submitted in total fulfillment for the degree o

Master of Information Technology

in the

School of Computing and Information Systems

Faculty of Engineering and Information Technology

The University of Melbourne

Supervisor: Prof. Richard O. Sinnott

October 2023

Declaration

I certify that

- this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

- the thesis is 6351 words in length (excluding text in images, table, bibliographies and appendices).

Signed: Zicheng Mu

Date: 29/10/2023

Acknowledgement

I would like to appreciate Professor Richard O. Sinnott for offering me the opportunity to undertake such a meaningful project in my field of interest. Without the resources shared and assistance provided by him, my research would have faced unimaginable challenges.

I would also like to express my gratitude to my family for their unwavering support throughout my student career. Their encouragement and faith have saved me from countless mental breakdowns and consistently push me forward.

Also, the deepest appreciation to the University of Melbourne and the Spartan platform. This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

Contents

List of Figures

List of Tables

Abstract

In the 21st century, Australia experienced two devastating bushfires that impacted the nation's history. In 2009, the ”Black Saturday” bushfires claimed 173 lives, making it the deadliest wildfire incident in Australia's record. Subsequently, the ”Black Summer” bushfires, which spanned from 2019 to 2020, burned over 19 million hectares and resulted in an estimated financial loss of 80 billion AUD. Except for the environmental and economical impacts, these catastrophic bushfires also caused severe climate change and biodiversity issues. To mitigate the influences of potential future large-scale bushfires, monitoring and forecasting models are of vital importance. Satellite images, which encompass multiple spectral bands, collectively capture various features on earth surface. Such characteristics making them ideal inputs for a wide range of applications. The inception of Convolutional Neural Network (CNN) was initially inspired by biological processes, and Yann LeCun engineered the earliest model for recognizing hand-written digits. Recently, CNNs have shown remarkable efficacy in the field of image recognition. Therefore, the combination of satellite imagery and CNNs has become an effective solution to solve bushfire detection problems. In this research, we aim to exploit a variety of CNNs to productively identify areas that are at the highest risk of fires, specifically by analysing the levels of hydration and carbon content.

Keywords: Bushfire Detection, Deep Learning, Convolutional Neural Networks, Satellite Imagery

1 Introduction

Due to the climate, topography, and vegetation, south-eastern Australia is extremely vulnerable to wildfires. This region suffered from bushfires consistently, with two particularly devastating mega-fires occurring in the 21st century alone.

On 7th Feb 2009, the state of Victoria recorded its highest temperature, with Melbourne reaching 46.4°C (116°F). Initiated by the intense heat, the ”Black Saturday” bushfire spread across the state on that day. And it was intensified by a temperature decline and winds gusting at up to 100 km/h [1]. Over 300 individual fires happened on this date, and 13 of them developed into significant incidents. The ”Black Saturday” bushfire lasted over a period of 3 weeks, destroyed more than 2200 buildings, and burned over 450,000 hectares [2]. Furthermore, it caused 173 human fatalities, making it one of the most deadliest wildfire incident in Australia.

The other series of mega-fires are also known as the ”Black Summer” bushfires, which occurred during a period of record breaking temperatures and extremely low rainfall. The wildfires, which started in Jun 2019 at middle Queensland and lasted until in May 2020, comprised over 15,000 bushfires. Approximately 19 million hectares were burned and the estimated losses amounting to 80 billion AUD [3]. New South Wales (NSW), Victoria (VIC), and South Australia (SA) states were all affected by the ”Black Summer” bushfires, with multiple records refreshed. NSW had more burned area than in any fire season during the last 20 years, VIC had a season with the highest number of fires and area burned, and SA had the highest number of buildings destroyed in the past two decades [4].

Beyond the economic losses and human fatalities, the biodiversity and ecosystems were also damaged. Moreover, bushfires also have implications for climate change issues. Given the devastation caused by bushfires, monitoring them and forecasting their trends is of great significance. The combination of satellite data and CNNs has been proven to be one effective solution [5], [6], [7], [8].

Due to the nature of satellite imagery, it has drawn significant interest across a wide range of fields. Satellite images are captured periodically, resulting in a enormous accumulation over time. And multiple spectral bands are encompassed in those images, which means various features on the Earth’s surface are encapsulated. Such characteristics can be utilized by researchers to build image time-series that have both abundant useful information and consistent geographical location. Consequently, satellite imagery has emerged as a primary data source for numerous applications. There are multiple platforms from which sentinel satellite data can be sourced, including Google Earth Engine, Sentinel Hub, Planetary Computer, and the BDC platform.

Meanwhile, CNNs have achieved remarkable results in image recognition. The inspiration for CNNs from biological processes can be traced back to the 1960s. Experiments were conducted on the visual cortex of cats, and the result proved that specific neurons responded to specific regions of the visual field [9]. However, the popularity of CNNs in the field of deep learning was raised by Yann LeCun, who engineered the earliest CNN model for recognizing hand-written digits in 1989 [10]. Later in 1998, LeNet-5, which is considered to be the foundation work of CNNs, was proposed by Yann LeCun [11]. Nowadays, CNNs have achieved remarkable accomplishments in solving many classes of problems, such as image classification, object detection and identification, and image segmentation.

This research aims to utilize CNNs to create a model that can productively identify areas that are at the highest risk of fires, specifically by analysing the levels of hydration and carbon content extracted from satellite images. The rest of this paper is structured as follows: Several direct and indirect related works are provided with comprehensive reviews in section 2. Then, the datasets as well as the feature selection are discussed in section 3. Section 4 elaborated on the models created, and details of our experiments are shown in section 5. Then, section 6 analyzes the results of experiments and illustrates the reasons of performance gaps. In section 7, we pointed out the direction of future works. Finally, Section 8 provides a conclusion of our research.

2 Related Work

2.1 Convolutional Neural Networks

Similar to traditional Artificial Neural Networks (ANNs), CNNs are also structured with multiple layers stacked sequentially. These layers (also called hidden layers) incorporate neurons, which are able to self-optimize through learning, to carry out computational tasks. However, CNNs have several outstanding advantages: (1) CNNs typically reduce the size of the input as the depth increases, which means the number of parameters is reduced constantly. (2) Weights are shared across nodes and connections, which can further decrease the number of parameters. (3) Due to the nature of convolutinal and pooling layers, CNNs can extract abstract features more accurately as the input propagates toward deeper layers. As a result, CNNs are capable of addressing computational difficult problems with relatively simple structures, making it an ideal solution to deal with excessively large inputs.

Layers have different levels of depth in a CNN will learn different features. Taking face recognition shown in Figure 1 as an example, the first few shallow layers may learn features like edges, dots, and bright or dark patterns. As the input moves deeper in the network, features extracted could be shapes of eyes, noses, and etc. And in the final layers, highly abstracted faces will be learned.

CNNs are comprised of several kinds of layers, and the construction of a CNN is mostly by stacking layers sequentially. The training process often involve several epochs, and back-propagation is introduced at the end of each epoch. Basically, errors between predictions and labels will be back-propagated, and the model will optimize parameters and weights based on

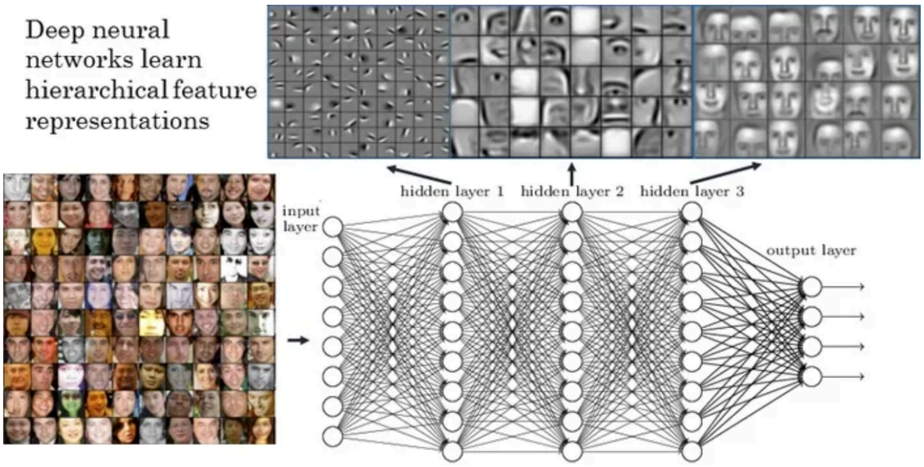

Figure 1: Features Learned in Different Layers.

Source: https://medium.com/@fenjiro/face-id-deep-learning-for-face-recognition-324b50d916d1

errors.

Following contents focus on illustrating different types of layers, introducing special techniques tailored for CNNs, and a few famous CNN models.

2.1.1 Convolutional Layer

As the name indicates, this layer involves convolution calculations. Notably, although researchers commonly refer to this operation as convolution, the mathematical procedure executed is technically known as cross-correlation. In this paper, we will also call this process convolution. Inputs to this layer can be either two-dimensional or three-dimensional. It is imperative that the dimensions of the filter (or kernel) used for convolution operations match those of the input.

Two-dimensional inputs can be simply regarded as flat pictures, and the corresponding filters are generally called windows. The most common filter size is 3x3, various other sizes can also be employed. The convolution process involves placing the filter at every feasible position on the image and computing the sum of the element-wise products. For three-dimensional inputs, both images and filters can be conceptualized as stacks or volumes. In this context, convolution involves the stack of filters sliding over the stack of images. Despite this added depth, the underlying convolution calculation remains consistent. After obtaining the scalar product at each position, activation functions are applied to derive the values in the output. A higher activation value indicates a better match to the particular feature represented by the filter.

While several activation functions are available, the Rectified Linear Unit (ReLU) is the most commonly used. It operates by retaining the maximum value; however, negative values will be replaced by 0.

After performing convolutions, the sizes of inputs are often reduced. Given that filter values signify the features to be learned, the resulting output is a more abstracted representation [12]. Yet retaining the original input sizes is achievable by employing padding (adding pixels with a value of ”0” around the image) and adjusting the stride (determining the step size the filter moves).

2.1.2 Pooling Layer

The pooling process is very similar to convolution: a filter, or a set of filters, glides over the image or stack of images. However, rather than computing scalar products, a certain value with in the filter is chosen according to the type of pooling and subsequently filled into the output.

Pooling layers achieve down-sampling by harnessing the principle of image local correlation, which refers to the inherent association among neighboring pixels within an image. This principle means that pixels that are close to each other in the same image generally have similar values due to the patterns of image. Consequently, pooling layers are relatively intuitive to use, and the key idea is to use a single value to represent the feature in an area.

Taking Maximum Pooling as an example, if a certain feature is detected in a certain area, then a big pixel value will be logged into the output. If this feature is not detected, the maximum value selected will be very small or even ”0”. Thus, pooling layers can reduce the sizes of images while retaining useful information [13]. There are several types of pooling options, such as Minimum, Average, and etc. Max Pooling and Avg Pooling are the most commonly used ones among them.

One impotant benefit of pooling layers is that there is no parameters or weights to learn, as these layers are only for down-sampling. Therefore, such layers save computational resources with no extra cost. In addition, setting strides and paddings are also possible in pooling layers, although paddings are rarely used.

2.1.3 Fully-Connected Layer

In a fully-connected layer, which is also known as the dense layer, every neuron is directly connected to every other neuron in the previous and subsequent layer. The main purpose of the fully-connected layer is that after many convolution and pooling layers, the output contains distributed representations for the input image, then features with stronger capabilities can be built with the current representations [14]. Neurons in such layers vote for the predictions based on the computation with weights and parameters learned, therefore these layers are generally placed at the end of CNN models.

Nevertheless, involving a large amount of parameters is one significant drawback of the fully-connected layer, which consequently leads to higher computational cost in training [15]. Applying proper dropout techniques, which set random subsets of activations to zero, can control the number of nodes and connections, as well as solving overfitting problems.

2.1.4 Skip Connections

Skip connections [16], also known as residual connections, are designed to train very deep neural networks. The concept of skip connections involves taking the output from one layer and, instead of only forwarding it to the immediately adjacent layer, also passing it to a deeper, non-adjacent layer. This approach ensures that deeper layers receive additional inputs, enabling them to extract key patterns from images more precisely. In reality, the training errors of normal CNNs decrease first and then increase, which indicates that having deeper neural networks does not always yield a better performance. But for Residual Networks that uses skip connections, the training errors decrease and eventually reach a plateau, allowing people to train very complex CNNs.

2.1.5 Transpose Convolutions

As previously noted, typical convolutional operations tend to either retain or reduce the spatial dimensions of the input. Nevertheless, it is also possible to enlarge the size of input by using a special type of convolution called Transpose Convolution [17], [18]. Rather than applying paddings and use filters to convolve over the input, Transpose Convolution operates directly on the output. This operation firstly pads the input to generate a larger image, and use the filter to decide the output of each pixel. In certain classes of problems, for example, semantic segmentions, it provides an effective up-sampling method. Spatial information lost during down-sampling can be retrieved by restoring the initial resolutions.

2.1.6 Transfer Learning

Transfer Learning involves leveraging pre-trained models to perform related tasks on own dataset. By using pre-trained weights and parameters, researchers can often achieve satisfying performance with minimum effort. Typically, the output layer of the pre-trained model is modified to align with the number of classes in the target dataset. For more challenging tasks, it might also be necessary to fine-tune the last few layers of the model.

2.2 LeNet-5

LeNet-5 was proposed by Yann LeCun for hand-written digit recognition in 1998 [10], and it is widely acknowledged as the foundation of modern CNNs. The architecture of LeNet-5 is shown in Figure 2. Although some of the design ideas no long fit into today's problem, it is still a good starting point of exploring CNNs.

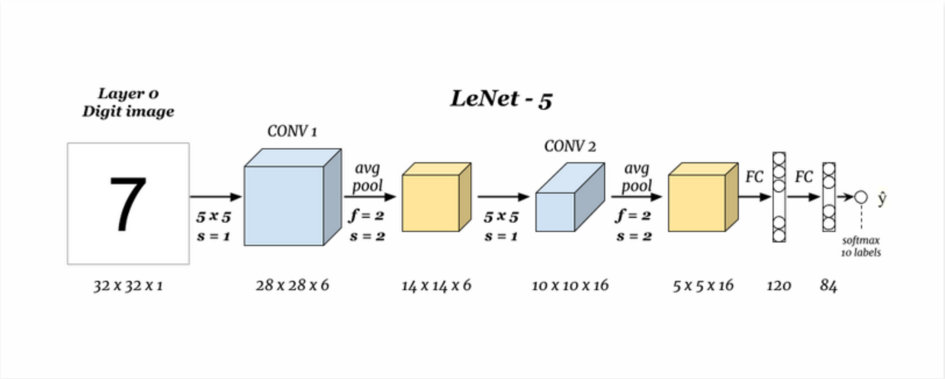

Figure 2: Architecture of LeNet-5.

Source: `https://www.philschmid.de/getting-started-with-cnn-by-calculating-lenet-layer-manually`

The first design principle worth mentioning is the structure of LeNet-5, in which every convolutional layer is immediately followed by a pooling layer. Secondly, the dimensions of images are gradually decremented with the increase of depth. These principles are commonly seen in modern CNNs, and it offers the advantage that the size reduction is controlled to a reasonable speed. Moreover, by gradually down-sampling the feature maps, models are able to capture representative features, while mitigating vanishing gradient problem.

The great LeNet-5 raised the popularity of CNN, but some of its design ideas are outdated now. It primarily used the Hyperbolic Tangent (tanh) as its activation function, which is replaced by ReLU today. Although both functions introduce non-linearity, ReLU is able to promote faster convergence. Besides, Average Pooling was the predominant option back to the days that LeNet-5 was proposed. However, Max Pooling is proved to better capture the most outstanding features in the feature maps and provide a form of translation invariance.

2.3 AlexNet

Another landmark CNN model is the AlexNet [19], which was trained for the ImageNet LSVRC-2010 competition. This dataset include over 1 million high-resolution images from 1,000 classes, and AlexNet achieved 37.5% and 17.0% error rates of top-1 and top-5 respectively. These error rates proved that such model is significantly outperforms the piror state-of-the-art.

Firgue 3 presents the architecture of AlexNet. It uses 5 convolutional layers and 3 fully-connected layers. Undoubtedly, these learned layers with parameters are used to extract features from images and calculate weights for voting. Besides, it uses 3 overlapping max pooling layers. As its name suggests, pixels covered by filters at each move are overlapped, which is realised by setting a stride smaller than the size of filter. By leveraging such pooling layers, the model can capture more information, enabling it to learn more complex features.

And there are also some novel features used in AlexNet. As mentioned above, early CNNs often use tanh as the activation function. Nevertheless, AlexNet chose ReLU, whose efficiency is proved by [20]. Models employing ReLU can train considerably faster than their counterparts using tanh. In addition, AlexNet utilized Local Response Normalization (LRN), which can prevent neurons that have a strong activation from dominating the learning process. In general, the output of activations will become more balanced and spread-out.

Furthermore, AlexNet adopts several techniques to reduce overfitting. The first technique is data augmentation, and will be elaborated in the later section. The second one is dropout [21], which refers to aborting the output of a neuron with a probability with 0.5. This technique reduces complex co-adaptations of neurons, making the input of next layer have a different architecture. And the model is forced to learn more robust features.

2.4 VGG

The Visual Geometry Group (VGG) from the University of Oxford thoroughly investigated CNNs of increasing depth with 3x3 filters, and validated that adding more layers can consid-

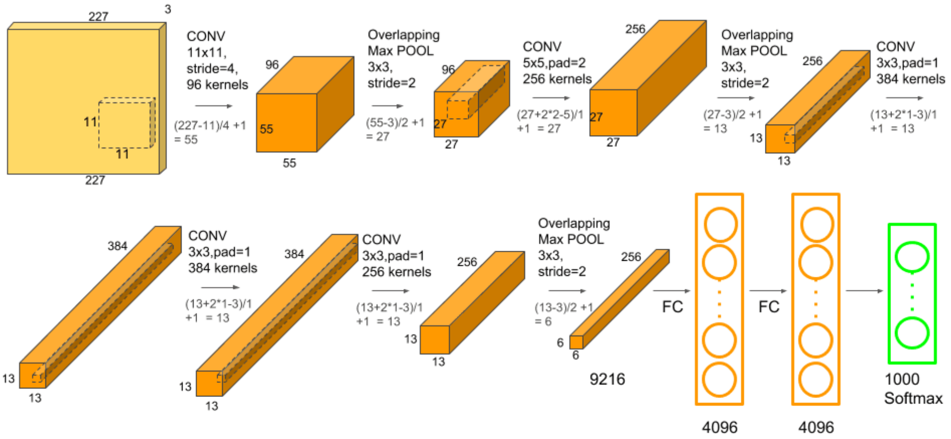

Figure 3: Architecture of AlexNet.

Source: https://www.kaggle.com/code/blurredmachine/alexnet-architecture-a-complete-guide

erably improve the performance [22]. They also published the two best-performing networks, popularly known as VGG-16 and VGG-19. Those networks have 16 and 19 layers respectively, which was very deep at that time.

In contrast to using a single large filter, the VGG team implemented several continuous 3x3 filters, while the actual effect is equivalent. Therefore, the computation efficiency is improved since less parameters are involved. The outstanding performances, as well as the publications by the VGG team, making the VGG-16 and VGG-19 suitable models for transfer learning. Users can download the pre-trained models and manually set the output layers to yield satisfactory results with less effort. However, although the number of parameters are relatively small by using small filters, the overall number of parameters is still very big due to their depth. This means more memory space is needed and model training may cost longer time.

2.5 U-Net

The U-Net was initially designed for biomedical image segmentation in 2015 [17], addressing the challenges posed by limited labeled training samples. The name of this network comes from the shape of its architecture, which is shown in Figure 6. We can find a shrinking path and a symmetric expansion path, together forming the shape of letter ”U”.

The left half of the architecture, which is the shrinking path, is also called the encoder. This path is constructed by stacking multiple convolutional layers and pooling layers, aiming to capture useful features from the images. And due to the nature of convolutional and pooling layers, the heights and widths of inputs are gradually reduced. The right half of the architecture, which is the expansion path, is also called the decoder. Up-sampling by Transpose Convolutions is applied in this path, and the heights and widths of images are increased. This process effectively retrieved details as well as spatial information within the images, assisting the model to utilize extra spatial context. Additionally, it can also be discovered that Skip Connections are involved. This technique allows model to utilize both low resolution but high level contextual information and high resolution but low level feature information.

Specifically, 2 recent researches exploited the U-Net on bushfire related problems. Lee et al. defined a pipeline architecture using Geoscience Australia Data Cube (AGDC) to identify wildfire affected areas in near real-time [8]. It introduces Sentinel-1 C-band dual-polarized Synthetic Aperture Radar (SAR) data to mitigate the impact of cloudy weather, captures useful features based on backscattering coefficients. And Brand et al. proposed a series of models to identify burned areas with a dataset that contains images from Indonesia and Central Africa [7]. The first two models are trained with data from each individual area, and gathered useful data about the wrongly classified data. Then, a global model is trained with data from all regions, and showed a surprisingly well generalization ability despite high dissimilarities between two areas. This research also demonstrates the feasibility of using multiple geographically separated datasets in a common detection framework.

3 Data

This section delves into the dataset utilized in our research, offering detailed expalnations of its source, features, and preparation process. We firstly introduce that data source, which is the Sentinel Hub, and satellite imagery. Then, all three indices incorporated in this study are thoroughly discussed. At the end of this chapter, we presented the specific areas selected, as well as the data augmentation technique.

3.1 Data Source

Satellite images cover several spectral bands, including visible, infrared, near-infrared, and short-wave infrared. By utilizing certain combinations of spectral bands, a variety of features can be extracted from these images. Therefore, many applications use satellite images to do scientific researches. Besides, several satellite image distributors are available, such as Sentinel Hub, Landsat, Google Earth Engine, Planetary Computer, and etc. In this research, we obtained images from Sentinel Hub by using its Processing API, which is a RESTful API that provides access to raw satellite data, rendered images, statistical analysis.

Sentinel Hub is a cloud platform designed for the European Space Agency's (ESA) Sentinel satellite data. We selected data from Sentinel 2 - L2A as the satellite's multispectral imager provides a versatile set of 13 spectral bands, covering all bands we need to calculate the indices for features. Indices used as features in this research include Normalized Difference Vegetation Index (NDVI), Normalized Difference Mositure Index (NDMI), while Normalized Burn Ratio (NBR-RAW) is chosen to obtain labels. Indices selected will be discussed below. Besides, all features are returned in the format of RGB image, and resolution is set to 512 x 512.

3.2 NDVI

The widely used NDVI, which normalizes green leaf scattering in Near Infrared (NIR) wavelengths with chlorophyll absorption in Red (RED) wavelengths, is a simple but poweful index for quantifying green vegetation [23]. Thus, we chose this index as an indirect indicator of carbon content, since healthy plants tend to have higher levels of carbon content. The formula of calculating NDVI is given below:

$$NDVI := INDEX(NIR, RED) = \frac{(NIR - RED)}{(NIR + RED)}$$

From the formula, we can infer that the value range of NDVI is between -1 to 1. Different values are correspond to different land features. Negative values that are close to -1 represent water body, while positive values that are close to 1 represent temperate and tropical rainforests. Values approaching 0 from either positive or negative side generally correspond to barren areas, while higher positive values (about 0.2 to 0.4) represent shrub and grassland.

3.3 NDMI

The NDMI is calculated by using the NIR and Shortwave Infrared (SWIR) bands. The SWIR band indicates variations in vegetation water content and the mesophyll structure within plant canopies. In contrast, the NIR reflectance is influenced by the internal configuration of leaves and their dry matter content, without being impacted by their water content [24]. By using such bands, variations induced by leaf internal structure and leaf dry matter content are avoided, ensuring more accurate results in obtaining the vegetation water content. The formula of NDMI is shown below:

$$NDMI := INDEX(NIR, SWIR1) = \frac{(NIR - SWIR1)}{(NIR + SWIR1)}$$

There is also another index called Normalized Difference Water Index (NDWI), however, this index is generally realted to water body. Thus, we chose the NDMI instead. Higher NDMI values indicate higher vegetation water contents.

3.4 NBR-RAW

A common challenge of all bushfire research is to obtain the labels of corresponding images. Due to the area of interest chosen, which represent the most severe bushfire-affected areas of the ”Black Summer”, we used the NBR-RAW index to obtain the labels of each area selected. This index is the most appropriate choice of detecting burned areas, and darker values indicate burned areas [25]. The formula is displayed below:

$$NBR - RAW := INDEX(NIR, SWIR2) = \frac{(NIR - SWIR2)}{(NIR + SWIR2)}$$

To details of labeling satellite images is as follows. From the formula we can find that the value is between -1 and 1, and the value can also represent the severity of bushfire. For each NBR-RAW image, we iterate through the pixels and calculate the value of the red channel. An area is labeled as ”Burned” once it satisfies the condition that more than 25% pixels have a NBR-RAW value greater than 0.25. By manually checking part of the real satellite images from the Sentinel Hub, errors in this method are acceptable.

3.5 Area of Interest

In this research, we mainly focused on the ”Black Summer” bushfires, as the timeframe of these bushfires is closer. Therefore, we selected the most bushfire-affected areas in the south-eastern coast of Australia, which are shown in Figure 4. In addition, every red dot in the picture represents a individual bushfire, and sizes can be regarded as the scale bushfires. According to this picture, all selected areas in the grid incorporate a relatively large number of wildfires, so the threshold of minimum burned pixels is set to 25%.

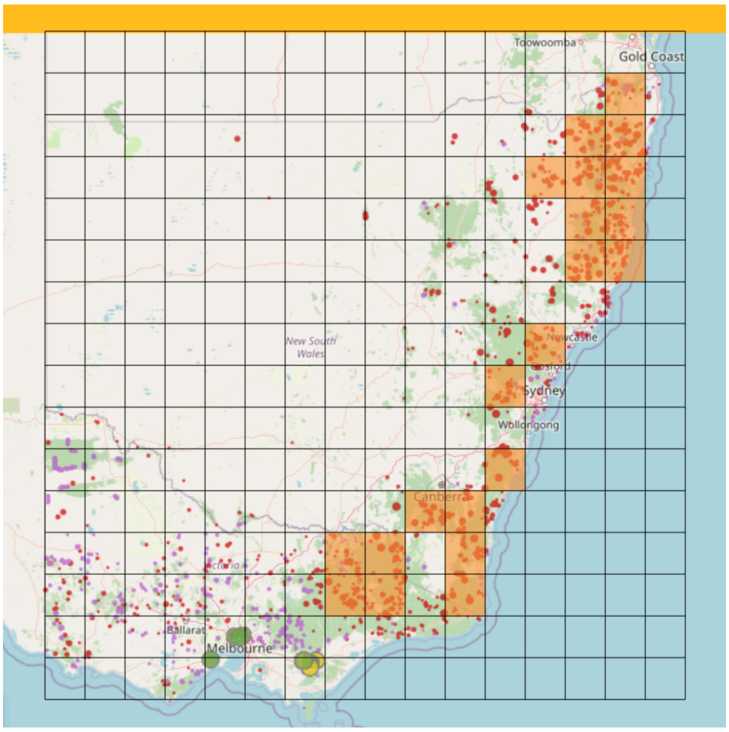

Figure 4: Area of Interest.

3.6 Dataset

Based on the areas selected, we prepared a dataset covered the time range from 1st Jul 2019 to 1st Jul 2020, which matches the time range of the ”Black Summer”. However, this dataset does not include images of the whole year, since the revisit time of Sentinel 2 - L2A satellite is 5 days.

This dataset include features of 1407 burned and 126 unburned satellite images, which means it is a strongly imbalanced dataset. The main reason is due to the way of obtaining labels, as a burned area will remained burned for a period of time. Empirically, plants will not recover until at least next Spring, or even Summer. Nevertheless, several models yield satisfactory results by using imbalanced dataset [26], [27].

3.7 Data Augmentation

Data Augmentation is a technique widely used in machine learning and deep learning, especially for image processing problems. It refers to artificially adding new samples to a dataset by applying transformations to the existing data. Common data augmentation methods include rotation, scaling, random cropping, flipping, color adjustments, and nois injection. [19] discussed the details of these methods.

To mitigate the negative effect caused by the imbalanced dataset, as well as increasing the the number of samples, we applied flipping and cropping techniques in this research. The flipping process is to mirror the image, while cropping is to randomly select a portion of the image. By implementing such techniques, both the size and diversity of the dataset is incremented.

Moreover, the advantages of using data augmentation is not limited to those mentioned above. Firstly, it is a less costly than searching new data, and leverages the existing data to the maximum degree. Also, by exposing the model to a broader set of data variations, it can enhance the generalization ability of the model and prevents overfitting problems.

4 Model Construction and Training

In this section, we delve into the architectural details and training methodologies adopted for the CNNs used in our research. Since some models were thoroughly introduced in previous sections, here, we emphasize the specific variants and configurations tailored for our study. Four CNN architectures were explored in our study: LeNet-5, ResNet, U-Net, and EfficientNet, each with its unique motivations, advantages and challenges.

4.1 LeNet-5

A variant LeNet-5 model is employed in our research. In our architecture, we incorporated 32, 64, and 128 filters in the convolutional layers, progressing from the shallower to the deeper layers. And we altered the activation function from tanh to ReLU because its proven efficiency in modern neural networks. The stride was set to 1 to ensure a more granular extraction of features at each layer. As for the pooling layers, we replaced average pooling with max pooling due to the same reason stated above. And finally, for the fully-connected layers, we used 128 and 1 neurons respectively, fitting into our binary classification problem.

It is worth mentioning that extensive fine-tuning was undertaken to test various parameter combinations. Moreover, a separate model incorporating data augmentation techniques was trained. However, these efforts seem to be a purely waste of time given that the performances of every model were equally disappointingly. So, LeNet-5 is chosen to be the baseline model.

4.2 ResNet

The degradation problem in deep neural networks refers to the decline in performance as the network depth increases. According to the research [16], this issue does not stem from overfitting but from the difficulties in optimizing very deep layers. To address this problem, the authors introduced the Residual Network (ResNet) concept. ResNet employs residual blocks and skip connections to mitigate the degradation problem, allowing deeper neural networks to outperform the shallower ones.

Figure 5 dipicts the residual block and skip connection. In traditional deep neural networks, layers attempt to extract features and learn the matching information from its input to output. In contrast, layers in residual blocks focus on learning the residual (or difference) between the input and output. This difference allows ResNet to ease optimization, avoid degradation problem, and learn adaptively. And as aforementioned, skip connections allow ResNet to receive high resolution images for obtaining extra detailed spatial information.

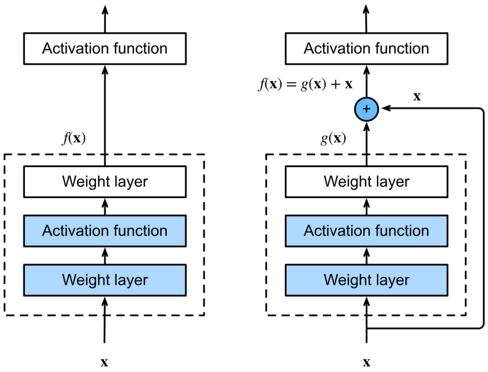

Figure 5: Residual Block and Skip Connection.

Source: `https://d2l.ai/chapter_convolutional-modern/resnet.html`

Additionally, ResNet, which addresses the performance degradation problem in deep neural networks, emerges as one of the most suitable architectures for transfer learning. It offers a variety of architectures with differing depths so that researchers can choose models based on the complexities of their tasks. Furthermore, those models are trained with comprehensive dataset encompassing a diverse set of images, e.g. ImageSet, allowing models to learn different features. Also due to this reason, ResNet has excellent generalization ability and is adaptable to a wide range of applications.

In our research, we investigated the most commonly used ResNet-50 and ResNet-101 because they both provide a good balance between computational efficiency and performance outcomes. Again, the suffix numbers in their names denotes their depth of the network. We trained models with and without data augmentation implemented for each type of network. The last two fully-connected layers are adjusted to have 1024 and 1 neurons separately to fit into the burned area identification task.

4.3 U-Net

Details about LeNet-5 are discussed in Section 2, so we will exclusively illustrate the U-Net model used in our research in this section. We basically followed the origin design of the U-Net. For the contracting path, the number of filters are 64, 128, 256, and 512 at different levels. For the expansion path, the number of filters decreased from 256 to 128, and finally 64 before delivered to the fully-connected layers. Besides, transpose convolutional layers, which can be regarded as the equivalent of max pooling layers from the contracting path in a reversed way, are used in the expansion path. Finally, skip connections are set to ensure deeper layers have extra input information about the position information of pixels.

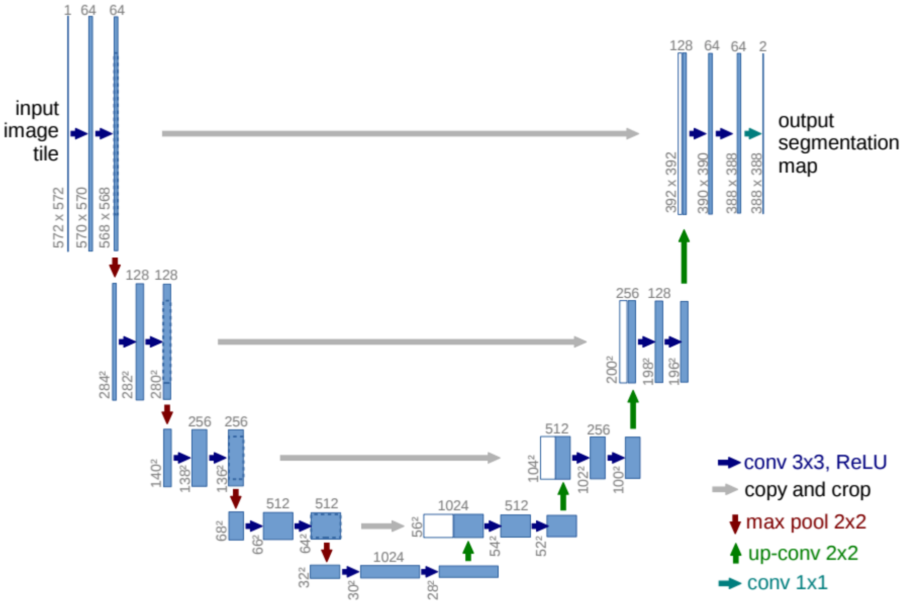

Figure 6: Architecture of the U-Net [17].

Same as all models above, two different U-Net models are trained. One is implemented the data augmentation techniques, while the other is not.

4.4 EfficientNet

As empirically observed, different scaling dimensions are not independent. Images with higher resolution generally require deeper neural networks since more features are encompassed and larger receptive fields are needed. Compound Scaling, together with EfficentNet [28] were proposed for uniformly scaling network width, depth, and resolution in a principled way by using a compound coefficient.

The idea behind Compound Scaling is that, the performance of networks by scaling in a single dimension is generally poor, while uniformly sacling in all dimensions (resolution, width, depth) may yield better results. The compound coefficient is specified to control the amount of resources available for model scaling, and another three parameters are involved to assign extra resources to to network width, depth, and resolution respectively. As illustrated in Figure 7, there is a distinct difference between the compound scaling method and conventional approaches. It is evident that the network resulting from compound scaling achieves a balanced harmony across these dimensions.

EfficientNet also has multiple architectures from B0 to B7, enabling users to choose desired depth. We chose EfficientNetB5 and EfficientB6 since their recommended image input resolutions are 456x456 and 528x528 separately. In addition, the finally fully-connected layers are again modified. Two versions of each EfficientNet are trained based on whether using data augmentation or not.

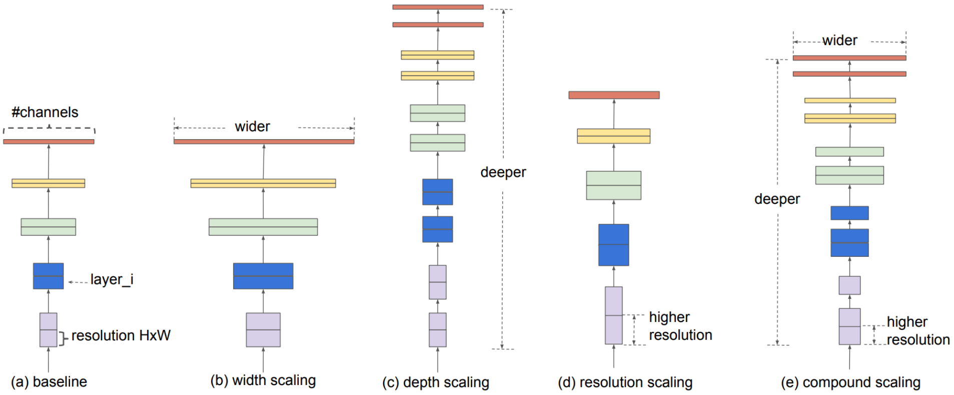

Figure 7: Compound Scaling [28].

5 Results and Analysis

We trained our models across 10 epochs using a batch size of 32. Each model was trained five times, and the average values are presented in Table 1. As previously mentioned, half of the models implement data augmentation techniques. These models are denoted with a ”DA” following their names. In this chapter, we will first introduce the chosen evaluation metrics, then discuss the performance of each model in detail and analyze the underlying reasons. Three evaluation metrics are selected: accuracy, precision, and recall.

5.1 Evaluation Metrics

5.2 Evaluation Metrics

Accuracy is determined by dividing the number of correct predictions by the total number of predictions. It quantifies the proportion of instances that are accurately predicted.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

Precision measures the ratio of true positive predictions to all positive predictions. For instance, from the results table, it can be observed that over half of the models report a precision of 0.5896. This suggests that among all the images predicted as burned, approximately 59% are genuinely burned.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall, which is also termed as the True Positive Rate, is defined as shown below. It computes the proportion of true positives out of all positive instances. For instance, the EfficientNetB5 model with data augmentation achieves a recall of 0.9885. This implies that among all the burned images, a mere 0.0015% are misclassified as unburned.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

5.3 Result Interpretation

A closer examination of Table 1 reveals that over half of the models exhibit identical performance metrics: accuracy and precision at 0.5896, and recall at a perfect 1.0000. The accuracy

	Accuracy	**Precision**	**Recall**
LeNet-5	0.5896	0.5896	1
LeNet-5 (DA)	0.5896	0.5896	1
U-Net	0.5896	0.5896	1
U-Net (DA)	0.5896	0.5896	1
EfficientNetB5	0.5896	0.5896	1
EfficientNetB6	0.5896	0.5896	1
ResNet-50	0.5896	0.5896	1
ResNet-101	0.5896	0.5896	1
EfficientNetB5 (DA)	0.9327	0.9376	0.9885
EfficientNetB6 (DA)	0.9207	0.9207	0.9942
ResNet-50 (DA)	0.9273	0.9319	0.9927
ResNet-101 (DA)	0.9338	0.9342	0.9988

Table 1: Performances of Different Models

shows that a suboptimal classification ability, only about 60 percent of images can be truly classified. However, if we focus on the precision and recall, it is evident that the weights and parameters learned are predominantly for burned areas. Following the calculation of recall, the result suggests not a single burned area is predicted to be unburned. Also, according to the precision formula, around 40 percent images are false positives, which means unburned areas are mistakenly labeled as burned areas.

Six types of CNNs produced theses uniform output. These CNNs vary in architectures and complexities, and the total numbers of layers ranged from a few layers to over a hundred layers. This diversity in architecture suggests that complexity isn't the primary contributor to the observed performance. Then the only key influence to the performances is the dataset volume. Due to the fact that our dataset only have 1533 images, the size of the dataset is the main factor limiting the performances.

The phenomenon above can also be found by comparing the last eight rows of the table. When data augmentation techniques are employed in the corresponding models, all three evaluation metrics are improved. This again validates that having sufficient data input is one important prerequisite of training CNNs.

Nevertheless, by comparing only the models implementing data augmentation, we can infer that architectures and complexities can also cause impacts on performances. Variants of the LeNet-5 and U-Net models are relatively less complicated, making them unable to recognize features effectively. On the contrary, more complicated models (EfficientNetB5, EfficientNetB6, ResNet-50, and ResNet-101) yield satisfactory results on all evaluation metrics when supplemented with data augmentation.

The top-performing models in the table consistently achieve metrics around 93% for both accuracy and precision, and nearly 100% recall. Those values proved that our models can almost identify all burned images correctly, and making true predictions at the same time. Yet, there is no distinct difference in the values of metrics. This observation indicates that at least EfficientNetB6 and ResNet-101 may be over complicated. A more detailed discussion on EfficientNet and ResNet structures is available in earlier sections, highlighting that these highly complex architectures might lead to unnecessary computational resource consumption.

6 Future Works

The data engineering process is described in Section 3, where we elaborated on the assignment of labels to each image. Although the feasibility of this method was validated by manually comparing the labels with actual satellite images, it is specifically tailored to label areas severely affected by the ”Black Summer” bushfires. In other words, the limits used might not be appropriate for other areas or different bushfires. Moreover, we experimented with various limit values and selected the best one (0.25) based on our observations. However, the optimal limit might vary in different scenarios. These challenges point out a vital future direction: developing a robust automated approach for image labeling.

Besides, due to the compact timeframe, we could not pinpoint an effective and cost-efficient model for recognizing burned areas. Results proved that EfficientNetB6 and ResNet-101 are too sophisticated for the small dataset used. However, given that the only less sophisticated models tested are EfficientNetB5 and ResNet-50, it can not be guaranteed that these models did not consume resources for learning nothing. This uncertainty highlights the need for further optimization of the CNNs used to ensure they are exploited to the maximum extent.

Lastly, our research focuses solely on identifying burned areas from historical satellite images. An enhancement would be to extend this into a pipeline architecture that processes real-time data for accurate predictions.

7 Conclusion

In this research, we employed various CNN models to identify areas at the highest risk of wildfires. The basic background knowledge about CNN was provided, including the functionalities of each type of layers, as well as other techniques that can help to improve performances. Meanwhile, the whole process of establishing our dataset is explained. Satellite images were sourced from the Sentinel Hub via the Processing API, and features were extracted to construct our dataset. Indices like NDMI and NDVI were derived from these satellite images to gauge hydration levels and carbon content respectively, whereas NBR-RAW was used for image labeling. Our primary focus being the ”Black Summer” bushfires meant we manually selected our areas of interest.

For conducting the experiments, we elaborated on the following information. Principles, rationals, benefits, and challenges of four kinds of CNN architectures and six particular models are explained in detail. We trained a total of twelve CNNs and gave a thorough performance comparison and analysis. We stated two convincing reasons for the discrepancies in performance, with the predominate one being that models were not fed with enough data input, and the less important one being that some models are lacking in complexity. Furthermore, methods to bridge these gaps are also proposed.

Nevertheless, our study has some limitations. Firstly, the class labeling method is tailored for the ”Black Summer” bushfires and may not be universally applicable. Moreover, the ideal neural network that is both accurate and cost-efficient is not yet identified. In the future, modifying the model to a real-time detection pipeline would be immensely beneficial, especially for addressing urgent bushfire incidents.

References

[1] P. A. Cameron, B. Mitra, M. Fitzgerald, *et al.*, "Black saturday: The immediate impact of the february 2009 bushfires in victoria, australia," *Medical Journal of Australia*, vol. 191, no. 1, pp. 11–16, 2009.

[2] M. Cruz, A. Sullivan, J. Gould, *et al.*, "Anatomy of a catastrophic wildfire: The black saturday kilmore east fire in victoria, australia," *Forest Ecology and Management*, vol. 284, pp. 269–285, 2012.

[3] D. Celermajer, R. Lyster, G. M. Wardle, R. Walmsley, and E. Couzens, "The australian bushfire disaster: How to avoid repeating this catastrophe for biodiversity," *Wiley Interdisciplinary Reviews: Climate Change*, vol. 12, no. 3, e704, 2021.

[4] A. I. Filkov, T. Ngo, S. Matthews, S. Telfer, and T. D. Penman, "Impact of australia's catastrophic 2019/20 bushfire season on communities and environment. retrospective analysis and current trends," *Journal of Safety Science and Resilience*, vol. 1, no. 1, pp. 44–56, 2020.

[5] J. R. Bergado, C. Persello, K. Reinke, and A. Stein, "Predicting wildfire burns from big geodata using deep learning," *Safety science*, vol. 140, p. 105 276, 2021.

[6] S. H. Oh, S. W. Ghyme, S. K. Jung, and G.-W. Kim, "Early wildfire detection using convolutional neural network," in *International Workshop on Frontiers of Computer Vision*, Springer, 2020, pp. 18–30.

[7] A. Brand and A. Manandhar, "Semantic segmentation of burned areas in satellite images using a u-net-based convolutional neural network," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 47–53, 2021.

[8] I. K. Lee, J. C. Trinder, and A. Sowmya, "Application of u-net convolutional neural network to bushfire monitoring in australia with sentinel-1/-2 data," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 573–578, 2020.

[9] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.

[10] Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[13] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.

[14] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, p. 495, 2017.

[15] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, Ieee, 2017, pp. 1–6.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[23] S. Hub, *Normalized difference vegetation index*. [Online]. Available: `https://custom-scripts.sentinel-hub.com/sentinel-2/ndvi/`.

[24] S. Hub, *Normalized difference moisture index*. [Online]. Available: `https://custom-scripts.sentinel-hub.com/sentinel-2/ndmi/`.

[25] S. Hub, *Nbr-raw (normalized burn ratio)*. [Online]. Available: `https://custom-scripts.sentinel-hub.com/sentinel-2/nbr/`.

[26] M. J. Sousa, A. Moutinho, and M. Almeida, “Wildfire detection using transfer learning on augmented datasets,” *Expert Systems with Applications*, vol. 142, p. 112 975, 2020.

[27] C. Lai, S. Zeng, W. Guo, X. Liu, Y. Li, and B. Liao, “Forest fire prediction with imbalanced data using a deep neural network method,” *Forests*, vol. 13, no. 7, p. 1129, 2022.

[28] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.