WILDFIRE RISK ASSESSMENT USING CONVOLUTIONAL NEURAL NETWORKS AND MODIS CLIMATE DATA

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Sean Nesbit

June 2022

COMMITTEE MEMBERSHIP

TITLE:	Wildfire Risk Assessment Using Convolutional Neural Networks And MODIS Climate Data
AUTHOR:	Sean Nesbit
DATE SUBMITTED:	June 2022
COMMITTEE CHAIR:	Jonathan Ventura, Ph.D. Professor of Computer Science & Software Engineering
COMMITTEE MEMBER:	Paul Anderson, Ph.D. Professor of Computer Science & Software Engineering
COMMITTEE MEMBER:	Andrew Fricker, Ph.D. Professor of Social Sciences

ABSTRACT

Wildfire Risk Assessment Using Convolutional Neural Networks And MODIS Climate Data

Sean Nesbit

Wildfires burn millions of acres of land each year leading to the destruction of homes and wildland ecosystems while costing governments billions in funding. As climate change intensifies drought volatility across the Western United States, wildfires are likely to become increasingly severe. Wildfire risk assessment and hazard maps are currently employed by fire services, but can often be outdated. This paper introduces an image-based dataset using climate and wildfire data from NASA's Moderate Resolution Imaging Spectroradiometer (MODIS). The dataset consists of 32 climate and topographical layers captured across 0.1 ° by 0.1 ° tiled regions in California and Nevada between 2015 and 2020, associated with whether the region later saw a wildfire incident. We trained a convolutional neural network (CNN) with the generated dataset to predict whether a region will see a wildfire incident given the climate data of that region. Convolutional neural networks are able to find spatial patterns in their multi-dimensional inputs, providing an additional layer of inference when compared to logistic regression (LR) or artificial neural network (ANN) models. To further understand feature importance, we performed an ablation study, concluding that vegetation products, fire history, water content, and evapotranspiration products resulted in increases in model performance, while land information products did not. While the novel convolutional neural network model did not show a large improvement over previous models, it retained the highest holistic measures such as area under the curve and average precision, indicating it is still a strong competitor to existing models. This introduction of the convolutional neural network approach

expands the wealth of knowledge for the prediction of wildfire incidents and proves the usefulness of the novel, image-based dataset.

ACKNOWLEDGMENTS

An enormous thank you to:

- My family for always supporting me and setting me up for continued success
- My friends who provided much needed stress relief
- Dr. Ventura for guiding me through this thesis and providing invaluable feedback
- My committee for taking the time to review and evaluate this endeavour
- The entire Computer Science & Software Engineering department for such an incredible program

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Motivation

In California, the 2020 wildfire season burned 4.2 million acres of land [1] and threatened over 2 million homes [2]. The Legislative Analyst's Office estimated a total CalFire spending budget of approximately $2.5 billion, more than tripling their budget since 2005 [3]. New estimates for total wildfire damages from California's 2018 wildfire season cost the economy an amount estimated at upwards of $148 billion [4]. With increasing volatility regarding drought conditions due to climate change [5], California has struggled to manage increasingly active fire regimes.

Changing global climate trends threaten the natural cycles in our forests. Westerling showed how since 1970, spring seasons have been increasingly earlier in the year and noted that early spring seasons result in an increase in large-fire frequency [6]. The team observed a strong inverse correlation between snow-melt measured via streamflow gauges and wildfire severity. Importantly, the earlier spring snow-melt started, the more severe and numerous the wildfires would be during the summer. The team also found that warm spring and summer temperatures were often followed by drier, less precipitous winters. Other researchers have simulated temperature scenarios given a spectrum of estimates for emissions data, resulting in end-of-century temperature outcomes between +1.5°C and +4.5°C [7]. These findings, compounded by increasing drought severity [5] and volatility in yearly precipitation [8], suggest wildfires will continue to pose a growing challenge facing California.

Traditional wildfire risk assessment largely relies on maps drawn by government agencies such as the United States Forest Service. However, with the emergence of new and robust classification methods, increased research efforts have gone into developing risk assessment models using publicly available data. Combined with the rapidly growing amount of historical data available online, new opportunities are being afforded to redefine how wildfire risk assessment is approached. Specifically, various neural network designs such as convolutional neural networks remain a largely untapped area of study.

1.2 Design

In this thesis, we introduce and analyze a new approach for wildfire risk prediction. The approach utilizes a convolutional neural network that interprets climate satellite imagery retrieved with Google Earth Engine, with the images primarily acquired from NASA's Moderate Resolution Imaging Spectroradiometer (MODIS) aboard the Terra and Aqua satellites. The satellites produce near-real-time images which are free to the public, providing a wealth of invaluable data about the climate of our world. The neural network then generates a risk assessment for the given geospatial area using the corresponding climate images. The network can be given current climate images of a region and will estimate the wildfire risk, enabling it to be deployed as a real-time predictor of wildfire risk.

Previous research attempts in the field of wildfire risk assessment have minimally addressed the use of convolutional neural networks as an option for processing climate data. The subset of research articles that use MODIS data to predict wildfire risk often uses logistic regression, decision trees, and neural network types such as artificial neural networks. We believe that utilizing convolutional neural networks is necessary to advance the research space from strictly using tabular climate data to using images that can provide a more nuanced and spatial representation of an area. The primary difference is that images can represent the climate using a two-dimensional space that traditional logistic regression and artificial neural networks cannot. This adds an additional layer of information for the network to process that may allow it to better find patterns in the causes of wildfires.

There did not appear to be a dataset that organizes geospatial images containing multiple climate products with wildfire burn labels. The first step of this research was to use the Google Earth Engine and MODIS to collect a wide array of climate product images commonly associated with wildfires. This generates a novel dataset to train a network on to find the inherent differences between areas that burned and areas that did not burn depending on the climate data given. This data and its collection methods may also be used for future reproductions of the work in this paper and for further research in the interdisciplinary study of machine learning and environmental science.

Chapter 2

BACKGROUND

2.1 Wildfires

A wildfire is the unplanned or unintentional burning of any wildland area. Wildfires can be a combination of ground fires, surface fires, and crown fires. Ground fires burn subsurface material such as roots, peat, coal, or other combustible organic matter. Ground fires can spread underneath large areas, erupting into surface fires [9]. Surface fires burn brush and other small plants above the soil. As surface fires grow they can reach up through the canopy and ignite the tops of treas. Crown fires are fires that spread across the canopy, quickly jumping from tree to tree as the wind pushes burning material across the treetops with high velocity [10]. This spread is compounded with surface fire updrafts forcing burning material up from the forest floor, continuously igniting the canopy, while burning debris falls down to ignite unburned areas on the ground [11].

The biophysical drivers of wildfire, or ignition factors, play an important role in when and how an area will burn. Characteristics such as vegetative combustibility, fuel availability, and moisture content affect an area's ability to ignite. Fuel moisture content and the dryness of an area are primarily a product of climate information such as temperature, humidity, and precipitation, making moisture content a prime target in risk analysis. These biophysical factors have a significant effect on naturally caused wildfires but are slightly less important in determining outcomes for human-caused wildfires [12].

Wildfire ignition can be broadly categorized as either naturally caused or human-caused. Previous research suggests human-caused wildfires account for 60% of all land area burned by fires, with only 8% of the land being burned by events such as lighting. Additionally, human-caused wildfires make up 85% of the total global number of wildfires, with the ecoregion of *Mediterranean California (MC)* seeing human-caused fires represent 97% of the region's wildfires [13].

2.1.1 Mitigation Techniques

One of the primary tactics to reduce wildfire damage is through firefighting and mitigation techniques. Wildfire prevention, prediction, and suppression are three segments of firefighting that are critical to minimizing the devastating effects of wildfires.

2.1.1.1 Wildfire Prevention

One method of preventing wildfire spread is limiting its ability to ignite and grow. Firefighting agencies and forest services employ various forest management techniques to reduce the severity and area of a wildfire. A primary objective of the US Forest Service is to promote "healthy forests" by reducing highly combustible fuel and promoting climax and old-growth species [14]. Brush removal, or thinning from below, is the most common form of reducing combustible material. Material can be removed in a variety of ways such as controlled burns or mechanical treatments like mowing, scattering, mastication, and lopping. Forest management often includes developing barren fire roads to act as breaks in the forest to prevent larger wildfires from spreading too quickly [15]. These fire roads also enable access to more remote locations of the forest for easier management. The various federal and state forestry management services also ensure that they are not disrupting the biological diversity of the ecosystem, making the task of forest management incredibly difficult. This also includes reforestation efforts in areas harmed by wildfire and excessive logging. These actions allow forestry management services to support the health of our forests while providing sustainable approaches to reducing wildfires.

2.1.1.2 Wildfire Prediction

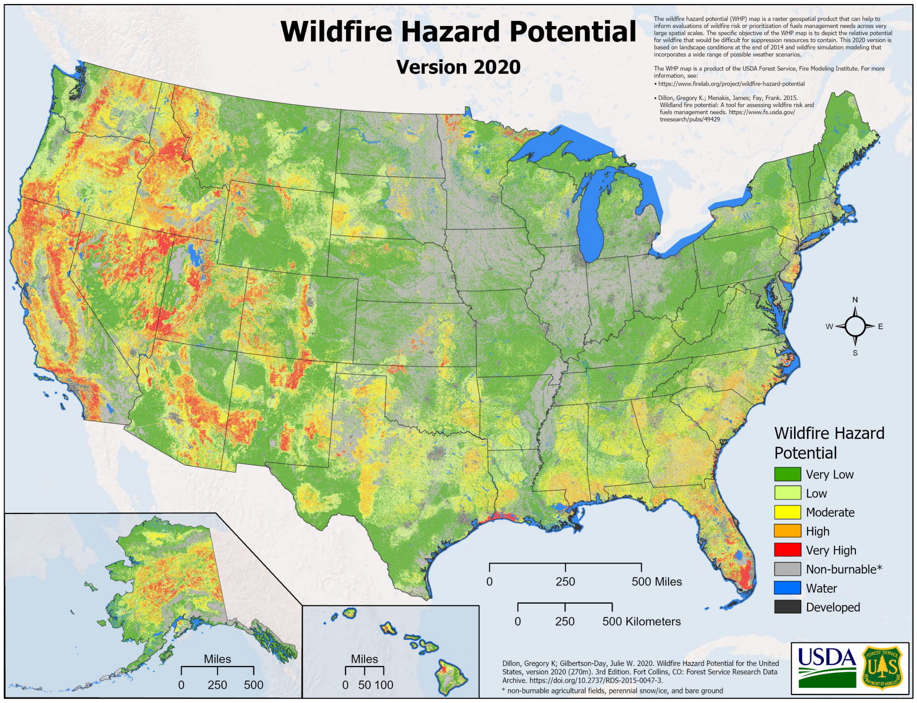
Even with mitigation efforts, wildfire ignition may still occur. A powerful tool in the firefighting arsenal is the ability to predict where fires may occur and the risk they pose. Fire prediction and risk forecasting is a critical sub-field of firefighting where forestry management services use weather, climate, fuels, and personnel assets to estimate risks of given areas as well as the services' preparedness to fight wildfires in different regions [16]. Understanding how risk levels determine the number of resources allocated to different areas covered by the fire service. Resources distributed according to risk assessments can impact response time and give firefighters the best chance to prevent the spread of devastating fires.

Traditional implementations of wildfire risk assessment calculate risk by using a combination of various hazards that increase the probability of wildfires. Hazards include weather, topography, sunlight exposure, vegetation, fuel, and crown fire potential [17]. Risk is also dependent on the Wildland Urban Interface (WUI), which is the area of land where an undeveloped wildland begins to blend with housing and other human structures. These areas and the people occupying them become the most threatened when wildfires occur, therefore understanding risk is critical for safety.

2.1.1.3 Wildfire Suppression

In the event of a wildfire, firefighters must be able to quickly respond to fires in order to have the best chance of containing and extinguishing the fire. Fire spotting, or fire recognition, is a task that firefighting agencies must employ to rapidly detect and confirm wildfires. Large amounts of research are currently being done to provide modern, state-of-the-art image recognition software to these agencies which will help them respond quickly. Computer vision algorithms and neural networks can accurately detect these wildfires with minimal interaction, which is ideal for surveying large amounts of forests remotely. Developments like the use of mounted cameras on fire-watch towers employ these algorithms to detect wildfires many kilometers away with true-positive rates up to 99.7% and true-negative rates up to 99.8% [18]. These modern approaches are enabling firefighters to rapidly and robustly detect wildfires.

Detection, however, is only the first step in suppressing a wildfire. Once the crews are on-site, they must determine the appropriate methods of attacking the wildfire. The Incident Commander will direct various personnel to begin either indirect attacks or direct attacks. Indirect attacks focus on creating barriers that the fire cannot cross. Primarily this will be accomplished by creating firelines by wetting combustible material, clearing material, or burning material in a controlled manner. Fire retardants may also be dispensed to prevent combustion from occurring. While indirect attacks can provide an enormously beneficial defense, fires may still cross firelines due to falling trees, floating embers, or changing wind directions. The Incident Commander may also have crews begin direct attacks. Direct attacks focus on cooling down the fire by dumping water or fire retardant from helicopters and air tankers. Ground crews also establish anchor points near the fire front, clearing combustible material and even removing trees, while spraying the fire from the ground [19]. Once a wild-



The wildfire hazard potential (WHP) map is a raster geospatial product that can help to inform evaluations of wildfire risk or prioritization of fuels management needs across very large spatial scales. The specific objective of the WHP map is to depict the relative potential for wildfire that would be difficult for suppression resources to contain. This 2020 version is based on landscape conditions at the end of 2014 and wildfire simulation modeling that incorporates a wide range of possible weather scenarios.

The WHP map is a product of the USDA Forest Service, Fire Modeling Institute. For more information, see:

- https://www.firelab.org/project/wildfire-hazard-potential
- Dillon, Gregory K.; Menakis, James; Fay, Frank. 2015. Wildland fire potential: A tool for assessing wildfire risk and fuels management needs. https://www.fs.usda.gov/treesearch/pubs/49429

Dillon, Gregory K; Gilbertson-Day, Julie W. 2020. Wildfire Hazard Potential for the United States, version 2020 (270m). 3rd Edition. Fort Collins, CO: Forest Service Research Data Archive. https://doi.org/10.2737/RDS-2015-0047-3.

* non-burnable agricultural fields, perennial snow/ice, and bare ground

Figure 2.1: United States Department of Agriculture Forestry Service Wildfire Hazard Potential (2020) - Classifications Map

fire has been successfully suppressed, firefighting crews must "mop up" the area by extinguishing remaining embers and confirming there are no subsurface fires.

2.2 Traditional Wildfire Prediction Methods

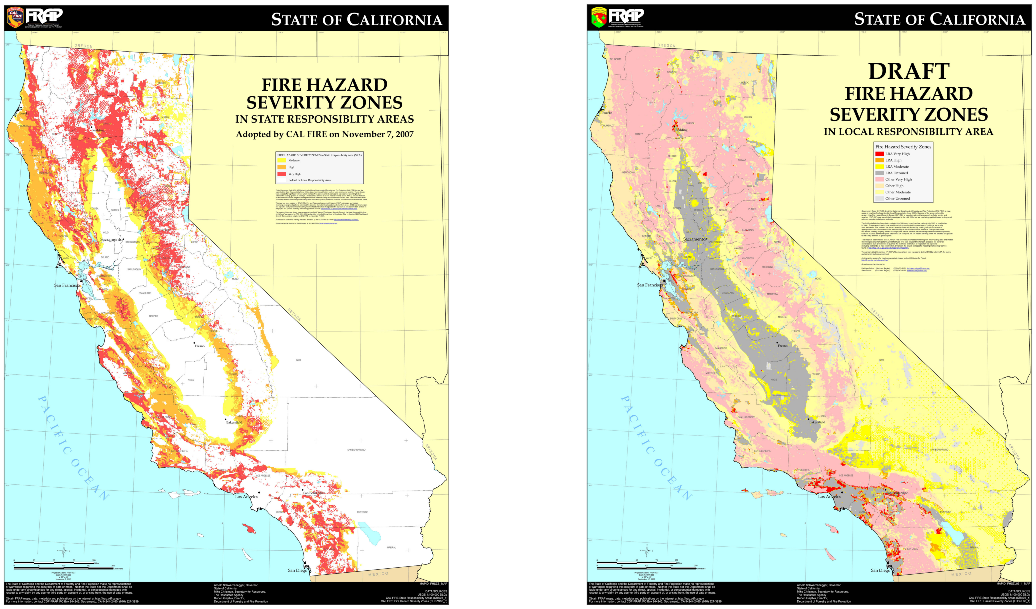
2.2.1 United States Forest Service

Every year the United States Forest Service releases its Wildfire Hazard Potential (WHP) map, Figure 2.1. This map covers the United States with a resolution of 270 meters, classifying the wildfire hazard potential into continuous values, which are often represented as five risk categories: very low, low, moderate, high, and very high [20]. This map, while not inherently a "risk" map, is intended to provide insight into where vegetation is likely to cause intense wildfires and where wildfire prevention methods such as brush removal should be targeted.

The Wildfire Hazard Potential map is derived from geospatial datasets such as LANDFIRE, with the Large Fires Simulator (FSim) quantifying the scores. LANDFIRE is a dataset developed by the U.S. Fire Service and U.S. Department of the Interior which provides geospatial data on vegetation, combustible fuel, fire regimes, and topographical information across the United States [21]. The U.S. Forest Service uses these products in the Large Fires Simulator to generate the annualized, 270 m resolution Wildfire Hazard Potential map. FSim is able to simulate the growth and behavior of hundreds of thousands of wildfires producing a hazard index for each location. FSim is also able to depict the effects of various suppression techniques as well as the effects of various wildfire durations and sizes. This enables FSim to calculate burn probabilities and burn intensities which impact the susceptibility of homes, habitats, and watersheds [22]. While FSim is able to accept weather conditions, the U.S. Forest Service does not include weather predictions as inputs in its annualized Wildfire Hazard Potential map, as it is neither a risk map nor a near-real-time hazard map. The resultant Wildfire Hazard Potential map can then be used by the various federal and state forest services to plan and execute their prevention activities.

2.2.2 CalFire

In California, CalFire similarly develops its own Fire Hazard Severity Zones map, as seen in Figure 2.2, using very similar techniques to U.S. Forest Service. The Fire Hazard Severity Zones maps designate hazard scores for state and local responsibility areas. Hazard scores are generated by taking into account factors such as fire history, combustible fuel, estimated flame lengths, terrain, typical weather, and floating ember potential. However, CalFire's Fire Hazard Severity Zone maps were last updated between 2007 and 2010, leaving much to be improved on with changing climate patterns and modern hazard and threat analysis methods.



(a) State Responsibility Map

(b) Local Responsibility Map

Figure 2.2: CalFire Fire Hazard Severity Zones Map - California (2007)

2.3 Classification

Classification algorithms are a form of problem-solving algorithms where data is grouped into a number of distinct categories called classes. These algorithms may be designed by hand or trained using machine learning and take an input and mathematically apply it across the classification function. This produces a result denoting the class the data is predicted to represent. A common type of classification algorithm is binary classification. Binary classifiers will determine if the input is in the positive or negative class. This is useful for the myriad of problems that can be shaped into simple yes or no questions [23]. Classification can occur in any N-dimensional space, with N representing the number of input features, dividing the space into regions that represent the positive and negative classes.

2.3.1 Logistic Regression

One of the most recognized classification algorithms is Logistic Regression. Logistic regression attempts to fit a sigmoid function to a set of data in order to classify the data into distinct classes [24]. The s-shaped sigmoid function outputs a continuous value between 0 and 1, representing the negative and positive classes respectively. This output is said to be "probability-like," indicating that while it is not a *true* probability, it does represent the model's confidence. The closer the value is to 1, the more likely – or confidently – the value belongs to the positive class. Similarly, the closer the value is to 0, the more likely the value belongs to the negative class. Values near 0.5 signify that the model is not confident about the input's class. This makes logistic regression a powerful classification algorithm for a wide variety of problems. However, logistic regression struggles with higher dimensional data [25]. As the number of features in a dataset grows, the problem becomes increasingly non-linear, making it harder for logistic regression to capture the multicollinearity between the features.

2.3.2 Decision Trees

Decision trees are a type of classifier that relies on cascading rules to determine the class label. Decision trees are incredibly powerful as they employ a training concept called Information Gain [26]. Information gain is the measure of change in entropy or the change in a tree's orderedness. The less entropy – or more ordered – a tree is, the more accurate a prediction will be. This allows the tree to determine which sets of decisions result in the best overall predictive capability. However, like logistic regression, as we increase the number of features, decision trees have a harder time determining the splits that reduce entropy.

2.3.3 Gradient Boosting

Decision trees with gradient boosting are an incredible example of a dimension-resistant model. Gradient boosted trees operate by training trees on the successes and failures of the prior tree. It combines weak learners into strong learners through ensemble learning. Each time a tree is generated, a loss function is applied to determine which decisions were the most correct, and which were the most incorrect. Weights of the incorrect decisions are then changed to attempt to lower the loss. Through successive iterations, this grows the tree into a strong learner [27]. Because of this ability to iteratively generate stronger decision trees, gradient boosting has been incredibly successful and consistently takes top spots in data science applications.

2.3.4 Neural Networks

Outside of logistic regression, Artificial Neural Networks – or ANNs – are one of the most common classification algorithms for wildfire risk prediction. Artificial neural networks are a biology-inspired system that attempts to mimic the basic function of the neurons in our brain. These networks are comprised of nodes that form a layer, and many layers forming a network. A given layer takes input values from the previous layer and outputs values to the next layer, transforming the data using weights and biases attached to each node. As data feeds forward across the network, these transformation operations are performed, outputting a value or set of values on the other end of the network. Depending on the output activation function, this output can be treated as a class probability [28].

Learning occurs as the data then moves backwards across the network, with gradient descent attempting to determine which weights to augment [29]. This process is highly similar to the trimming of synapses in our brain that allows us to form optimal pathways. The more a network is trained, the more the weights and biases in the network will begin to correctly predict the class.

Artificial neural networks have taken the world by storm as computing power grows and continued research advances their development. These classifiers are versatile and powerful and are applied liberally across research papers as the new baseline for machine learning algorithms.

2.3.4.1 Forward Propagation

Neural networks operate in two stages: forward propagation and backward propagation. Forward propagation is the prediction stage where inputs are converted to outputs. No learning occurs in this stage, the output is merely a product of the input and the current weights and biases in the network.

When an input value is passed into the network, the value is applied against a node. This node aggregates and sums all received inputs. The aggregate sum is then multiplied by the node's weight and the node's bias is added. Weights and biases are trainable parameters and are used to modify values as they flow forward through the network. The node then applies this result to the activation function. An activation function helps the model learn complex patterns by determining when to pass a value to the next neuron, and what the scale of the passed value should be. The output is then given as input to the next layer. Each node can output its results to many nodes, similar to how a biological neuron in the brain can attach to many other neurons. This process continues for each hidden layer within the network [30].

As values flow forward through the network, they are modified and scaled, until the final layer is reached. The amount of nodes in the final layer is equivalent to the number of output classes desired. If a numerical or continuous value is being predicted, a linear activation function enables the network to predict a continuous result. For classification problems, softmax, tanh, or sigmoid are often used to predict discrete classes [31]. Softmax, for example, shows a “probability-like” confidence, illustrating which class the network believes the input is most likely to represent.

2.3.4.2 Backward Propagation

Backward propagation is the learning stage when training occurs in the network. After values have flowed forward through the network during forward propagation, the weights and biases must be updated depending on how correct the network’s answers were to the truth.

Backward propagation starts at the output layer of the network and works its way back to the start. A loss function is applied to the predicted value to measure how far off the network was from predicting the true value. The network then employs gradient descent, an iterative optimization algorithm [29], to determine how far the weights and biases should be nudged in order to decrease the value of the loss function. This iterative stepping ensures that the network eventually converges at a local minimum. This is applied to every layer to slowly update the parameters within the network. Ideally, after each backward propagation stage, the network gets closer to predicting the correct value for any given input [32].

2.3.4.3 Dropout & Regularization

A common fault of neural networks is their tendency to overfit. When training a network, the weights and biases are updated according to the input data. This causes the network to predict previously seen data with incredible accuracy, however, any unseen data often fails to reach desired accuracies. As a model overfits, the hyperplane dividing the features becomes increasingly higher-dimensional, signaling that the model is memorizing the data points explicitly rather than learning the implicit patterns [33]. One way to approach reducing the problem of overfitting is to expand the dataset to be larger and more representative of the intended use. Architecturally, dropout and regularizers are two of the most used techniques to reduce the effects of overfitting. The process of generalization by restricting a model's constraints and discouraging complexity is known as regularization.

Dropout is a regularization technique that attempts to mimic the robustness of ensemble learning. Dropout layers can be added to the model architecture and will deactivate neurons during the training phases. These dropout layers slow down the learning of the model and force the network to develop new pathways to yield results. This effectively creates a "new model" during each pass, simulating an ensemble learning model [34].

Regularizers are mathematical operations applied to the cost function to reduce overfitting. L1 or Least Absolute Shrinkage and Selection Operator regression acts as a feature selector by reducing the weight on unimportant features to zero [35]. L1 regression adds a sum of the absolute value of the weights to the cost function. Because this is a non-differentiable piecewise function, it is often slower computationally than L2 regularization. L2 or Ridge regularization instead uses a square of the sum of weights to penalize the cost function and reduce the weights of unimportant features to *near* zero [36]. L2 is more computationally efficient than L1 as it can be solved using matrix operations, a fundamental piece of neural network math. Both regularization techniques prove useful in reducing the overfitting of machine learning models with each being selected for use on a case-by-case basis.

2.4 Convolutional Neural Networks

In recent years, convolutional neural networks have taken the spotlight through their widespread use in visual classification problems and computer vision applications [37], [38], popularized by the success of AlexNet in the 2012 ImageNet Large Scale Visual Recognition Challenge [39]. These networks play a pivotal role in the processing of images, from networks that can classify types of food [40] to networks helping doctors detect melanoma [41].

Convolutional neural networks are distinctly different from previous classification algorithms as they operate on images and capture the image's spatial information, rather than one-dimensional vectors of data, [42]. Convolutional neural networks apply a sliding window filter, referred to as a kernel, to an image. This kernel acts as a feature extractor for the image, determining which combination of patterns results in an important feature. Convolutional layers output another similarly sized image with areas of discriminative information highlighted as larger values. Pooling layers step across the convolutional layer's output image and chooses pixels to include or ignore in the pooling layer's output depending on the pooling layer's function. The most common pooling function is MaxPooling which takes the highest value out of a given area and outputs it to the next layer. For classification problems, the image is often fed through multiple convolutional layers and pooling layers to eventually reach a fully-connected neural network that outputs the class probabilities.

This combination of convolving an image and downsampling is the basic structure of a convolutional neural network. Once the image is sufficiently small, it can be flattened into a vector and applied to a fully connected neural network (FCNN), and output into a class confidence value. A loss function is applied and the data backpropagates using gradient descent, and the weights, biases, and kernels are updated. As this process continues, kernels emerge that are able to extract the most relevant features for the desired task, and the weights of the FCNN are able to convert the flattened vector of relevant features into an increasingly accurate confidence value.

2.5 Satellite Imagery

In 1990, the United States Congress passed the Global Change Research Act [43] which introduced the Global Change Research Program, a program dedicated to studying the changes in our global environment and measuring their effects on society. NASA's Earth Observing System (EOS) contributes to the Global Change Research Program through its fleet of active satellites. With the data collected by these satellites, NASA and scientists around the world can study the impacts of climate change and develop models to predict future effects.

EOS satellites extend across a variety of temporal and spatial resolutions which provides NASA with incredible real-time data. Spatial resolutions define the area of observation. For example, an instrument on a satellite may capture images at a resolution of 250 m, meaning each pixel represents a 250 m by 250 m area. Spatial resolutions may also be measured in degrees (e.g. 0.5°), indicating the latitude and longitude area per pixel. Satellite-based instruments also incorporate a temporal resolution. Temporal resolutions define the time between observing the same area twice. Referring to our example, the instrument may collect an observation above the same geographic area once per day, indicating our instrument has a 250m 1-Day spatiotemporal resolution.

2.5.1 Moderate Resolution Imaging Spectroradiometer (MODIS)

The Moderate Resolution Imaging Spectroradiometer (MODIS) is an instrument aboard the Terra and Aqua EOS satellites. The instrument captures light radiated from Earth as 36 spectral bands, ranging from 0.4 μm to 14.4 μm in wavelength. MODIS is able to record these spectral bands in 250 m, 500 m, and 1 km resolutions depending on the band. MODIS collects data 24/7, sending the images back to NASA in near-real-time. Both Terra and Aqua are positioned in geocentric, sun-synchronous, polar orbits. Terra orbits South to North during its sun-side transit, crossing the equator at 10:30 AM on each orbit. Aqua orbits North to South during its sun-side transit and crosses the equator at 1:30 PM on each orbit [44]. Together, they provide valuable daily snapshots of the entire Earth.

MODIS has a wide variety of products, including surface reflectance, albedo, temperature, leaf cover, vegetation, land cover, thermal anomalies, and others. These daily products enable researchers to study the long-term trends in the Earth's climate. Retrospective studies using MODIS products have the ability to correlate human or natural events with changes in the climate, while prospective studies can use the vast archive of historical data to predict future trends and shifts in the climate.

2.5.2 Google Earth Engine

Google Earth Engine is a research tool available for academic use, providing researchers with API access to petabytes worth of satellite and geospatial data. This data is compiled into massive catalogs stretching back decades, containing image suites like Landsat, Sentinel, MODIS, and others. Google Earth Engine harnesses Google's cloud computing infrastructure to provide rapid computation on large amounts of data.

Google Earth Engine has an online code editor where scripts can be built and run, outputting results to either a console or overlaid on an interactive world map. Datasets can be extracted from Earth Engine Snippets. These snippets reference Earth Engine Images or Earth Engine Image Collections. Each Image or Image Collection has one or many bands which represent the individual data layers on the image. Once a dataset has been retrieved, the user can filter the data, reduce and aggregate the data, apply geographical bounding boxes to the data, and a myriad of other operations. Google Earth Engine also supports a Python library that wraps its API, making all of Google Earth Engine accessible from Python scripts and notebooks.

Chapter 3

RELATED WORKS

Various classification algorithms have had success at utilizing climate data as a predictor for forest fire risk. These classification algorithms are the fundamental building blocks in creating a modern, scalable solution to wildfire risk assessment. In this section, we introduce a few of the most significant research papers in the field, highlight the classification algorithm they used, and discuss the conclusions while noting what lessons can be learned. Through this, we aim to illustrate a sample of the research occurring in this field that impacted the design of this thesis's methods.

3.0.1 Logistic Regression

In the 2016 paper "Learning to predict forest fires with different data mining techniques" by Stojanova et al. [45], a team of researchers from Slovenia and Macedonia designed a dataset to predict wildfires in three regions of Slovenia. The team used satellite data from ALADIN, MODIS, and LANDSAT, as well as elevation and topographical data. The researchers broke these regions down into 1 km by 1km tiles with an 18-day window to collect data and determine whether a fire occurred. The dataset measured between 129 and 159 products depending on the region, including gross primary production, meteorological data, precipitation, wind, solar radiation, evapotranspiration, humidity, canopy cover height, and more. The team then used five different algorithms, including logistic regression, to classify the data by whether a fire occurred in the area or not. While the logistic regression did not beat any of the other algorithms in any of the three regions, it had a strong accuracy of 77%, 83%, and 84% and recall score of 56%, 85%, and 85% for the three regions. These regression accuracies were only between 1% and 4% off from the strongest classifiers in each of the regions. The researchers, however, were unable to discern which of the myriad of features played the most important role, leaving significant feature analysis to be completed in future works. The study demonstrates just how effective a simple regression model can be in comparison to stronger models.

The paper highlights several important factors. First, the breadth of products used in the dataset likely capture a wide array of climate and topographical scenarios that may be linked to wildfires. Having such a wide breadth of features ensures there are plenty of patterns for the algorithms to discover. Second, the logistic regression model is shown to be a strong competitor to other more advanced algorithms such as bagging and boosting algorithms.

3.0.2 Decision Trees and Feature Importance

In the paper "Fire Prediction Based on CatBoost Algorithm" by Zhou et al. [46], a team of researchers analyzed climate, geographical, and human features to determine forest fire risk. The team first uses a feature selection method based on Gradient Boosting Decision Trees, finding that land surface temperature was the strongest predictor of wildfires, followed by infrared index, distance from roads, wind speeds, topographical elevation data, and vegetative index, with the rest of the features trailing significantly behind. The team then used Principal Component Analysis (PCA) to modify the dataset to maximize the variance between important features and reduce the weight of unimportant features [47]. These steps reduced the feature set by 13 features while keeping 99% of the information power of the dataset. The paper goes on to use gradient boosting to grow trees sequentially using the classification error of the previous tree to inform the next tree on how to predict the revised residuals. This process can yield incredibly accurate results. The researchers saw such success, demonstrating an accuracy of 83% and precision of 82%. These results illustrate the powerful ability of binary classifiers and gradient boosted decision trees.

The feature importance table which described the importance index of wildfire impact factors is incredibly valuable. Many of the wildfire impact factors listed such as land surface temperature, elevation, vegetative index, and water vapor pressure (being related to humidity) are retrieved for the MODIS dataset generated in this thesis.

3.0.3 Artificial Neural Networks

Many wildfire prediction papers have used artificial neural networks to predict risk assessment. One paper published in 2009 by Maeda et al. titled "Predicting forest fire in the Brazilian Amazon using MODIS imagery and artificial neural networks" [48] paved the way for researchers attempting to classify wildfire occurrences given historical climate data from MODIS. The researchers used the Normalized Difference Vegetation Index (NDVI) 16-day composite images for two weeks in April, two weeks in May, and one week in August as data vectors to be analyzed by the network in order to predict wildfires. They tested nine artificial neural network architectures that used one input layer, one hidden layer, and one output layer, testing a hidden layer neuron count of 4, 6, 8, 10, 12, 14, 16, 18, and 20. These networks were then tested on five geographical situations, such as areas that burned one year and not the next, areas that burned and converted to agricultural areas the next, etc. The team concluded with accuracies in the various situations ranging from 78% to 87%, determining the 14-neuron hidden layer to output the smallest mean-squared-error when classifying forest fires.

Several years later, the 2012 paper "Application of artificial neural networks and logistic regression to the prediction of forest fire danger in Galicia using MODIS data" by Bisquert et al. attempted a similar neural network-based approach to wildfire prediction [49]. The paper solidifies the usefulness of artificial neural networks in out-competing other algorithms such as logistic regression. Using an ablation study, the researchers were also able to analyze which features of MODIS data were most important in determining whether a fire would occur. The study found that the artificial neural network produced the highest accuracy and precision with the input combinations of fire history, land surface temperature, and period of the year. The model was able to correctly predict whether a fire occurred with 76% accuracy and had a precision score of 66%.

Artificial neural networks are continuously proven to be effective when determining wildfires across a variety of datasets. This strong performance makes a good baseline to compare future models against. Additionally, the feature analysis completed in the 2012 paper paves the way for continued refinement of MODIS datasets for wildfire prediction use.

3.0.4 Convolutional Neural Networks

One of the only papers discussing convolutional neural networks and their application to wildfire risk assessment using MODIS data is the 2021 paper "Forest Fire Risk Prediction from Satellite Data with Convolutional Neural Network" by Santopaolo et al. [50]. The paper examined several features, including normalized difference vegetation index, normalized difference water index, accumulated rainfall, land surface temperature, land cover type, elevation, as well as the previous year's fire mask. The team used a spatial resolution of 1 km and a temporal resolution of eight days. The model was comprised of six convolutional layers with a 3x3 kernel and ReLU activation function with each layer being followed by a 2x2 stride max-pooling layer. The final layer output an image using the sigmoid activation function. This predicted image represented an estimated fire mask. To validate the results, the researchers used the pixel-wise mean-squared error to determine loss. The researchers analyzed two geographic locations, the island of Sicily and the Los Angeles Area. While the overall results appear very good, the model performed much more effectively in the Los Angeles area. The researchers theorize that this is due to Sicily's lack of vegetative area which was a driving feature in the model. The model operating on Sicily resulted in a loss of 0.7, whereas the loss for the Los Angeles area is 0.002.

This paper is foundational in defining the research space of neural network-developed fire masks. The authors found a clear and convincing approach to building high-resolution fire masks. The strong link between land cover, elevation, vegetative index, and wildfire risk was yet again proven by these findings. While the predictive fire mask approach is very useful for precision mapping of potential wildfires, a classification approach is still necessary in order to create a broader, region-based risk map.

Chapter 4

SYSTEM DESIGN

4.1 Problem Definition

The classifiers discussed above have significant importance to this research. Many of the listed studies paved the way in determining the most significant MODIS features to help predict wildfire risk. They demonstrate the viability of a host of different classification algorithms in tackling the growing problem of wildfires. Through all of the previous research, we still believed there were gaps to fill. We focused on expanding the breadth of wildfire prediction research occurring around convolutional neural networks, as Santapolo's [50] approach to wildfire prediction utilizing convolutional neural networks was one of the very few making headways in wildfire risk assessment.

We took key features from this paper and extrapolated the area of interest to all of California, rather than the Los Angeles area, and train the network to be more resilient to differences in vegetation indices by expanding the MODIS products. Additionally, we trained the model output to a class probability rather than a raster as the intention is to estimate region-based wildfire risk rather than estimate a burned area mask. The goal is that this would generalize better, allowing the model to be used broadly as a wildfire predictor.

4.1.1 Convolutional Neural Network Advantages

The decision to utilize convolutional neural networks stems from their unique ability to detect patterns in two-dimensional space. As discussed previously, wildfires are largely a product of their physical environment, therefore the representation of each climate feature as a two-dimensional environment allows for the relationships and patterns within the region to be examined. Non-convolutional classification algorithms are unable to achieve these benefits, therefore, convolutional neural networks enable us to reach conclusions that have not been found through previous research.

4.1.2 Goals

There are three primary goals of this research. The first is to build a dataset of multi-layered climate images, labeled by whether a burn has occurred in the imaged region during the following weeks after the climate data was recorded. No previous research attempt appeared to utilize a dataset with such expansive climate and topographical images, offering a new perspective on this task. This addresses shortcomings in previous research attempts where less encompassing datasets are used. Additionally, the ability to generate datasets for future reproduction efforts is critical, therefore we released the dataset creation code which can be found linked in Appendix A. This dataset creation code can then be used to validate the findings in this study or can be used to generate datasets in other geographic locations for wildfire prediction attempts outside of this area of study.

The second goal is to generate a never-before-seen model which utilizes a convolutional neural network to predict whether a wildfire will occur in an area given this specific set of climate information. The model will output a confidence value that can be associated with either a positive (burned) or negative (unburned) prediction. The closer the confidence value is to the label, the higher the “risk” of the predicted label. Together, these two objectives will provide useful information for furthering the research surrounding wildfire prediction.

The third and final goal is to determine which features are most important to the dataset by performing an ablation study. This study would place similar features into groups which are then successively removed from the dataset. The effect of removing these groups from the dataset will give us an indication of whether the features helped or hindered the overall performance. Through this study, we can help determine which climate features may warrant use in future studies.

4.2 Tile Building

Before any climate data was retrieved, we determined how the data should be represented. Since convolutional neural networks accept statically sized images as their inputs, it can be asserted that the images collected must all be of equal sizes. This means the intended area of study must be divided into uniformly sized tiles. However, because the spherical surface of the Earth must be projected onto a two-dimensional plane, there will be mild stretching effects in order to fit the regions into flat images. Moreover, since longitudinal lines are relatively closer together towards the Earth's poles, the farther the tile is from the equator, the less relative area will be collected inside those tiles.

For simplicity and consistency in representing geographic areas, boundaries will use the geographic coordinate system (longitude and latitude) to define the area of study. Each tile's side length will be 0.1° in order to capture a small yet informative area. If the tile side length was too small, we risk losing the two-dimensional patterns of the land area studied, and if the side length was too large, it will not be useful in determining precise locations where a wildfire is likely to occur. The side length of 0.1° allows for enough topographical information per tile while still being discrete enough to provide useful, fine-grained results on where forest service efforts should be directed.

4.2.1 Spatiotemporal Region of Interest

Previous studies have analyzed the geographic regions of Galatia [49], Sicily, Los Angeles [50], Slovenia [45], and many other areas impacted by seasonal wildfires. Given the recent increase in attention to California's wildfires, this dataset represents only California and Nevada. There appears to be no other major study utilizing MODIS climate data for wildfire prediction in this area, making this a pioneering study for both the novelty of the dataset, as well as the application of the convolutional neural network.

The bounding box stretches from 32.8°N, 124.2°W to 42.0°N, 114.4°W. These coordinates define the southwestern and northeastern points respectively. This region entirely captures the state of California and, due to California's shape, almost captures the entirety of Nevada. Given the decided side length of 0.1°, we conclude that there will be 92 tiles east-west, and 98 tiles north-south yielding 9016 total tiles. A tile can then be solely represented by the coordinates of its southeastern corner, as it is known that the side length will extend 0.1°to the north and west. For reference, each of these 0.1°x0.1°tiles is approximately 9.9 km wide by 11.1 km tall at their widest in the southern-most tiles, and 8.3 km wide by 11.1 km tall at their narrowest in the northern-most tiles, or roughly 100 km^2. This size allows there to be sufficient climate information in each tile, while simultaneously representing enough locations to provide targeted, regional wildfire risk information.

For the temporal resolution, we collected data from California's peak fire season. Since the majority of wildfires, often including the largest wildfires, occur during peak wildfire season, it best represents the most critical time to predict wildfires [51]. Additionally, restricting the climate data to a tighter range of months may help prevent potential underfitting due to large differences in the climate data between winter months and summer months. California's wildfire season ranges from approximately early summer through the beginning of fall, therefore the months of July through October were collected when building the dataset. The wildfire seasons of 2015 through 2020 were sampled to produce a dataset most similar to California's current climate.

4.2.2 Filtering Unnecessary Tiles

Due to California's shape, the bounding box selected covers areas of the Pacific Ocean. As wildfires do not occur over the ocean, these tiles should be removed from consideration. Eliminating the unnecessary tiles will help reduce the size of the dataset and may help boost model performance by removing trivial data that can skew weights unexpectedly.

In order to identify desirable tiles, we must determine whether a tile is a "land tile" or a "water tile." The MODIS Water Mask product is a global mask utilizing the Shuttle Radar Topography Mission's Water Body dataset and 250 m resolution MODIS data. Using Google Earth Engine, we retrieved the mask information for each of the 9016 generated tiles. The water mask is an Earth Engine Image consisting of 1's where a pixel is water and 0's where the pixel is land. Given a tile's bounding box coordinate, we can generate statistics about the region using Google Earth Engine's *ee.Image.ReduceRegion()* function. This function accepts the water mask image, the region of interest, and a reducer. To quickly determine the proportion of the region that is water, we can pass the *ee.Reducer.mean()* reducer operation to determine the average pixel value over the region. Therefore, the examined region will reduce to a value between 0 and 1, representing the proportion of water a region contains.

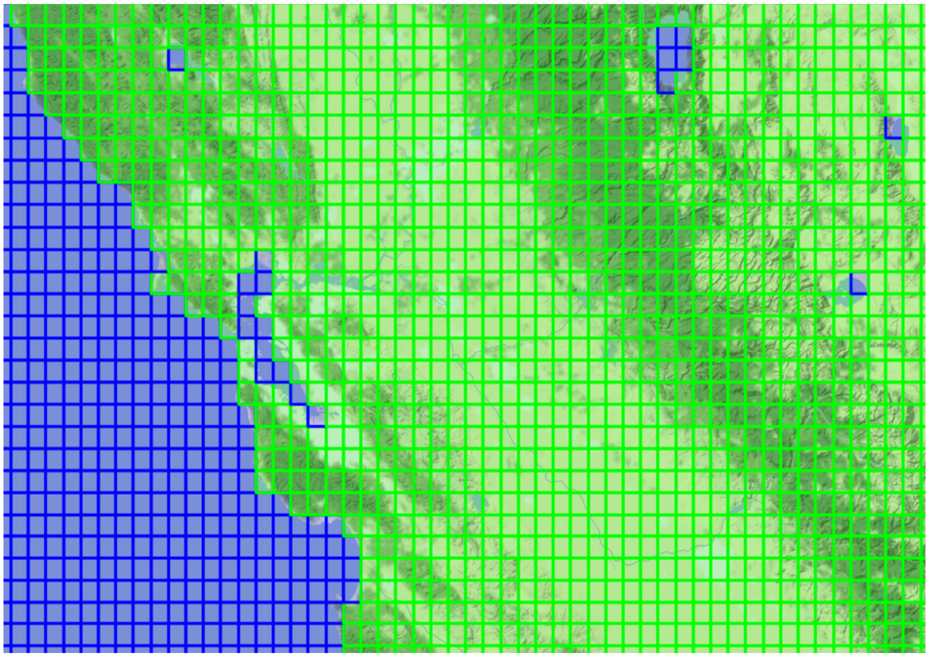


Figure 4.1: Land (green) and water (blue) tiles over the San Francisco Bay Area

The dataset creation tool automatically saves a newline-separated file of tile coordinates and water proportions. A default threshold of 50% water was selected to eliminate a tile. Thus, all tiles with less than 50% water are considered valid land tiles. This reduces the number of tiles from 9016 to 6910 tiles, or approximately a 23% decrease in tiles. Strictly land tiles are used during the rest of the dataset creation. Land tiles and water tiles can be seen in Figure 4.1, highlighting areas such as oceans, lakes, and bays that will be excluded from the dataset.

4.2.3 Determining Wildfire History

Once the desired tiles were selected by the Water Mask product, we labeled the land tiles with whether they were observed to contain a wildfire or not contain a wildfire during a given time frame. The dates ranging from July through October of 2015 through 2020 were divided into one-week intervals. July through October generates 18 one-week intervals, multiplied over six years resulting in 108 date intervals. Applying

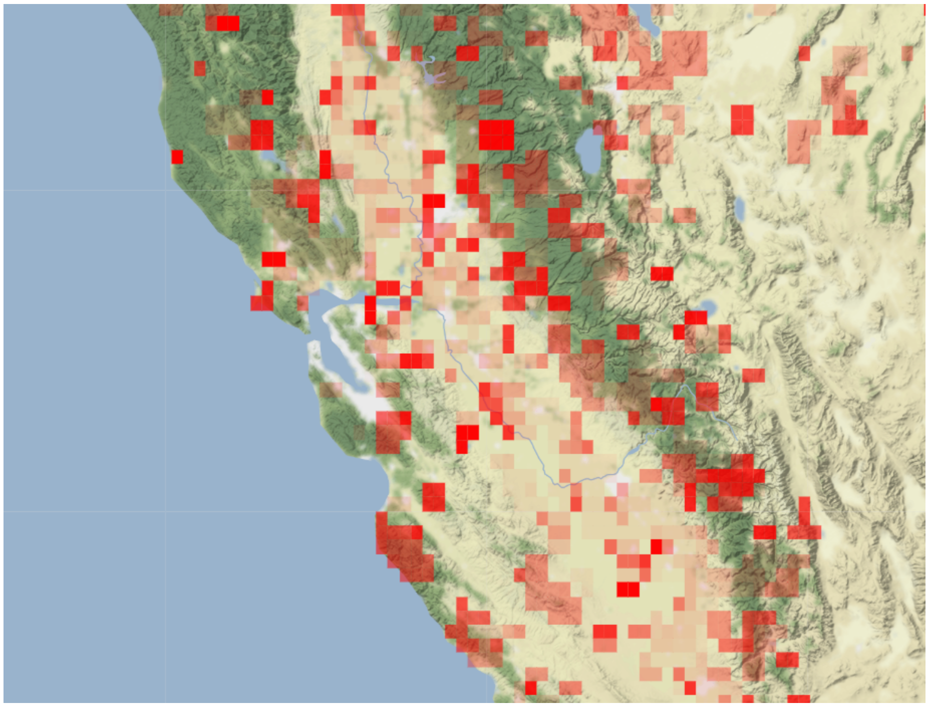


Figure 4.2: Fire history of Northern California where recent fires are more opaque (2015-2020)

these 108 date intervals to the 6910 tiles yields almost 750,000 unique possible data points.

The MODIS Thermal Anomalies & Fire product is a global, daily, 1 km resolution mask that provides an image of the Earth where the wildfires are estimated to have occurred over a time frame. The mask consists of pixels of value 0 or 1, representing whether there was not or was a wildfire. Reducing a region with *ee.Reducer.sum()* returns the total number of pixels within the area that were estimated to contain a wildfire at some point during the time frame measured. Should an area contain a number of wildfire pixels greater than a set tolerance threshold, the area is deemed to have burned. Since there is no research regarding appropriate threshold pixel values for estimating wildfires as opposed to other types of fires such as building fires or bonfires, we selected a threshold greater than 0 pixels to constitute a wildfire incident.

In order to make the data transferable and expandable, we store each of the 108 date intervals in their own JSON files. Each file contains a two-dimensional dictionary where the keys are strings of the longitude and latitude coordinates of the south-western corner of the tile, the values are null or a string of the date interval (e.g. json[”-117.6”][”33.9”] = ”2015-07-01”), and the files are named with their date interval (e.g. `firedata-YYYY-mm-dd.json`). With this information, fire data JSON files can be trivially combined into a large, singular dictionary. The combined dictionary contains the same keys with the values being represented as lists of dates when a fire occurred on that tile. A tiled map of wildfire history weighted by recency can be seen in Figure 4.2. In this image, wildfire incidents that are more recent are more opaque. Tiles are also stacked cumulatively if a wildfire occurred on that tile multiple times. This creates an interesting visualization of wildfires as a product of both recency and frequency.

4.3 Dataset Creation

Before data collection began, we decided how an individual data point would be stored. Convolutional neural networks accept image-like formats. Each piece of data must have a static width, height, and depth. Width and height define the area of the image, while the third dimension, depth, defines the channels of an image. Normally, the channels of an image are RGB, representing the red, green, and blue channels. However, convolutional neural networks can accept any static depth. This opens the door for a unique use case where the depth component of the data is distinct climate bands over the same land area, rather than color channels making up the image. Thus, the width and height components encompass the 0.1° by 0.1° tile, and the depth will define the individual climate layers across that geospatial area. This enables there to be many climate, topographical, and historical images packed into a single piece of data.

Each layer is a 64 px by 64 px image, with each pixel being stored as a single byte. This yields exactly 4 kilobytes per layer. There will be 32 unique climate layers which will be discussed below. Therefore, each data point is 64 kilobytes. Given that there are roughly 750,000 data points, the size of the entire dataset would be approximately 98 gigabytes. Due to its prevalence in data science and compatibility with Keras, NumPy will be used to format and save each data point. Data points will be stored based on their class label. After retrieving and stacking the climate layers, the multi-channel image will be stored as a three-dimensional `.npy` file. These files are binary files that reduce the storage overhead and provide fast read capabilities through the NumPy library.

4.3.1 Climate Layers

There are ten Google Earth engine snippets that comprise the 32 climate, topographical, and historical bands. These layers provide insight into the climate of the region during the date interval examined. Since the date interval associated with the class label indicates the date that there either was a wildfire or was not a wildfire, climate data must be collected across a date range prior. This has two main benefits: first, it prevents the existence of a wildfire from altering the climate conditions, and second, it enables the climate data to signal a future wildfire event. Because several of the products in Table 4.1 have a temporal resolution of 16 days, a date range of 17 days was selected to fully capture the array of selected bands and will begin 17 days before the start of the Thermal Anomaly & Fire detection window.

Table 4.1: Dataset products by name, snippet, band codes, and spatiotemporal resolution

Climate, Topological, and Historical Products			
Product Name	**Google Earth Engine Snippet**	**Band Code**	**Resolution**
Land Surface Temperature	MODIS/006/MOD11A2	[’LST_Day_1km’]	1km 8-Day
Leaf Area Index	MODIS/006/MCD15A3H	[’Fpar’, ’Lai’]	500m 4-Day
Gross Primary Productivity	MODIS/006/MOD17A2H	[’Gpp’]	500m 8-Day
Evapo-transpiration	MODIS/006/MOD16A2	[’ET’, ’LE’, ’PET’, ’PLE’]	250m 16-Day
Vegetation Index	MODIS/006/MOD13Q1	[’EVI’]	250m 16-Day
Land Cover	MODIS/006/MCD12Q1	[’LC_Type1 (0-16)’]	500m Yearly
Vapor	NCEP_RE/surface_wv	[’pr_wtr’]	2.5° 6-Hr
Elevation	CGIAR/SRTM90_V4	[’elevation’]	90m
Precipitation	UCSB-CHG/CHIRPS/ DAILY	[’precipitation’]	0.5° Daily — 6-Month Summed
Fire History	FIRMS	[’T21’]	1km near-real-time — 0-1yr, 1-5yr, 5-10yr

To download the data, each product is first filtered to the climate date window. For most products, this results in an *ee.ImageCollection* object. ImageCollections requires an aggregate operation to reduce them to a single *ee.Image*. The *median()* aggregator takes the median pixel value of all Images in the ImageCollection across the filtered range and reduces them to a single Image. Each band is then scaled to 0-255 using the min and max band values found on the product’s description page. The image is then downloaded as a 64x64 png by retrieving the thumbnail image using Google Earth Engine’s *ee.Image.getThumbURL()*. The returned image, a multi-channel png, must be converted to a single-channel, gray-scale image using OpenCV’s *cv.cvtColor()*.

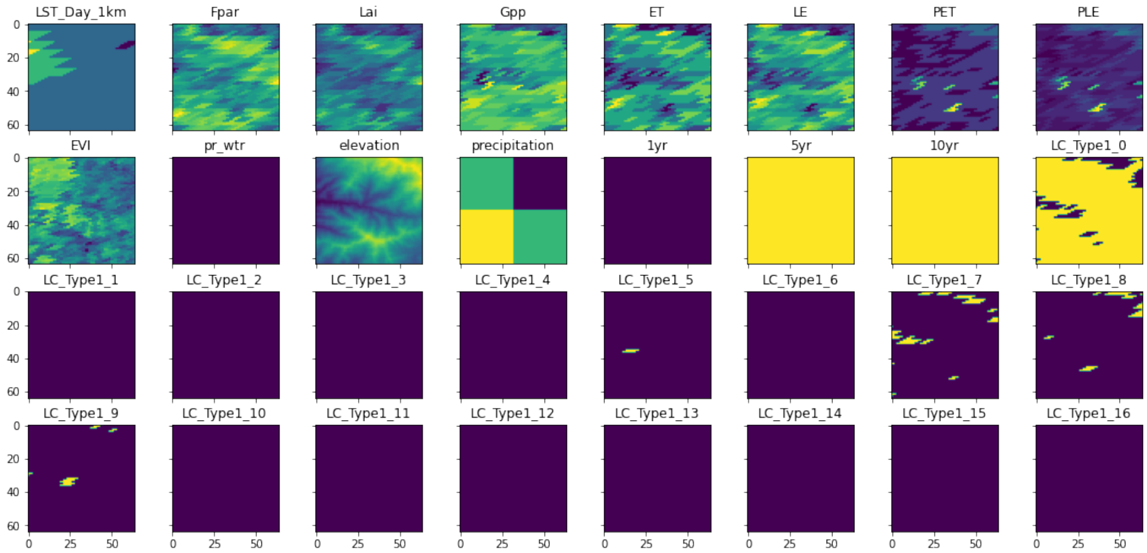


Figure 4.3: An example of a burned multi-channel climate image separated into their distinct climate layers

This results in the desired 64x64x1 image that can be stacked on the other climate layers to create the 32-channel final image.

Several products require slightly more derivation to generate the desired layers. Precipitation data is collected across the previous six months leading up to the start of the burn date. Each pixel value in the region is averaged across the entire window and scaled from 0-255. The layer is then downloaded and converted to gray-scale like the other climate layers. This process captures a snapshot of the potential moisture content of the soil and vegetation. From this, the model can make inferences about the patterns and relationships between the spring and early summer rains and the potential for wildfire.

Additionally, three fire history layers are defined. These layers use the same Thermal Anomalies & Fire product that is used to generate the class labels. These fire history layers are analyzed across varying windows of the ten previous years. The first layer measures whether a wildfire occurred in the one year leading up to the class label date interval. The second layer measures whether a wildfire occurred during the date window spanning from five years prior to one year prior. The third fire history layer measures whether a wildfire occurred during the date window spanning from ten years prior to five years prior. If a wildfire is detected during any of these windows, the entire layer is filled with 1s, if there is no wildfire detected, the layer is filled with 0s. The goal of these layers is to enable the model to determine whether the area is prone to recurring wildfires, whether the area has burned recently and therefore may be unlikely to burn again soon, or whether an area has not burned indicating it may not be prone to burning.

An example of a burned image is shown in Figure 4.3, with each layer being displayed individually. Due to how Python's Matplotlib displays images, some layers may appear blank, but do contain climate data. Matplotlib will set the minimum value of the image as the darkest color. For example, *precipitation* can only have 4 unique values due to its spatial resolution, this causes the lowest value, regardless of the true numeric value, to be a dark segment of the displayed image. Additionally, *1yr*, *5yr*, and *10yr* fire history bands contain either a 0 (black) if there was no wildfire during that period, or a 1 (yellow) if there was a wildfire during that period. The 17 land cover type layers are binary representations of where that land cover occurs over the region.

4.3.2 Burned vs Unburned Split

When retrieving a thumbnail image through Google Earth Engine's API, there is a slight processing time while the URL is generated. There is also a delay while fetching the image from the URL. Due to there being nearly 750,000 data points and 16 product retrievals per data point, there are approximately 12 million calls to the Google Earth Engine API. This significantly slows down the data collection process, presenting a large bottleneck.

Downloading data points by naively sampling the set of possible tiles and date intervals results in a lopsided dataset, with the proportion of burned to unburned data being approximately 1:99. Instead, data was downloaded by first separating the burned tiles from the unburned tiles and sampling each set equally. This generates burned class data more quickly than naively sampling. This also produces a dataset that has a 1:1 split, allowing the model to have more burned data to train. Data can be downloaded equally from each class until the desired size of the dataset is reached, or the positive class is entirely sampled. In our case, we downloaded approximately 6000 examples, 3000 from each class. We also downloaded an additional 100 positive examples and 10,000 negative examples to form the testing dataset. This allows us to test on real-world conditions with the correct data balance.

4.4 Convolutional Neural Network Architecture

Due to the established size of the data, developing the architecture for the neural network was fairly straightforward. The model is based on the VGG16 architecture developed by Karen Simonyan and Andrew Zisserman for the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The network (as seen in Figure 4.4) uses five blocks, the first two blocks repeat two 3x3 convolutional layers and a 3x3 max-pooling layer, and the final three blocks repeat three 3x3 convolutional layers and a 3x3 max pooling layer, all followed by a flattening layer and two fully-connected dense layers.

Because VGG16 was designed to classify 1000 classes, it does not perform well on this binary classification task. Therefore, a smaller model had to be designed to accommodate the data. This model, shown in Figure 4.5, is based on the VGG16 architecture and employs a similar five-block approach. This smaller model, however,

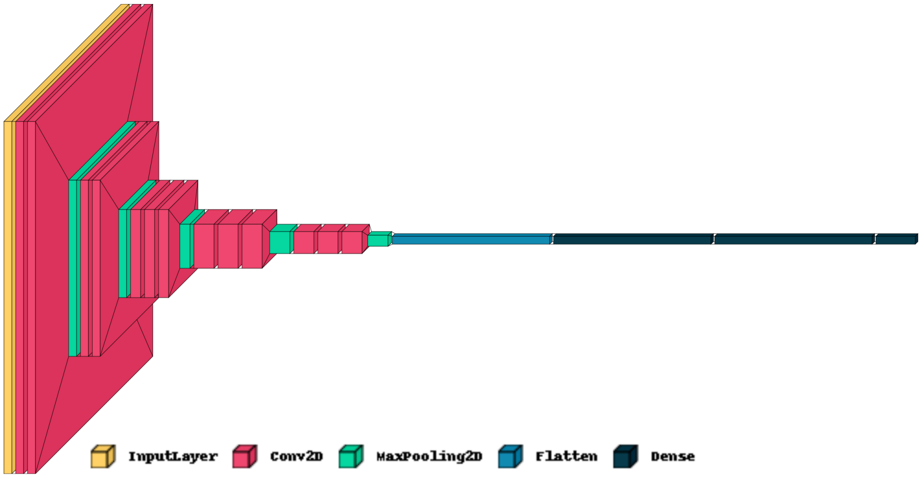


Figure 4.4: Visualization of the VGG16 convolutional neural network architecture

does not use double convolutional layers like VGG16, and also includes two dropout layers to reduce overfitting. Each block is a 3x3 convolutional layer followed by a max-pooling layer, with the two dropout layers positioned after the second and fourth blocks. The model finishes with a flattening layer and two dense layers, outputting the confidence in the two classes.

The ReLU activation function is used on all convolutional layers to mitigate vanishing gradients as well as decrease training time. The convolutional layers use 32, 64, 128, 128, and 256 filters respectively. This enables the model to first identify the patterns in the input layers, then apply more abstract concepts to the emergent patterns. By increasing the number of filters, the network can recognize the growing number of possibilities and differences in the data. Then, the network is flattened into a fully-connected neural network and outputs the prediction through a softmax activation function. Softmax is chosen as the output activation function because it provides the "probability-like" distribution for the network's prediction. In total, this results in 839,906 trainable parameters within the model.

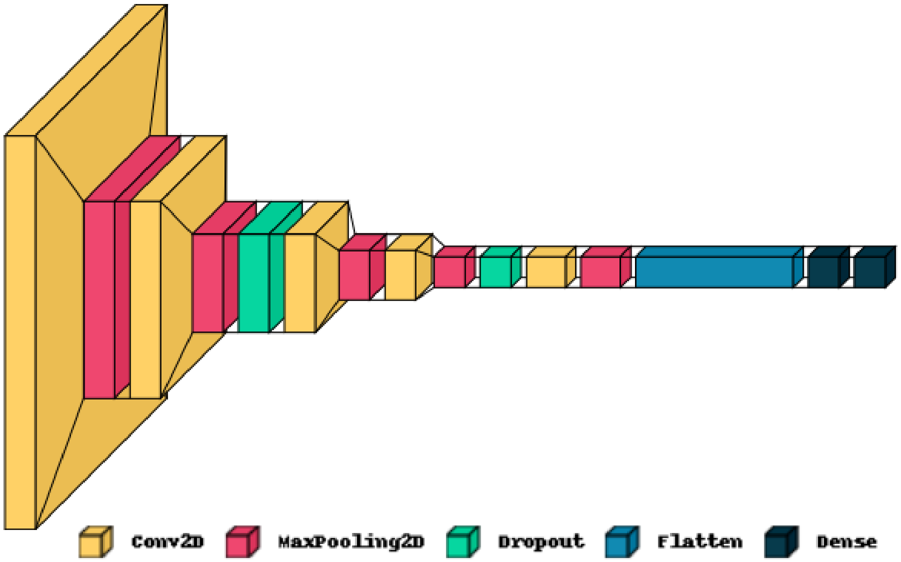


Figure 4.5: Visualization of the utilized convolutional neural network architecture

Chapter 5

RESULTS

5.1 Evaluation Criteria

To define the results of the model, Accuracy, Loss, F1-Score, Precision, and Recall, as well as various graphical derivatives, will be used as the primary performance metrics for evaluation. It is important to look at multiple metrics, as accuracy does not paint a complete picture of the model's performance.

True positives are the number of times the model correctly identified the positive class. True negatives are the number of times the model correctly identified the negative class. False positives are the number of times the model incorrectly predicts the input to be in the positive class. False negatives are the number of times the model incorrectly predicts the input to be in the negative class. These four values can be used to create F1-score, precision, and recall, which help unlock more information about how the model performs.

Accuracy is normally the first metric analyzed as it is, in many cases, a good predictor of how correct the model is. Accuracy tells us what proportion of all predictions were of the correct class. Accuracy is defined as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}}$$

However, accuracy does not indicate where a model may be insufficient. Thus, precision and recall are used to further determine how the model is performing. Precision is the measure of how correct a model is in its positive predictions. Precision is a useful metric to analyze when the cost of a false positive is high. Imagine the case where fingerprint matching software is used in crime analysis. If the software incorrectly matches two fingerprints, it may lead to an innocent person being wrongfully accused or convicted of a crime. In such a case, high precision is critical to ensuring that positive predictions only occur when the value is in fact positive. Minimizing the false positives rate maximizes the precision score. Precision is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{\text{True Positives}}{\text{Total Predicted Positives}}$$

However, should the model almost never predict the positive class unless it was absolutely certain, one may be erroneously led to believe the model is performing well. In actuality, it may be the case that the model is incorrectly selecting the negative class the majority of the time to increase precision. While this may be optimal in the previous example, we can imagine the case where a doctor is trying to identify people with a highly contagious sickness. There is a very large cost if the doctor incorrectly predicts the patient was healthy, or a negative case, when the patient was actually sick, or a positive case, as the sickness could harm the individual without treatment and rapidly spread to many more people. The recall score represents the proportion of true positives predicted compared to the total number of true positives. Minimizing the number of false negatives maximizes the recall score. Recall is defined as:

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

Since both of these scores are needed to have an accurate view of a model's performance, a third metric is used to combine precision and recall into a single score. The F1-score can be used as a representation of the overall power of a model. The F1-score is defined as:

$$\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

In wildfire risk assessment, recall is a more important metric than precision. False negatives are extremely harmful when predicting if an area is likely to burn, therefore a high recall score is the first objective. Should an area be incorrectly thought to be safe from wildfires, lives, homes, and fragile ecosystems may be lost due to slow response times, preparedness, and equipment allocations. However, if an area is incorrectly thought to be at risk to burn but does not burn, minimal value is lost. Therefore, the model's recall score will be primarily examined with respect to acceptable accuracy, precision, and F1-score.

5.2 Base Results

The convolutional neural network was trained with the novel satellite image dataset. This dataset contained 6,000 images with each image being comprised of the 32 climate channels. 3,000 images were from the positive burned class, and 3,000 images were from the negative unburned class. This results in a 50% class-based split, differing vastly from the true distribution where the unburned class represents roughly 99% of the examples and the burned class is roughly 1% of the examples. Therefore, the dataset had to be rebalanced in order to aid the network in discovering the patterns of both classes.

Separately, 100 positive class images and 10,000 negative class images were collected for use in a test dataset to evaluate the model through various performance metrics. This resulted in a test set that is similar to the true proportion of positive and negative data found in the real world. We augmented the training set by rotating the images by 90°, 180°, and 270°. This resulting dataset was then also copied and vertically

Table 5.1: Convolutional neural network model performance metrics on the novel image-based climate dataset

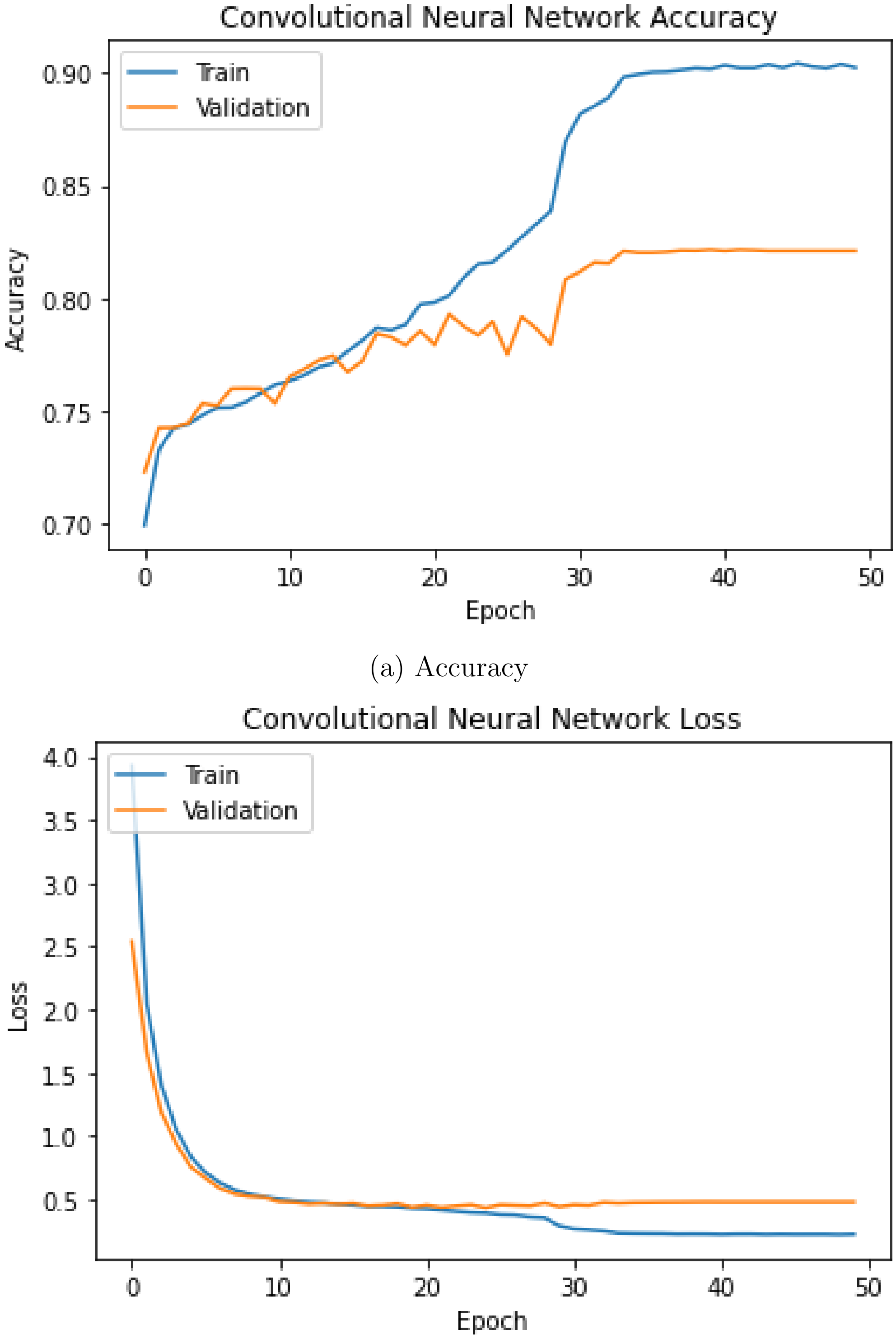
Performance Metrics - Base CNN			
Accuracy	**F1-Score**	**Precision**	**Recall**
0.734	0.056	0.029	0.800

flipped, yielding an eight times increase in the amount of data, or 48,000 total images. We use a 20% validation split for training, leaving the model with 38,400 images to train on.

The network was trained across 50 epochs, with the highest scoring validation accuracy model being selected. Figure 5.1 depicts the model's validation accuracy and validation loss during training. The model's accuracy jumps above 70% after the first epoch and slowly climbs a few percent higher throughout the next 25 epochs, indicating the model quickly made inferences about the difference between the classes and was then able to find slightly more nuanced patterns.

The validation loss drops to about 0.5 before completely leveling off, while the training loss dives to about 0.25, with their divergence indicating the model begins to overfit to the training data during the second half of training. To help the model make more minute adjustments as training progressed, the learning rate was reduced each time the loss plateaued for four epochs. This leads the model to slow down any drastic changes and settle into a comfortable validation accuracy and loss.

Table 5.1 shows the performance metrics achieved by the model. Notably, the model produces a recall score of 0.800, indicating that it identified 80% of all positive cases in the testing set. The model results in a precision score of 0.029, indicating that of all positive predictions, 2.9% of them were true positives. These recall and precision scores result in an F1-score of 0.056, illustrating the overall power of the model.



(a) Accuracy

(b) Loss

Figure 5.1: Convolutional neural network model accuracy and loss over the 50 epochs of training

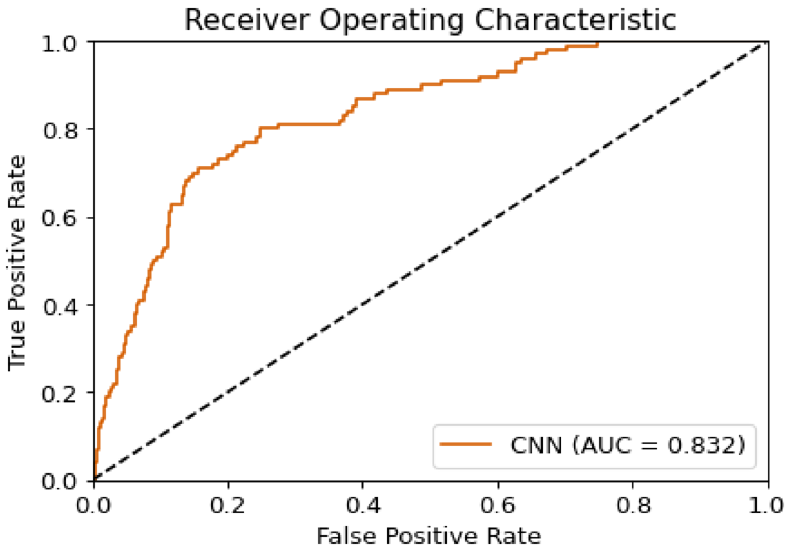


Figure 5.2: Convolutional neural network model Receiver Operating Characteristic curve

Performance metrics can be also represented as a Receiver Operating Characteristic (ROC) curve. ROC curves define the relationship between the model's false positive rate against the true positive rate at various decision thresholds. Ideally, the ROC curve hugs the top left corner of the graph and gives the largest area under the curve. This would signify that the model has a very high true positive rate while maintaining a low false positive rate. Most classification models tend to fall somewhere between the top left corner of the graph and the identity line bisecting the graph. Models can be compared both visually to determine where the ideal true positive and false positive points are, and numerically by comparing the value for the area under the curve.

Figure 5.2 graphs the convolutional neural network's Receiver Operating Characteristic curve. The curve boasts an area under the curve (AUC) score of 0.832, indicating the model has a good discriminative ability between true positives and false positives.

In addition to Receiver Operating Characteristic curves, the Precision-Recall curve can be used to plot the precision scores against the recall score as the decision thresh-

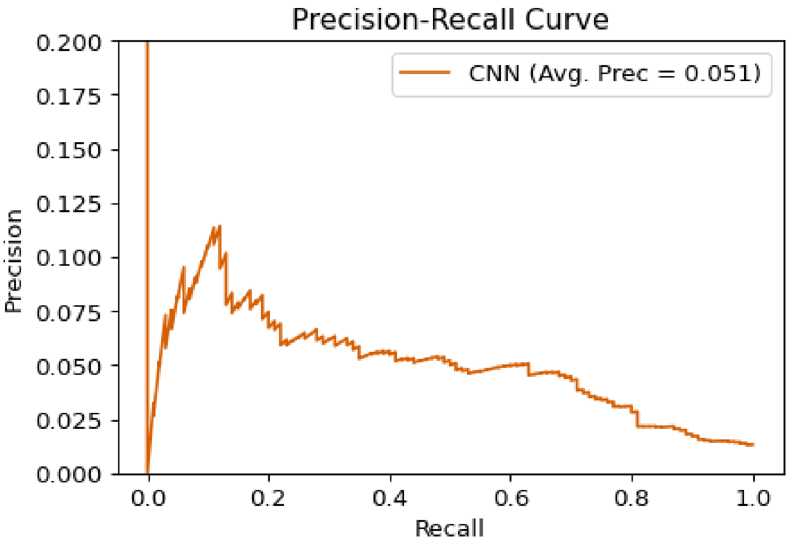


Figure 5.3: Convolutional neural network model Precision-Recall curve

old changes. Precision-Recall curves are another way to view the performance of a model in a slightly different way than ROC curves. Where ROC curves allude to precision and recall through their representation of true positive rate and false positive rate, Precision-Recall curves directly depict the two performance scores. Figure 5.3 shows this curve for the proposed convolutional neural network model. The closer the curve is to the top right corner, the better a model is performing as it is able to score both high precision and a high recall. Across all recall values, the precision-recall curve generates an average precision score of 0.051. This precision is slightly higher than the precision score measured for the base performance as the average precision is raised by the model's higher precision at the lower recall values.

5.3 Tuning Recall

As mentioned previously, the recall score depicts the model's ability to capture the positive class. A higher recall score indicates that the model is classifying a larger proportion of these positive examples. As depicted in the Precision-Recall curve, we

Table 5.2: Performance metrics of the convolutional neural network model when the recall score is tuned to 0.900

Performance Metrics - Recall Tuned to 0.900			
Accuracy	**F1-Score**	**Precision**	**Recall**
0.517	0.036	0.018	0.900

can modify the confidence threshold at which the model predicts the positive class. This causes a trade-off between precision and recall. As we increase the recall value, we decrease the precision.

A firefighting agency or forest service might request the performance metrics of the model when capturing a specific percentage of the positive class. An example of this would be an agency that wants a model that is able to predict 90% of all wildfires. The agency would then determine the success of the model through the F1-score and precision. In this example, we can tune the recall score to 0.900 by modifying the classification threshold and analyzing the resulting performance metrics. Table 5.2 shows the performance of the model at this static recall score. The accuracy of the model drops to 51.7% indicating that a smaller proportion of all predictions belong to the correct class. The precision score also drops to 0.018, signaling that of all the positive predictions, only 1.8% of them are true positives. The F1-score also decreases to 0.036, indicating that overall the model is weaker at predicting the positive class at this tuned recall score.

5.4 Comparison Results

To prove the model's effectiveness we compare this new convolutional neural network approach to previous research. We utilize the methods architecture of Bisquert [49] where the researchers applied both a logistic regression model and an artificial neural network model to predict wildfire risk in Galacia, Spain. The study used MODIS data, comprising of features like fire history, enhanced vegetation index, land surface temperature, period of the year, as well as several combined features to predict whether an area was at risk to burn. As mentioned by Santopaolo in their comparison of wildfires in Sicily and Los Angeles, the geographical region of study plays an important part in determining wildfire prediction performance [50]. Therefore, we are unable to directly compare our results to other research attempts as they analyze different regions.

To compare previous models to the convolutional neural network approach, the previous models are applied to the new dataset. This both tests the validity of the novel satellite image dataset and the power of the convolutional approach. First, a simple logistic regression model was developed that takes the 32 features as inputs and outputs two classes using softmax. This logistic regression model contains 66 trainable parameters. Additionally, four single-layer perceptron artificial neural network models were designed. The study only provided that their artificial neural networks had a single hidden layer, but did not discuss the number of neurons in that layer. However, this may not be relevant as the number of inputs into the network will differ from the original paper's architecture, as this novel dataset does not contain the same amount of inputs. This would then affect how the data is processed by the network, resulting in an entirely different structure. The four artificial neural networks used in this comparison contain 32, 64, 128, and 256 nodes in the single hidden layer respectively. This results in 1122, 2242, 4482, and 8962 trainable parameters for the artificial neural networks. Using four different architectures provides a broad approach that captures various network complexities.

Table 5.3: Performance metrics of all comparison models and the convolutional neural network model

Comparison Models Performance Metrics				
Model	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Logistic Regression	0.835	0.060	0.032	0.530
Artificial Neural Network 32 Neurons	0.975	0.046	0.037	0.060
Artificial Neural Network 64 Neurons	0.946	0.078	0.047	0.230
Artificial Neural Network 128 Neurons	0.960	0.069	0.045	0.150
Artificial Neural Network 256 Neurons	0.756	0.054	0.028	0.710
Convolutional Neural Network	0.734	0.056	0.029	0.800

Because each of the comparison models accepts a one-dimensional array as input in, the satellite image dataset is flattened to fit this shape. Each image contains 32 two-dimensional climate layers stacked on top of each other and must be reduced into an array of 32 floats. This can be trivially accomplished by taking the average value for each two-dimensional climate layer. Since taking the average negates the spatial resolution of the dataset, data augmentation was not applied to the logistic regression and artificial neural network models. Original, flipped, or rotated images would result in identical examples when taking the average of the two-dimensional space. This flattened data is then used to train each of the comparison models.

The relevant performance metrics for all models discussed in this section can be found in Table 5.3. When looking at accuracy alone, it appears that the 32-neuron artificial neural network has the highest accuracy at 97.5%. However, because we utilize the real-world, imbalanced dataset, it may be trivial for the models to retain a high accuracy by strictly picking the negative class. Given the approximately 1% positive case rate in the testing dataset, should the model pick the negative class on every

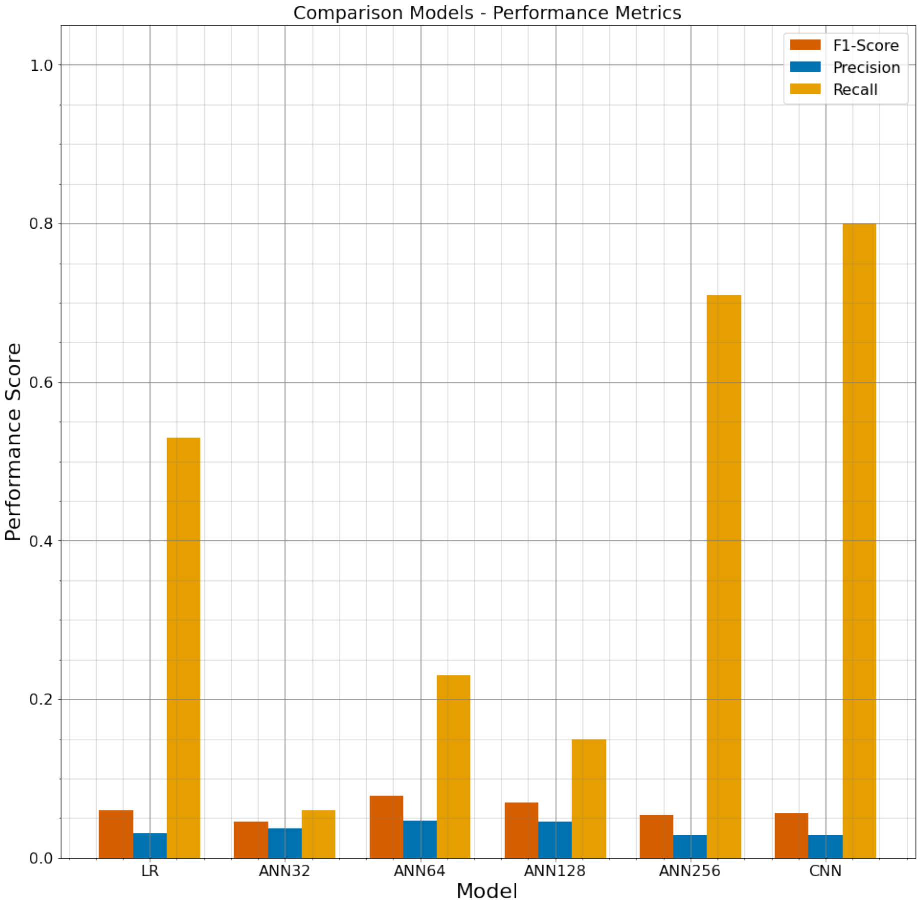


Figure 5.4: Bar chart of the performance metrics of all comparison models and the convolutional neural network model

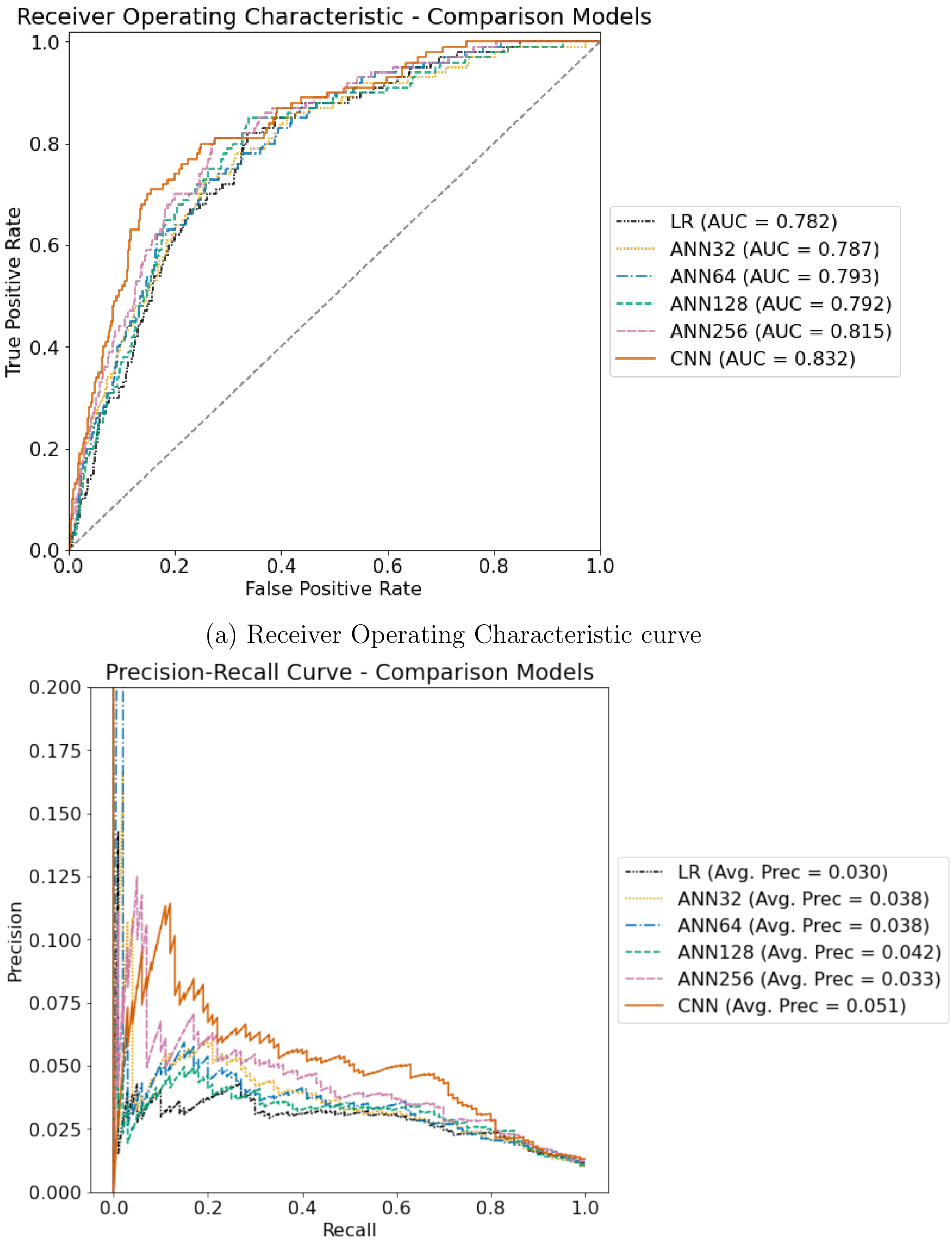
prediction, the model would achieve a roughly 99% accuracy. Therefore, it is more useful to look at the relative F1-scores which combine precision and recall, telling us which model has the highest overall power in predicting the positive class.

When looking at F1-score, we can see that the 64-neuron model has the highest F1-score at 0.078, generated from a precision of 0.047 and recall of 0.230. This indicates that the 64-neuron artificial neural network is able to identify 23% of the positive examples in the testing dataset with each positive prediction made by the model having a 4.7% likelihood of being a true positive. The convolutional neural network has the fourth-highest F1-score at 0.056. The convolutional neural network has a precision score of 0.029 and recall of 0.800, indicating that the network is able to identify 80% of the positive examples in the testing dataset with each positive prediction made by the model having a 2.9% likelihood of being a true positive.

These results are also depicted visually in Figure 5.4.

Figure 5.5 shows the ROC curve and Precision-Recall curve for the five comparison models and the new convolutional neural network model. The convolutional neural network scores the highest area under the curve, beating out the 256-neuron artificial neural network by 1.7%, and the 64-neuron neural network by 3.9%. We can see that in the lower false positive rates the convolutional neural network produces a slightly higher true positive rate than the other models. As the false positive rate increases, the models roughly equal out. This increased performance in the lower false positive rates helps bolster the overall holistic score of the area under the curve when compared to the other models.

This analysis is reflected in the Precision-Recall curve with the convolutional neural network scoring the highest average precision across all recall values at a score of 0.051. This is 0.9% higher than the 128-neuron model and 1.3% higher than the 64



(a) Receiver Operating Characteristic curve

(b) Precision-Recall curve

Figure 5.5: Comparison models performance metric curves

Table 5.4: Performance metrics of all comparison models and the convolutional neural network model when the recall score is tuned to 0.900

Comparison Models Performance Metrics - Recall Tuned to 0.900				
Model	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Logistic Regression	0.455	0.032	0.016	0.900
Artificial Neural Network 32 Neurons	0.471	0.033	0.017	0.900
Artificial Neural Network 64 Neurons	0.503	0.035	0.018	0.900
Artificial Neural Network 128 Neurons	0.498	0.034	0.017	0.900
Artificial Neural Network 256 Neurons	0.516	0.035	0.018	0.900
Convolutional Neural Network	0.517	0.036	0.018	0.900

and 32-neuron models. We can see that the convolutional neural network performs betters than the other models at some of the lower recall scores, such as the recall values between 0.1 and 0.2, indicating the convolutional neural network is better at classifying positive classes confidently at these lower recall scores. As the recall score increases the models begin to converge, indicating similar precision performance, especially after the recall score of 0.8.

We also compare the results when tuning the recall score of each model to 90%. This would illustrate the performance of each model in the hypothetical scenario mentioned above. The agency could then pick which model performed best and suited their wildfire prediction need. The results of this scenario are shown in Table 5.4. With the recall scores tuned to 0.900, the models all perform roughly the same. The highest F1-score is the convolutional neural network at 0.036, however, this is only 0.1% higher than the 64 and 256-neuron models, and at most 0.3% higher than the logistic regression model, suggesting that at this recall score there is minimal

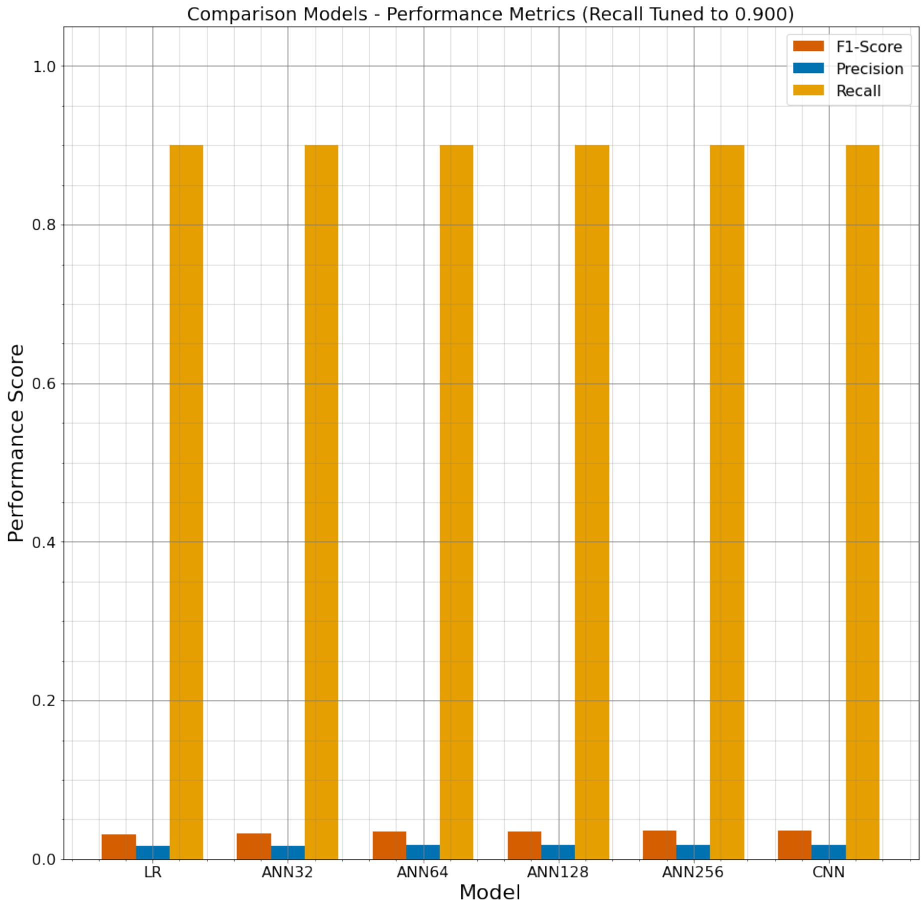


Figure 5.6: Bar chart of the performance metrics of all comparison models and the convolutional neural network model when the recall score is tuned to 0.900

difference between the models' ability to correctly predict the positive class. These results are also depicted visually in Figure 5.6.

Overall, the convolutional neural network underperforms compared to various other models when inspecting the base performance on the 50% class confidence threshold. When analyzing the receiver operating characteristic and precision-recall curves, the convolutional neural network outperforms the other models, retaining a higher area under the curve and average precision. This indicates the model has slightly stronger holistic scores than the other comparison models. Finally, the convolutional neural network marginally outperforms the other models when tuning the recall score, however, this difference is negligible and does not signify any meaningful difference between the models.

Once again, it is important to note that these models are tested using the real-world proportion of positive to negative data of approximately 1% positive to 99% negative. When we define the testing set to use a 50% positive to 50% negative split of 100 positive and 100 negative examples, the results become less skewed towards the lower precision values.

Table 5.5 shows the performance metrics on the 50% class confidence threshold for each of the models when using the 50% positive to 50% negative example testing set. While this testing set is not applicable to real-world conditions, it does help us interpret how these models are performing. In this experiment, the convolutional neural network yields the highest F1-score, at 0.762, as well as the highest recall at 0.800, indicating that the model was able to find patterns in the data. This demonstrates that the results seen in Table 5.3 and Table 5.4 are using some found inference to predict the positive class, however, because the imbalance is so large, false positives greatly distort the precision and F1-scores.

Table 5.5: Performance metrics of all comparison models and the convolutional neural network model when using a test set containing 50% positive and 50% negative examples

Comparison Models Performance Metrics - 50:50 Testing Data				
Model	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Logistic Regression	0.670	0.616	0.736	0.530
Artificial Neural Network 32 Neurons	0.525	0.112	0.857	0.060
Artificial Neural Network 64 Neurons	0.605	0.368	0.920	0.230
Artificial Neural Network 128 Neurons	0.545	0.248	0.714	0.150
Artificial Neural Network 256 Neurons	0.720	0.717	0.724	0.710
Convolutional Neural Network	0.750	0.762	0.727	0.800

5.5 Feature Analysis

Because the dataset brings in so many new features, it is necessary to determine which features have the largest effect on predicting wildfires. A simple ablation study was conducted to successively remove groupings of similar layers and measure the model's performance metrics. The grouped layers can be seen in Table 5.6. Groups are sorted in order of which group will be removed at each step. Evapotranspiration is the first group to be removed in the ablation study, followed by land information, water content, fire history, and finally vegetation.

Results are derived from the original convolutional neural network with no classification threshold modification. Each of the ablation models was trained across 50 epochs, and the epoch with the highest validation accuracy was selected. Results of the ablation study can be seen in Table 5.7.

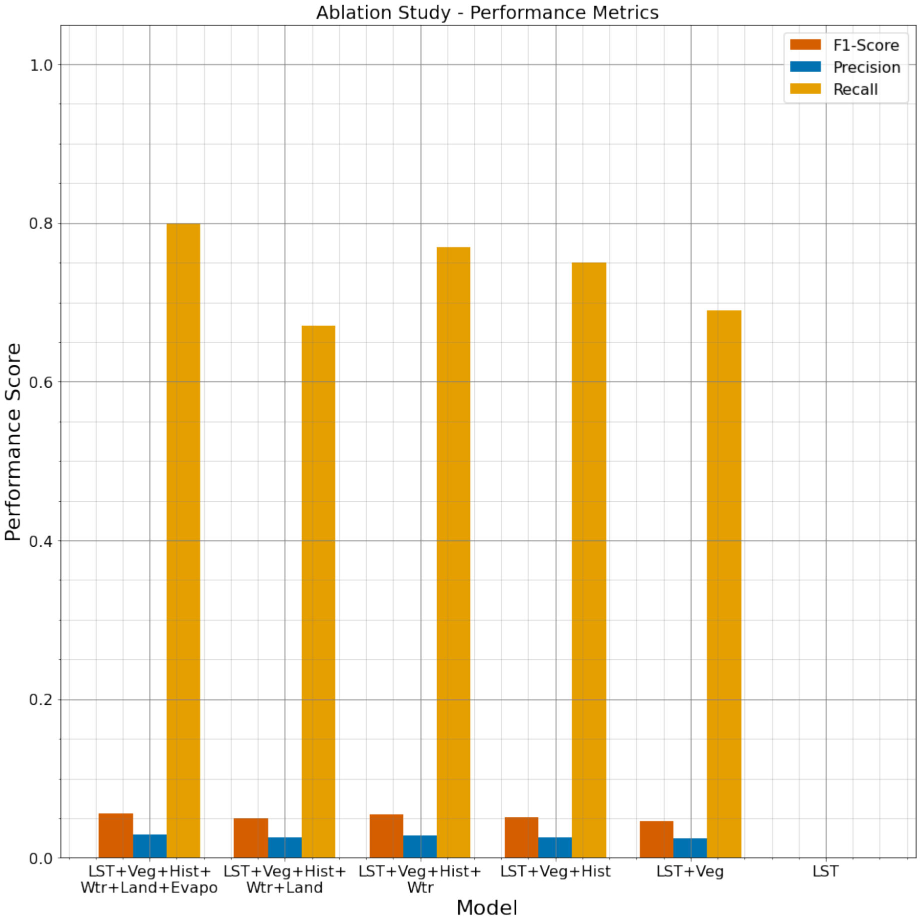


Figure 5.7: Bar chart of the feature analysis ablation study performance metrics using the convolutional neural network model

Table 5.6: Climate layer feature groups removed in the feature analysis ablation study

Feature Groups For Ablation Study	
Group Name	**Climate Layers**
Evapotranspiration	ET, LE, PET, PLE
Land Information	LC_Types, elevation
Water Content	pr_wtr, precipitation
Fire History	1yr, 5yr, 10yr
Vegetation	Fpar, Lai, Gpp, EVI
Land Surface Temperature	LST_Day_1km

The five-group model scored the highest accuracy at 74.7%, 1.3% higher than the six-group model's 73.4%. However, the five-group model has a 0.6% lower F1-score, indicating that the six-group model is overall more powerful than the five-group model. This implies that evapotranspiration products are beneficial in predicting where wildfires will occur.

There is a decrease in the accuracy (-1.1%) when removing land information products. This results in the four-group model achieving an accuracy of 73.6%. The recall and precision scores both increase slightly, with the F1-score of the four-group model being 0.5% higher than the five-group model. This indicates that the removal of land information products increases the overall power of the model when predicting the positive class.

When removing water content products, both the accuracy and F1-score decrease, indicating that products such as six-month summed precipitation and columnar water vapor positively affect the model's ability to predict wildfires. The three-group model is overall less powerful than the four-group model across all metrics.

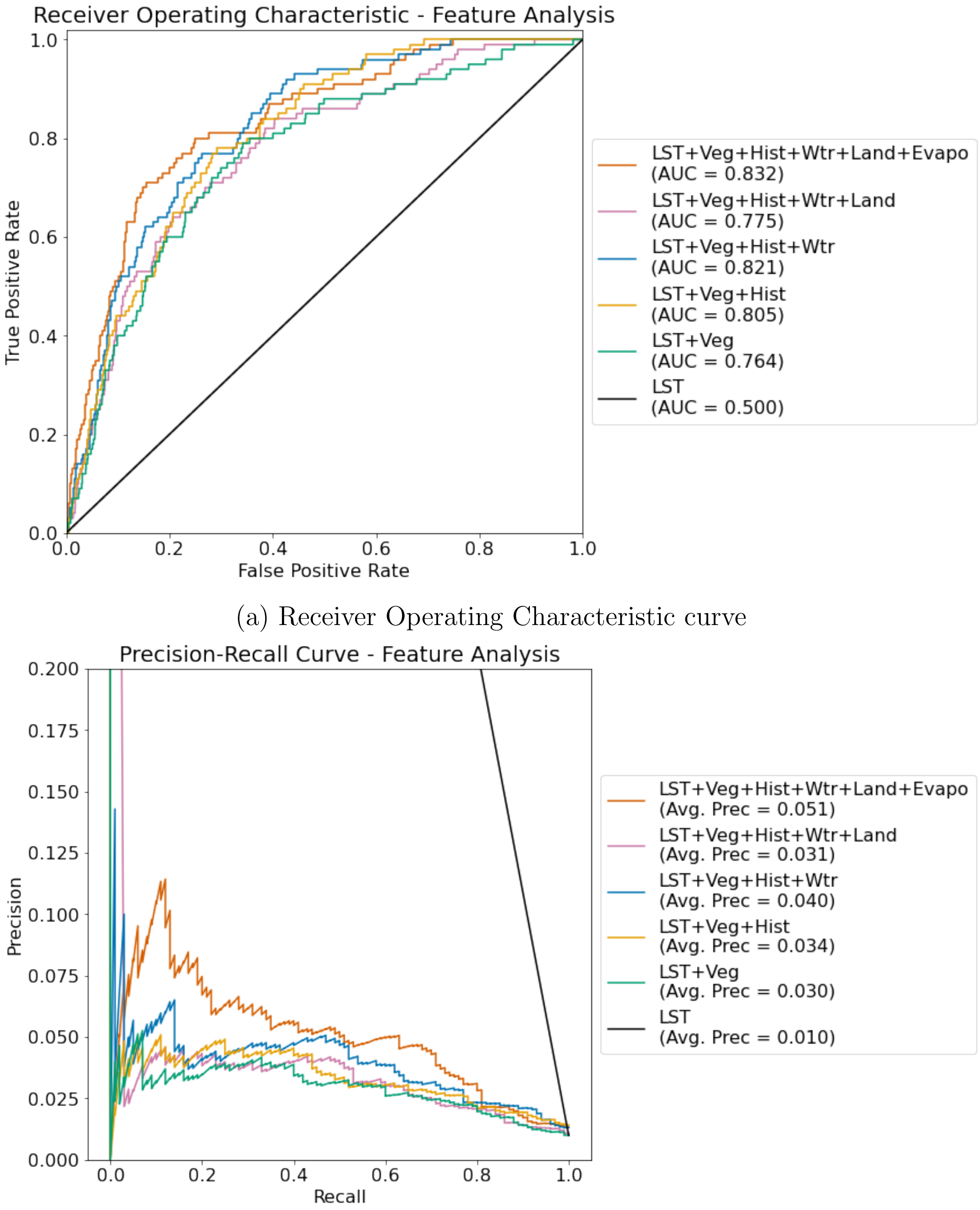
Table 5.7: Feature analysis ablation study performance metrics using the convolutional neural network model

Feature Analysis Ablation Study Performance Metrics Table					
Group Count	**Groups**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
6	LST+Veg+Hist+ Wtr+Land+Evapo	0.734	0.056	0.029	0.800
5	LST+Veg+Hist+ Wtr+Land	0.747	0.050	0.026	0.670
4	LST+Veg+Hist+ Wtr	0.736	0.055	0.028	0.770
3	LST+Veg+Hist	0.722	0.051	0.026	0.750
2	LST+Veg	0.723	0.047	0.024	0.690
1	LST	0.990	0.000	0.000	0.000

There is a decrease in precision and recall resulting in a 0.4% decrease in F1-score when removing fire history products. This two-group model performs worse than the three-group model, indicating that fire history products are useful in determining where a wildfire will occur.

Finally, when removing vegetation products, the one-group model of just land surface temperature is unable to predict the positive class. The accuracy is 99% as the model will only ever predict negative. With the imbalanced testing data, this results in a 99% accuracy, but also a score of 0 in F1-score, precision, and recall. These results are also depicted visually in Figure 5.7.

Figure 5.8 shows the ROC and Precision-Recall curves for each of the models. The six-group model scores the highest area under the curve at 83.2%. When removing evapotranspiration products, the area under the curve drops to 77.5%. The three-group and four-group models break above 80% area under the curve. The two-group



(a) Receiver Operating Characteristic curve

(b) Precision-Recall curve

Figure 5.8: Feature analysis ablation study performance metric curves

model containing just land surface temperature and vegetation products has an area under the curve of 76.4%, while the one-group model becomes the identity line at 50% area under the curve. These results are consistent with the analysis from the performance metrics table above.

These results are mirrored in the precision-recall curve with the six-group model having the highest average precision, while the rest of the models follow the analysis and traits described above.

Evapotranspiration products, water content products, history layers, and vegetation products all result in increases in predictive ability. Land cover products cause a decrease in predictive ability. The impact of land surface temperature could not be determined as it was the last feature of the ablation study. Overall, it appears that the six-group model is the most successful in predicting where wildfires will occur.

Chapter 6

DISCUSSION

6.1 Summary

This paper has several major accomplishments which expand the breadth of research surrounding wildfire risk modeling. The first accomplishment was developing an entirely new image-based dataset that captured a comprehensive set of climate features. The publicly available dataset creation tool (Appendix A) provides a simple way to segment geographical areas which are then parseable by machine learning models. This dataset creation tool manages the division and collection of climate data over geographic and temporal areas. It stacks the collected climate layers into a single three-dimensional image, splits the data into its respective burned and unburned classes, and saves the geographic location and date information in the file name. This creates an incredibly useful and expandable dataset that could even be applied outside the scope of wildfire research.

The second accomplishment was developing a convolutional neural network model that is novel to the research space. While the developed model was competitive with the models proposed by previous approaches and wielded the highest holistic measures such as area under the curve and average precision, it ultimately did not outperform other models by meaningful amounts. Ideally, the network would have been able to determine stronger spatial patterns within the image which could have enhanced the predictive ability of the model. This lack of increase may have been because there was not enough data that the model was trained on and expanding the dataset could further boost the performance by finding more subtle patterns. It also may have been because there are simply minimal significant patterns in the two-dimensional space that determines whether an area would burn. The analyzed tile sizes were 0.1° by 0.1° or approximately 9 km by 9 km. This area may be too small to discern more meaningful patterns. Increasing the area of each tile to capture broader patterns would reduce the usefulness of small, granular results, causing a potential trade-off between performance and resolution. However, the convolutional approach is a distinctly unique attempt and warrants valid consideration for further studies and applications.

Additionally, while the F1 and precision scores appear low, they are consistent with the results measured by Gholami [52], where a team of researchers found average precision values between 0.002 and 0.013 for the Nilgiris and Sathyamangalam regions of India when testing on imbalanced data. The Uttara Kannada region had average precision scores of 0.32 to 0.4 while testing on imbalanced data, again suggesting that geographic location plays a large role in the performance of the model, as well as that low precision values may be expected. This signifies that the results achieved in our study are within the bounds of results produced by previous studies.

The final accomplishment was the dataset feature analysis using the developed convolutional neural network. This feature analysis consisted of an ablation study that removed successive groupings of similar layers. This ablation study illustrated that land cover information products like land cover type and elevation had an overall negative impact on the model's performance. Evapotranspiration products, water content products, and historical fire data appeared to give the model a boost in performance. Vegetation data such as leaf area index (LAI), fraction of photosynthetically active radiation (FPAR), gross primary product (GPP), and enhanced vegetative index (EVI) demonstrated the largest increase in performance, indicating those products are crucial predictors of wildfire. These findings may be useful in additional studies which can continue to refine the dataset and determine the set of optimal products to include in wildfire detection.

6.2 Technical Challenges

6.2.1 The Dataset

One of the largest obstacles of this project was designing and downloading the dataset. First, determining which layers should be included was an ever-growing task. At countless turns, it seemed as though bringing in one additional layer may cover specific cases and boost performance, leading to the continuous pressure of feature-creep around the dataset. Similar difficult design decisions also arose through determining time-frames for how to retrieve the data. For instance, the decision to collect wildfire data over one week versus two weeks, or collecting precipitation data across the previous 6 months, rather than beginning at a fixed date. These decisions accumulated to steer the direction of the final dataset, resulting in a unique product that could have been designed in countless other ways.

The second obstacle surrounding the dataset was the retrieval of the images in a fast, reliable manner. Initially, a different third-party library was used to retrieve images, however, it was significantly slower than the current approach and was extremely limited in the products offered. A switch to Google Earth Engine was quickly made in order to mitigate some of these issues arising. However, Google Earth Engine's Python API still didn't offer ideal solutions for retrieving large quantities of images. Data was collected using image thumbnail URLs, as the size of the images requested was suitable to that endpoint. This endpoint is not traditionally intended to be used for data collection purposes, so a better, more robust solution should be examined for further replications. The thumbnail URLs also had to be individually processed for each climate layer across every tile, resulting in lots of small tasks which likely wasted computing resources. Expanding the geographic area of the downloaded climate image and slicing it into tiles after downloading the image may result in faster processing speeds.

6.2.2 The Network

Optimizing neural networks can be a very arduous task. Throughout this development and research for this paper, the network was continuously tweaked to squeeze out every last drop of performance. First, a network architecture had to be designed. This was originally based on the VGG16 architecture, however, since VGG16 was designed for 1000 classes, the size and depth were too large for this binary classification problem. This leads to no material improvement over a true random approach. A similar, yet smaller, design had to be employed. This design went through continuous iterations of adding convolutional and max-pooling blocks, double convolutional layers, and various dense layers. In the end, the design used in this thesis appeared to be the best architecture, however, there are countless permutations that were left untested.

In addition to testing dozens of model architectures, hyperparameters had to be tuned to further optimize the model. This included filter size, filter stride, padding, number of neurons in the dense layers, and more. The addition of overfitting prevention mechanisms such as dropout layers and regularization were also tested to slow down the model's overfitting and allow the validation accuracy to climb. Altogether, there was a large, constant effort to adjust the model to have the highest performance.

6.3 Limitations

6.3.1 Lack Of Data

One of the main limitations of this research stemmed from the lack of data we acquired in our limited time frame. We were unfortunately not able to produce actual prediction maps due to not having a complete set of data over a given time frame. Therefore, we could not compare our results directly to a wildfire or forest service's wildfire hazard, risk, or prediction map.

6.3.2 Data Imbalance

Additionally, the real dataset is imbalanced, with approximately 1% of the examples belonging to the positive class and approximately 99% of examples belonging to the negative class. This presents various problems when training and testing the network. The network was trained using a 50% positive to 50% negative split as this allows the network to best learn the patterns of the positive class. To test the results of our model, we then used the imbalanced data. This shows how the model performs in real-world situations. Because the imbalance is so large, having a sufficient amount of data becomes difficult. We used only 100 positive examples which required 10,000 negative examples to keep the correct balance of data. 100 positive examples is a smaller amount than we ideally wanted in our testing set, but increasing the number of positive examples causes the number of negative examples required to increase dramatically. This imbalance also can lead to precision and F1-scores dropping rapidly with each false positive. It is hard for the model to be extremely precise when the imbalance is so large, as there are many more opportunities for false positives than with an evenly balanced dataset.

The data imbalance also prevents us from easily doing k-fold cross-validation. K-fold cross-validation is not usually completed on imbalanced data as it makes learning on the entire dataset difficult. Because of this imbalance, only one model for each architecture is able to be trained. K-fold cross-validation attempt to account for variance between model performance when trained on different segments of data. Since we were unable to perform k-fold cross-validation, we may have a slight variance in model performance due to the segment of data selected for training which may affect the final results.

6.3.3 Ablation Permutations

Since our ablation study only measured one set of permutations, there may be slight differences in feature group performance when changing the combinations measured. Therefore, we are unable to determine with certainty whether the results of the ablation study hold true in other feature group pairings. This is highlighted by the one-group model where land surface temperature is alone not able to predict wildfires, however, we cannot say whether land surface temperature helped or hindered performance in combination with other feature groups.

6.3.4 Data Leakage

Because there is both a spatial and temporal aspect to this data, there is potential data leakage when randomly sampling the dataset for training. A geographic region may be represented in the dataset many times over the 108 one-week intervals. During training, the model may learn patterns relating to that specific region's climate patterns, rather than the general climate patterns that occur. Therefore, there is an argument that including the same region multiple times at different date intervals can cause the network not to generalize on the climate data. However, this can be viewed as both a negative and a positive effect. From the negative viewpoint, the model may be less able to generalize and consequently be unable to be applied to different regions outside of California and Nevada. From the positive viewpoint, the model is able to remember each region throughout the temporal aspect and can understand the region-specific trends that cause wildfires. This enables the model to be more highly specialized in the California and Nevada region.

In this research, we did not mask spatial regions to create the training and testing dataset. Our primary goal is to allow the network to train as best as it can on every region to produce the most accurate wildfire predictor for California and Nevada. Future research may decide to use region masking to determine the effectiveness of the model on unseen geographic areas.

6.3.5 Positive Labels

To define the positive and negative classes we utilize the MODIS Fire & Thermal Anomalies product. This product is susceptible to producing false positives from controlled burns, bonfires, or any other large fires that are not wildfires. This likely results in these false positive labels making their way into the dataset. These false positive labels may then be detracting from the network's ability to predict and find patterns relating to wildfires. Removing these instances would take large-scale, manual labeling to verify each anomaly is a true wildfire incident.

6.4 Future Works

6.4.1 Expanding The Dataset

As there are minimal papers utilizing convolutional neural networks and MODIS images to predict wildfires, there is still an enormous amount of research left to do in this area. Expanding the dataset to include the most valuable layers and capture the most comprehensive array of climate patterns is likely one of the highest value next steps that can be taken. There is a host of additional products that can be brought into the dataset, as well as many other Google Earth Engine products from various sources that can supplement the MODIS images. Including these additional layers may allow the model to find new combinations of features that boost performance.

One of the most interesting and potentially useful groups of products would be California's Basin Characterization Model (BCM). The BCM aggregates various climate inputs to measure runoff, recharge, evapotranspiration, and sublimation. These products play a major part in the hydrological cycles in the California region [53]. These cycles determine climatic water deficit which helps define drought severity and its effect on vegetation. The BCM contains historical and predictive monthly data at 270 m resolution, possibly providing a more in-depth and rigorous analysis of the hydrology of the region than MODIS images can. While the BCM is not available through Google Earth Engine, TerraClimate is and measures similar products such as climatic water deficit. Either of these sources could provide additional value to the dataset.

In our dataset, we utilize vegetation indices to attempt to model the fuel components of a region. Dried or dead underbrush might not be represented in these indices. Preliminary searches do not reveal any datasets that track underbrush density. We could create an alternative dataset to measure underbrush density by analyzing vegetation growth throughout the spring and determining the difference between the summer months. This would measure the amount of vegetation that died or dried out between the spring growth and the selected summer date. This dataset could be used to measure spring season phenology, or the date at which spring starts. As mentioned previously, Westerling showed earlier spring seasons are correlated with larger and more numerous fires during the summer [6]. Measuring these two products may provide more context into the spring conditions leading up to and impacting wildfires.

6.4.2 Addressing Data Imbalance

One of the largest obstacles we faced was the imbalance of positive and negative examples in our dataset. There are a several common techniques for addressing data imbalance to help models perform better.

The technique employed in this research is to oversample the positive class. This brings the distribution of positive and negative examples closer together. We utilized 3,000 positive examples and 3,000 negative examples which results in an even split for training. With a more balanced training set, the model is able to learn the features of the positive class better than when training on the purely imbalanced data. Future works may try taking different distributions of positive and negative data to see which has the best results. A study with splits of 1:99, 25:75, 50:50, 75:25, and 99:1 may provide a good assortment of results to determine where the optimal distribution of data lies. The testing set should remain as the 1:99 split as this allows us to see the real application of the models.

A technique that doesn't involve changing the balance of the dataset is to change the weighting of the loss function for the positive and negative class. By weighting the loss for the positive class higher than the loss for the negative class, we can punish the network more when misidentifying the positive class, forcing it to learn faster in order to correct these mistakes. This would allow the model to train on the imbalanced dataset without having to modify the distribution of data.

Finally, increasing the amount of data in the dataset would help bring in a larger number of positive examples from which the network can find more patterns. This would help the network become more confident in its positive predictions and therefore help the overall performance of the model. Performing these three techniques in future studies would provide useful insight into how best to approach the imbalance in this dataset to optimize results.

6.4.3 Additional Ablation Permutations

A more comprehensive feature analysis also should be completed with an exhaustive ablation study to determine which products were the most important and which combination of products resulted in the highest performance. Additionally, further analysis should be completed to determine whether the negative effects of evapotranspiration and water products as well as the positive effect of historical fire data and land data are reproducible. This would prove to be a critical follow-up to validate the utility of the dataset for future studies.

6.4.4 Other Dataset Uses

Moreover, dataset-focused studies could concentrate on the applicability of this dataset to areas outside of the Western United States. Comparing the performance of different geographical areas and climates may reveal strengths and weaknesses in the model's predictive ability across the globe.

The dataset can also be used for applications outside of the field of wildfire prediction. MODIS data can be used for a myriad of climate-related tasks; the network may be trivially tweaked to instead predict other features, or swapped out with other network types to predict time-series data into the future.

This convolutional neural network approach should also continue to be tested to confirm the findings in this paper. A study verifying the usefulness of several different convolutional neural network architectures would shed more light on the feasibility of this approach. Such an ancillary study may determine a more optimal architecture for this dataset.

6.4.5 Generating Maps

Finally, because the primary objective of the study is to determine wildfire risk, research that compares historical wildfire risk maps to wildfire risk maps generated by the network would be an incredibly valuable validation study. Should the maps be comparatively similar, the model would likely be a faster and easier way to rapidly generate risk maps on a more frequent basis, requiring less work and oversight.

6.5 Closing Remarks

Overall, this study explored a wide breadth of research surrounding wildfire risk assessment, MODIS climate data, feature importance, and convolutional neural networks, and it opens the door to many new research opportunities. While the novel convolutional neural network model did not show a compelling improvement over previous models, it expands the wealth of knowledge in this area and introduced the convolutional neural network approach for the prediction of wildfire incidents.

BIBLIOGRAPHY

[1] California Department of Forestry and Fire Protection (CAL FIRE). *2020 incident archive*. Cal Fire Department of Forestry and Fire Protection.

[2] *2021 Verisk FireLine State Risk Report - California*. Verisk, 2021.

[3] *State Wildfire Response Costs Estimated to Be Higher Than Budgeted*. California Legislative Analyst's Office, 2020.

[4] Daoping Wang, Dabo Guan, Shupeng Zhu, Michael Mac Kinnon, Guannan Geng, Qiang Zhang, Heran Zheng, Tianyang Lei, Shuai Shao, Peng Gong, and et al. Economic footprint of california wildfires in 2018. *Nature Sustainability*, 4(3):252–260, 2020.

[5] Sergio M Vicente-Serrano, Juan-I Lopez-Moreno, Santiago Beguería, Jorge Lorenzo-Lacruz, Arturo Sanchez-Lorenzo, José M García-Ruiz, Cesar Azorin-Molina, Enrique Morán-Tejeda, Jesús Revuelto, Ricardo Trigo, et al. Evidence of increasing drought severity caused by temperature rise in southern europe. *Environmental Research Letters*, 9(4):044001, 2014.

[6] Anthony L Westerling, Hugo G Hidalgo, Daniel R Cayan, and Thomas W Swetnam. Warming and earlier spring increase western us forest wildfire activity. *science*, 313(5789), 2006.

[7] Daniel R Cayan, Edwin P Maurer, Michael D Dettinger, Mary Tyree, and Katharine Hayhoe. Climate change scenarios for the california region. *Climatic change*, 87(1):21–42, 2008.

[8] Daniel L Swain, Baird Langenbrunner, J David Neelin, and Alex Hall. Increasing precipitation volatility in twenty-first-century california. *Nature Climate Change*, 8(5):427–433, 2018.

[9] Andrew L Sullivan. Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi-physical models. *International Journal of Wildland Fire*, 18(4):349–368, 2009.

[10] BW Butler, MA Finney, PL Andrews, and FA Albini. A radiation-driven model for crown fire spread. *Canadian Journal of Forest Research*, 34(8):1588–1599, 2004.

[11] AM Grishin, AD Gruzin, and EE Gruzina. Aerodynamics and heat exchange between the front of a forest fire and the surface layer of the atmosphere. *Journal of Applied Mechanics and Technical Physics*, 25(6):889–894, 1984.

[12] Michael Bowden John Carpenter Maureen T. Brooks Karen L. Abt Ronda Sutphen Jeffrey P. Prestemon, Todd J. Hawbaker and Samuel Scranton. Wildfire ignitions: A review of the science and recommendations for empirical modeling. *United States Department of Agriculture - Forest Service*, 2013.

[13] Jennifer K Balch, Bethany A Bradley, John T Abatzoglou, R Chelsea Nagy, Emily J Fusco, and Adam L Mahood. Human-started wildfires expand the fire niche across the united states. *Proceedings of the National Academy of Sciences*, 114(11):2946–2951, 2017.

[14] James K. Agee and Carl N. Skinner. Basic principles of forest fuel reduction treatments. *Forest Ecology and Management*, 211(1-2):83–96, 2005.

[15] Susan Charnley, Thomas A. Spies, Ana M. Barros, Eric M. White, and Keith A. Olsen. Diversity in forest management to reduce wildfire losses: implications for resilience. *Ecology and Society*, 22(1), 2017.

[16] Joe H Scott, Matthew P Thompson, and David E Calkin. A wildfire risk assessment framework for land and resource management. 2013.

[17] Joe H Scott, Matthew P Thompson, and David E Calkin. Risk assessment methods. *Oregon Natural Hazards Workgroup*, June 2005.

[18] ByoungChul Ko, JunOh Park, and Jae-Yeal Nam. Spatiotemporal bag-of-features for early wildfire smoke detection. *Image and Vision Computing*, 31(10):786–795, 2013.

[19] Suppressing wildland fires. *Environment and Natural Resources.*

[20] Gregory K Dillon and Julie W Gilbertson-Day. Wildfire hazard potential for the united states (270-m), version 2020. 2020.

[21] Landfire products table. *Landscape Fire and Resource Management Planning Tools.*

[22] Fsim-wildfire risk simulation software. *FSim-Wildfire Risk Simulation Software — Missoula Fire Sciences Laboratory.*

[23] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and Bruno Arnaldi. A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4(2):R1, 2007.

[24] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[25] Thomas P Ryan. *Modern regression methods*, volume 655. John Wiley & Sons, 2008.

[26] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[27] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[28] Bayya Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[29] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[30] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[31] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.

[32] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[33] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[35] Alexandre Belloni and Victor Chernozhukov. 1-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39(1):82–130, 2011.

[36] Warren S Sarle et al. Stopped training and other remedies for overfitting. *Computing science and statistics*, pages 352–360, 1996.

[37] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, and Mohammed Bennamoun. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8(1):1–207, 2018.

[38] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

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

[40] Keiji Yanai and Yoshiyuki Kawano. Food image recognition using deep convolutional network with pre-training and fine-tuning. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2015.

[41] Shivangi Jain, Nitin Pise, et al. Computer aided melanoma skin cancer detection using image processing. *Procedia Computer Science*, 48:735–740, 2015.

[42] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[43] *Global Change Research Act of 1990.* Public Law 101-606(11/16/90) 104 Stat. 3096-3104, 1990.

[44] A Savtchenko, D Ouzounov, S Ahmad, J Acker, G Leptoukh, J Koziana, and D Nickless. Terra and aqua modis products available from nasa ges daac. *Advances in Space Research*, 34(4):710–714, 2004.

[45] Daniela Stojanova, Panče Panov, Andrej Kobler, Sašo Džeroski, and Katerina Taškova. Learning to predict forest fires with different data mining techniques. pages 255–258, 2006.

[46] Fangrong Zhou, Hao Pan, Zhenyu Gao, Xuyong Huang, Guochao Qian, Yu Zhu, and Feng Xiao. Fire prediction based on catboost algorithm. *Mathematical Problems in Engineering*, 2021, 2021.

[47] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[48] Eduardo Eiji Maeda, Antonio Roberto Formaggio, Yosio Edemir Shimabukuro, Gustavo Felipe Balué Arcoverde, and Matthew C Hansen. Predicting forest fire in the brazilian amazon using modis imagery and artificial neural networks. *International Journal of Applied Earth Observation and Geoinformation*, 11(4):265–272, 2009.

[49] Mar Bisquert, Eduardo Caselles, Juan Manuel Sánchez, and Vicente Caselles. Application of artificial neural networks and logistic regression to the prediction of forest fire danger in galicia using modis data. *International Journal of Wildland Fire*, 21(8):1025–1029, 2012.

[50] Alessandro Santopaolo, Syed Saad Saif, Antonio Pietrabissa, and Alessandro Giuseppi. Forest fire risk prediction from satellite data with convolutional neural networks. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, pages 360–367. IEEE, 2021.

[51] fire season climatology - national wildfire coordinating group.

[52] Shahrzad Gholami, Narendran Kodandapani, Jane Wang, and Juan Lavista Ferres. Where there's smoke, there's fire: Wildfire risk predictive modeling via historical climate data. In *Annual Conference on Innovative Applications of Artificial Intelligence (IAAI)*, 2021.

[53] Lorraine E Flint, Alan L Flint, James H Thorne, and Ryan Boynton. Fine-scale hydrologic modeling for regional landscape applications: The california basin characterization model development and performance. *Ecological Processes*, 2(1), 2013.

APPENDICES

Appendix A

DATASET CREATION TOOL

The code used in this thesis can be found in the public repository:

https://github.com/sfnesbit/Wildfire-Risk-Assessment

This includes notebooks for the following tasks:

- Dataset creation tools
- Data visualization tools
- Experiments
  - Convolutional neural network
  - Comparison models
  - Feature analysis