

# **Canada Wildfire Next-Day Spread Prediction Tools Using Deep Learning**

by

Xiang Fang

A thesis  
presented to the University of Waterloo  
in fulfillment of the  
thesis requirement for the degree of  
Master of Applied Science  
in  
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

© Xiang Fang 2024

## **Examining Committee Membership**

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor(s):                   Mark Crowley  
  Professor, Dept. of Electrical and Computer Engineering,  
  University of Waterloo

Reader:                         Zhou Wang  
  Professor, Dept. of Electrical and Computer Engineering,  
  University of Waterloo

Reader:                         Kate Larson  
  Professor, Dept. of Computer Science, University of Waterloo

### **Author's Declaration**

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

Wildfires have become a pressing issue globally, with their increasing frequency and intensity causing significant environmental, economic, and human impacts. Traditional wildfire prediction methods, while useful, often fall short in time complexity or simulation on heterogeneous landscapes. This thesis explores the application of deep learning models, especially convolutional networks, to improve the accuracy and reliability of wildfire spread predictions. By leveraging advanced machine learning techniques, this research aims to enhance the current prediction capabilities and provide better tools for Canadian wildfire management and mitigation.

Utilizing a comprehensive dataset from various sources, this thesis integrates multiple features such as weather data, vegetation types, and topographical information. The research introduces a novel module for fusing multi-modal data, which enhances the performance of U-shape deep learning models like U-Net. Additionally, an innovative U-shape network structure with atrous convolution and new attention implementation was developed to further improve prediction accuracy. The thesis also proposes an enhancement method that amplifies grouped error pixels for element-wise error computation for model training. The novel data fusion module proposed in this thesis has been proven to improve the baseline model on the F1 score, while the new model I suggest outperformed the baseline model and its two variants on the same metric. In the final part of the thesis I proposed various additional enhancement methods to improve performance further, it has shown its statistical significance under certain conditions when applied to BCELoss.

By enhancing the predictive capabilities of wildfire spread models, this thesis offers valuable insights for emergency responders and policymakers, aiding in better resource allocation and risk mitigation strategies. The deep learning methodologies developed in this study are versatile and have potential applications in other fields requiring spatial data predictions, such as intelligent healthcare, flood forecasting, and disease spread modelling.

## **Acknowledgements**

I would like to thank my family for their support and encouragement throughout my master's studies. Their belief in me has been invaluable.

I am also grateful to my supervisor, Mark Crowley, for his guidance and assistance. His expertise has been essential in the development of this thesis.

Additionally, I appreciate the help from researchers outside the university who assisted with data collection and provided background training. Their contributions have been greatly appreciated.

# Table of Contents

<b>Examining Committee</b>	ii
<b>Author's Declaration</b>	iii
<b>Abstract</b>	iv
<b>Acknowledgements</b>	v
<b>List of Figures</b>	x
<b>List of Tables</b>	xii
<b>1 Introduction</b>	1
1.1 Thesis Tasks . . . . .	2
1.2 Research Frame Design . . . . .	3
<b>2 Background Knowledge</b>	5
2.1 Wildfire Background Information . . . . .	5
2.1.1 Fuel Type in FBP System . . . . .	6
2.1.2 Fire Weather Index System . . . . .	6
2.2 Machine Learning Background Information . . . . .	7
2.2.1 Machine Learning and Deep Learning . . . . .	7
2.2.2 Convolutions and Convolution Neural Networks . . . . .	7

2.2.3	Attention Mechanisms . . . . .	10
2.2.4	U-Net, U-Net++, Swin-UNet . . . . .	13
2.3	Significance Testing . . . . .	14
2.3.1	Significance Testing and the P-Value . . . . .	14
2.3.2	Paired Samples T-Test . . . . .	14
<b>3</b>	<b>Literature Review for Wildfire Prediction Tools</b>	<b>16</b>
3.1	Traditional Tools . . . . .	16
3.1.1	Rothermel Fire Spread Equations . . . . .	16
3.1.2	Fire Behavior Prediction (FBP) System . . . . .	17
3.1.3	FARSITE . . . . .	18
3.1.4	Prometheus, Burn-P3 and Burn-P3+ . . . . .	18
3.2	Machine Learning Literature Review . . . . .	18
3.2.1	Machine Learning in Fire Spread Predictions . . . . .	19
3.2.2	Machine Learning in Fire Occurrence Predictions . . . . .	20
3.2.3	A Review of Machine Learning Applications in Wildfire Science and Management . . . . .	20
<b>4</b>	<b>Data Acquisition and Processing</b>	<b>22</b>
4.1	National Wildfire Dataset . . . . .	22
4.2	Burned Area and Hotspot . . . . .	25
4.2.1	Sight Window, Mask and Resample . . . . .	29
4.2.2	Estimated Burning Areas . . . . .	31
4.2.3	Cumulative Burning Areas . . . . .	31
4.3	Fuel . . . . .	32
4.4	Elevation . . . . .	33
4.5	Fire Weather Index . . . . .	34
4.6	Tebular(1D) Features . . . . .	35
4.6.1	Agency . . . . .	35
4.6.2	Temporal Feature . . . . .	35
4.6.3	Fire Cause . . . . .	35

<b>5 Fusion of Heterogeneous Features</b>	<b>37</b>
5.1 Block Structure . . . . .	38
5.2 Performance . . . . .	39
5.2.1 Ablation study on 1D features . . . . .	39
5.2.2 Controlled Experiment with MLP . . . . .	40
5.2.3 Experimental Result Analysis . . . . .	41
<b>6 Burned Area Prediction</b>	<b>42</b>
6.1 Experiment Design . . . . .	42
6.1.1 Hyperparameters . . . . .	42
6.1.2 Loss . . . . .	43
6.1.3 Optimizers . . . . .	43
6.1.4 Evaluation Metrics . . . . .	44
6.1.5 Additional Settings . . . . .	44
6.2 Past Work Reproduction . . . . .	45
6.2.1 ANN . . . . .	45
6.2.2 DCIGN . . . . .	46
6.2.3 FireCast . . . . .	47
6.2.4 Experimental Result and Analysis . . . . .	47
6.3 Comparing Convolutional U-Shape Networks and Attention-Based U-Shape Networks . . . . .	48
6.3.1 U-Net . . . . .	48
6.3.2 Extensions . . . . .	50
6.3.3 Experimental Result . . . . .	53
<b>7 AA-Unet, a New Multi-scale, Convolutional Attention Based Architecture</b>	<b>55</b>
7.1 AA-Unet . . . . .	56
7.2 Atrous Spatial Pyramid Convolution (ASPC) . . . . .	57

7.3	Windowed Convolutional Attention (WCA) . . . . .	58
7.4	Multi-Head WCA . . . . .	60
7.5	Experimental Result . . . . .	61
7.6	Ablation Study . . . . .	62
<b>8</b>	<b>Spread Error Attenuation Field (SEAF)</b> . . . . .	<b>64</b>
8.1	Introduction . . . . .	64
8.2	Core Idea . . . . .	66
8.2.1	SEAF Kernel . . . . .	66
8.2.2	Layer Definition . . . . .	66
8.2.3	Weight Calculation of Pixels . . . . .	67
8.2.4	OR Operation After Spread Using Error Mask (optional) . . . . .	68
8.2.5	Additional Scaling Factor (optional) . . . . .	68
8.2.6	IoU Scaling Factor (optional) . . . . .	68
8.2.7	Experimental Result . . . . .	69
<b>9</b>	<b>Conclusions and Future Works</b> . . . . .	<b>72</b>
9.1	Conclusions . . . . .	72
9.2	Future Works . . . . .	72
9.2.1	Dataset Making . . . . .	73
9.2.2	Data Augmentation . . . . .	73
9.2.3	Multi-Class Prediction . . . . .	73
9.2.4	Comparison to traditional models . . . . .	73
<b>References</b>		<b>74</b>

# List of Figures

1.1	The Frame of Prediction Model . . . . .	3
1.2	An example of ground truth(L) and prediction output(R) . . . . .	4
2.1	CFFDRS [27] . . . . .	6
2.2	FWI System [28] . . . . .	7
2.3	2D Convolution Demonstration (kernel=3 stride=1) . . . . .	9
2.4	Atrous Convolution Demonstration on kernel size = 3 . . . . .	10
2.5	Scaled Dot-product Attention [36] . . . . .	11
2.6	Multi-head Attention [36] . . . . .	12
2.7	Original U-Net [31] . . . . .	13
4.1	Canadian provincial and territorial boundaries . . . . .	23
4.2	KDE of burned areas from 1994 to 2021 with 95 percent mark . . . . .	24
4.3	KDE of 95 percent burned areas from 1994 to 2021 . . . . .	25
4.4	The Bounding Box and affine transformation Matrix . . . . .	30
4.5	Estimated Burning Areas . . . . .	31
4.6	Cumulative Burned Areas . . . . .	32
4.7	Elevation from Multiple Tiles . . . . .	34
5.1	Hybrid Bottleneck Block in U-shape encoder-decoder . . . . .	37
5.2	Hybrid Bottleneck Block . . . . .	38
5.3	Controlled Experiment on Resnet 18 . . . . .	40

6.1	U-Net+HBN Structure . . . . .	49
6.2	5-fold loss curve of U-Net . . . . .	50
6.3	Feature aggregation on $X_{de}^2$ in U-Net 3+ . . . . .	51
6.4	Swin Transformer Block [7] . . . . .	52
6.5	5-fold loss curve of U-Net 3p(L) and Swin-Unet(R) . . . . .	53
7.1	AA-Unet+HBN Structure . . . . .	56
7.2	Atrous Spatial Pyramid Convolution (ASPC) . . . . .	57
7.3	Windowed Convolutional Attention (WCA) . . . . .	58
7.4	Data Flow in WCA . . . . .	59
7.5	Data Flow in Multi-head WCA . . . . .	60
8.1	A SEAF Example with grouped error . . . . .	65
8.2	A SEAF Example without grouped error . . . . .	65
8.3	SEAF Kernel . . . . .	66
8.4	Grid A(Left) & Grid B(Right) . . . . .	67
8.5	An example of prediction: Ground Truth(L), AA-Unet trained on BCELoss(C) and trained on SEAF-BCELoss(R) . . . . .	70

# List of Tables

1.1	Fire Sample Metadata . . . . .	4
4.1	Burned Areas in Hectares (Canada, 1994-2021) . . . . .	26
4.2	List of Abbreviations . . . . .	27
4.3	Fire Event Counts Per Province, Territory and National Park Canada (1994-2021) . . . . .	28
4.4	Hotspot Yearly Counts (1994-2021) . . . . .	29
4.5	Canadian Forest FBP Fuel Types . . . . .	33
4.6	Fire Cause Values and Descriptions . . . . .	36
5.1	Performance Metrics of the Ablation Study . . . . .	40
5.2	T-test on U-Net+HBN and U-Net . . . . .	40
5.3	Perfomance Metrices of the Controlled Experiment . . . . .	41
6.1	ANN . . . . .	45
6.2	DCIGN . . . . .	46
6.3	FireCast . . . . .	47
6.4	Performance Metrics for Reproduced Models . . . . .	48
6.5	Performance Metrics of U-Net/U-Net 3P/Swin-Unet with HBN . . . . .	53
6.6	T-test on U-Net 3p and Swin-Unet (with/without HBN) . . . . .	54
7.1	Performance Metrics of AA-Unet, U-Net, U-net 3+ and Swin-Unet . . . . .	61
7.2	T-test on AA-Unet/U-Net with HBN . . . . .	62

7.3	Performance Metrics of the Ablation Study on AA-Unet . . . . .	62
8.1	Performance Metrics of BCELoss/SEAF-BCELoss on AA-Unet+HBN . . .	69
8.2	T-test on AA-Unet using SEAF-BCELoss/BCELoss with HBN . . . . .	70

# Chapter 1

## Introduction

The wildland fires (wildfires in short) in Canada are becoming an increasingly severe issue. In recent years, they have become more frequent and intense, affecting vast areas and resulting in great financial loss to the country. As a result of climate change, the weather has become hotter and drier, worsening the situation by increasing the possibility of ignition and spread, resulting in higher costs for fire management and recovery. With certain conditions, burning pollution such as ozone or carbon monoxide [6] could spread for very long distances and leave a long-term threat to human health. These trends emphasize the importance of fire behaviour prediction. Due to the complexity and randomness of fire hazard factors such as weather, human activity and fuel type, this prediction task is challenging.

Traditional wildfire prediction methods mainly rely on the statistics from historical data and expert judgment. However, the dynamic nature and complexity of wildfires in recent years, especially with large and multi-dimensional datasets obtained from satellites and other sources, may require more adaptive and sophisticated methods. Additionally, traditional methods could have less tolerance for error, distortion and disappearance in datasets.

Machine learning techniques, particularly deep learning, have shown their great performance in many fields. By transforming the wildfire behaviour prediction task into specific classification or regression tasks, deep learning models, introduce the great potential of dealing with several features, identifying complex relationships and adaptive learning. Some recent research has suggested the potential of deep learning technologies to outperform traditional methods in some wildfire prediction tasks. For example, Ghali and Akhloufi [15] provided a comprehensive review of deep learning solutions on several

wildland fire-related tasks including prediction, reflecting the capabilities for deep learning models learning complex wildfire behaviour.

This thesis focuses on exploring national-level wildfire spread prediction tasks using deep learning models. The research aims to improve the comprehensive performance of wildfire spread predictions and contribute to saving wildfire control and reaction costs such as resource allocation, damage prevention or minimization, rescue and evacuation. The research converted full procedures from data collection to model training and analysis. In data preparation, several sets of data with different attributes were gathered and merged by using different processing methods to ensure good alignment. Then a new module for fusing features with different dimensions was introduced, which provides an efficient way to feed data to U-shape deep learning models. After that, I proposed a novel structure based on U-Net which has the potential to not only help wildfire prediction but also spatial prediction/classification/segmentation tasks in other fields. Finally, an enhancement method for the 2D criterion function relying on element-wise error was proposed.

## 1.1 Thesis Tasks

The research aims to provide deep learning solutions with better performance than existing tools on the wildfire next-day (step-wise) burned area prediction task. In this thesis, the task is identified as binary classification on the 2D grid from two consecutive days of data. The model takes several features from one day and then predicts the combination of the burning area and the burned area of the next day.

The wildfire input features in this thesis contain both spatial(2D) and tabular(1D) features which are not compatible with most of the deep learning models that were designed to take single-modal input. The first task is about the efficient solution of data fusion in the prediction task. The performance of the solution could be valued by comparison to a simple implementation of a fusion module using Multi-Layer Perceptron (MLP).

Considering the situation that deep learning techniques have not been commonly adopted in this field, the second goal is to choose a modern model that performed well in a similar task and test its behaviour on the wildfire prediction task. The U-Net [32] structure has shown its high efficiency in medical imaging segmentation and the design has influenced the progress in multiple research fields. Because of the similarities between medical imaging segmentation and fire spread shape segmentation, U-Net was chosen to be the baseline model. Based on the idea of a baseline model, I investigate the possibility of better performance on variants of the U-Net structure. UNet 3+ increases the number of connections

among layers and Swin-Unet replaces the convolution module with a windowed attention module.

The last task is bringing better prediction performance which is the main value of this research. Inspired by the ideas of the above models, a new U-shape network containing two key modules was proposed. This network was tested against the three existing models to prove its strength. Besides the prediction model, I have also explored the possibility of enhancing grouped error during loss calculation. A method was recommended which can amplify localized error pixels during element-wise error calculation. It was tested on the loss function adopted in this thesis.

## 1.2 Research Frame Design

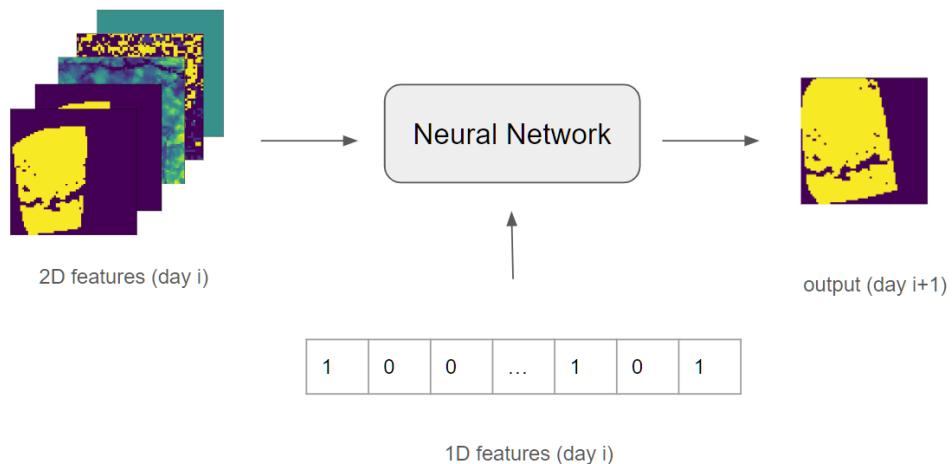


Figure 1.1: The Frame of Prediction Model

The prediction model takes two inputs and generates one output. One input contains five spatial features including ideal geographical shape, weather, vegetation and historical shape of fire. Another input contains three supplementary tabular features regarding metadata such as the region or the reason for fire occurrence. The model is expected to predict the shape of the next-day burned area as a spatial map (grid map).

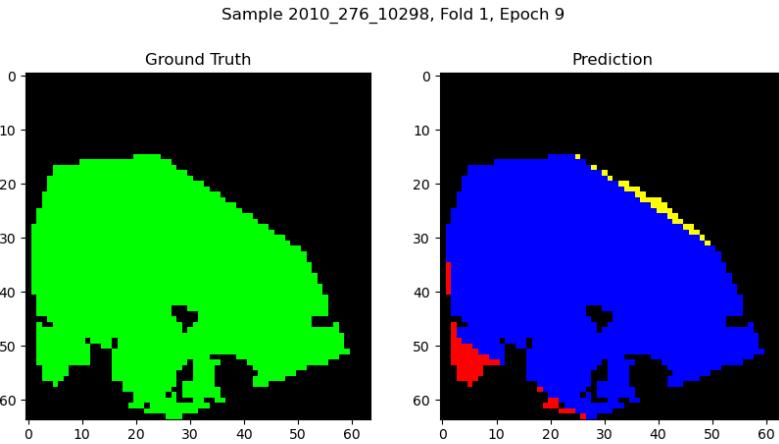


Figure 1.2: An example of ground truth(L) and prediction output(R)

Table 1.1: Fire Sample Metadata

NFIREID	YEAR	AGENCY	FIRECAUS	Total Burned Are (Hectares)	Burning Days	Estimated Start Date	Estimated End Date	Geometry Shape
205	2010	AB	Lightning	26961.4665	78	6/15/2010	9/1/2010	multi-polygon

Figure 1.2 shows an illustration of the content of output from prediction models. The input sample comes from a day of a wildfire that happened in Alberta in 2010. A brief metadata is provided through table 1.1.

The figure shows the ground truth shape of the next-day burned area on the left and the prediction for it on the right. The ground truth is historical data from the dataset and the prediction is from the prediction mode. This thesis has covered several deep learning models but they have the same output configurations. The red and yellow in prediction output stand for false negative(burned pixels not reported by model) and false positive(reported pixels which did not burn). These two kinds of false predictions were not acquired and classified from the model's output. They were computed through extra scripts for visual analysis.

# Chapter 2

## Background Knowledge

This chapter aims to provide background knowledge related to the studies of this thesis. By explaining relevant topics, it is expected to help the readers understand the contents of the following chapters regarding my work.

### 2.1 Wildfire Background Information

Wildland fires, which are also known as wildfires or forest fires, are uncontrolled fires occurring in areas covered by various vegetation. Much area in Canada is covered with forests and plant life, making these regions highly susceptible to wildfires. The unpredictable nature of the causes of wildfires makes it challenging to predict the precise occurrence of these fires. Reliable predictions of wildfire growth are crucial as they could lead to an efficient allocation of resources during firefighting efforts.

The intensity and spread of fires are influenced by three primary factors, known as the “fire behaviour triangle”: Fuel, Weather, and Topography. Canada has employed a Canadian Forest Fire Danger Rating System (CFFDRS) [27] as the national fire risk system, which includes two critical tools for predicting and managing wildfires: the Fire Behavior Prediction (FBP) Fuel System and the Fire Weather Index (FWI).

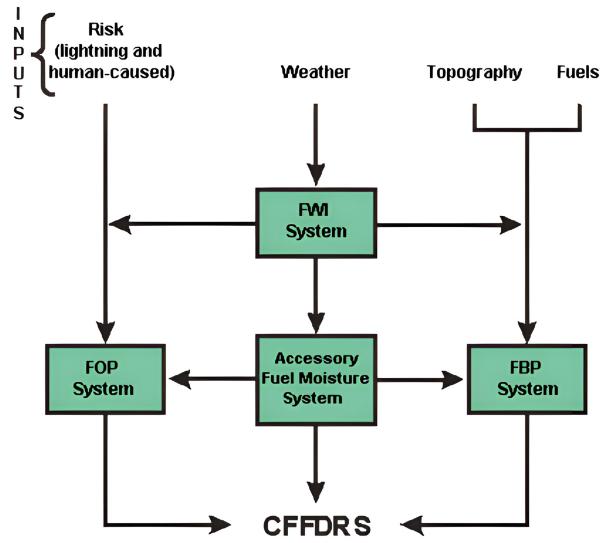


Figure 2.1: CFFDRS [27]

### 2.1.1 Fuel Type in FBP System

The Fire Behavior Prediction (FBP) Fuel System is one of the key components of the CFFDRS. It offers detailed predictions of wildfire behaviour based on specific fuel types and conditions. The FBP system categorizes fuels into various types based on their physical characteristics. Each class is unique regarding forest floors and organic layers, fuels at the surface and ladder, also the stand structure and composition.

### 2.1.2 Fire Weather Index System

The Fire Weather Index (FWI) [28] is another core component of the CFFDRS. It assesses the risk and potential behaviour of wildfires based on daily weather observations. The FWI system integrates several sub-indices to provide a comprehensive assessment of fire danger. FWI is the final output in the system and is a numerical wildfire threat indicator, used as a general index of fire danger throughout the forested areas of Canada.

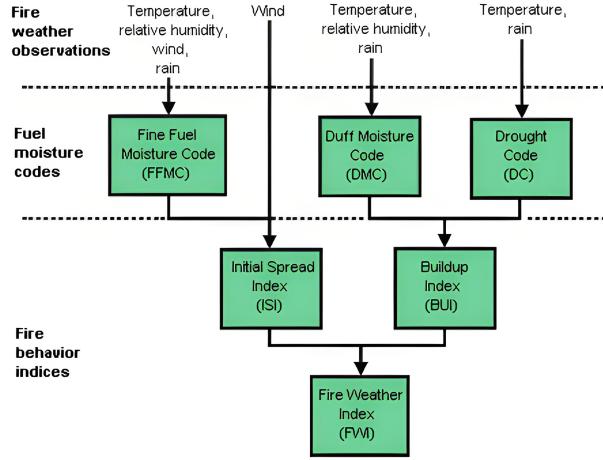


Figure 2.2: FWI System [28]

## 2.2 Machine Learning Background Information

### 2.2.1 Machine Learning and Deep Learning

Machine Learning is a subset of the Artificial Intelligence (AI) field of study that enables computers to learn the pattern of data and perform tasks such as classification, prediction, decision-making and much more. Deep learning is a subset of machine learning that mainly focuses on neural networks. Deep learning has shown its great capabilities for many tasks and its greater potential to outperform traditional tools in some fields.

### 2.2.2 Convolutions and Convolution Neural Networks

#### Convolutions and CNN

The traditional convolution is a mathematical operation that takes two functions and produces a third function. The convolution of two continuous functions  $f(t)$  and  $g(t)$  is defined as:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau \quad (2.1)$$

where:

- $(f * g)(t)$  is the convolution of  $f(t)$  and  $g(t)$ .
- $\tau$  is a dummy variable of integration.

In the context of deep learning, convolution involves applying a kernel (filter) to an input and assigning a weighted sum to each pixel from its local area. This concept is widely used in Convolutional Neural Networks (CNN). The convolution in deep learning focuses on the concept of weighted sum and ignored some properties in the original meth definition, such as the flip of the second function.

$$Y(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i \cdot s_i + m - p_i, j \cdot s_j + n - p_j) \cdot K(m, n) \quad (2.2)$$

The equation 2.2 is a 2D convolution process in machine learning operation. The convolution result of each pixel in  $Y$  is a sum of element-wise product of  $X$  and  $K$ , where:

- $Y$  is the output feature map.
- $X$  is the input feature map.
- $K$  is the convolution kernel (filter).
- $M$  and  $N$  are the dimensions of the kernel.
- $s_i$  and  $s_j$  are the strides along the height and width, respectively.
- $p_i$  and  $p_j$  are the padding sizes along the height and width, respectively.

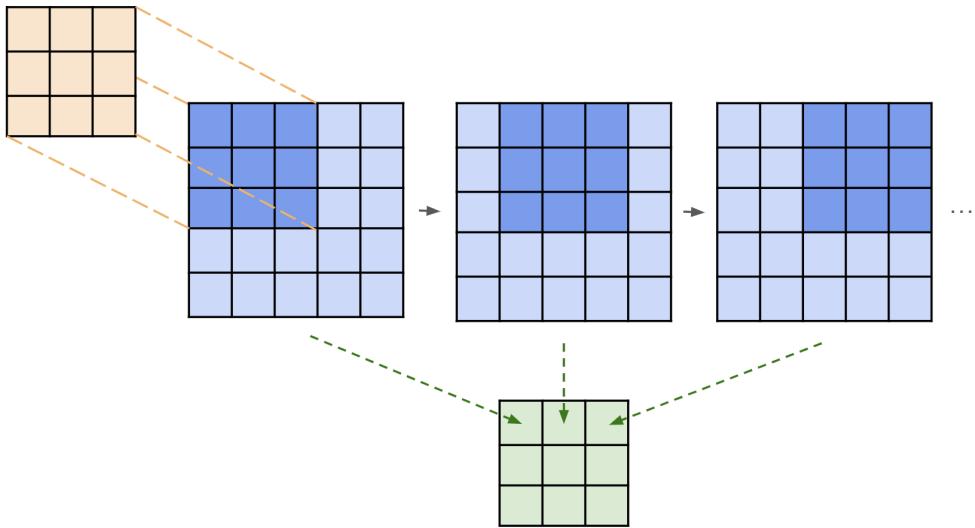


Figure 2.3: 2D Convolution Demonstration (kernel=3 stride=1)

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed to process and analyze visual data. They utilize convolutional layers to learn spatial hierarchies of features from input images. By applying various filters across the input, CNNs can detect patterns such as edges, textures, and more complex structures at different layers. These learned features are crucial for tasks like image classification, object detection, and segmentation, making CNNs highly effective in computer vision applications.

### **Deconvolution and Transposed Convolution**

Deconvolution is the reverse process of convolution and it requires the arguments in convolutional kernels. Transposed Convolution is a computing process that follows the procedure of deconvolution but uses independent kernels. In this thesis paper, both terms “deconvolution” and “transposed convolution” refer to the definition of transposed convolution.

## Atrous Convolution

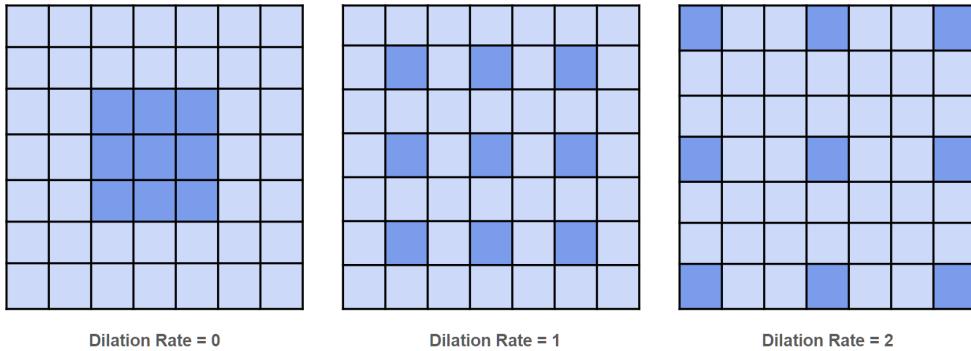


Figure 2.4: Atrous Convolution Demonstration on kernel size = 3

Atrous convolution (dilated convolution) is a convolution method that increases the field of view of the model without greatly increasing the computational cost. The atrous kernel is sparse to cover a large scale of data without losing details. A common implementation of atrous convolution is Atrous Spatial Pyramid Pooling (ASPP) proposed with the model Deeplabv3 [9]. ASPP applies multiple atrous convolution layers in parallel, which enables sampling at different scales and the capability of capturing better long-distance information.

### 2.2.3 Attention Mechanisms

Attention mechanisms are an advanced component of deep learning models that allows models to focus on specific parts of data by observing relationships among components of the data. This is particularly useful in many tasks where the important information is not uniformly distributed across all the data. Attention mechanisms enable models to dynamically weigh the importance of different parts of the input data. This is achieved by calculating attention scores that reflect how strongly each part of the data should influence the output.

#### Self-Attention and Cross Attention

Self-attention is a mechanism by which a model weighs the elements in the output from the last layer by their relevance to each other. For example, in a natural language processing

task, the model analyses each word token and calculates the relevance to the other word tokens.

Cross-attention is similar to self-attention but the relevance score is not necessarily computed from only the output of the last layer in the model but also from other layers. The Transformer model uses self-attention in the encoder and decoder part and uses cross-attention in its decoder part.

### Scaled Dot-product Attention

Scaled Dot-Product Attention

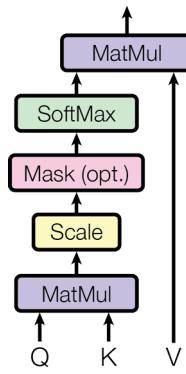


Figure 2.5: Scaled Dot-product Attention [36]

Scaled dot-product attention is an implementation of the attention mechanism in the Transformer model [36]. The input of the module goes through three different transformation matrices and becomes three tensors Query, Key and Value ( $Q, K, V$ ). The  $Q$  and  $K$  can be regarded as the representation of the “question” side and “answer” side and thus the dot matrix production gives a relation score (compatibility in the author’s word) between the two.

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.3)$$

The scaled dot-product attention mechanism calculates attention scores by taking the dot product of the  $Q$  and  $K$  matrices, dividing by the square root of the dimension of the Key vectors ( $d_k$ ) to mitigate the effect of large dot product values, and applying a

softmax function to obtain the attention weights. These weights are then used to compute a weighted sum of the V matrix, producing the final output.

The scaling using the factor  $d_k$  is to reduce the vanishing gradient problem for terms with high compatibility in dot products with large attribute dimensions. The reason for the problem is many terms are multiplied during the matrix product, resulting in the summation of them approaching the hard limit of the activation functions.

## Mult-head Attention

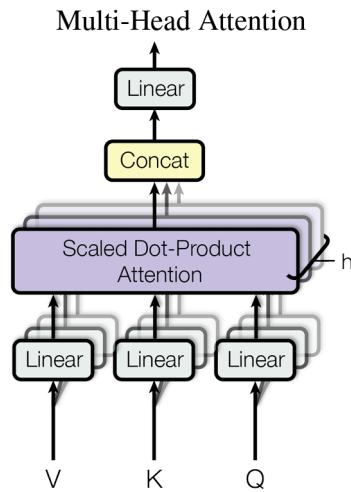


Figure 2.6: Multi-head Attention [36]

The Multi-head attention mechanism projects the input to several different small attribute spaces (similar to channels in CNN) called heads instead of one big attribute space [36]. The size of the attribute space in each head  $d_k$ ,  $d_v$  is computed by dividing the original model's attribute space by head numbers. The computation cost of multi-head attention is similar to single-head attention with full dimensionality but adding diverse projections.

## 2.2.4 U-Net, U-Net++, Swin-UNet

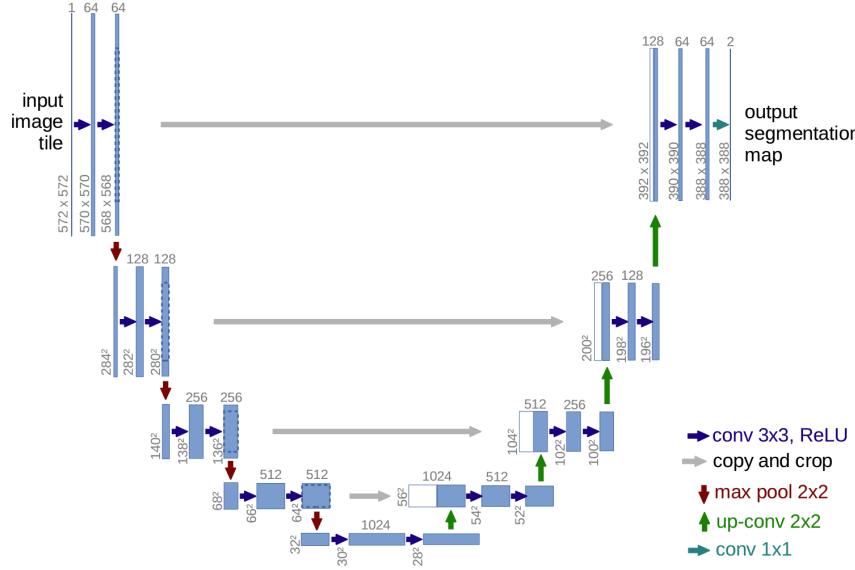


Figure 2.7: Original U-Net [31]

U-Net is a specific type of Convolutional Neural Network structure that was initially developed for biomedical image segmentation tasks [31]. The network consists of a downsampling path (encoder) and an upsampling path (decoder) with connections between the encoding block and decoding block on the same level.

U-Net++ (U-Net 3+/U-Net 3p) is a U-Net-based architecture based on another variant of U-Net called U-Net++ [19]. U-Net 3+ increases the number of connections to each block, with attention to improving the ability to capture complex patterns across different layers in the model.

Swin-UNet is a variant that integrates features of the Swin Transformer into the U-Net architecture [7]. The Swin Transformer is a Transformer architecture that is designed to solve some problems that the Vision Transformer might encounter, for example, the presence of large variations in visual entities, or the difference in resolution between text token and image pixels. The Swin Transformer's core idea is performing self-attention within different patch sizes along the depth and adopting sliding windows to solve the lack of connection among patches. Swin-UNet inherits those features and puts them into the U-Net architecture.

## 2.3 Significance Testing

The T-test is a statistical hypothesis test used to determine if there is a significant difference between the means of two groups. It is widely used in various fields, including biomedical research, social sciences, and machine learning, to compare the performance of different models or treatments. The T-test can be classified into several types, but the most commonly used ones are the Independent Samples T-test, Paired Samples T-test, and One-Sample T-test. This section will focus on the Paired Samples T-test, which is particularly relevant when comparing the performance of two models on the same dataset.

### 2.3.1 Significance Testing and the P-Value

Significance testing is a statistical technique used to determine whether the observed effect in data is genuine or if it could have occurred by chance. A key concept in significance testing is the p-value. The p-value is the probability of obtaining results at least as extreme as the observed results, assuming that the null hypothesis ( $H_0$ ) is true. The null hypothesis typically states that there is no effect or no difference between groups.

A low p-value (typically smaller than or equal to 0.05) indicates that the observed effect is unlikely to have occurred by chance, leading to the rejection of the null hypothesis. Conversely, a high p-value suggests that the observed effect could plausibly occur under the null hypothesis, and thus, the null hypothesis is not rejected. In this research, 0.05 was used as the common threshold for the judgement of significant superiority between two models regarding a performance metric.

### 2.3.2 Paired Samples T-Test

The Paired Samples T-test, also known as the dependent T-test, is used when the samples are dependent, meaning that each pair of observations is related in some way. In the context of machine learning, this is often the case when comparing two models on the same test data, where each observation (e.g., prediction error) from one model is paired with a corresponding observation from the other model.

The Paired Samples T-test assesses whether the mean difference between these paired observations is significantly different from zero. The null hypothesis ( $H_0$ ) of the test states that there is no difference between the means of the paired observations, while the alternative hypothesis ( $H_1$ ) states that there is a significant difference.

The equation of the T-test is shown below:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \quad (2.4)$$

where:

- $\bar{d}$  is the mean of the differences between paired observations.
- $s_d$  is the standard deviation of the differences between paired observations.
- $n$  is the number of pairs.

The T-test is particularly useful because it provides a method to assess whether the observed differences between two groups are statistically significant or likely to have occurred by chance. In machine learning, it is crucial to determine if one model significantly outperforms another. The T-test offers a rigorous statistical framework for making such comparisons, ensuring that any reported improvements are not due to random variability in the data.

# Chapter 3

## Literature Review for Wildfire Prediction Tools

### 3.1 Traditional Tools

This section introduces traditional tools that have been widely used in wildfire prediction tasks. These simulators generally incorporate spatial (multi-dimensional) and environmental (scalar, categorical label) data to model wildfire behaviour and spread. Usually, their behaviour highly relies on the quality and precision of the historical data.

#### 3.1.1 Rothermel Fire Spread Equations

The Rothermel fire spread model, developed by Richard Rothermel in the 1970s [33], provides a mathematical framework for estimation of the rate of fire spread in various conditions. While Richard Rothermel developed the original model in 1972, and Frank A. Albini made notable modifications in 1976 [2], Patricia L. Andrews has furthered this work through continued research and updates [3]. Her contributions have likely involved integrating new scientific insights, improving computational methods, and adapting the model to contemporary fire management needs. This ongoing work ensures that the Rothermel model remains relevant and effective in predicting wildfire behaviour under modern conditions.

The rate of fire spread  $R$  can be calculated using the following basic equation:

$$R = \frac{I_R \xi (1 + \phi_w + \phi_s)}{\rho_b \epsilon Q_i} \quad (3.1)$$

where:

- $R$  is the rate of spread (m/min).
- $I_R$  is the reaction intensity (kW/m<sup>2</sup>).
- $\xi$  is the propagating flux ratio.
- $\phi_w$  is the wind coefficient.
- $\phi_s$  is the slope coefficient.
- $\rho_b$  is the bulk density of the fuel (kg/m<sup>3</sup>).
- $\epsilon$  is the effective heating number.
- $Q_i$  is the heat of pre-ignition (kJ/kg).

While the Rothermel fire spread model is widely used for its comprehensive approach, it has several limitations. One major limitation is the assumption of uniformly distributed fuel, which does not account for complex fuel arrangements in real-world scenarios. Additionally, the model does not consider changes in fuel moisture content over time or the effects of varying weather conditions beyond wind and slope. These simplifications can lead to inaccuracies in predicting fire behaviour in heterogeneous landscapes and dynamic environmental conditions.

### 3.1.2 Fire Behavior Prediction (FBP) System

The Canadian Fire Behavior Prediction System is a framework for Canadian wildland fire behaviour prediction developed by the Canadian Forest Service in the 1980s [17]. The system was designed to provide a systematic approach to predict wildfire behaviours under various environmental conditions and help decision-making concerning fire control and other tactics. The core of the FBP system is an empirical fire spread model that was developed by analyzing fire behaviour in different Canadian forest fuels.

### **3.1.3 FARSITE**

FARSITE was developed in the United States [11] and is a fire modelling system predicting the spread and behaviour of wildland fires using topography, fuel types and weather conditions. The model uses Rothermel's fire spread equations for calculating fire growth and behaviour spatially and temporally. The model outputs include fire spread rate, flame length and the shape of the fire perimeter over time, which aid firefighting strategies and management decisions. The model possesses the ability to simulate complex fire behaviours, but its performance can be limited under extreme conditions. For example, K. Zigner et al. believed FARSITE has inherent limitations on spotting algorithms when carrying out research with extreme downslope winds.

### **3.1.4 Prometheus, Burn-P3 and Burn-P3+**

Prometheus is a Canadian wildfire behaviour prediction system [35] designed for the conditions of Canadian forests. The system has a similar input as FARSITE. Prometheus is based on the Canadian Fire Behavior Prediction (FBP) system but it integrates deeply with the GIS system which enables it to provide detailed simulations of fire growth with spatial and temporal changes.

Burn-P3 [29] (Probabilistic Prometheus Project) integrates a probabilistic component into wildfire prediction based on Prometheus. The output of Burn-P3 includes probabilistic fire growth maps and the uncertainties of the simulation are achieved by using Monte Carlo simulation techniques. This ability enables the system to provide potential wildfire behaviour under different scenarios and support strategic planning and decision-making in firefighting.

Burn-P3+ is an open-source package on SyncroSim which aims to update and replace Burn-P3. Burn-P3+ has the scalability to bigger landscapes and scenarios. By introducing good compatibility on crossing platforms and enhanced user interfaces with more detailed analytics, the tool is more accessible and effective for management and decision-making. Besides Prometheus, Burn-P3+ provides Cell2Fire as another optional model for use.

## **3.2 Machine Learning Literature Review**

This section introduces some past research carried out in the wildland fire field. Researchers have tried various models for fire growth prediction tasks from classic machine

learning models [25], deep learning models [37] to reinforcement learning models [14], and have brought improvement in timeliness, prediction accuracy, and adaptability to complex terrains, and changing conditions. However, the section focuses on introducing experiments involved with deep learning models, especially CNNs. Due to the similarity of tasks, fire occurrence predictions (FOP) are also included in this section.

### 3.2.1 Machine Learning in Fire Spread Predictions

J. Hodges and B. Lattimer (2019)[18] stated that the computational cost of past models in predicting wildland fire spread across various landscapes is huge and this deeply affects the efficiency of mitigation strategies development using simulations. They proposed a Deep Convolutional Inverse Graphics Network (DCIGN) that gives an increment of computation speed by at least two orders of magnitude. The experimental data was generated by simulation tools and contained both homogeneous landscape samples and heterogeneous landscape samples with some weather input randomly generated. The former is generated by the Rothermel fire spread equations 3.1.1 and the latter is generated using FARSITE 3.1.3. They ran the DCIGN over 10 thousand times to predict fire growth after 6 to 24 hours on samples and recorded model performances in terms of precision, sensitivity, and F-measure of 0.97, 0.92, and 0.93.

Since the current state-of-the-art wildfire spread models still rely on mathematical growth predictions and physics-based models, which are difficult and computationally expensive to run, Radke et al (2019) [30] presented a system named FireCast which could efficiently combine AI and GIS data. The model is trained on a large dataset augmented from real historical data and has been proven to outperform FARSITE under both wet and dry conditions on the test set regarding accuracy, recall and F-score. The author stated that FireCast outperformed modern software and is computationally inexpensive.

Singh et al.(2023) [34] apply several machine-learning techniques on next-day wildfire growth predictions and they highlighted the potential of machine-learning models in such tasks. The dataset includes satellite images, weather, and geographical conditions aggregated across the United States from 2012 to 2020. In their report, all methods achieved a relatively low RMSE score while the decision tree performed the best among the three models. However, the paper does not include either highly detailed experimental results or a clear explanation of the ANN structure used for the dataset.

### **3.2.2 Machine Learning in Fire Occurrence Predictions**

Prapas et al. (2021) [24] explore the application of deep learning techniques to predict daily wildfire danger. A data cube containing several features across spatial and temporal coordinates is made from historical data. The authors employed a suite of deep learning models including Random Forest, Convolutional Neural Networks, Long Short-Term Memory(LSTM), and Convolutional LSTM (ConvLSTM) that take different dimensions of the data cube and predict a next-day burned map. In the experiment result, ConvLSTM achieved the highest AUROC of 0.926, and LSTM reached the highest F1 of 0.751.

The need for novel wildfire warning and management tool systems is emphasized by F. Huot et al. (2020) [20] because of the growth of fire scales and the duration of fire season. The authors aggregated historical data from multiple sources into 10 feature channels and the prediction task, or “fire likelihood estimation” in the authors’ words, is defined as three different tasks: daily image segmentation, aggregated image segmentation, and sequential aggregated image segmentation. Four different models were used in all three tasks: encoder, U-Net, encoder LSTM and U-net LSTM. Among all the performances, the autoencoder achieved the best result when performing daily segmentation and gave AUC of 0.83.

### **3.2.3 A Review of Machine Learning Applications in Wildfire Science and Management**

Jain et al. [22] provide a comprehensive scoping review that enhances the awareness and understanding of machine learning (ML) methods among wildfire researchers and managers. Their paper illustrates the diverse challenges in wildfire science that can be addressed using ML data science techniques, offering a thorough examination of ML applications within the field.

The review begins by outlining some widely used ML methods in wildfire science, including random forests, artificial neural networks, decision trees, support vector machines, and genetic algorithms. The authors categorized ML applications into six primary problem domains: fuels characterization, fire detection and mapping, fire weather and climate change, fire occurrence, susceptibility and risk, fire behaviour prediction, fire effects, and fire management. This categorization showcases the extensive range of wildfire-related issues that can be tackled using ML approaches.

Jain et al. discussed the strengths of various ML approaches, such as their ability to handle large datasets and complex relationships, making them particularly suitable for

wildfire science. However, they also highlight limitations related to data size, computational requirements, generalizability, and interpretability. Despite these challenges, the authors identified significant opportunities for future research, particularly in the application of advanced ML methods like deep learning and agent-based learning. They emphasized the importance of high-quality, freely available wildfire data for developing effective ML models.

In conclusion, the paper calls for increased collaboration between wildfire researchers and ML practitioners to ensure realistic and effective modelling of fire processes. The authors stressed that the wildfire research community must actively contribute to creating relevant datasets and developing ML-based tools for wildfire science and management. By providing a detailed review of ML applications, this paper aims to improve the utilization of these methods, ultimately contributing to more effective wildfire prediction, prevention, and management strategies.

# Chapter 4

## Data Acquisition and Processing

### 4.1 National Wildfire Dataset

This thesis studies most of the recorded wildfires that happened in Canada's national boundaries, ranging from 1994 to 2021 and across all the provinces. With reference to the architecture of current fire behaviour prediction tools and the consideration of data availability, the data set in this thesis uses 2D and 1D features coming from multiple different sources. The data was collected in different file formats, geographical coordinate systems, resolutions and time ranges, which requires appropriate pre-processing. On the one hand, it is technically not easy to acquire data at both high spatial and temporal resolution, on the other hand, some features like burned area shapes are not feasible for real-time data collection. It needs time to compute and construct from other "raw data". Additionally, some features for fire events are more vulnerable to natural interference. For example, when the geographical area of the occurrence of a fire event is cloudy, the heat source would not be fully observed by satellites, resulting in noise in data.

Considering the above reasons, the dataset used in this thesis focuses on features that are feasible to acquire in real-time or short time range, and without heavy computation. Ideally, well-processed data could help the training process and potentially provide better metrics, but also reduce the practical value of the prediction model in terms of model robustness to noise and other interference.

The boundary of Canadian provinces and territories references the shape file provided by ArcGIS hub [5], to filter data sources covering global areas. All the fire events studied in this thesis are within the total boundary shape.



Figure 4.1: Canadian provincial and territorial boundaries

The dataset consists of six 2D features and three 1D features extracted from five datasets. The daily burned area and cumulative daily burned area are computed from the National Burned Area Composite (NBAC) database [13] from the CWFIS Datamart. The fuel map comes from a dataset that is designed for Canadian Wildland Fire Information System (CWFIS) using multiple sources [4][1][12][10]. The daily Fire Weather Index (FWI) comes from the Global Fire Weather Index Dataset which is by the works of Megan M. et al. [26] and Piyush J. et al. [21]. The elevation map is acquired from the dataset published by Jeffrey N, Laurence H [23]. This dataset supports the paper presented by Laurence H. et al. regarding a high-resolution global elevation map with buildings and forests removed by machine learning techniques [16]. The three 1D features including month, fire causes and agency are from NBAC.

The size selection of the sight window (the edge length measured in meters) of each sample is determined by the statistic of burned area in the raw dataset. The distribution of the burned areas from 1994 to 2021 is shown in the kernel density estimate(KDE) chart. The KDE is a good method for visualizing the distribution of some features in a dataset. Figure 4.2 shows the complete statistic of burned area sizes in the dataset and it is obvious from the curve that some extreme values exist around the main peek. When the sight window is large enough to cover all the burned areas, it also means for the majority of the fire events, the tensor would contain a great portion of blanks. Instead, 95 percent of the densest areas in the distribution were used for the decision of sight window. The “95 percent interval” shown in Figure 4.3 is [552.4465, 13673.0519]. To cover those areas, a  $64 \times 64$  grid with a pixel resolution of 200m is established. The max capacity for the sight

window is 16384 hectares.

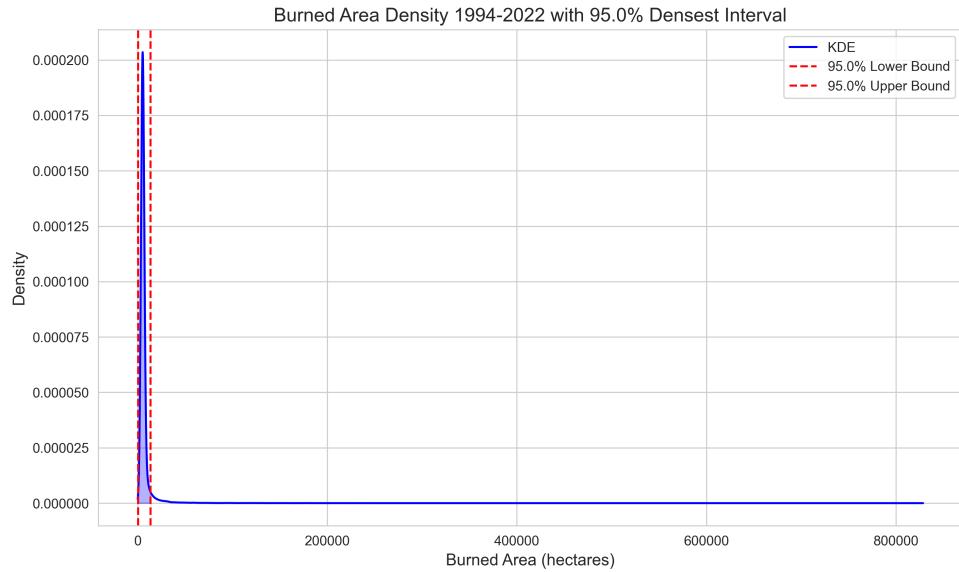


Figure 4.2: KDE of burned areas from 1994 to 2021 with 95 percent mark

A sequence of procedures for data pre-processing has been performed to integrate and process every feature into the final dataset. To acquire that, a fire event dataset is constructed. Firstly, separate fire events are generated from the burned area polygon and put in the center of a bounding box. Then I mapped all the hotspots matching spatial and temporal characteristics of fire events into the bounding box and rasterized them. The rest of the features were transformed into the coordinate reference system (CRS), NAD83/Canada Atlas Lambert (geographical/projection), which is also the CRS of the burned area and hotspot dataset. After that, they are filtered according to the burning days in each fire event and masked by the bounding box. Finally, resampling is performed to align the data with  $64 \times 64$  grid resolution. The procedure works well when the data is assumed to be none-sparse, integrated and without missing/noise data. I addressed a few challenges when some features did not meet this assumption, and solved them using the minimum change I had to make (which are described in detail in each section). Once the fire events are built, samples are extracted from the fire events.

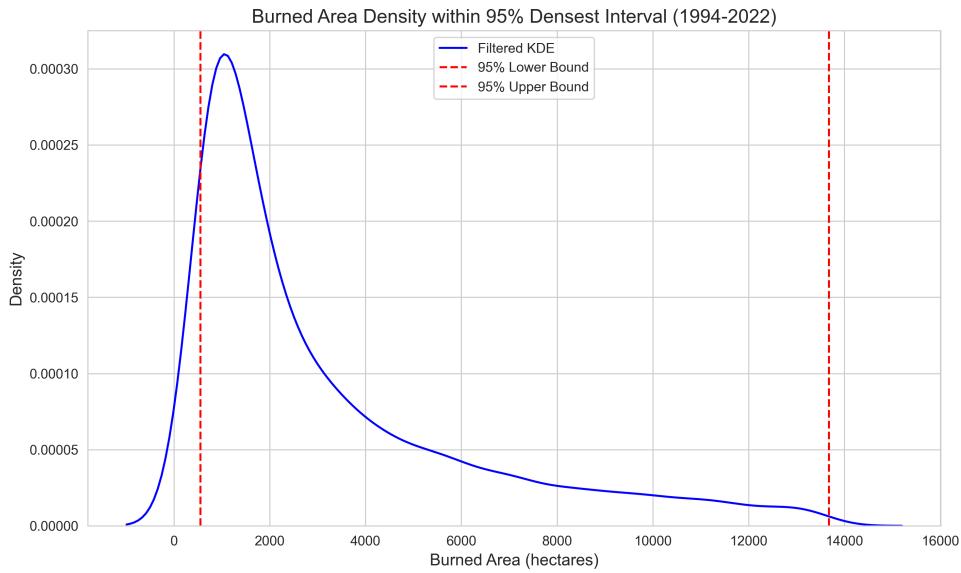


Figure 4.3: KDE of 95 percent burned areas from 1994 to 2021

## 4.2 Burned Area and Hotspot

The burned area raw dataset from NBAC contains a large shapefile of all the fire events that happened from 1994 to 2021 in the record. Each line of record stores the final burned shape of a fire in polygon/multi-polygon. This feature is not acquired by direct observation but computed from multiple data sources. Polygon-form burned area could be an ideal representation for daily fire growth but unfortunately, this is not available in daily resolution. The scale Range is from 1:5,000 Minimum to 1:150,000,000.

The burned area is useful in providing statistical information in terms of fire scales, month/year of occurrences, and provinces of occurrences. Besides that, it could be used as a limiter for the estimated daily burned area, which will be mentioned later. The limiter makes sure every generated 'ground truth' does not exceed the real fire zone.

The hotspot raw dataset from Fire M3 contains a list of shapefiles of yearly records of hotspots. The dataset aggregated multiple sources: Advanced Very High Resolution Radiometer (AVHRR) imagery, Moderate Resolution Imaging Spectroradiometer (MODIS) imagery, and Visible Infrared Imaging Radiometer Suite (VIIRS) imagery. Their resolution ranges from 30m to 1km. The satellite sensors record the intensity of electromagnetic radiation from Earth in various spectral wavelengths or channels and capture specific infrared wavelengths which are commonly emitted by fires.

The hotspot data are available in daily resolution and were used for generating estimated burned areas in the thesis. Based on the daily estimation, I also built an estimation of the cumulative (aggregated) burned area. However, the dataset does have some limitations. Firstly, there is a small chance that false alarms (from land or the edge of the winds) survive the detection algorithm. Then, the heat emission of fires under thick clouds or smoke cannot be detected by the satellites. Additionally, the spatial resolution between some samples and the target resolution in the thesis introduces some errors. For example, a hotspot recorded in  $1\text{km} \times 1\text{km}$  resolution could represent a small fire that is only a quarter of the pixel size. The super-resolution problem is not considered as a task in this thesis.

The NBAC and M3 dataset still gets updated and it could be different from the version of the dataset used in the thesis.

The table below shows the burned area distribution in hectares each year. There are 16868 fire events on record, the biggest of them is over 800 thousand hectares (8 thousand square kilometres) and the smallest of them is less than 1 hectare. The occurrences of fire events are geographically classified into Provinces, Territories, or Parks Canada (PC) and the distribution is not balanced. Most of the wildfires happened in the northwest of Canada.

Table 4.1: Burned Areas in Hectares (Canada, 1994-2021)

Year	Min	Max	Average	Year	Min	Max	Average
1994	6.761	360195.731	16348.354	2008	0.023	114841.654	2909.756
1995	12.284	828531.701	12856.509	2009	0.195	58672.605	1321.774
1996	8.375	72169.247	3679.471	2010	0.202	425654.060	3895.844
1997	5.755	57202.368	4037.882	2011	0.022	462658.278	2600.072
1998	0.019	151624.095	6376.614	2012	0.006	156827.090	1700.867
1999	10.871	173179.680	4677.231	2013	0.001	501737.419	4840.114
2000	2.619	40826.968	2549.448	2014	0.001	596459.121	4984.619
2001	1.007	109751.662	2685.210	2015	0.002	223766.963	2384.260
2002	10.444	237805.075	7440.566	2016	0.044	437720.153	1936.674
2003	0.100	70085.166	3663.625	2017	0.003	410381.645	2780.829
2004	0.222	305192.257	4280.320	2018	0.006	101856.894	1640.976
2005	0.009	69061.903	3003.048	2019	0.127	309709.627	4736.121
2006	0.032	184514.981	2830.211	2020	0.070	51460.041	514.406
2007	0.150	96801.267	3105.507	2021	0.078	381328.856	3996.310
				Total	0.001	828531.701	4206.308

Table 4.2: List of Abbreviations

Abbreviation	Description
AB	Alberta
BC	British Columbia
MB	Manitoba
NB	New Brunswick
NL	Newfoundland and Labrador
NS	Nova Scotia
NT	Northwest Territories
NU	Nunavut
ON	Ontario
PC	Parks Canada
QC	Quebec
SK	Saskatchewan
YT	Yukon Territory

Table 4.3: Fire Event Counts Per Province, Territory and National Park Canada (1994-2021)

Year	Total	NT	MB	YT	SK	QC	ON	BC	AB	NL	PC	NU	NB	NS
1994	296	68	60	45	37	31	30	12	4	1	8	0	0	0
1995	453	34	80	19	56	124	114	12	7	1	6	0	0	0
1996	458	51	43	12	9	193	129	7	0	11	3	0	0	0
1997	154	11	18	3	8	93	18	2	0	1	0	0	0	0
1998	632	97	41	32	206	64	129	28	24	2	9	0	0	0
1999	267	13	60	15	18	53	93	5	1	1	8	0	0	0
2000	366	61	52	32	20	72	95	15	14	1	4	0	0	0
2001	456	36	61	27	44	97	149	13	20	1	8	0	0	0
2002	517	62	82	42	54	72	132	23	44	2	4	0	0	0
2003	601	60	87	53	52	131	157	30	22	1	8	0	0	0
2004	414	60	53	26	45	88	95	31	9	4	3	0	0	0
2005	568	49	83	48	65	136	116	32	32	2	5	0	0	0
2006	375	41	40	38	45	65	99	26	13	1	7	0	0	0
2007	520	52	83	37	48	126	112	27	28	2	5	0	0	0
2008	535	44	79	42	57	122	126	32	30	1	2	0	0	0
2009	492	55	55	49	46	109	126	32	16	2	2	0	0	0
2010	453	44	61	39	49	115	101	25	14	1	3	0	0	1
2011	466	54	61	42	50	110	108	29	9	2	1	0	0	0
2012	458	54	62	50	47	97	106	31	6	2	3	0	0	0
2013	521	70	84	51	65	116	92	28	8	3	4	0	0	0
2014	580	48	106	48	57	126	123	27	42	1	2	0	0	0
2015	471	52	86	42	47	108	87	35	11	2	1	0	0	0
2016	416	54	62	43	53	86	88	20	7	2	1	0	0	0
2017	551	68	82	52	49	113	125	29	21	2	10	0	0	0
2018	1121	19	391	19	70	185	52	324	47	0	13	0	1	0
2019	335	45	46	36	39	74	20	19	44	0	11	1	0	0
2020	409	15	143	4	12	178	36	10	4	2	3	0	0	2
2021	974	92	148	39	220	134	93	199	39	0	9	1	0	0
Total	16868	2066	3893	826	1786	2254	1559	1703	2033	107	345	13	174	109

Table 4.4: Hotspot Yearly Counts (1994-2021)

Year	Yearly Total	Year	Yearly Total
1994	47,012	2008	34,151
1995	68,918	2009	35,697
1996	16,886	2010	54,230
1997	5,154	2011	45,856
1998	38,265	2012	86,075
1999	19,334	2013	174,845
2000	9,613	2014	311,128
2001	5,082	2015	539,233
2002	31,165	2016	298,913
2003	82,181	2017	1,039,334
2004	99,157	2018	813,089
2005	66,482	2019	545,337
2006	48,973	2020	987,400
2007	60,553	2021	1,842,085
		Total	7,406,148

#### 4.2.1 Sight Window, Mask and Resample

The burned area and hotspot data are in NAD83 / Canada Atlas Lambert format, which are the geographical and projection coordinate systems desired. The main challenge here is to map burned areas and hotspots based on spatial and temporal aspects to provide the base features for each fire event.

Functions in the Geopandas Python library were used to detect if a geometric point is within a geometric polygon/multi-polygon in a GeoDataFrame. In the M3 hotspot dataset, each hotspot is stored as an individual record. If a hotspot was detected within the burning days of a fire event, its geometric point was then compared to the burned area of that fire event, and only matching hotspots were saved with each fire event.

When calculating the burning days, there are four related attributes: SDATE, EDATE, AFSDATE and AFEDATE. SDATE and EDATE are the dates of the first/last hotspot detected. AFSDATE and AFEDATE are fire start/end times reported by the agency. Both sources contain null values representing the missing data during collection. To address the problem, I took the min and max values from all four attributes, where the unnecessary two attributes (Blue terms in 4.1) were used in case the first two attributes were both null.

$$\text{min\_burning\_days} = \min(\text{SDATE}, \text{AFSDATE}, \text{EDATE}, \text{AFEDATE}) \quad (4.1\text{a})$$

$$\text{max\_burning\_days} = \min(\text{EDATE}, \text{AFEDATE}, \text{SDATE}, \text{AFSDATE}) \quad (4.1\text{b})$$

The selection of the sight window is based on the geographical centroid of the geometric polygon of burned areas. For each geographical centroid, I defined a bounding box by minimum and maximum values on the X and Y axes and created a transform for the sight window. A transform is an affine transformation matrix as shown in the figure. The first row of the matrix contains the transformation of the X-axis and the second row contains which of the Y-axis. A and E are the pixel width and height based on the scale of the area. B and D are the rotation and are considered as 0 in this situation. C and F are the geographical coordinates of the top-left corner, which is the position (0, 0) of the grid. Then, the geometric polygons and hotspots were rasterized based on transforms. Pixels inside the shapes were assigned to 1 and those that were not were filled with 0.

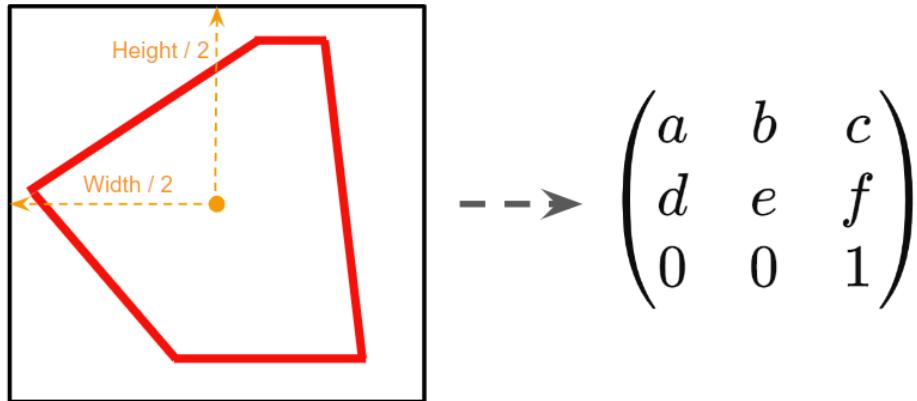


Figure 4.4: The Bounding Box and affine transformation Matrix

$$x' = a \cdot x + b \cdot y + c \quad (4.2\text{a})$$

$$y' = d \cdot x + e \cdot y + f \quad (4.2\text{b})$$

#### 4.2.2 Estimated Burning Areas

To generate the daily burning area shape, I computed the convex hull from daily hotspot data in the eight windows. A convex hull of a shape is the smallest convex set that contains it. For each day in a fire event, contours were computed from discrete hotspots and transformed into a rasterized convex hull filled with 1. The method was expected to generate close enough approximate shapes of the daily burning area when the raw data was collected with limitations of resolution.

The final burned area shapes from the NBAC dataset were used as a ground truth filter on the estimation output. This keeps the estimation in a reasonable range and helps correct false burning at non-fuel-type locations.

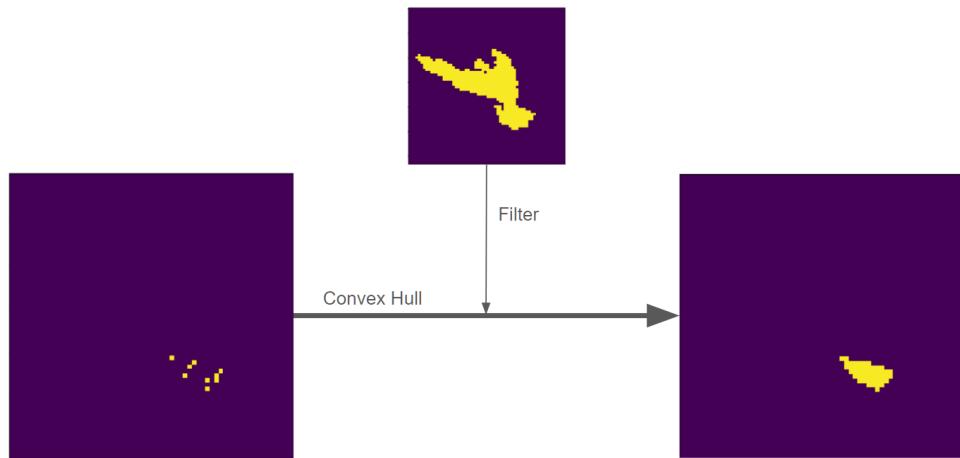


Figure 4.5: Estimated Burning Areas

#### 4.2.3 Cumulative Burning Areas

The hotspot dataset reflects the burning location during the fire event but does not provide any historical information before that time point. To assist the learning process, I added cumulative burned area shapes besides the daily burning area shapes. The extra feature provides more time-sequence information and helps the model prevent predicting burned pixels once again. As shown in figure 4.6, the images in the first row show the fire growth using estimated daily burning areas and the images in the second row show the same event using cumulative estimated burning areas.

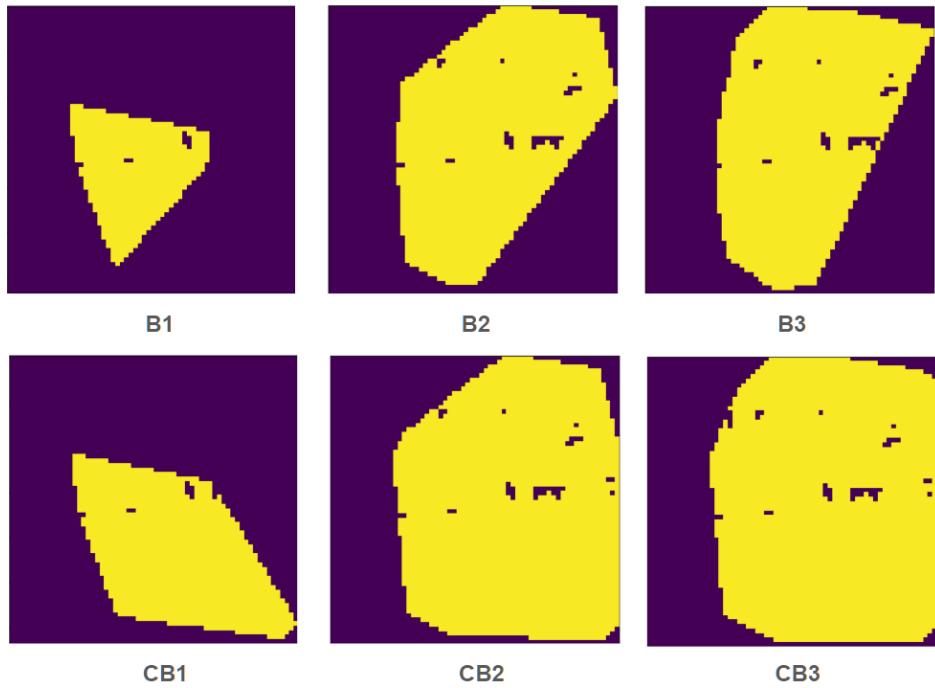


Figure 4.6: Cumulative Burned Areas

### 4.3 Fuel

The raw dataset of the fuel map provided a national map of Canadian FBP fuel types developed from public data sources. The map is updated to the year 2017 and is not available in high temporal resolution. In this thesis, I assumed fuel distribution does not change much and applied the fuel map of 2017 to all years from 1994 to 2021.

The raw dataset is stored in GeoTIFF format which is a rasterized data format. The empty value is filled by 65535 in the file and I changed it to 0 for the convenience of training tasks. The transformation has been carried out from its original CRS, NAD83 / Canada\_Lambert\_Conformal\_Conic, to the target CRS. Using the bounding box acquired from the burned area, I masked the Canadian fuel map and cropped it. Finally, the rasters were resampled based on the nearest neighbour.

It is not explicitly included in the models' structure in the rest of the paper, but an embedding layer of output size 8 was used to project categorical fuel values.

Table 4.5: Canadian Forest FBP Fuel Types

FBP Fuel Code	Description
1	C1
2	C2
3	C3
4	C4
5	C5
7	C7
11	D1
12	D2
13	D1/2
31	O1
101	Non-fuel
102	Water
106	Urban or built-up area
425	M1 C25
525	M2 C25
625	M1/2 C25
635	M1/2 C35
650	M1/2 C50
665	M1/2 C65

## 4.4 Elevation

The raw Elevation dataset from FABDEM is a global elevation map available at 1 arc second grid spacing, which is approximately 30 meters. It uses WG484 as CRS and was transformed properly to the target CRS. The filling value of the elevation map was 65535 and was changed to 0.

The world map has been split into tiles with the size of 1 degree  $\times$  1 degree. This brought challenges to the searching and mapping of the feature. To tackle the problem, I computed reversely by getting a list of tiles needed from the transform of each fire event. This greatly reduced the searching time cost when there were 19011 tiles in Canada.

The complete elevation feature for each fire event came from a list of tiles. Each tile was masked, cropped, and resampled using the same transformation. Each tile contains a fragment of the required information, and I performed an XOR-like operation on all the

transformed rasters to integrate the data. Due to the error introduced by CRS transformation and resampling, the intersection areas of tiles were likely to contain abnormal values. I performed three steps of data cleaning. Pixels with NaN values, infinite values and abnormal values were replaced by the nearest neighbours in this order, where the extreme value is defined as when a pixel's elevation is over 9000(meters) or below -500(meters). There were 149 NaN values, 8 infinite values, and 534 fire events with abnormal values cleaned in this way.

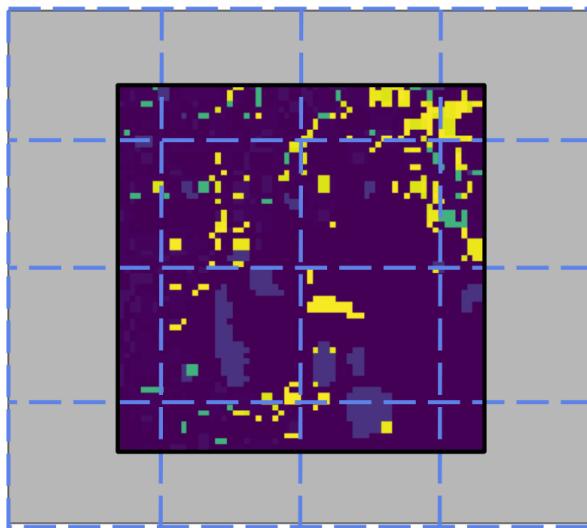


Figure 4.7: Elevation from Multiple Tiles

## 4.5 Fire Weather Index

The Fire Weather Index (FWI) raw dataset is a list of NetCDF files containing daily FWI data. It was stored in WGS84. I extracted the coordinates of each FWI record, transformed it into a geometric point, then reformatted the data into GeoDataFrame and changed the CRS.

The data is global but very sparse. A reasonable explanation for the weather-related data, the weather stations provide finite coverage. I introduced a nearest neighbour interpolation with an expanding window to solve this problem. It should be noted that advanced interpolation would perform better but considering the map is national, it could take great computation cost under high spatial resolution.

Besides a sight window, I created a sampling window which was initialized as a copy of the sight window. When there was not a valid value in the sampling window, it was expanded by a factor = 0.1, which means each side of the window extended by 0.1 of the current length to both directions. The area expansion rate was then 1.44 and it could grow to over 38 times as big as the sight window in 10 iterations. I limited the maximum number of iterations to 10 and made the algorithm take an average if multiple valid values appear in one expansion. If there was not any valid value within 10 iterations, a fire event is considered abandoned.

## 4.6 Tebular(1D) Features

Besides the rasterized features above, the thesis added three scalar parameters acquired along with the burned area from the NBAC database. They are agency, temporal feature and fire cause. The 1D features were integrated as a 1D array with a length of 50. The idea of introducing tabular features was inspired by the simulation setting in Burn-P3 [29] and was expected to provide a context of the fire event and help the prediction.

### 4.6.1 Agency

The agency is a categorical feature of jurisdiction indicating which province, territory or national park a fire event was located. There are 12 labels of provinces and territories and 21 labels of national parks. The feature was stored in one-hot encoding with a length of 33.

### 4.6.2 Temporal Feature

I intended to include a temporal feature to boost the learning process. A temporal feature could be in day, month, or season. I chose months to be the temporal feature and stored it in one-hot encoding with a length of 12.

### 4.6.3 Fire Cause

The fire cause was included in cases when potential differences exist among wildfires from different sources. This feature was also stored in one-hot encoding, which took 5 digits.

Table 4.6: Fire Cause Values and Descriptions

<b>Value</b>	<b>Description</b>
0	Undefined: Cause of fire unavailable
1	Other: Not covered by another category but resulting from known causes (e.g. “miscellaneous”, “incendiary”)
2	Lightning: Lightning
3	Industry: e.g. forestry, oil and gas, rail, agriculture, prescribed burn
4	Human: Nonindustrial human activity, e.g. recreational (campfire, ATV, etc), arson

# Chapter 5

## Fusion of Heterogeneous Features

Standard CNN structures often face significant challenges when it comes to integrating heterogeneous data sources, such as combining spatial (2D) data with auxiliary tabular (1D) data. The problem could be crucial when supplementary information can enhance model performance but is not easily incorporated into the standard CNN frameworks. I want a way to integrate spatial features and tabular features in the middle of U-shape networks without high dependency on the MLP. To address the problem, I introduce a novel module called Hybrid Bottleneck.

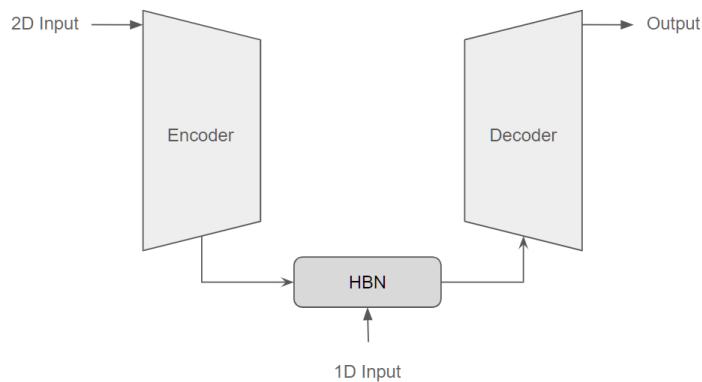


Figure 5.1: Hybrid Bottleneck Block in U-shape encoder-decoder

There are three components in U-shape networks, they are encoder, bottleneck and decoder. The encoder is a sequence of encoder blocks performing feature extraction and

representation learning. The decoder is a sequence of decoder blocks performing downstream tasks. Taking U-Net as a representation of U-shape networks, a bottleneck block used in the original U-Net architecture [31] is a module with two convolutional layers that perform further feature extraction from the output of the encoder while keeping the spatial resolution of the image. I introduced a Hybrid Bottleneck Block, a bottleneck block adding this Hybrid Bottleneck in order to utilize supplementary 1D information during high-level feature extraction in the bottleneck block.

## 5.1 Block Structure

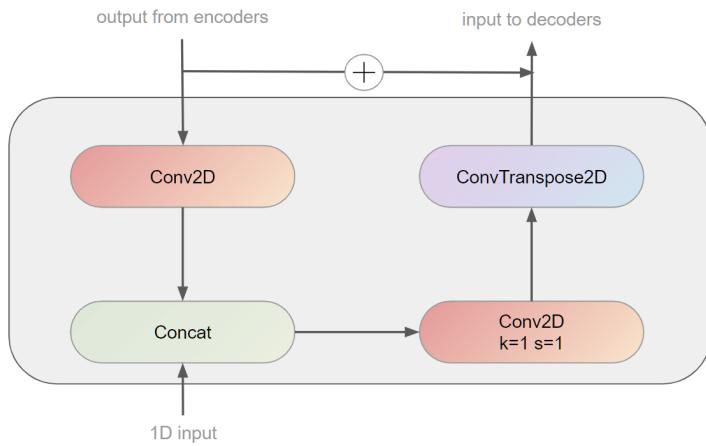


Figure 5.2: Hybrid Bottleneck Block

The Hybrid Bottleneck block enhances the model's capability to leverage both spatial and auxiliary data effectively. It comprises four layers: A convolutional layer for compressing the size of spatial data, a concatenation layer to connect tabular and compressed spatial data, a  $1 \times 1$  convolutional layer for data fusion, and a transpose convolutional layer for decompressing the data back to the same size as the input. Each convolutional layer is by default followed by a batch normalization and a RELU activation function for stable training.

The first convolutional layer is designed to take any spatial input and compress the spatial features from  $(H, W)$  to  $(1, 1)$ . This procedure could be performed through more than one convolutional block but the intention was to process small internal spatial data such as  $(2, 2)$  or  $(4, 4)$ . The spatial features could be considered flattened after that

but extra dimension removal might be needed (squeezing) before the data is ready for concatenation. After concatenation, the second convolutional layer functions similarly to a fully connected layer. It uses a 1x1 kernel and restores the dimension of the input. During this step, the extra dimension brought by tabular features is compressed into the original dimensions. At last, a transposed convolutional layer restores the size of images and makes the data shape unchanged from the input. I added a residual connection to reduce the potential gradient vanishing.

Embedding the hybrid bottleneck block into U-shape networks enables the utilization of auxiliary tabular data and improves the model’s performance in tasks requiring the integration of heterogeneous data sources. It keeps the data shape and thus does not introduce extra work like inserting or removing dimensions.

The hybrid bottleneck is a scalable structure since each layer in the block could be replaced by multiple layers of any size as long as the symmetry remains.

## 5.2 Performance

I have carried out two experimental studies of the Hybrid Bottleneck to prove the superiority of adopting it. The first study is about the advantage of utilizing tabular supplementary data in the research rather than only using spatial features. The second study is about the superiority of using a Hybrid Bottleneck structure instead of using an MLP layer during the fusion of heterogeneous data.

### 5.2.1 Ablation study on 1D features

The ablation study was carried out on U-Net which is the baseline model in this thesis. The treatment model was trained only on five spatial features and the control model was trained on five spatial features and three tabular features with Hybrid Bottleneck (HBN).

As is shown in table 5.1, U-Net has reached an F1 score of 0.9309 which is a good performance. After applying the Hybrid Bottleneck, the F1 score of the model has achieved 0.9312.

I carried out the T-Test on the two models and as is shown in the table 5.2, the test showed that the U-Net with the Hybrid Bottleneck is significantly better than the U-Net without the Hybrid Bottleneck, thus proving the strength of this module.

Table 5.1: Performance Metrics of the Ablation Study

Model+Data	Loss	Accuracy	Precision	Recall	Specificity	ROC AUC	PR AUC	F1 Score
U-Net	0.088542	0.968302	0.984371	0.88302	0.995516	0.989216	0.97749	0.93093
U-Net+HBN	0.087765	0.968377	0.982626	0.884932	0.995006	0.9895	0.977889	<b>0.931222</b>

Table 5.2: T-test on U-Net+HBN and U-Net

Model1	Model2	T-statistic	P-value	Significance
U-Net+HBN	U-Net	2.5415	0.0110	YES

### 5.2.2 Controlled Experiment with MLP

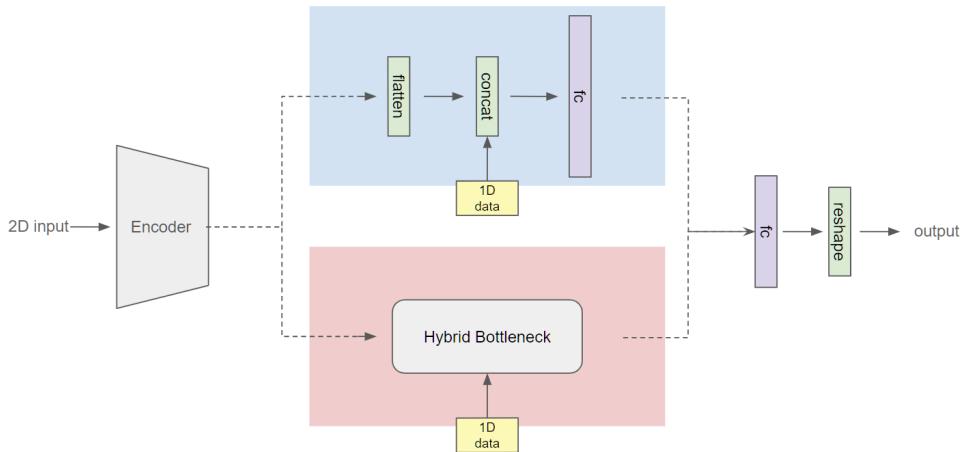


Figure 5.3: Controlled Experiment on Resnet 18

The Multilayer Perceptron (MLP) has shown high compatibility in many machine learning tasks. To prove the strength of the Hybrid Bottleneck in feature fusion, I proposed a controlled experiment. As is shown in Figure 5.3, a simplified model architecture was used for the wildfire prediction task. The architecture consists of an encoder, for which I used the encoder part in Resnet18, a feature fusion block, and the minimum combination of layers for downstream tasks.

The implementation of the fusion block was limited to the least necessary components and did not introduce any extra computation. The first implementation using MLP consists of a flattened layer, a concatenation of hybrid features and a MLP layer. The MLP layer functions similarly to the 1x1 convolutional layer from the Hybrid Bottleneck, which adjusts the size of data. The second implementation uses the Hybrid Bottleneck and has the same output size as the first implementation method.

Table 5.3: Performance Metrics of the Controlled Experiment

Model+Fusion Method	Loss	Accuracy	Precision	Recall	Specificity	ROC AUC	PR AUC	F1 Score
Resnet18+HBN	0.122746	0.957954	0.941683	0.880842	0.982561	0.982799	0.965945	0.910203
REsnet18+MLP	0.147394	0.944312	0.895637	0.871437	0.967567	0.979975	0.952931	0.883328

As is shown in table 5.3, with finite feature extraction of a simpler encoder, The Hybrid Bottleneck has shown strength against MLP implementation on utilizing supplementary data. The model using a Hybrid Bottleneck showed an absolute superiority on every metric, especially more than 2% higher on the F1 score.

### 5.2.3 Experimental Result Analysis

The ablation study of the Hybrid Bottleneck revealed the effectiveness of introducing hybrid features over single features in the prediction task on the best network structure I proposed in the thesis. However, due to the strong ability in feature extraction of the best neural network, the performance increment is noticeable but small. In the controlled experiment carried out on the Resnet18 encoder, I focused on the function of the fusion module and suppressed the rest parts in the model. The matrices proved the strength of the Hybrid Bottleneck in utilizing supplementary data when it could greatly assist the learning process of downstream tasks.

# Chapter 6

## Burned Area Prediction

This chapter elucidates the design of the next-day burned area prediction task, the model used, and the experimental result. I reproduced three deep-learning models from the existing literature reviews and then researched U-Net to seek better solutions. Extending from U-Net, I assessed the convolution and attention mechanism by testing two variants of the original U-Net.

Similar to the ablation study conducted in the last chapter, for any model that utilizes the HBN module in this chapter (the U-shape models since section 6.3), the metrics table recording model performances will include both models with HBN and models without HBN. When the name of a model is mentioned in a paragraph, it stands for the model with HBN structure in default (“model without the HBN structure” stated otherwise). When without the HBN structure, the model only takes spatial input.

### 6.1 Experiment Design

#### 6.1.1 Hyperparameters

The hyperparameters for my model were carefully selected through a combination of grid search and empirical testing.

- **Learning Rate:** The learning rate was set to  $\alpha = 0.001$ , chosen to balance the trade-off between convergence speed and stability.
- **Batch Size:** A batch size of 32 was used to ensure efficient utilization of GPU memory while maintaining a stable training process.

- **Number of Epochs:** The model was trained for 30 epochs, with early stopping criteria based on validation loss to prevent overfitting.
- **K-Fold Cross-Validation:** To ensure robust evaluation, I employed 5-fold cross-validation. This approach helps in assessing the model's generalization ability and mitigating overfitting.

### 6.1.2 Loss

Binary Cross Entropy Loss (BCELoss) was selected to measure the performance of my binary classification/segmentation model.

$$\mathcal{L}_{\text{BCE}}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (6.1)$$

where:

- $y_i$  represents the true label for the  $i$ -th sample, indicating whether a fire occurred ( $y_i = 1$ ) or not ( $y_i = 0$ ).
- $\hat{y}_i$  represents the predicted probability of fire occurrence for the  $i$ -th sample.
- $N$  is the total number of samples.

The advantages of using the BCE loss function in wildfire spread prediction include its effectiveness in handling imbalanced datasets by focusing on the prediction probability for each class, providing a smooth gradient which facilitates the training of deep learning models, and its direct relation to probabilistic outputs, making it suitable for binary classification tasks such as predicting fire spread probability on pixels.

### 6.1.3 Optimizers

The AdamW optimizer was chosen due to its adaptive learning rate capabilities and the added benefit of weight decay regularization. The weight decay parameter was set to 0.0001.

#### 6.1.4 Evaluation Metrics

To assess the performance of my model, I used the following evaluation metrics:

- **Accuracy:** This metric measures the proportion of correct predictions.
- **Precision:** Precision evaluates the number of true positive predictions made out of all positive predictions.
- **Recall:** Recall measures the number of true positive predictions made out of all actual positives.
- **Specificity:** Specificity assesses the number of true negative predictions made out of all actual negatives.
- **ROC AUC:** The Receiver Operating Characteristic Area Under the Curve (ROC AUC) provides a measure of the model's ability to distinguish between classes.
- **PR AUC:** The Precision-Recall Area Under the Curve (PR AUC) is used to evaluate the trade-off between precision and recall.
- **F1 Score:** The F1 score, which is the harmonic mean of precision and recall, was used as the main metric for evaluation.

#### 6.1.5 Additional Settings

Other important settings and configurations include:

- **Data Preprocessing:** The data was standardized using the statistical standard deviation and mean of the train set. The same standardization was applied to the train set and test set.
- **Hardware and Software:** The experiments were conducted on a NVIDIA GeForce RTX 4090 Laptop GPU, with the implementation done in Python using the Pytorch libraries.
- **Early Stopping:** Early stopping was applied in all training processes. The model with the lowest validation loss during all epochs was selected in each fold to prevent overfitting and ensure optimal performance on the test.

## 6.2 Past Work Reproduction

### 6.2.1 ANN

Singh et al.(2023) [34] voted the existence of research gaps in wildfire growth prediction and attempted to prove the practical value of deep learning algorithms through an Artificial Neural Network(ANN) structure.

The dataset in their research has a highly similar specification to the dataset in this thesis. The ANN model was designed for layers of  $64 \times 64$  spatial features with a pixel resolution of 1km. However, the paper lacks detailed level information on the model and the data flow. The block chart regarding model structure seems to be unfit for the input data. The largest layer in their model is a fully connected layer with 128 neurons which is a relatively small amount even compared to the number of pixels in a single spatial feature ( $64 \times 64$ ). Additionally, the authors did not mention much about the reshaping operations during their experiment which brought difficulty in reproduction.

With the missing information, I have added the least change necessary to fit the model with my data, which includes flattening and concatenation of the input, an extra fully connected layer for outputting prediction, and reshaping. The entire model consists of four blocks. Each block is a combination of a fully connected layer, an activation function, and a dropout layer except the last block which only has a fully connected layer.

Table 6.1: ANN

Block	Layer	Parameters	Input Size	Output Size
1	Linear		$12 \times 64^2 + 50$	128
1	RELU			
1	Dropout	p=0.5		
2	Linear		128	64
2	RELU			
2	Dropout	p=0.5		
3	Linear		64	32
3	RELU			
3	Dropout	p=0.5		
4	Linear		32	$64^2$

### 6.2.2 DCIGN

The Deep convolutional inverse graphics network (DCIGN) was proposed by J. Hodges and B. Lattimer. [18] The model was proposed as an efficient prediction model competing with traditional models regarding both time and computation cost, especially in heterogeneous landscapes with complex terrain.

The DCIGN is an encoder-decoder convolutional model. It was designed for 13 channels of  $50 \times 50$  spatial features and outputs two layers. I have changed the minimum arguments to fit the model with my data. The authors of the DCIGN added a dropout rate of 0.5 at the input layer to reduce the over-fitting problem. The feature extraction happened in two convolution blocks, which used a relatively uncommon large kernel with half padding. Similar to traditional CNN, a fully connected layer was added for downstream tasks. In the original network structure, the output was 2 layers of images. An additional transposed convolutional layer was inserted to compress the shape of the output.

There are two main reasons I did not apply the Hybrid Bottleneck in the DCIGN. The first reason is I did not intend to introduce much modification while a concatenation at the fully connected layer could do its job. The second reason is that the DCIGN did not compress the image size enough and it could bring a great portion of extra computation. The Hybrid Bottleneck intends to create a lightweight component possessing the functionality of feature fusion.

Table 6.2: DCIGN

Block	Layer	Parameters	Input Size	Output Size
0	Dropout	p=0.5		
1	Conv2d	k=10 s=1 p=half	(12, 64, 64)	(32, 64, 64)
1	ReLU			
1	MaxPool2d	k=2 s=2	(32, 64, 64)	(32, 32, 32)
2	Conv2d	k=3 s=1 p=half	(32, 32, 32)	(64, 32, 32)
2	ReLU			
2	MaxPool2d	k=2 s=2	(64, 32, 32)	(64, 16, 16)
3	Linear		$64 \times 16^2$	$2 \times 64^2$
3	Tanh			
3	ConvTranspose2d	k=10 s=1 p=half	(2, 64, 64)	(1, 64, 64)

### 6.2.3 FireCast

The FireCast model was proposed by Radke et al (2019) [30] is another deep-learning tool aiming to compete with the commonly used traditional fire prediction models. The model has been proven to outperform random prediction and FARSITE on their dataset. Since the model already includes fully connected layers, we performed concatenation as data fusion on the second fully connected layer directly.

The FireCast is a traditional CNN model using an encoder-decoder structure. In the encoder part, convolutional layers and pooling layers compress image size and fully connected layers perform downstream tasks. Two fully connected layers were inserted in the decoder and the data fusion happened between two layers by concatenation. The modified model inherited the original design but changed the size of the layers.

Table 6.3: FireCast

Block	Layer	Parameters	Input Size	Output Size
0	AvgPool	k=2 s=2	(32, 64, 64)	(32, 32, 32)
1	Conv2d	k=3 s=2 p=1	(32, 32, 32)	(32, 16, 16)
1	Sigmoid			
1	MaxPool2d	k=2 s=2	(32, 16, 16)	(32, 8, 8)
1	Dropout	p=0.5		
2	Conv2d	k=3 s=2 p=1	(32, 8, 8)	(64, 4, 4)
2	Sigmoid			
2	MaxPool2d	k=2 s=2	(64, 4, 4)	(64, 2, 2)
2	Dropout	p=0.5		
3	Linear		$64 \times 2^2$	$128 \times 2^2$
3	Linear		$128 \times 2^2 + 50$	$64 \times 64$

### 6.2.4 Experimental Result and Analysis

Among the three reproduced models. FireCast reached the best performance overall and was superior to the other two according to the table. The ANN model gave moderate performance and seemed to be not very compatible with the dataset when I used the same image sizes as their research. It reached an accuracy score of less than 90 percent. In terms of F1 score, it had a gap greater than 15 percent to the DCIGN, and a gap greater than 25 percent to the FireCast. This reflects the limited capabilities of this model on

Table 6.4: Performance Metrics for Reproduced Models

Model	Loss	Accuracy	Precision	Recall	Specificity	ROC AUC	PR AUC	F1 Score
ANN	0.3616	0.8334	0.692119	0.531867	0.929622	0.869662	0.678822	0.584773
DCIGN	0.254213	0.900614	0.832601	0.721183	0.957872	0.918157	0.819351	0.756338
FireCast	0.180651	0.932399	0.878606	0.836151	0.963113	0.970856	0.927653	0.856813

an unbalanced dataset. The limitation could come from the scale of the model and the pure composition of fully connected layers in the structure. DCIGN showed an improved performance compared to the ANN model. The accuracy of DCIGN has reached 90.06% and the F1 score has reached 75.63%. The FireCast outperformed the other two models and achieved an accuracy score of 93.24%. A gap of over 10% on the F1 score compared to the second model showed a better balance between positive and negative pixel classification.

## 6.3 Comparing Convolutional U-Shape Networks and Attention-Based U-Shape Networks

In this section, I delve into a comparative analysis of convolutional U-shape networks and attention-based U-shape networks by experimenting with U-Net, U-Net3+, and SwinUNet. The objective is to understand how these different architectures perform in the context of image segmentation, particularly in predicting wildfire growth.

U-Net serves as the baseline model in this analysis, providing a foundation to confirm the efficiency of the U-shape structure for the given task. U-Net3+ and SwinUNet represent two advanced extensions of this architecture, incorporating different mechanisms to enhance performance. U-Net3+ builds upon the traditional convolutional approach by adding deeper supervision and additional skip pathways, aiming to improve segmentation accuracy and robustness. SwinUNet, on the other hand, integrates self-attention mechanisms, enabling the model to dynamically focus on relevant features across the spatial dimensions of the image. Through the study of these two extensions, I hoped to explore the possibility of improvement on the baseline model.

### 6.3.1 U-Net

U-Net is a type of Convolutional Neural Network (CNN) invented by Olaf Ronneberger et al. in 2015. [31] It was primarily designed for biomedical image segmentation but has since

found applications in various fields requiring precise localization and context understanding. By leveraging its ability to capture intricate details and contextual information from spatial data, U-Net can process satellite imagery and other geospatial inputs to forecast the spread of wildfires.

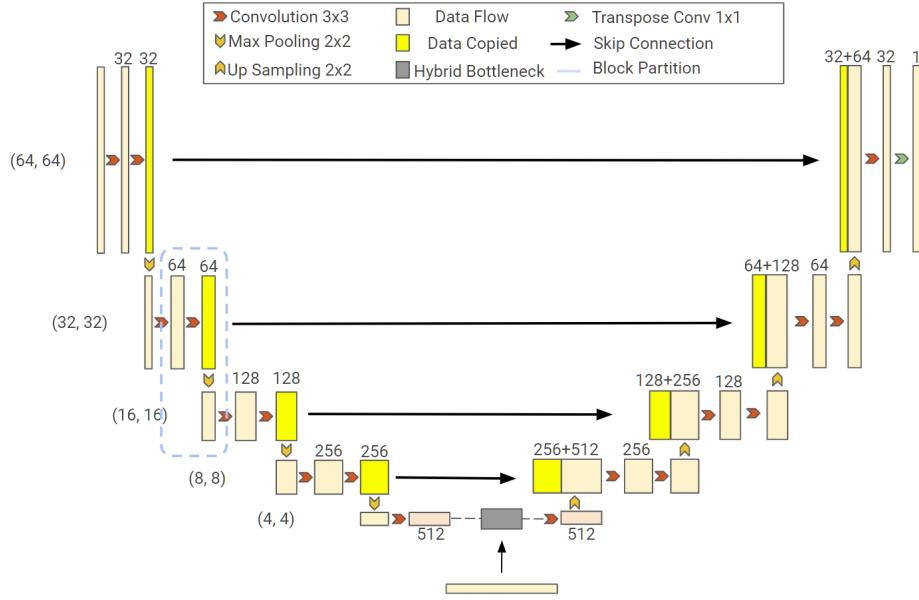


Figure 6.1: U-Net+HBN Structure

The U-Net structure in this thesis was a modification based on the original U-Net design proposed by the authors [31]. From the functional perspective, This U-Net consists of an encoder, a Hybrid Bottleneck, and a decoder. From the perspective of image shapes, each half of the U-Net contains four convolutional blocks at four levels. There are three components in each block. The first convolutional layer adjusts the size of channels, and the second convolutional layer extracts features without changing the shape of the data, then the last layer adjusts the spatial dimensions and sends it to the next layer. The independence among different functional components benefits the modification of the network structure. The bottleneck block in the original U-Net contains two convolutional layers. I assigned them to encoder and decoder separately, then inserted the Hybrid Bottleneck structure between two layers, renaming them the Hybrid Bottleneck block. I added a transposed convolutional layer in addition to the original U-Net to generate a  $64 \times 64$  probability map of the next day's burned area. The skip connection combines low-level and high-level features along with the decoder.

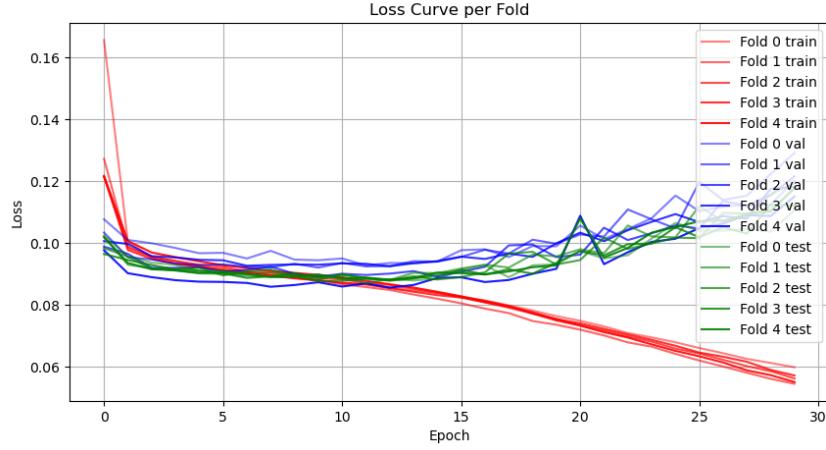


Figure 6.2: 5-fold loss curve of U-Net

The metrics of the experiment were already shown in the table 5.1. The U-Net model exhibits exceptional performance with a loss of 0.087765, indicating high prediction accuracy. It achieved an accuracy of 96.84%, precision of 98.26%, and recall of 88.49%, demonstrating its effectiveness in identifying both positive and negative cases. The specificity of 99.50% and high ROC AUC (98.95%) and PR AUC (97.79%) further confirms its robustness. The F1 score of 93.12% highlights U-Net's balanced performance, making it promising for precise and accurate wildfire spread prediction.

Through the loss curve during the training process, the U-net tended to overfit the dataset after approximately 12 epochs. The train loss kept going down while the validation loss went up. This was well controlled by the early-stop strategy based on the validation records.

### 6.3.2 Extensions

#### U-Net 3+

U-Net3+ is an advanced variant of the traditional U-Net architecture designed to enhance performance in image segmentation tasks. [19] One of the key innovations in U-Net3+ is its approach to aggregating multi-scale feature maps at each level of the decoder.

$$X_{De}^i = \begin{cases} X_{En}^i, & i = N \\ \mathcal{H} \left( \left[ \underbrace{C(\mathcal{D}(X_{En}^k))_{k=1}^{i-1}}_{\text{Scales: } 1^{\text{th}} \sim i^{\text{th}}} C(X_{En}^i), \underbrace{C(U(X_{De}^k))_{k=i+1}^N}_{\text{Scales: } (i+1)^{\text{th}} \sim N^{\text{th}}} \right] \right), & i = 1, \dots, N-1 \end{cases} \quad (6.2)$$

In this equation,  $X_{de}^i$  represents the feature map at the  $i$ -th level of the decoder, while  $X_{en}^i$  denotes the feature map from the corresponding encoder level. In this thesis,  $N=5$  where the fifth level stands for the bottleneck block. If treating the bottleneck block as a whole entity, there would be only one computation happening at this level thus  $X_{De}^i = X_{En}^i$  when  $i = N$ . For other levels ( $i = 1, \dots, N-1$ ), the decoder feature map  $X_{de}^i$  is obtained by concatenating the downsampled feature maps from the encoder at all preceding scales ( $1^{\text{st}} \sim i^{\text{th}}$ ), the feature map from the current encoder level  $X_{en}^i$ , and the upsampled feature maps from the decoder at all subsequent scales ( $(i+1)^{\text{th}} \sim N^{\text{th}}$ ). This concatenated stack is then processed by a function  $\mathcal{H}$ , which is a convolutional layer with a  $1 \times 1$  kernel in this research. This dense concatenation approach ensures that each decoding layer benefits from both fine-grained details and high-level contextual information, enhancing the model's ability to produce accurate and detailed segmentation outputs.

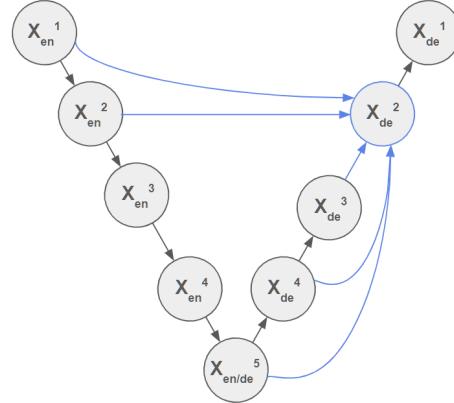


Figure 6.3: Feature aggregation on  $X_{de}^2$  in U-Net 3+

## Swin-Unet

Swin-UNet is a novel variant of the U-Net architecture that integrates Swin Transformers to enhance image segmentation performance. [7] Unlike traditional convolution-based U-

Net models, Swin-UNet leverages the power of self-attention mechanisms provided by Swin Transformers. This allows the model to dynamically capture and focus on relevant features across various spatial dimensions of the input image.

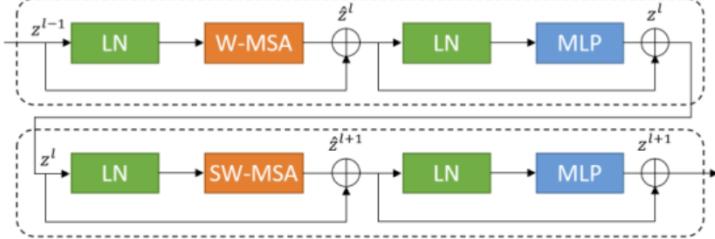


Figure 6.4: Swin Transformer Block [7]

$$\begin{aligned}
 \hat{z}^l &= W - MSA (LN (z^{l-1})) + z^{l-1} \\
 z^l &= MLP (LN (\hat{z}^l)) + \hat{z}^l \\
 \hat{z}^{l+1} &= SW - MSA (LN (z^l)) + z^l \\
 z^{l+1} &= MLP (LN (\hat{z}^{l+1})) + \hat{z}^{l+1}
 \end{aligned} \tag{6.3}$$

Swin-UNet replaces each regular convolutional block in the U-Net structure with two consecutive windowed multi-head self-attention blocks. For each pair of consecutive blocks at the same level, the first multi-head self-attention block divides the input image into non-overlapping windows and performs self-attention within each window independently. To enhance the model's ability to capture long-range dependencies across different regions of the image the second windows are shifted by a fixed number of pixels, and self-attention is performed again. This shifting mechanism is called shifted windowed multi-head self-attention. It ensures that the attention computation spans across the boundaries of the original windows, allowing the model to learn relationships between distant parts of the image.

### 6.3.3 Experimental Result

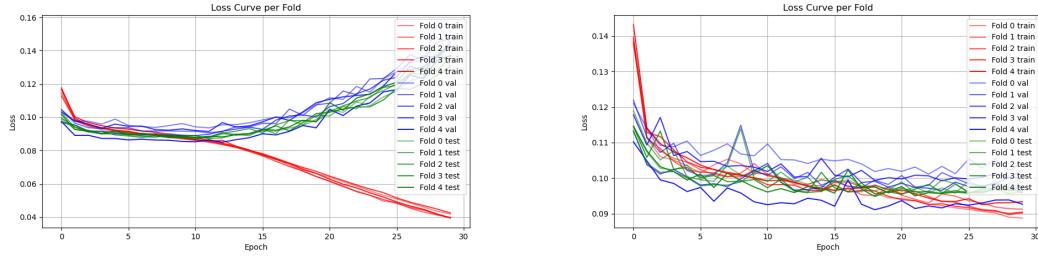


Figure 6.5: 5-fold loss curve of U-Net 3p(L) and Swin-Unet(R)

Table 6.5: Performance Metrics of U-Net/U-Net 3P/Swin-Unet with HBN

Model	Loss	Accuracy	Precision	Recall	Specificity	ROC AUC	PR AUC	F1 Score
U-Net	0.088542	0.968302	0.984371	0.88302	0.995516	0.989216	0.97749	0.93093
U-Net+HBN	0.087765	0.968377	0.982626	0.884932	0.995006	0.9895	0.977889	0.931222
U-Net 3p	0.087982	0.968429	0.984148	0.883778	0.995442	0.98934	0.97778	0.931242
U-Net 3p+HBN	0.087617	0.968456	0.982225	0.885663	0.994876	0.989454	0.977989	<b>0.931434</b>
Swin-Unet	0.095491	0.967722	0.989644	0.875763	0.997067	0.986288	0.973336	0.929214
Swin-Unet+HBN	0.09571	0.967662	0.986882	0.878033	0.996264	0.986269	0.973432	0.929263

The analysis (table 6.5) shows that while all three models perform exceptionally well, U-Net3+ slightly outperforms the others when considering the F1 score. This indicates that the enhancements in U-Net3+, such as the addition of deeper supervision and more extensive skip connections, provide a meaningful improvement over the original U-Net. SwinUNet, despite its innovative use of self-attention mechanisms, falls slightly behind in terms of F1 score due to its lower recall, suggesting it might be less effective in capturing all true positives in the dataset.

The convolutional network models tend to overfit the data but early-stop effectively prevented it and kept the generalization of the models. The loss score of Swin-Unet shows a larger fluctuation compared to the other two models. Instead of a smooth decrease, it oscillates but converges better than the other two models. A slight over-fitting can still be observed, but the Swin-Unet has shown its advantage in training convergence and stability, which is probably due to the regularization in the structure.

Table 6.6: T-test on U-Net 3p and Swin-Unet (with/without HBN)

Model1	Model2	T-statistic	P-value	Significance
U-Net 3p+HBN	U-Net 3p	5.6771	<0.0001	YES
Swin-Unet+HBN	Swin-Unet	1.3956	0.1629	NO

T-tests on the two variant models were conducted, and the result (table 6.6) reflects that the U-Net 3p has a significantly better F1 score when with the HBN structure. However, the HBN only brought a very slight increase in performance to Swin-Unet.

In conclusion, while U-Net3+ achieves the highest F1 score, indicating the best balance between precision and recall, SwinUNet's stable convergence and resistance to overfitting make it a strong contender, especially in scenarios where overfitting is a concern.

# Chapter 7

## **AA-Unet, a New Multi-scale, Convolutional Attention Based Architecture**

In the realm of medical image segmentation, U-Net has established itself as a highly effective architecture, particularly for tasks requiring precise localization and delineation of structures. From the last chapter, the U-Net structure and its advanced variants have also performed well in the wildfire growth prediction task.

From U-Net 3+, it was concluded that the aggregation of multi-scale feature maps helps the inference process. Additionally, although Swin-Unet fell behind the convolutional U-shape networks by a small gap, the attention mechanism has revealed its unique strength. I sought to explore the room for improvement based on U-net, particularly in capturing multi-scale context and enhancing feature selection through attention mechanisms. To address these limitations, this chapter proposes AA-Unet.

The chapter starts by introducing the structure of AA-Unet, and then explains the novel points in this model in detail. Finally, it supports the idea with the experimental results of comparison to models from the last chapter as well as the ablation study.

## 7.1 AA-Unet

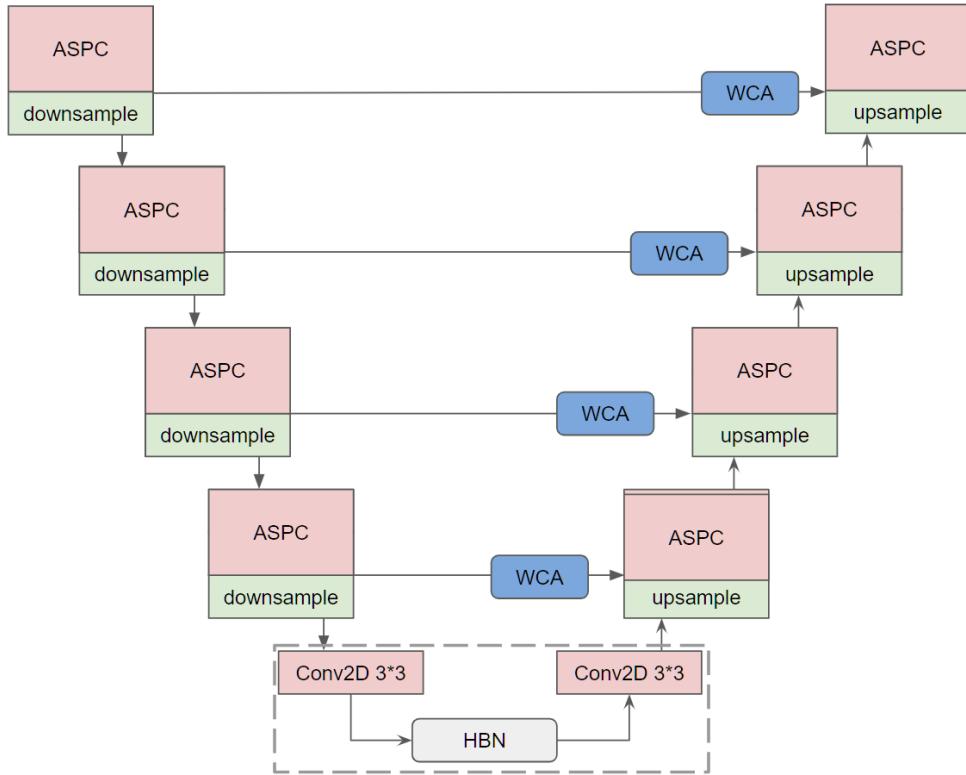


Figure 7.1: AA-Unet+HBN Structure

The AA-Unet architecture is an enhancement of the traditional U-Net, designed to improve segmentation performance by integrating advanced convolutional techniques. The encoder path of the AA-Unet consists of multiple layers of Atrous Spatial Pyramid Convolution (ASPC) Blocks, which replace the standard convolutional layers found in the original U-Net. These ASPC layers employ dilated convolutions with varying dilation rates, capturing multi-scale features and providing a richer representation of the input image. The decoder path mirrors the encoder's structure, utilizing transpose convolutions to upsample the feature maps. A key innovation in the AA-Unet is the replacement of the standard concatenation-based skip connections with convolutional attention mechanisms. These attention mechanisms dynamically weigh the features from the encoder before merging them with the corresponding decoder features, enhancing the model's ability to focus on the most relevant information. AA-Unet is a network that not only preserves the spatial hier-

archies of the original U-Net but also leverages the power of multi-scale feature extraction and attention-based refinement.

## 7.2 Atrous Spatial Pyramid Convolution (ASPC)

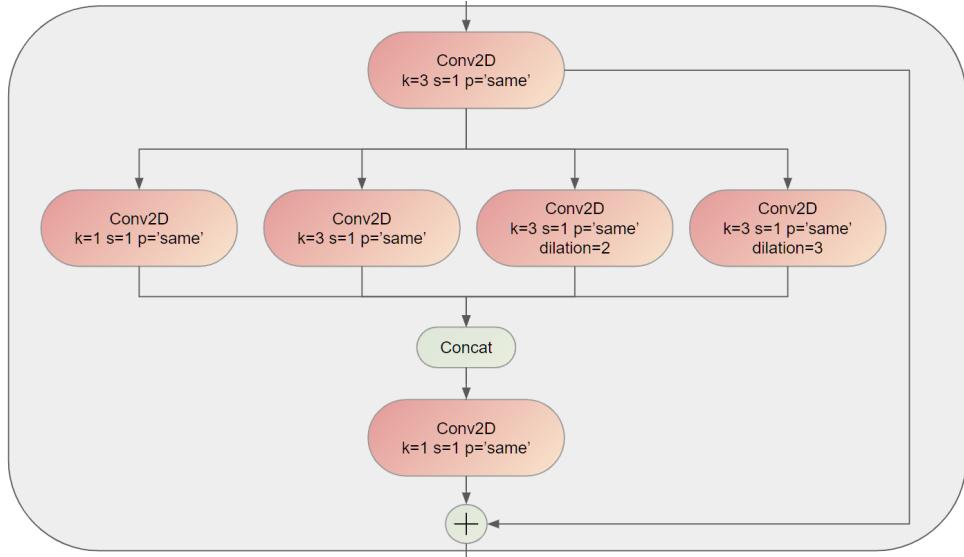


Figure 7.2: Atrous Spatial Pyramid Convolution (ASPC)

The Atrous Spatial Pyramid Convolution structure is a crucial component of the AA-Unet, designed to enhance the model's ability to capture multi-scale contextual information. The idea was inspired by the Atrous Spatial Pyramid Pooling (ASPP) technique [9], but ASPC modifies and extends this concept by focusing on convolutions rather than pooling operations. This structure allows for more effective feature extraction across multiple scales without reducing the spatial resolution of the feature maps.

Unlike traditional convolutional layers that use fixed kernel sizes, atrous convolutions introduce gaps between the kernel elements, enabling the receptive field to expand without increasing the number of parameters or the computational cost. In the AA-Unet, the ASPC layers are strategically placed within the encoder path, replacing standard convolutional blocks. Starting with a standard  $1 \times 1$  convolutional layer, I extract information at four different scales by adjusting the dilation rate. Multi-scale information is aggregated through a  $1 \times 1$  convolutional layer. To further enhance the learning capacity and stability of the

network, residual links are incorporated within the ASPC layers. In figure 7.2, p='same' means that padding is automatically calculated and filled surrounding the image to keep spatial dimension after convolution operations.

### 7.3 Windowed Convolutional Attention (WCA)

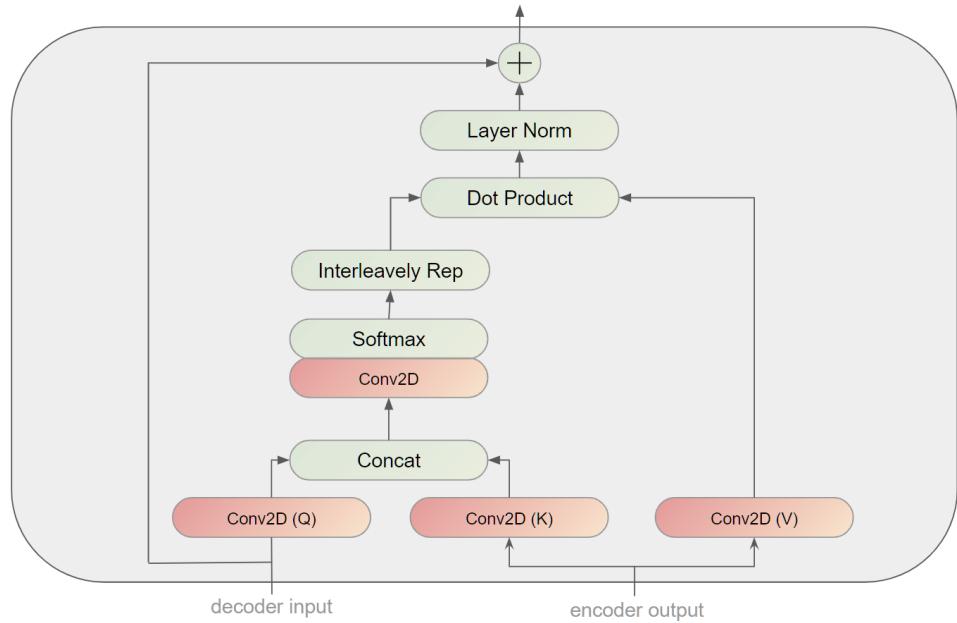


Figure 7.3: Windowed Convolutional Attention (WCA)

In the AA-Unet, the skip connections between the encoder and decoder paths have been significantly enhanced with the introduction of the Windowed Convolutional Attention (WCA) mechanism. This new approach is inspired by the principles of attention [36] but is implemented using convolutional layers.

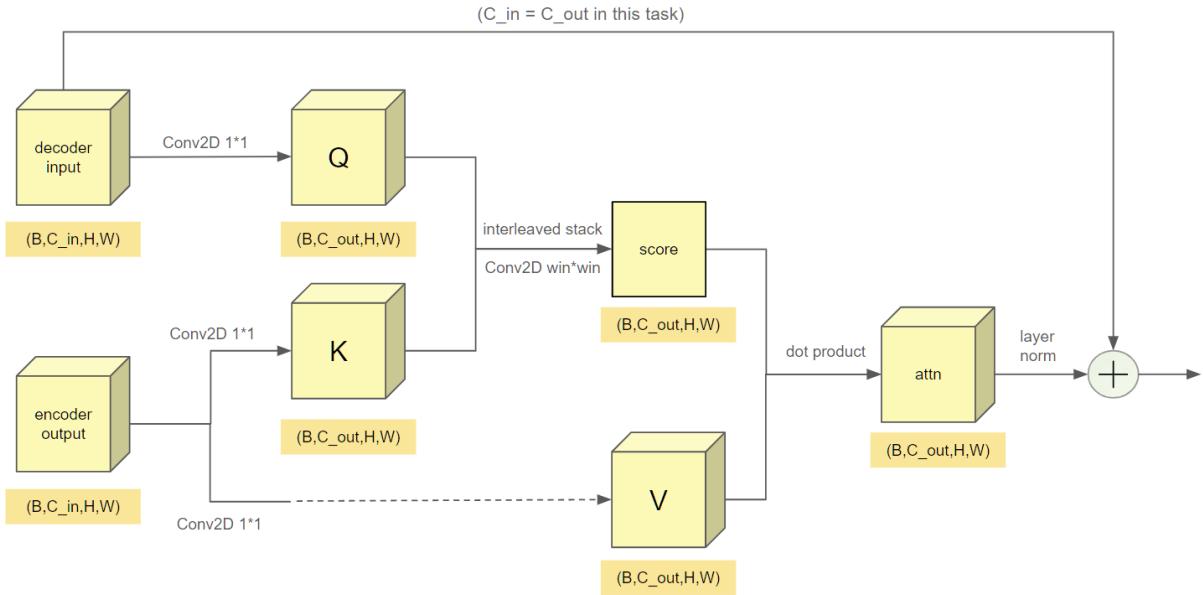


Figure 7.4: Data Flow in WCA

The structure of the Windowed Convolutional Attention mechanism is as follows:

- 1. Replacing the Projection Matrices for QKV:** In traditional attention mechanisms, Q (Query), K (Key), and V (Value) are generated using projection matrices. In the Windowed Convolutional Attention, Q, K, and V are instead derived from three separate  $1 \times 1$  convolutional layers. Q comes from the decoder input, and K, V come from the encoder output at the same level.
- 2. Generating the Attention Score:** K and V are stacked together and passed through a convolutional layer with a kernel size equal to the window size. The output of this convolutional layer is then processed through a softmax operation to produce a single-layer attention score.
- 3. Element-wise Product:** An element-wise matrix product (dot product) was used to apply the score weight on V.

Compared to traditional attention mechanisms that rely on global context and projection matrices for computing Q, K, and V, the Windowed Convolutional Attention in

AA-Unet offers some advantages. Benefiting from the computational efficiency and simplicity of convolutions, WCA could project and perform windowed attention with a lighter spatial computation cost.

## 7.4 Multi-Head WCA

Building upon the original WCA, the multi-head WCA introduces several key modifications to enhance feature representation and model performance. This advanced approach leverages the benefits of multiple attention heads to capture diverse features and improve the model's performance in segmenting complex structures.

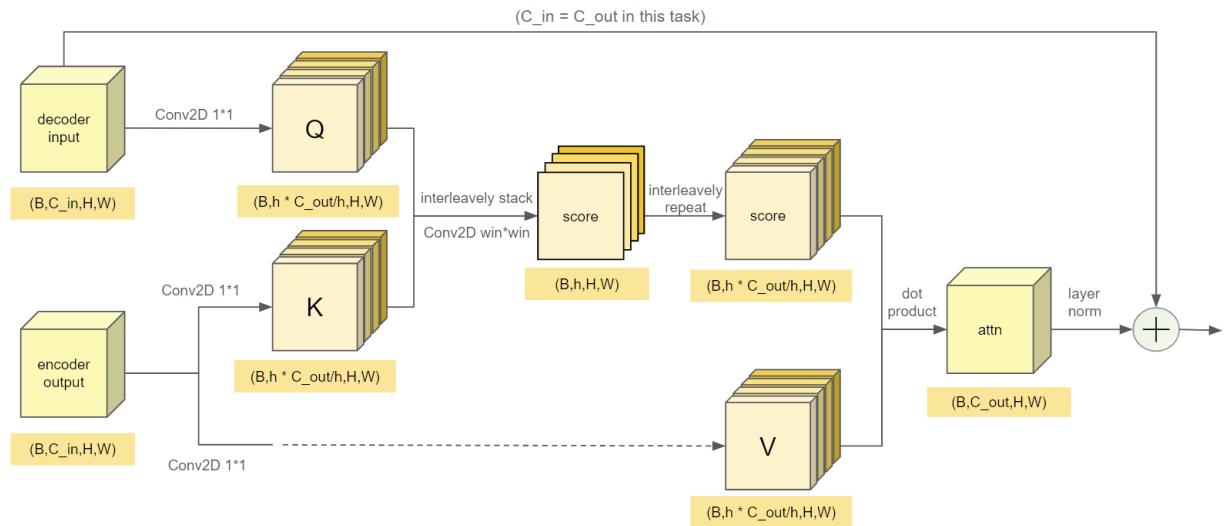


Figure 7.5: Data Flow in Multi-head WCA

In the multi-head WCA, Q and K are derived from the decoder and encoder inputs, which is similar to the original WCA from the last section. However, instead of handling these matrices as single entities, the multi-head approach stacks Q and K in an interleaved manner, ensuring that features with the spatial index are adjacent channel-wise (For example, point(C=0, i=1, j=3) and point(C=1, i=1, j=3)). Once stacked, a grouped convolutional layer is applied to the interleaved Q and K. By setting the group parameter to the number of attention heads, the grouped convolution effectively partitions the channels into multiple groups, each corresponding to a different attention head. This operation

enables the model to compute attention scores that represent various perspectives of the input features. The resulting attention scores, comprising multiple channels corresponding to the different heads, undergo a softmax operation to ensure they sum to one within each head. These multi-head attention scores are then repeated in an interleaved manner to match the dimensionality of  $V$ . Specifically, the first ( $total_{channels}/num_{heads}$ ) features in  $V$  correspond to the first head, and so on, ensuring that the attention scores are correctly aligned with their corresponding features in  $V$ . As in the original WCA, the adjusted feature maps are then passed through a layer normalization step to stabilize the training process, and a residual link is added to maintain effective gradient flow and enhance model stability. By leveraging multiple attention heads, the multi-head WCA captures a wider range of features and interactions within the input data, allowing the model to attend to different aspects of the features simultaneously.

## 7.5 Experimental Result

Table 7.1: Performance Metrics of AA-Unet, U-Net, U-net 3+ and Swin-Unet

Model	Loss	Accuracy	Precision	Recall	Specificity	ROC AUC	PR AUC	F1 Score
U-Net+HBN	0.087765	0.968377	0.982626	0.884932	0.995006	0.9895	0.977889	0.931222
U-Net 3p+HBN	0.087617	0.968456	0.982225	0.885663	0.994876	0.989454	0.977989	0.931434
Swin-Unet+HBN	0.09571	0.967662	0.986882	0.878033	0.996264	0.986269	0.973432	0.929263
AA-Unet	0.086757	0.968625	0.985697	0.883129	0.995908	0.989815	0.97843	0.931593
AA-Unet+HBN	0.086323	0.968692	0.985361	0.883718	0.995808	0.989976	0.978659	<b>0.931772</b>

In the experiment, I used AA-Unet with head numbers equal to  $2^{layer-1}$  (1,2,4,8). The AA-Unet achieved the highest F1 Score of 93.18%, surpassing U-Net (93.12%), U-Net 3+ (93.14%), and Swin-Unet (92.93%). This indicates that AA-U-net provides the best overall performance in terms of accurately and reliably identifying positive cases while minimizing false positives and false negatives.

In addition to its superior F1 Score, AA-Unet also demonstrated strong performance in other key metrics. It recorded the lowest Loss value of 0.086323, indicating better model fitting and lower prediction errors. The model achieved a high Accuracy of 96.87%, reflecting its reliability in making correct predictions. For Precision and Recall, AA-Unet scored 98.54% and 88.37%, respectively, showing a good balance in identifying true positives while maintaining high correctness in positive predictions. AA-Unet excelled in Specificity with a value of 99.58%, the highest among the compared models, indicating its effectiveness

in correctly identifying negative instances. The model also performed well in ROC AUC (98.99%) and PR AUC (97.87%), demonstrating its strong capability in distinguishing between positive and negative classes and maintaining a good precision-recall trade-off.

Table 7.2: T-test on AA-Unet/U-Net with HBN

Model1	Model2	T-statistic	P-value	Significance
AA-UNet+HBN	U-Net+HBN	1.9095	0.05621	NO
AA-UNet+HBN	AA-UNet	-2.807664319	0.0049	NO

Additionally, I have carried out the T-test on the significance of AA-Unet against the baseline model regarding the F1 score. The P-value is close to the threshold of 0.05 but still not enough to prove the statistical significance of the AA-Unet. However, AA-Unet without HBN tends to perform better than AA-Unet with HBN. This relationship was also observed in the ablation study on Swin-Unet.

## 7.6 Ablation Study

An ablation study was conducted to prove the effectiveness of different architectural components on the performance of the AA-Unet model. Specifically, the study compared the performance of the following models: the baseline U-Net, AA-Unet without the Atrous Spatial Pyramid Convolution (ASPC) structure, AA-Unet without the Windowed Convolutional Attention (WCA) structure, and the full AA-Unet.

Table 7.3: Performance Metrics of the Ablation Study on AA-Unet

Model	Loss	Accuracy	Precision	Recall	Specificity	ROC AUC	PR AUC	F1 Score
U-Net+HBN	0.087765	0.968377	0.982626	0.884932	0.995006	0.9895	0.977889	0.931222
AA-Unet+HBN without ASPC 3p	0.087617	0.968456	0.982225	0.885663	0.994876	0.989454	0.977989	0.931434
AA-Unet+HBN without WCA	0.087543	0.968453	0.982441	0.885431	0.994946	0.989541	0.978017	0.931411
AA-Unet+HBN	0.086323	0.968692	0.985361	0.883718	0.995808	0.989976	0.978659	<b>0.931772</b>

The baseline U-Net achieved an F1 Score of 93.12%, serving as a reference for evaluating the improvements introduced by the AA-Unet components. The AA-Unet without the ASPC structure showed a slight increase in recall (88.57%) compared to the baseline but a minor decrease in precision (98.22%). The AA-Unet without the WCA structure showed a similar trend of a slight increase in recall (88.54%) and a small decrease in precision

(98.24%). However, both models achieved a higher F1 than the baseline model but lower than the complete AA-UNet. This evidence illustrates that both novel components in AA-UNet have brought a performance boost in the prediction task.

The ablation study highlights the significant contributions of both the ASPC and WCA structures in enhancing the performance of the AA-Unet. The complete AA-Unet, integrating both enhancements, consistently outperforms the baseline U-Net and its simplified variants across multiple metrics, particularly the F1 Score, underscoring the effectiveness of these architectural innovations in accurately predicting wildfire growth.

# Chapter 8

## Spread Error Attenuation Field (SEAF)

This chapter proposes a new idea regarding enhancing specific types of error metrics computation for image segmentation tasks. The effectiveness of this idea has not been fully proven through the experiment, however, I would like to share the concept for its hidden potential.

### 8.1 Introduction

The Spread Error Attenuation Field (SEAF) is introduced as a novel approach to enhance error/loss calculations for 2D segmentation tasks conducted on a 2D grid. The essence of SEAF is to impose an additional penalty on clusters of erroneous predictions that are spatially close to each other, which could be particularly useful in domains such as forest wildfire prediction because the additional punishment will push the model to pay higher attention to the growing fire edges, providing precise burned area shapes. This approach aims to incentivize the model to learn and predict patterns with higher resolution rather than making broad area estimations.

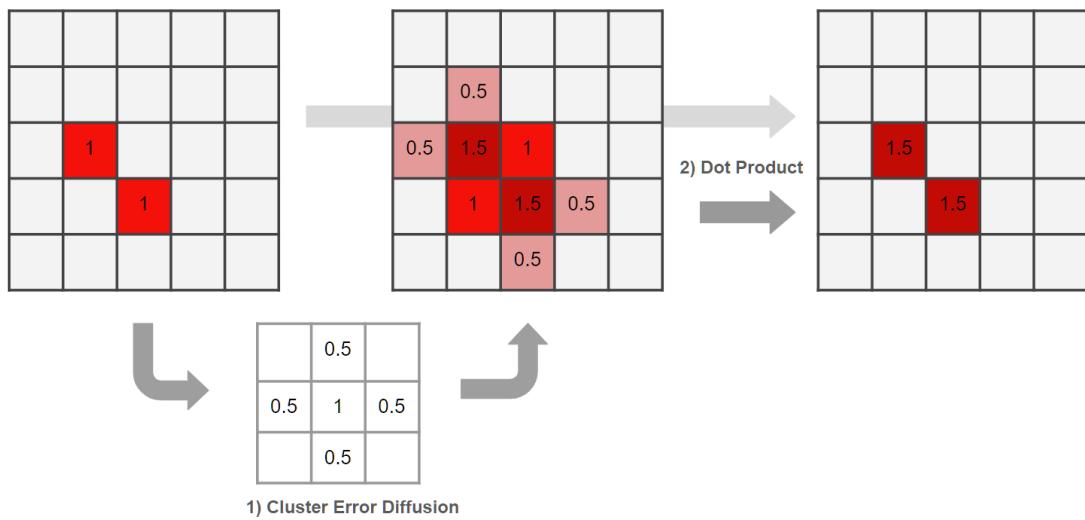


Figure 8.1: A SEAF Example with grouped error

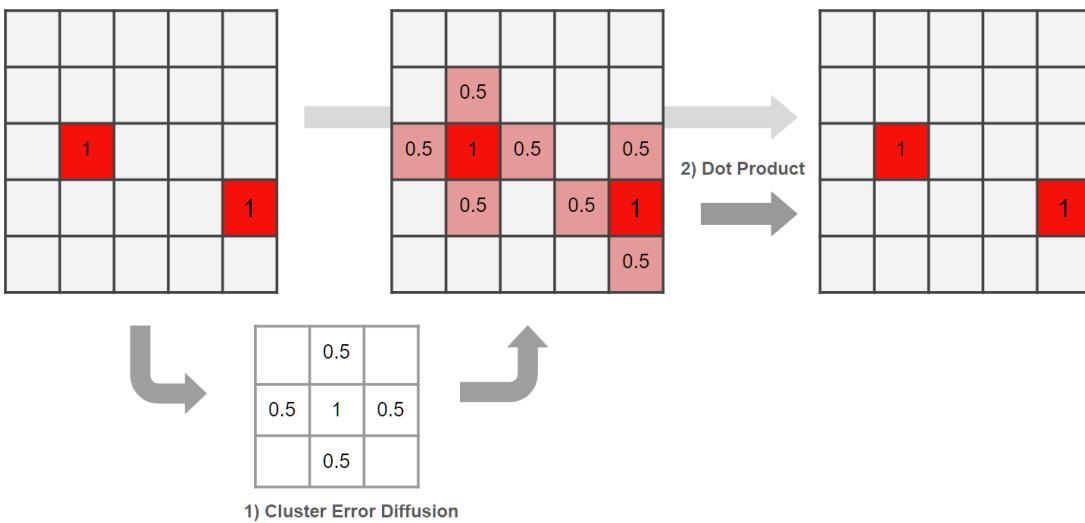


Figure 8.2: A SEAF Example without grouped error

## 8.2 Core Idea

To encourage the model to learn and predict patterns in high resolution instead of making an area estimation, errors appearing in groups shall get penalized harder. The primary objective of SEAF is to penalize grouped prediction errors more severely to encourage the model to improve its spatial accuracy. The result of SEAF is an enhanced form of the error mask, which could be used for BCELoss in my experiments or any criterion functions relying on it.

### 8.2.1 SEAF Kernel

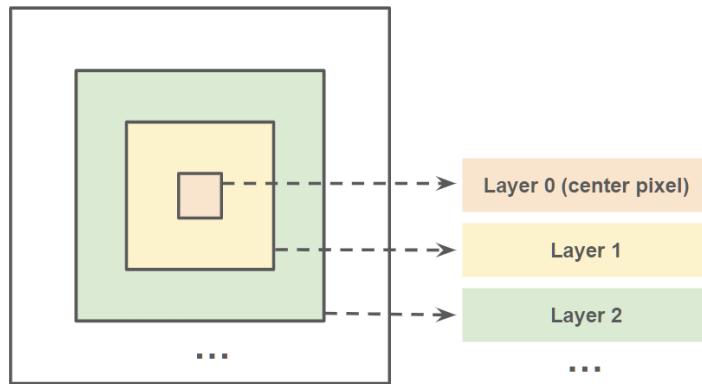


Figure 8.3: SEAF Kernel

A convolutional kernel is designed with a weighted influence that decreases with distance from the center pixel. Each pixel within the kernel's range contributes to the total penalty based on its distance from the center, thus defining layers of influence.

### 8.2.2 Layer Definition

The layers are defined based on the distance from the central pixel, and this definition is flexible. For instance, instead of assigning a single distance to each layer (e.g., layer one for pixels at distance 1), layers can encompass a range of distances (e.g., layer one for pixels at distances 1-2). Two common methods for defining these layers are:

1. Geometric Perspective (Grid A): Defines layers clearly based on geometric distances.

2. **Manhattan Distance (Grid B):** Utilizes the Manhattan distance to allow flexible weight distribution, which was adopted in my thesis project.

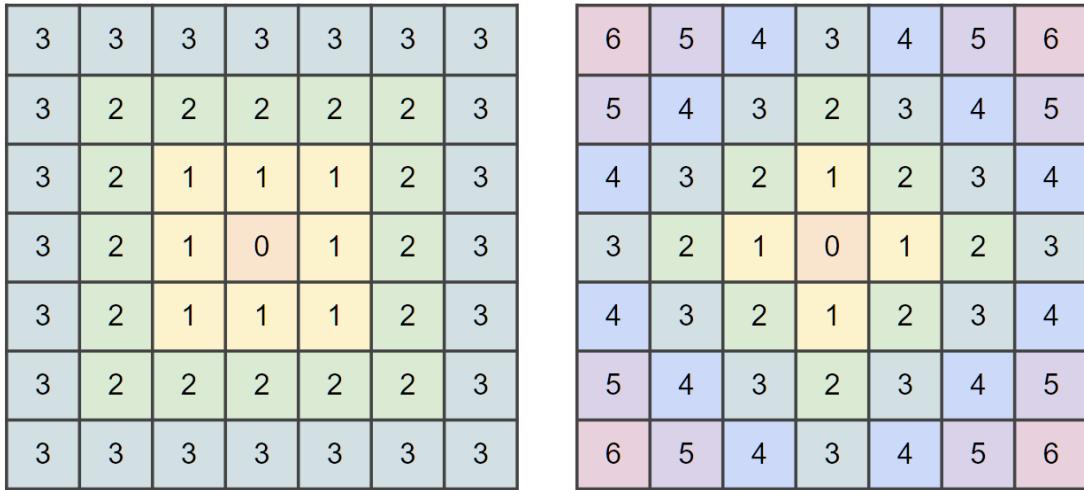


Figure 8.4: Grid A(Left) & Grid B(Right)

### 8.2.3 Weight Calculation of Pixels

To ensure that the influence of surrounding pixels does not exceed that of the center pixel, the weight summation in any layer  $n$  should not be greater than that of the previous layer  $n - 1$ . This attenuation is governed by an attenuation factor  $\alpha$ , ideally equal to or greater than 1. The weight function for any pixel  $p$  in layer  $n$  is defined as follows:

$$W_p^n = \begin{cases} 1 & \text{if } n = 0 \\ \frac{1}{2 \times \alpha^n \times \text{num\_of\_P}(n)} & \text{if } n > 0 \end{cases} \quad (8.1)$$

Here, the  $\frac{1}{2}$  term accounts for the double counting of the same error during convolution.

#### 8.2.4 OR Operation After Spread Using Error Mask (optional)

The SEAF method involves an optional element-wise product between the diffused error and the original error to capture the spread of cluster errors. Ideally, I expect a binary error mask instead of the original error, but applying a threshold function would make the error not differentiable and hinder the gradient descent. Alternatively, I could apply the original error as a soft version of the error mask. This operation is expected to help in identifying and penalizing clustered errors effectively.

#### 8.2.5 Additional Scaling Factor (optional)

The original idea regarding the relationship between outer layers and inner layers was that outer layers should always have smaller total weights, which means that the pixel's distance from the center is expected to be negatively correlated to its influence even under extreme cases when a layer only consists of wrong pixels. However, that case is relatively rare to happen. When trying to reduce the weight of the outer layers, the extra punishment produced by each pixel gets smaller. I implemented a constant scaling factor to tune the starting value of layer weights, and the experimental result showed that on some occasions, a scaling factor smaller than 1 (amplifying the sum of weight) helps perform better in prediction.

While the sum of weights for layer  $x$  should ideally not exceed that of layer  $x - 1$ , rare cases may arise where most or all surrounding pixels are errors. This becomes less likely as the number of pixels in layers increases, warranting consideration of additional scaling factors if needed.

#### 8.2.6 IoU Scaling Factor (optional)

To further enhance the SEAF method, I explored the incorporation of the Intersection over Union (IoU) metric as an additional dynamic scaling factor. This approach adjusts the penalization of prediction errors based on the spatial overlap between predicted and actual burned areas. It could be treated as a superior scaling factor and constant, weighting the error based on the relationship between prediction and ground truth masks.

By applying  $\frac{1}{\text{IoU}}$  as a scaling factor, the model becomes more sensitive to spatial prediction errors. Lower IoU values, indicating poor overlap, result in higher penalization, thus encouraging more accurate spatial predictions. This method provides continuous and contextual feedback, dynamically adjusting the error penalty based on the severity of spatial mismatches, which helps the model consistently improve its prediction accuracy.

### 8.2.7 Experimental Result

Table 8.1: Performance Metrics of BCELoss/SEAF-BCELoss on AA-Unet+HBN

Loss Function	Option	K	A	S	Loss	Accuracy	Precision	Recall	Specificity	ROC AUC	PR AUC	F1 Score
BCELoss	-	-	-	-	0.086323	0.968692	0.985361	0.883718	0.995808	0.989976	0.978659	0.931772
SEAF-BCELoss	-	5	0.5	1	0.393705	0.968688	0.982313	0.886534	0.994904	0.989272	0.978068	<b>0.931966</b>
SEAF-BCELoss	-	5	1	1	0.095054	0.96869	0.984807	0.884242	0.995638	0.989249	0.978019	0.931806
SEAF-BCELoss	-	5	2	1	0.045126	0.968619	0.982875	0.885728	0.99507	0.989264	0.978029	0.931769
SEAF-BCELoss	-	7	0.5	1	1.206955	0.968582	0.981799	0.886576	0.994751	0.989176	0.977913	0.931754
SEAF-BCELoss	-	7	1	1	0.128644	0.968637	0.98359	0.885127	0.995286	0.989199	0.977972	0.931762
SEAF-BCELoss	Error Mask	5	0.5	1	0.059271	0.968306	0.979177	0.887891	0.993967	0.989634	0.978185	0.931291
SEAF-BCELoss	Error Mask	5	1	1	0.01833	0.968378	0.980063	0.887372	0.994228	0.989751	0.978338	0.9314
SEAF-BCELoss	Error Mask	5	2	1	0.011218	0.968492	0.981678	0.886299	0.994721	0.989852	0.978493	0.931552
SEAF-BCELoss	Error Mask	7	0.5	1	0.157708	0.967679	0.972382	0.891779	0.991899	0.989219	0.977547	0.930314
SEAF-BCELoss	Error Mask	7	1	1	0.02245	0.968285	0.978073	0.888841	0.993637	0.989804	0.978382	0.931317
SEAF-BCELoss	IoU Scaling	5	0.5	1	0.258219	0.968567	0.984282	0.8842	0.995489	0.987863	0.976907	0.931552
SEAF-BCELoss	IoU Scaling	5	1	1	0.073914	0.968531	0.984698	0.88366	0.995614	0.987295	0.976461	0.931441
SEAF-BCELoss	IoU Scaling	5	2	1	0.043008	0.968535	0.98313	0.885136	0.995149	0.987409	0.976563	0.931556
SEAF-BCELoss	IoU Scaling	7	0.5	1	0.759363	0.968483	0.98142	0.886527	0.994637	0.9878	0.976872	0.931552
SEAF-BCELoss	IOU Scaling	7	1	1	0.094468	0.968454	0.981581	0.886273	0.994678	0.988042	0.97706	0.931475

Table 8.2: T-test on AA-Unet using SEAF-BCELoss/BCELoss with HBN

LossFunc1	LossFunc2	T-statistic	P-value	Significance
SEAF-BCELoss	SEAF	4.6438	<0.0001	YES

I conducted the experiment on enhanced BCELoss using SEAF. Three groups with different optional procedures and parameters were carried out. From the experimental result 8.1, the base SEAF-BCELoss could optimize the performance with some parameters. Among the base SEAF group, the AA-Unet achieved the highest F1 score in the thesis which is 93.20%(0.931966) when K(Kernel Size)=5, A(Attenuation Rate)=0.5, S(Scaling Factor)=1. The two optional procedures I proposed, the error mask and the IoU scaling, did not achieve higher scores from observation. For example, when applying the optional error mask in SEAF-BCELoss, the F1 score of AA-UNet dropped significantly to 93.03%.

Picking the best settings (K=5, A=0.5, S=1) in SEAF from the table 8.1, I conducted the T-test on AA-Unet trained using SEAF-BCELoss and original BCELoss regarding the F1 score, and the result (table 8.2) shows a clear significance that SEAF-BCELoss helped AA-Unet achieve higher F1 score. The T-test further demonstrates the potential of SEAF.

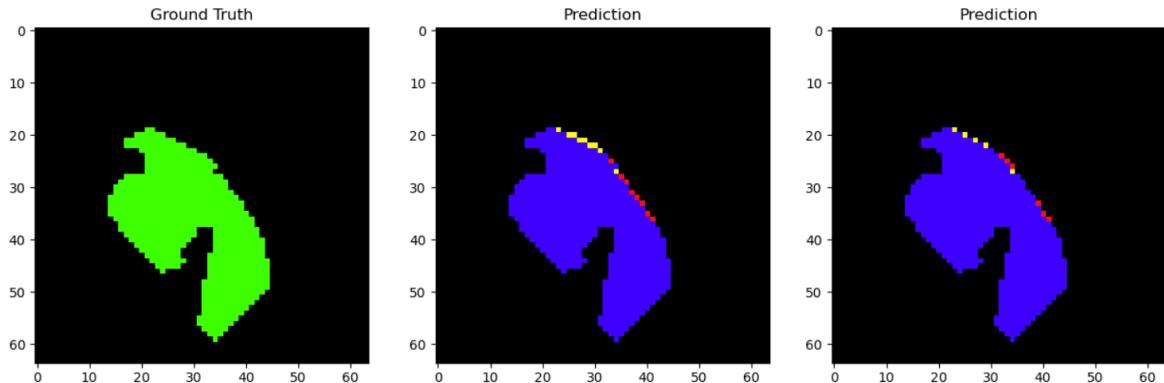


Figure 8.5: An example of prediction: Ground Truth(L), AA-Unet trained on BCELoss(C) and trained on SEAF-BCELoss(R)

Figure 8.5 shows an example of the visual result of how the SEAF changes the model's prediction. The input comes from a fire event that happened in BC, in 2015. The SEAF-BCELoss has reduced the false positive (yellow) and false negative (red) pixels predicted by the AA-Unet compared to just the BCELoss. This visual result provides a straight illustration of the functionality of the SEAF.

The results showed that while some combinations of parameters in the SEAF method resulted in improved performance, most attempts did not yield significant performance gains. Therefore, this idea has not been proven to be widely effective yet. However, the approach still shows potential and value, warranting further exploration and refinement.

# Chapter 9

## Conclusions and Future Works

### 9.1 Conclusions

In this thesis, the implementation of wildfire growth prediction tools using deep learning techniques was discussed. Features from multiple data sources were integrated and used to create a national dataset of wildfires in Canada from 1994 to 2021. To fuse the 2D and 1D features, the Hybrid Bottleneck structure was represented, which shows an advantage against MLP implementation. After that, prediction tools using U-Net and two of its extension variants were successfully built. All three models presented a satisfying performance during the test phase. Based on that, a convolutional neural network model called AA-UNet was offered, which deeply utilizes multi-scale and multi-level features with two enhancement modules. At last, I suggest SEAM, an interesting idea regarding error computation enhancement, which has not been fully validated but has potential for future exploration.

### 9.2 Future Works

The completion of this experiment is attributed to each of the procedures mentioned in the thesis. Reflecting on the experiments, here are some possible future works that could help the perfection of the research:

### **9.2.1 Dataset Making**

I came across different spatial resolutions during data collecting, and the most simple up-sampling method was adopted. The quality of data could be limited by the super-resolution challenge during the data processing, and there is room for improvement on this topic. Regarding the FWI features, the nearest neighbours algorithm was used to calculate approximation. The interpolation algorithm is a good direction for refining the extraction of the FWI layer. Extra features could also contribute to the model training.

### **9.2.2 Data Augmentation**

The wildfire dataset obtained in this thesis has finite samples. Data augmentation is a good method to provide a variety of samples. For example, randomly switching some pixels of the image as a form of noise could be adopted.

### **9.2.3 Multi-Class Prediction**

In the thesis, the prediction of binary burned areas has been investigated. If splitting the next-day cumulative burned area into two further sub-parts: the next-day burning area, and the rest cumulative burned area, the prediction models would face bigger challenges but also present more practical value to the field.

### **9.2.4 Comparison to traditional models**

Unfortunately, I was unable to conduct a comparative experiment between my deep learning implementation and traditional simulation models like Prometheus or FARSITE. Demonstrating a significant advantage of the deep learning model over these traditional models would provide compelling evidence of the potential benefits of deep learning in addressing Canadian wildfire issues. Hard access to certain datasets and specialized expertise requirements of some tools increase the difficulty of conducting this experiment. Ongoing research by our lab will be working with modelling experts in wildfire planning in the government to carry out these simulations in the future.

# References

- [1] Agriculture and Agri-Food Canada. Semi-decadal land use. <http://www.agr.gc.ca/atlas/landu>, 2021. Goverment of Canada.
- [2] Frank A Albini. *Estimating wildfire behavior and effects*, volume 30. Department of Agriculture, Forest Service, Intermountain Forest and Range . . . , 1976.
- [3] Patricia L Andrews et al. The rothermel surface fire spread model and associated developments: A comprehensive explanation. 2018.
- [4] André Beaudoin, Pierre Bernier, Luc Guindon, Philippe Villemaire, Xiaojing Guo, Graham Stinson, Thomas Bergeron, Steen Magnussen, and R.J. Hall. Mapping attributes of canada's forests at moderate resolution through k nn and modis imagery. *Canadian Journal of Forest Research*, 44:521–532, 05 2014.
- [5] Esri Canada. Provinces and territories of Canada. [https://hub.arcgis.com/datasets/d3fef65386df4e63b02d6e23bb98a1ee\\_0/about](https://hub.arcgis.com/datasets/d3fef65386df4e63b02d6e23bb98a1ee_0/about), 2024. ArcGIS Hub.
- [6] Health Canada. Wildfire smoke and your health. <https://www.canada.ca/en/health-canada/services/publications/healthy-living/wildfire-smoke-health.html>, 2024. Accessed: 2024-07-12.
- [7] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022.
- [8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

- [9] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.
- [10] Leckie D.G. Tinis S. Dyk, A. and S. Ortlepp. Canada's national deforestation monitoring system: System description. url<https://ostrnrcandostrnrcan.ca/handle/1845/244040>, 2015. Natural Resources Canada, Canadian Forest Service, Pacific Forestry Centre.
- [11] Mark A Finney. *FARSITE, Fire Area Simulator—model development and evaluation*. Number 4. US Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.
- [12] National Fire Database – Agency FireData. Canadian Forest Service. <http://cwfis.cfs.nrcan.gc.ca/ha/nfdb>, 2016. Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre, Edmonton, Alberta.
- [13] Canadian Forest Service. National burned area composite. <https://cwfis.cfs.nrcan.gc.ca>, 2023. Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre, Edmonton, Alberta.
- [14] Sriram Ganapathi Subramanian and Mark Crowley. Using spatial reinforcement learning to build forest wildfire dynamics models from satellite images. *Frontiers in ICT*, 5:6, 2018.
- [15] Rafik Ghali and Moulay A. Akhloufi. Deep learning approaches for wildland fires using satellite remote sensing data: Detection, mapping, and prediction. *Fire*, 6(5), 2023.
- [16] Laurence Hawker, Peter Uhe, Luntadila Paulo, Jeison Sosa, James Savage, Christopher Sampson, and Jeffrey Neal. A 30 m global map of elevation with forests and buildings removed. *Environmental Research Letters*, 17(2):024016, 2022.
- [17] Kelvin G Hirsch. *Canadian forest fire behavior prediction (FBP) system: user's guide*. Number 7. 1996.
- [18] J. Hodges and B. Lattimer. Wildland fire spread modeling using convolutional neural networks. *Fire Technology*, pages 1–28, 2019.
- [19] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected

- unet for medical image segmentation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1055–1059. IEEE, 2020.
- [20] F. Huot, R. Hu, M. Ihme, Qing Wang, J. Burge, Tianjian Lu, Jason Hickey, Yi-Fan Chen, and John R. Anderson. Deep learning models for predicting wildfires from historical remote-sensing data. *ArXiv*, abs/2010.07445, 2020.
  - [21] Piyush Jain, Dante Castellanos-Acuna, Sean CP Coogan, John T Abatzoglou, and Mike D Flannigan. Observed increases in extreme fire weather driven by atmospheric humidity and temperature. *Nature Climate Change*, 12(1):63–70, 2022.
  - [22] Piyush Jain, Sean CP Coogan, Sriram Ganapathi Subramanian, Mark Crowley, Steve Taylor, and Mike D Flannigan. A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4):478–505, 2020.
  - [23] Laurence Hawker Jeffrey Neal. FABDEM v1-2. <https://doi.org/10.5523/bris.s5hqmjcdj8yo2ibzi9b4ew3sn>, 2023. University of Bristol.
  - [24] Spyros Kondylatos, Ioannis Prapas, Michele Ronco, I. Papoutsis, Gustau Camps-Valls, M. Piles, M. Fernandez-Torres, and N. Carvalhais. Wildfire danger prediction and understanding with deep learning. *Geophysical Research Letters*, 49, 2022.
  - [25] Natasha Markuzon and Stephan Kolitz. Data driven approach to estimating fire danger from satellite images and weather information. In *2009 IEEE Applied Imagery Pattern Recognition Workshop (AIPR 2009)*, pages 1–7, 2009.
  - [26] Megan McElhinny, Justin F Beckers, Chelene Hanes, Mike Flannigan, and Piyush Jain. A high-resolution reanalysis of global fire weather from 1979 to 2018—overwintering the drought code. *Earth System Science Data*, 12(3):1823–1833, 2020.
  - [27] Natural Resources Canada. Canadian forest fire danger rating system (cffdrs). <https://cwfis.cfs.nrcan.gc.ca/background/summary/fdr>, 2024. Accessed: 2024-05-21.
  - [28] Natural Resources Canada. Canadian forest fire weather index (fwi) system. <https://cwfis.cfs.nrcan.gc.ca/background/summary/fwi>, 2024. Accessed: 2024-05-21.
  - [29] Marc-André Parisien, VG Kafka, KG Hirsch, JB Todd, SG Lavoie, PD Maczek, et al. Mapping wildfire susceptibility with the burn-p3 simulation model. 2005.
  - [30] David Radke, Anna Hessler, and Daniel Ellsworth. Firecast: Leveraging deep learning to predict wildfire spread. pages 4575–4581, 2019.

- [31] Olaf Ronneberger, Philipp Fischer, and Thomas 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*, pages 234–241. Springer, 2015.
- [32] Olaf Ronneberger, Philipp Fischer, and Thomas 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*, pages 234–241. Springer, 2015.
- [33] Richard C Rothermel. *A mathematical model for predicting fire spread in wildland fuels*, volume 115. Intermountain Forest & Range Experiment Station, Forest Service, US . . ., 1972.
- [34] Aman Singh, Rakesh Yadav, Gadug Sudhamshu, Aryan Basnet, and Rahmat Ali. Wildfire spread prediction using machine learning algorithms. *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–5, 2023.
- [35] Cordy Tymstra, RW Bryce, BM Wotton, SW Taylor, OB Armitage, et al. Development and structure of prometheus: the canadian wildland fire growth simulation model. *Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre, Information Report NOR-X-417.(Edmonton, AB)*, 2010.
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [37] Zhong Zheng, Wei Huang, Songnian Li, and Yongnian Zeng. Forest fire spread simulating model using cellular automaton with extreme learning machine. *Ecological Modelling*, 348:33–43, 2017.