Gurik Mangat
Semiahmoo Secondary School
Surrey, British Columbia, Canada

Project ID: ROBO049

ROBO049 - FyreWatch: Deep Learning for Accurate Wildfire Environmental Conditions Detection

Introduction

The Problem

- Wildfires burn more than **7.0 million acres annually** in just the United States [1] alone. Certain standardized systems and indices such as the Canadian Forest Fire Weather Index System (Figure 1) or the National Fire Data System currently exist to rank fire danger based on quantitative and qualitative measures . However, the largest problem faced by these indices is **analyzing and utilizing data in a practical and effective way** [2]. Input from the various fire danger rating systems is evaluated by human firefighters who rely on past exposure to determine fire risk [3]. Although firefighter experience is an important aspect, it is subject to problems such as **human error, lack of experience, and the fact that humans simply cannot process as much information** as a computerized system. Furthermore, manually collecting field data and running computations is both **time intensive, and resource intensive**.

- **Past Efforts** - Although some efforts have been made to modernize firefighting by organizations such as the United States Geographical Service (USGS) when they developed the **GeoMAC** [4] system (Figure 2), or by **NASA with the FIRMS Active Fire Database**, both to digitally map wildfire situations, the need for modern computational methods that can rapidly and intelligently detect preemptive wildfire conditions still stands.

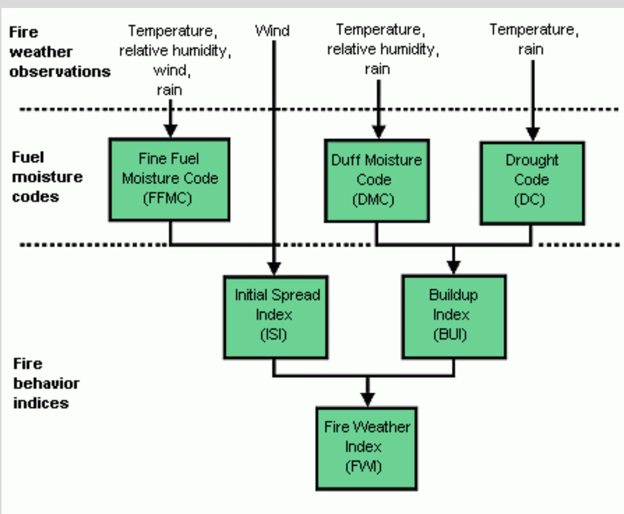


Figure 1: Diagram of the Canadian Forest Fire Weather Index System
Courtesy: Natural Resources Canada

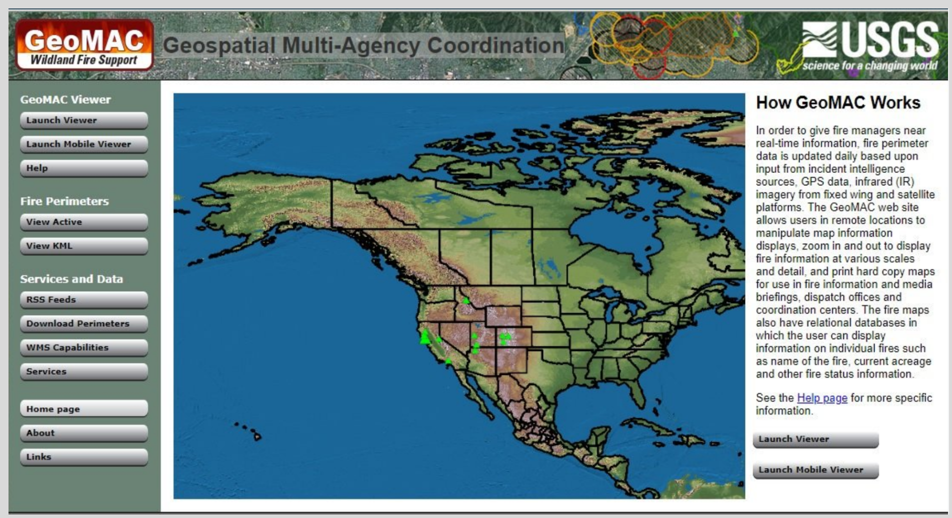


Figure 2: The since discontinued USGS GeoMAC wildfire mapping system
Courtesy: USGS GeoMAC Program

- **Better Data Collection** - The emergence of **cheaper satellite based data collection** methods (Figure 3 & 4) and greater **ease of access** to private launch services means that remotely sensed data and satellite imagery, which can be **acquired on a daily basis** [5] is a more viable option. Furthermore, this is a resource that fire departments already have access to. Remote sensing is also not prone to ground weather conditions and allows for more reliable data collection. Therefore developing a solution that can effectively analyze this data in large amounts will allow them to better utilise an **existing resource**.

Figure 3: Earth observation nanosatellites
Courtesy: NanoRacks

Figure 4: European Space Agency (ESA) Sentinel Observation Satellite
Courtesy: European Space Agency

Engineering Goal

As a result, the engineering goals of this project were to develop binary classification artificial neural networks and train them to a high degree of accuracy on satellite remotely sensed data to detect the presence of wildfire conditions, finally deploying those trained models into a prototype demonstration application leveraging the technology for firefighter field use.

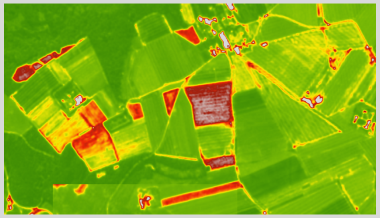
Development Procedure

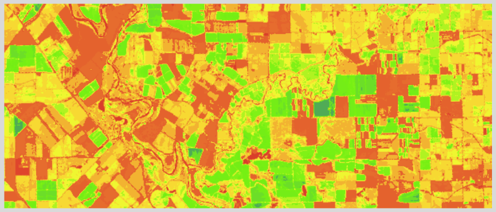
The procedure of this project was split into **four main steps**:

1. Determining which environmental conditions play a decisive role in wildfire environments so that they can be used as inputs for the neural network.
2. Obtaining data and constructing a dataset using the chosen inputs
3. Training a classification neural network on the gathered dataset for accurate detection of wildfire conditions
4. Integrating this trained machine learning model into a prototype desktop interface to use on the field.

1. Environmental Conditions

- After consulting wildfire ranking indices, the following environmental conditions were chosen as inputs for the neural network: **temperature, humidity, wind speed, soil temperature, soil moisture, and fuel moisture.**
- Normalized Difference Vegetation Index (NDVI) satellite imagery (Figures 5 & 6) was used for fuel moisture data. NDVI imagery provides a graphical representation of vegetation health by measuring the difference between near-infrared light (which vegetation reflects) and red light (which vegetation absorbs). Healthy vegetation that contains larger amounts of chlorophyll, water, and moisture has higher NDVI values.
- A ***K-means clustering*** **algorithm** was used to quantify this color-coded information in the image so that it could be added to a dataset. The RGB (red, green, blue) values of pixels in the image are taken and assigned to a nearest cluster with the largest cluster representing the most dominant colour. The centroid of each cluster provides the RGB values of the most dominant color and knowing this allows us to **quantifiably determine** whether the fuel moisture is primarily dry and high risk, or more moist with relatively low risk.





Figures 5 & 6: Sample NDVI Satellite Imagery used in this project.
Courtesy: OpenWeather

2. Dataset Construction

- To construct a dataset, the **NASA FIRMS Active Wildfire Database** was first used to locate a list of actively burning North American wildfires from the 2019 June-September season. Using this database, the geographical coordinates of those active fires were acquired, and environmental conditions data about their temperature, humidity, wind speed, soil moisture and soil temperature, were acquired through the **AgroDashboard API**.
- This API was also used to acquire remote sensing **high-resolution NDVI imagery** taken by the **Landsat-8 and Sentinel** satellites that corresponded to each of those locations. Those images were then processed by the K-means clustering algorithm to get the **RGB values** of their respective dominant colors.
- Both cases of fires and non-fires were collected, and through several steps of data augmentation the final dataset used for neural network training contained 2000 unique cases. This project also accounted for **regional data bias** by including balanced wildfire data from a multitude of varying regions, terrains, and topographies from across North America. **This is a distinct and crucial advantage over other wildfire prediction models as it exhibits the ability to dynamically generate accurate detections even in differing regions or conditions.**

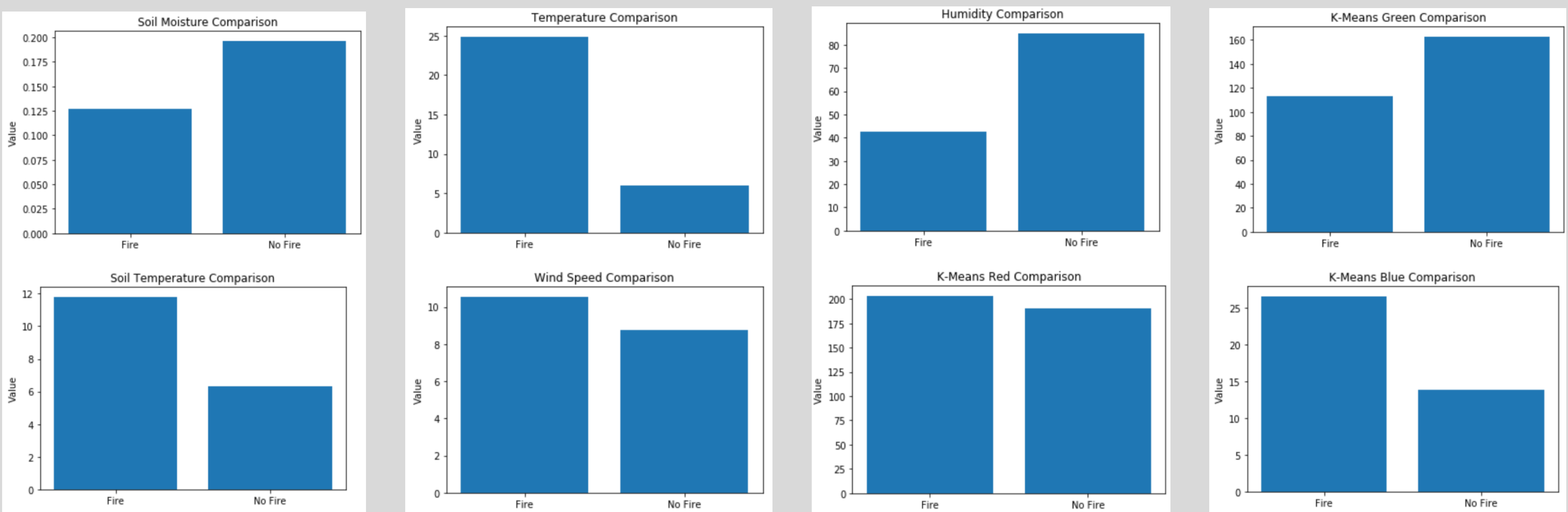


Figure 7: Graphical analysis completed on all individual data points in the dataset. Comparing their mean values with "fire" and "non-fire" classifications indicates to us that there is a distinct correlation to wildfires and can thus be used to determine whether conditions are present.

3. Artificial Neural Network

- In order to begin generating wildfire detections, a **classifier Artificial Neural Network (ANN)** was created.
- The network architecture used in this study contains **1 input layer with 8 neurons, 1 output layer, 2 hidden layers with 8 neurons each, 23 nodes, and 136 connections** (Figure 8). The Relu activation function was used for its subsequent faster learning rate [6] as opposed to other activation functions such as sigmoid or tanh. For optimization, the **Adam (Adaptive Moment Estimation) optimizer** was used in order to update attributes of the network, such as weights, during training.
- For a loss function, **binary cross-entropy** was used due to the classification nature of this problem. To split the dataset for training, it underwent a **70:20:10 split** where 70% of the dataset was used for training, 20% was used for validation, and the last 10% was used for testing purposes. This ratio is slightly different from the commonly used 80:20 split, but provides major benefits when training on relatively smaller sized datasets as it allows **model validation** to happen with greater accuracy on the network's different hyperparameters.
- A **dropout function with rate 0.20** after the first hidden layer was used to prevent excessive neuron co-adapting as this method **significantly reduces overfitting** in comparison to other regularization methods [8].
- **Accuracy and loss** of the ANN was measured and used as a benchmark for improvement during training, with an automated test run on the final completed model. This will be explored in further detail in section "Results".

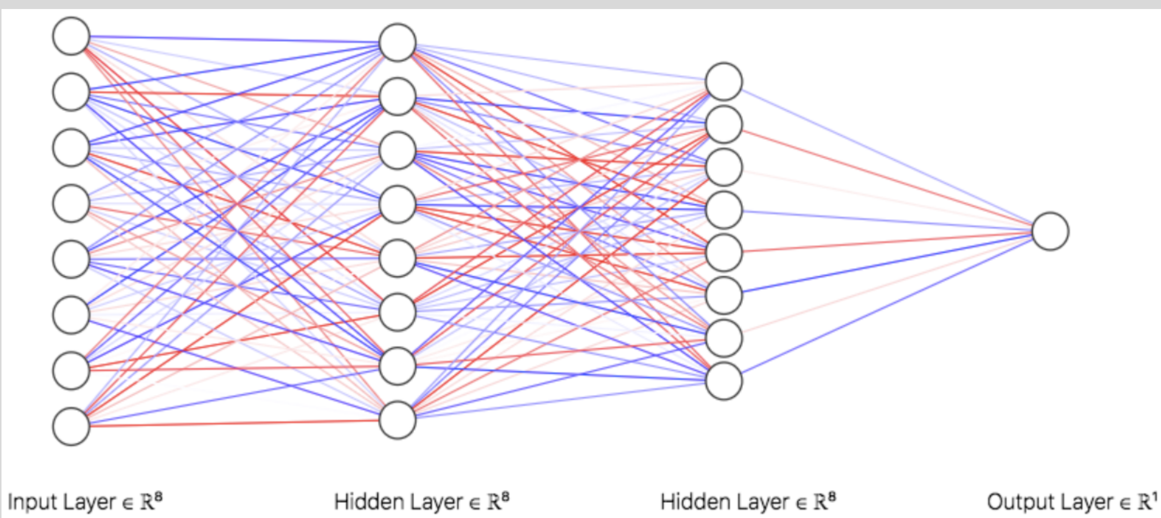


Figure 8: A diagram of the neural network architecture used in this study where each circle is a neuron and, and the coloured lines are weighted connections.

4. Application Interface

- In order to allow firefighters to utilize the algorithm to generate their own detections with **unique field-centric data**, a **prototype desktop app** (Figure 9) was built with the Tkinter graphics library. The app works by first prompting operators to upload their site's **latitudinal and longitudinal coordinates** which are then used to collect real-time, accurate weather data on humidity, wind speed, and temperature using the OpenWeather API.
- To aggregate the soil moisture and soil temperature data inputs, a **geometric-coordinate algorithm** is used to find the coordinates of a virtual one-hectare square around the firefighter which are then fed into the **OpenWeather Agriculture API**. This API allows us to call accurate **real-time** soil moisture and soil temperature data averaged over the hectare, acquired by **remote sensing sensors on the Landsat-8 and Sentinel satellites**.
- Finally, firefighters upload an **NDVI image** that corresponds to their input coordinates. That image is then quantified and analysed by the aforementioned **K-means clustering** algorithm. The field data is processed by the neural network which generates a prediction on whether or not fire conditions are present and a **detailed analytics page** is displayed. This page displays the prediction, along with all the data for the firefighters' own **situational awareness**. Below is a screenshot of the completed prototype application being used.

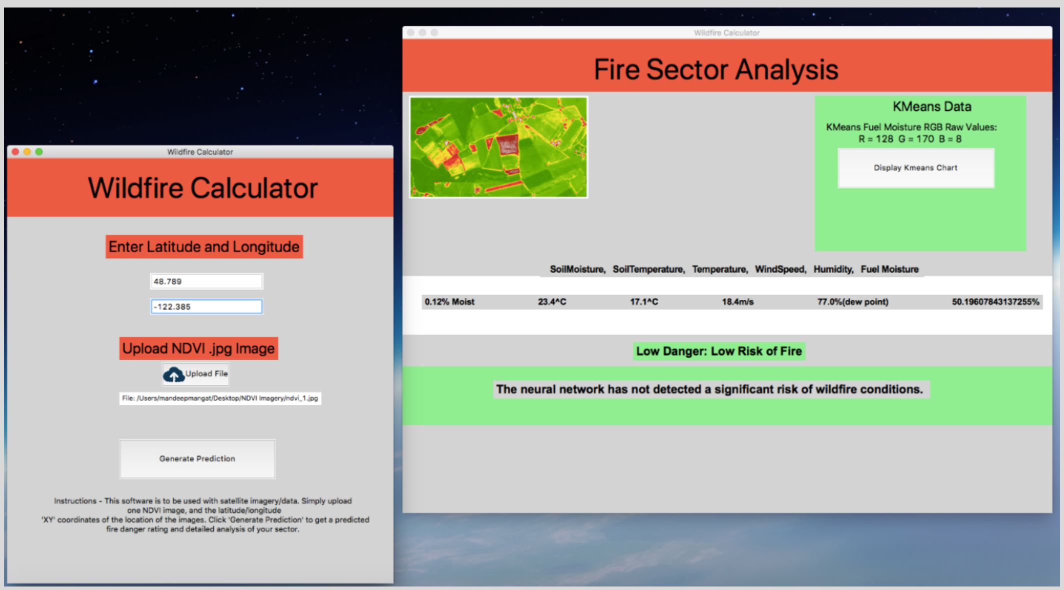


Figure 9: These are screenshots of the application in use. The first window is the data entry pane in which the firefighter uploads their coordinates and a corresponding NDVI image. The second window is produced after clicking the "Generate Prediction" button and displays the environmental conditions, fuel moisture data in the form of K-means dominant colour RGB values, and the neural network classification.

Results

- To ensure the accuracy and robustness of the neural network, 4 different metrics: **training accuracy, validation accuracy, training loss, and validation loss** were used along with a final **automated test** on the application. The first original, unoptimized model followed a conventional Pareto training validation split of 80:20, did not have a dropout layer, and a batch size of 100. This model (figure 10) shows **large signs of overfitting**. This model also had a validation accuracy of 98.33%, which can be slightly improved upon further.

- The next model leverages **fine-tuned hyperparameters** determined through repeated trials. These include a dropout function with rate 0.20 after the first hidden layer to account for the overfitting, a training:validation:test split of 70:20:10, and a batch size of 9. The Accuracy loss and Validation loss plots produced by this model after training (Figure 8), both converge, signifying no apparent signs of overfit. We also achieved a Validation accuracy of 98.61% and can see through the Accuracy graph (Figure 11), that this model trained in a stable and less volatile manner.

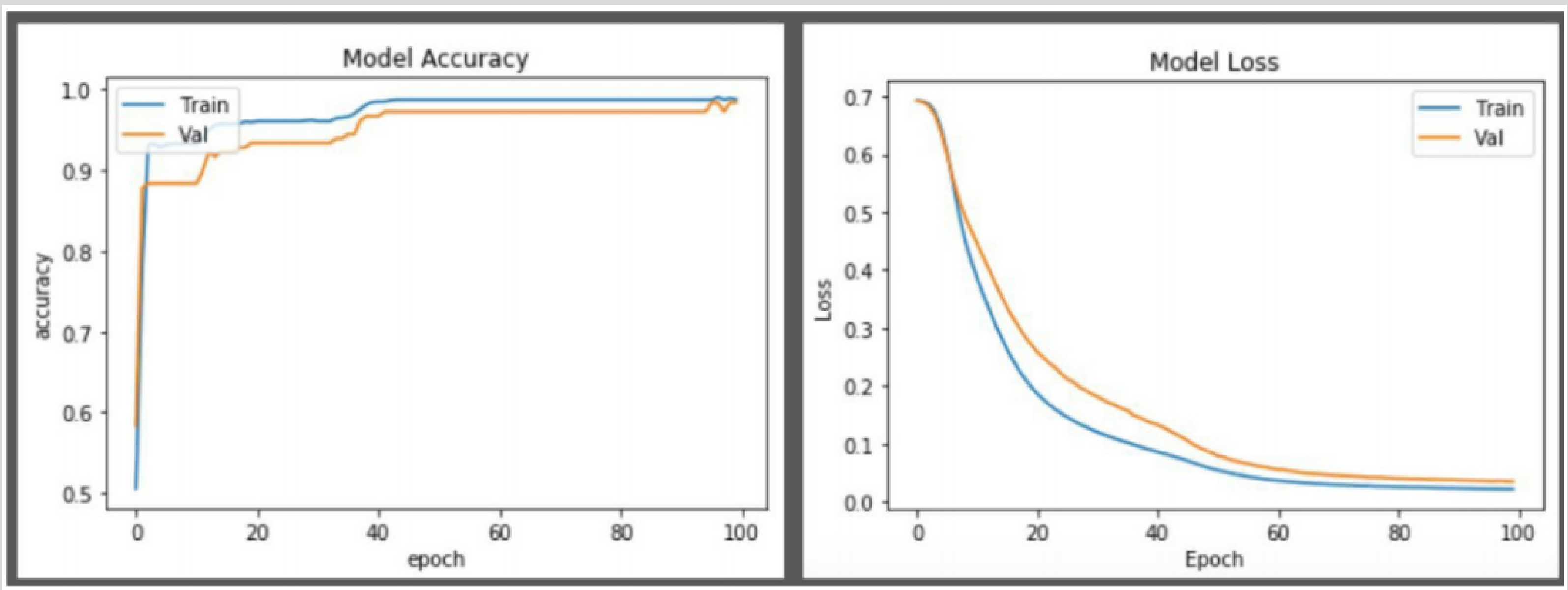


Figure 10: Accuracy and loss plots of unoptimized, original model using a data split of 80:20, no dropout function, and a batch size of 100.

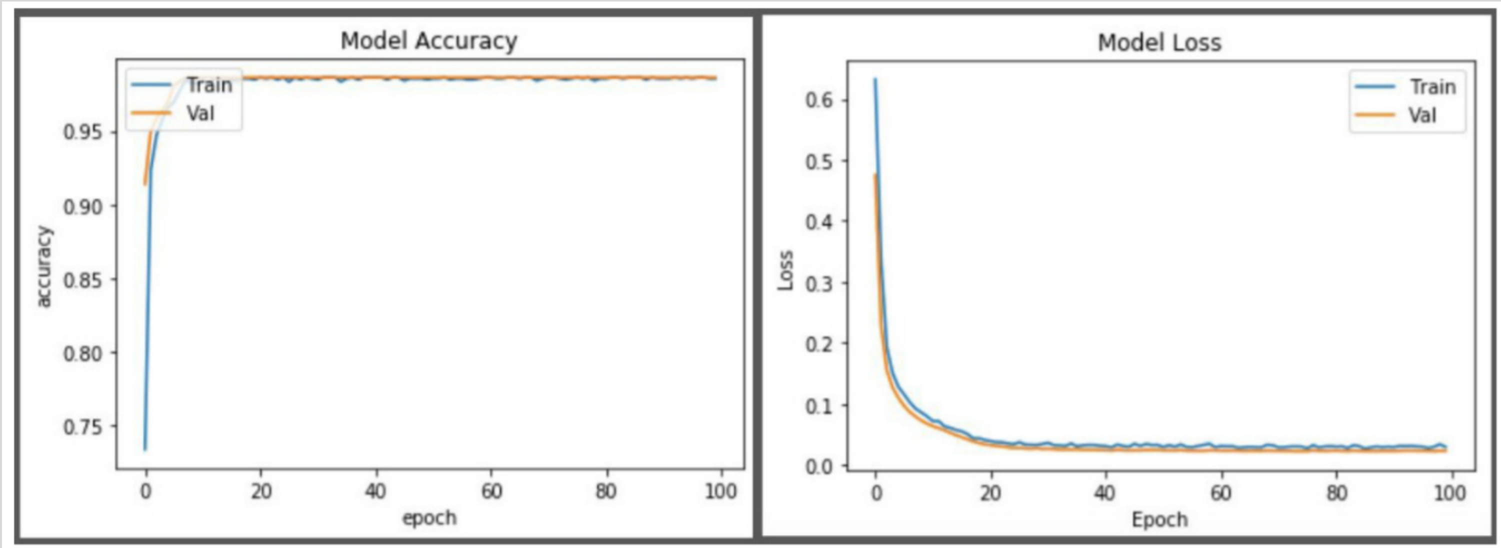


Figure 11: Accuracy and loss plots of the optimized, fine-tuned model including a dropout function with value 0.20, a data split of 70:20:10, and a batch size of 9.

- After model training and optimization was completed and a final model was produced, an automated test was run in order to determine the **final experimental accuracy**. We used our test set of 200 cases that was kept aside for testing during the initial 70:20:10 data split and achieved a final accuracy of 98%.
- The automated test correctly classified **98% of cases with a sub one-second runtime**, run locally on the machine **not relying on any external GPUs or computing power**. The prototype was able to meet its goal of detecting wildfire conditions through a **robust and reliable application interface** utilising remotely sensed data.

```
y_pred = classifier.predict(X_test)
# It's classified as "1" if the model is 95% sure that fire conditions are present. Else, it's "0"
y_pred=[1 if y>=0.95 else 0 for y in y_pred]
accuracy=0
for value in range (200): # The test dataset includes 200 cases
    if(y_pred[value]==list(y_test)[value]):
        accuracy+=1
print(y_pred) # Prints all the classifications made by the neural network
print("Final Test Accuracy: " + str((accuracy/200)*100) + "%") # Getting accuracy in percentage form
```

```
[1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0,
0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1,
1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1]
Final Test Accuracy: 98.0%
```

Our final completed model was able to correctly determine and classify 196/200 fire conditions cases and achieved a test accuracy of 98%

Figure 12: Code snippet of the automated test performed on the completed trained and optimized model.

Discussion

Previous Work

- Earlier research (Y.O. Sayad et al, 2019) had a similar approach by using NDVI imagery and LST (Land Surface Temperature) measurements as inputs for an artificial neural network.
- Another study (S.R. Coffield et al, 2019) utilised neural networks to predict the **final size of wildfires** after ignition. Although this was not the focus of this study, it is seen as an exciting future addition.
- No study to our knowledge used K-means clustering as a method for wildfire fuel moisture quantification, but as this study achieved high levels of accuracy, subsequent research could be conducted to go more in-depth into this unique technique's potential.

Improvement over Existing Methods

- Wildfire prevention effectiveness relies on knowledge about the spatio-temporal likelihood of fire occurrence and this is reflected in the use of a comprehensive set of environmental conditions as data points. A study mentioned earlier used LST as an input, however FyreWatch instead used **soil temperature and moisture data** as this is data which is remotely sensed and corroborates with above ground temperature and humidity.
- Moreover, aerial airborne technologies used for data collection such as UAVs and helicopters are **prone to environmental conditions like strong winds and heavy smoke**, both of which are very common in wildfire regions. This significantly limits flight availability, and the effectiveness of onboard instruments. FyreWatch uses frequent and high quality remotely sensed satellite data and imagery that fire departments already have. Therefore developing a solution that can effectively analyze this data in large amounts allows them to better utilise an existing resource. As a result, the FyreWatch application was an improvement over existing methods as it not only gave the ability to effectively detect wildfire conditions, but also integrated it into an **easy-to-use prototype application** that can allow firefighters to make more intelligent decisions out on the field.

Conclusion

- **With this research, artificial neural networks (ANNs) were successfully constructed and trained to classify regions on whether or not wildfire conditions were present based on environmental conditions inputs and our completed model achieved a final accuracy of 98% during testing.**
- A key aspect throughout the development of this project has been that FyreWatch focuses on **detecting areas of potential threat** rather than those where fires have already ignited, allowing for better anticipation of future fire risk. This predictive insight is a powerful application in a number of industries, from **more reliable data for insurance** and other financial processes, to **better wildfire planning by local governments**.
- The satellite remotely sensed data used by the application is **already accessible by wildland departments** and is **not prone to the same problems as other aerial methods**. In firefighting, knowing which areas are at a higher risk beforehand allows for **more accurate controlled burns, safer evacuation orders, and more geographically precise and efficient preventative measures** so that fires do not ignite in the first place. This study has successfully demonstrated the effectiveness of our technique and will provide major needed leaps to the fields of fire dynamics, sustainable forestry management, and wildfire danger mitigation.

References and Sources

Cover image: McNew, David. *The Guardian*, Getty Images, 15 Sept. 2020, www.theguardian.com/world/2020/sep/15/west-coast-fires-firefighters-oregon-washington-california.

[1] Hoover, K., & Hanson, L. A. (2019). Wildfire statistics. Congressional Research Service. https://fas.org/sgp/crs/misc/IF10244

[2] Hinds-Aldrich, M., Dr., Knight, M., Nicolosi, A., & Evarts, B. (2017). National Fire Data Survey: Findings on the State of the Existing American Fire Data Ecosystem. National Fire Protection Association. https://www.nfpa.org/~/ media/96AD 4BAEA2684EE3B2E3F73D80D1B60E.pdf

[3] United States Forestry Service. (1996). Wildland Firefighter Safety Awareness Study. TriData. https://www.nifc.gov/safety/safety_documents/phase1.pdf

[4] USGS. (2005). Wildfire Hazards—A National Threat. United States Geographical Survey. https://pubs.usgs.gov/fs/2006/3015/2006-3015.pdf

[5] Gillespie, T. W., Chu, J., Frankenberg, E., & Thomas, D. (2007). Assessment and prediction of natural hazards from satellite imagery. *Progress in Physical Geography: Earth and Environment, 31*(5), 459-470. doi:10.1177/0309133307083296

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90. https://doi.org/10.1145/3065386

[8] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, 15(56), 1929-1958.

Note: To ensure that the data collection did not infringe on any data use or privacy laws, it was accessed primarily through either open access public entities such as the NASA FIRMS Active Fire Database, or through commercially available APIs such as the OpenWeather Agro API.