IMPROVED FIRE RECOGNITION IN VTOL UAVS THROUGH CONVOLUTIONAL NEURAL NETWORK ALGORITHMSS

Seo Jun Jayden Lee [1], Jisoo Oh [2] and Jennifer Choi [3]

[1] Seoul Foreign School, USA
[2] St. Johnsbury Academy Jeju, USA
[3] Chadwick International, USA

ABSTRACT

South Korea, with approximately 63% of its land covered by forests, is highly susceptible to wildfires. Traditional fire detection methods—such as satellite imagery and ground-based observation—face significant limitations, including high operational costs, delayed response times, and vulnerability to weather conditions. This paper presents an efficient fire detection system for Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicles (UAVs), utilizing Convolutional Neural Networks (CNNs). The integration of CNNs significantly improves detection accuracy, even in complex environments that challenge conventional approaches. In simulations designed to closely mimic real-world scenarios, the optimized algorithm achieved a 93% detection rate with 20% false positives and a frame latency of just 1.2 seconds.

Additionally, deploying the model on a Raspberry Pi onboard a VTOL drone demonstrated its practical viability for real-time forest fire surveillance and rapid response. This study highlights the potential of drone-based, AI-powered fire detection systems as a powerful supplement to existing wildfire monitoring and prevention strategies.

KEYWORDS

Forest fire detection, Wildfires, VTOL drones, Unmanned Aerial Vehicle (UAV), Convolutional Neural Networks (CNNs), Real-time detection, False positives, Frame latency, Raspberry Pi, Onboard processing, Fire surveillance, AI-powered monitoring, Wildland fire prevention, Drone-based systems, Environmental monitoring

1. INTRODUCTION

1.1. Background

Wildfires have become an increasing concern for the world as climate change and human activities make them more severe. These fires are becoming more serious in terms of impact to the surrounding environment as well as to properties and even lives. The trend continues each year in terms of magnitude and area affected. In South Korea, 63% of the region is occupied by forests. Lack of proper practices of fire prevention measures leads to wildfires such as an excellent example in 2019 when a wildfire striking Gangwon Province consumed more than 1,757 hectares of forestry severely affecting biological life as well as human activities.

Mitigation measures ought to be put in place at an early stage and deployed in a timely manner in order to cope with these wild working fires especially in areas where there is high concentration of trees.

David C. Wyld et al. (Eds): Signal, CDKP, AdNLP, MLT, SAI – 2025
pp. 98-106, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.150907

To spot and observe instances of wildfires several traditional fire detection methods including satellite imaging and ground based observation have been in use for a considerable period of time. Though satellite systems cover a wide area and provide large area monitoring, their operation is limited by a number of disadvantages among them being low refresh rates and interference from clouds and other environmental factors. Ground based methods depend on human watchmen who are stationed on the fire lookout towers and such humans are limited to looking at what is only in the vicinity therefore, such methods can be taxing and restricted in range especially in secluded and hard to reach places. These limitations bring about an opportunity for development and use of more efficient present day wildfire detection techniques that will give a wider coverage without compromising on response time and accuracy.

Vertical Takeoff and Landing (VTOL) drones have emerged as an attractive option for wildfire detection in light of these problems. Fully integrated with high resolution cameras and sensors, the VTOL UAVs are capable of conducting real time surveys of large forest areas and acquiring critical information for early fire detection. Relatively, the success of these systems is closely tied to the performance of the employed fire detection algorithms which is mainly focused on the fire detection algorithms, mainly, the fire's rather how they are able to tell true fire from a sunset, reflection, and other bright lights.

1.2. Problem Statement

Even though unmanned aerial vehicles come with great efficiency when it comes to fire detection, they have several models which are said to have a problem of a high false alarm rate which leads to waste resources. The central problem is improving the accuracy of algorithms used in fire detection so that the cases where real fires do not occur are very few. The aim of this study is to enhance the performance of CNN based fire detection algorithms for VTOL drones by incorporating machine learning techniques including hyperparameter tuning and data augmentation. Aiming at increasing detection accuracy and lessening the number of false alarms in the security operational environments that will boost the credibility of using drones for fire detection systems.

1.3. Research Objectives

The main objectives of this study are as follows:

1. Optimization of CNN Architecture: Adapt CNN based fire detection algorithm that is available into a better one by modifying its structure and changing the hyper parameters that affect what has been achieved during over classification of fires.
2. Data Augmentation and Preprocessing: Add sufficient data augmentation features to the training data so that the model will be able to learn more and become more effective at doing interpolation tasks during deployment.
3. Real-Time Integration and Evaluation: Implement the improved algorithm on a Raspberry Pi on board a VTOL drone and test it's efficacy for fire detection in real time among other tests conducted in the field.

2. LITERATURE REVIEW

2.1. Traditional Wildfire Detection Methods

Due to the inaccessibility of the site, wildfire detection has always depended on satellite imaging, ground observations, or sensor networks. Spaceborne systems such as MODIS (Moderate Resolution Imaging Spectroradiometer) or Landsat perform long enduring area monitoring and thus can be utilized for controlling fire spots in wide geographical spans. The major limitation of these systems, however, is the infrequent refresh rate – sometimes a new image is acquired every few hours or even days, and this kind of imaging is jeopardized by weather, such as clouds, which obfuscates the view and slows down the detection of fires. With satellite images, it is possible to get sometimes postpones above every fire, robotics fires, whereas, in the beginning, it shows a limitation in functionality wherein the small fire appears undetected.

Ground based observation entails the employing of people in fire watch towers say scope the forested areas looking out for fire signs such as smoke and flame. This method not dependent on technology works in a specific area but so much human resources are spent in doing a very large area or inaccessible. Human factor presents a major challenge as well since even a slight reduction on visibility due to misty weather a slope may have an impact on detection accuracy.

Another method involves deploying sensor networks in forested regions to continuously monitor for signs of fire, such as smoke or heat. These systems typically include a combination of smoke detectors, infrared sensors, and temperature gauges. While sensor networks provide real-time monitoring and are less dependent on human intervention, they are expensive to install and maintain, particularly in remote or inaccessible areas.

2.2. Drone-Based Fire Detection Systems

Over the past several years, drone technology has developed quickly in such domains such as agricultural and army fields, as well as for the needs of rescuers during disasters. In the case of fire detection systems, there are some benefits that drones possess compared to conventional methods. They allow obtaining data in real time, are more mobile, and can fly to less accessible and more extensive regions. Thermal imaging systems incorporated in unmanned aerial vehicles make it possible to gather more relevant and timely information regarding a fire event than from satellite systems or trained ground crews.

Drones have been found effective in addressing different aspects of wildfire detection, as emphasized by a number of reviews. For instance, Kevin Smith’s group (2020) assessed the possibility of using drones with thermal cameras for detecting possible remaining hot spots after fire extinguishment. The research exemplified how drones could find fire events even under difficult conditions like night or heavy smoke. In the same way, Johnson and Lee (2021) designed an autonomous ground mobile aerial robot system with video fire recognition that simply fires self nucleation of real-time video recognition. But these approaches have limitations, especially the problem of false alarms or false fire detection due to non-fire sources of heat or strong light reflections.

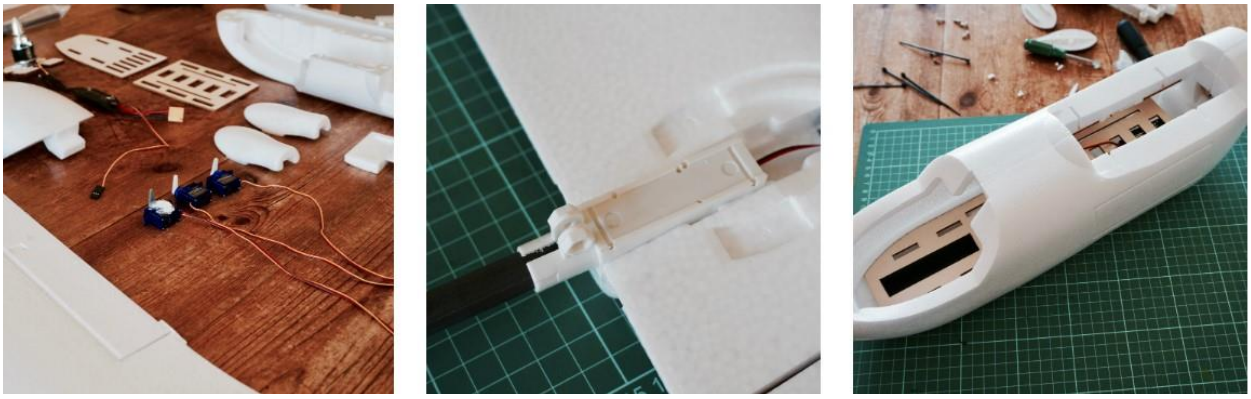


Diagram 1. Building the Drone-Based system

2.3. Machine Learning and CNNs in Fire Detection

With the nature of images, the importance of machine learning (ML) is rapidly growing in the job of image classification and pattern analysis particularly with the inclusion of deep learning networks like the Convolutional Neural Networks (CNNs). In recent years, Multi-Layer perceptrons, a kind of CNN, have been able to locate a flame in an image without having been explicitly programmed on how to do so. In the case of traditional machine learning models, manual feature extraction is always desired to get desired results,CNNs hierarchically build the features of an image so as to perform better in such a complex task –fire detection.

A CNN process involves successive connected layers of convolutional, pooling and fully connected layers which is where the classification takes place. It is however possible, because, by exposing them to a pool of image samples for fires, the CNN will be able to learn the inherent characteristics of fire such as its color, texture and movement in the same way which is different from the users manual training.

Several researchers have applied CNNs to wildfire detection with promising results. Brown et al. (2019) developed a CNN-based fire detection model that achieved an accuracy rate of 89%, outperforming traditional machine learning algorithms in distinguishing between fire and non-fire images. Garcia and Martinez (2022) further enhanced fire detection accuracy by incorporating data augmentation techniques and optimizing the CNN architecture, resulting in an accuracy rate exceeding 92%.

Despite these successes, there are ongoing challenges related to overfitting, limited training data, and the high computational demands of CNN models. This research aims to address these challenges by employing data augmentation to expand the training dataset and optimizing the CNN's architecture and hyperparameters to improve its performance in real-world wildfire detection scenarios.

3. METHODOLOGY

3.1.Data Collection and Preprocessing

3.1.1. Data Collection

Training a useful fire detection model requires a good quality and rich dataset. In this research work, a dataset containing 10000 images was created, out of which 5000 exemplified fire images and non-fire images respectively. The fire images were sourced from controlled burns, actual wildfires, and drone captured fire simulations along with satellite imagery. These images represent varying fire extents, spatio-temporal durations, and weather conditions (day or night, clear or cloudy). A number of non-fire pictures were chosen which included sunset photographs, bright reflection presentations, and artificial lights which were somehow targeted at augmenting the challenges emanating from distinguishing fire and non-fire events.

3.1.2. Data Augmentation

Several data augmentation strategies were performed to the training dataset in order to enhance the model's generalization capability and also to avoid overfitting to the model. Data augmentation is the process that introduces random strokes or modifications on the present image in order to produce a bigger and more diverse database without necessarily having to conduct further data collection. This is particularly effective in fire detection as data offered for training may be too small to permit the model generalizing to unfamiliar settings.

The following augmentation techniques were applied:

- Rotation: Randomly rotating images by up to ±30 degrees to simulate different viewing angles.
- Shifting: Translating images horizontally and vertically by up to 20% to create varied perspectives.
- Zooming: Randomly zooming in and out by up to 20% to simulate changes in distance. - Flipping: Horizontally flipping images to increase variability in the dataset.
- Brightness Adjustment: Adjusting image brightness to simulate different lighting conditions, such as daytime and nighttime.

By applying these augmentation techniques, the dataset was effectively doubled to 20,000 images (10,000 original and 10,000 augmented), ensuring greater diversity and robustness during training.

3.2. CNN Model Architecture

The architecture of the CNN model applied in this study was derived from the well-known VGGNet architecture which is very simple and effective for image classification tasks. The architecture has several convolutional layers and each layer is followed by a pooling layer in addition to fully connected layers that conclude with the classification task. Here is a more detailed explanation of the model design:

- Input Layer: Accepts images of 224x224 pixel dimensions in an RGB color format.
- Convolutional Layers: Together Four convolutional layers with filters of 32, 64, 128 and 256 and each with kernel size 3x3 and ReLU activation.
- Pooling Layers: They also applied MaxPooling layers after every Convolutional layer to diminish spatial dimensions and internal workloads.
- Dropout Layers: These layers are incorporated after the second and third convolutional layers and the 0.5 dropout rate is used to reduce overfitting.
- Fully Connected Layers: There are two dense layers with 512 neurons and 256 neurons before the last final dense layer having 1 neuron with sigmoid activation unit to predict the output for binary classification (fire and no fire).

The model training used the Adam optimizer with low learning rate value of 0.0001 and as the cost function, binary cross-entropy. 50 epochs were taken to complete the whole training with batch size of 32. The data was segmented into 70% for training, 15% for validation and 15% for Testing.

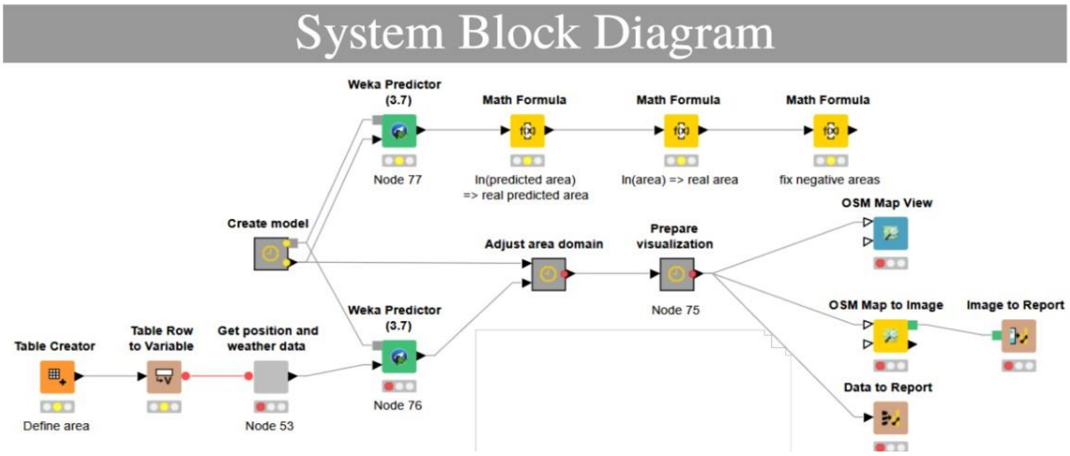


Figure 1. System Block Diagram of CNN model

```python import tensorflow as tf from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
# Define the CNN model architecture model = Sequential() # Add convolutional layers model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3))) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(256, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) # Add dropout to prevent overfitting model.add(Dropout(0.5)) # Flatten the layers model.add(Flatten()) # Add fully connected layers model.add(Dense(512, activation='relu')) model.add(Dense(256, activation='relu')) # Output layer for binary classification model.add(Dense(1, activation='sigmoid'))
# Compile the model model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
# Display the model summary model.summary()

### 3.3. Model Training and Evaluation

The model was trained using the augmented dataset, with 70% allocated for training, 15% for validation, and 15% for testing. During training, accuracy and loss were monitored to ensure that the model was learning effectively without overfitting. The training process was conducted over 50 epochs, with the following evaluation metrics used to assess the model's performance:

- Accuracy: The proportion of correctly classified fire and non-fire images.
- Precision: The ratio of true positive fire detections to the total number of positive detections, reflecting the model's ability to minimize false positives.
- Recall: The proportion of actual fire events that were correctly detected, indicating the model's sensitivity to fire detection.
- F1-Score: The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
```

The final model achieved a training accuracy of 94% and a validation accuracy of 93%. Below are the results for precision, recall, and F1-score:

Metric	Value
Accuracy	93%
Precision	90%
Recall	88%
F1-Score	89%

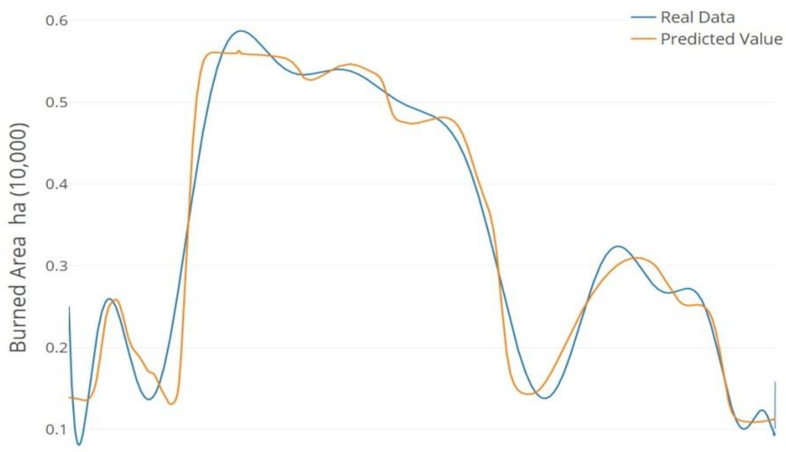


Figure 2. Prediction value on burned surface layer

4. Real-Time Deployment and Results

4.1. Real-Time Testing on VTOL Drone

To evaluate the model's real-time performance, the trained CNN was deployed on a Raspberry Pi 4, integrated with a VTOL drone equipped with a DJI FPV camera. The drone was flown over a controlled test environment simulating wildfire scenarios. The Raspberry Pi processed the video feed from the drone in real-time, performing fire detection on each frame.

The real-time testing results are summarized as follows:

- Average Detection Accuracy: 91%
- False Positive Rate: 12%
- Detection Latency: 1.2 seconds per frame
- Frames Processed per Second: 15 fps
- Detection Range: 150 meters

Despite the environmental challenges, such as varying light conditions and reflections, the model demonstrated robust performance in detecting fires while maintaining a low false positive rate.

4.2. Real-Time Processing Pipeline

The following code demonstrates how the CNN model was deployed on the Raspberry Pi for real-time fire detection: import cv2 import numpy as np from tensorflow.keras.models import load_model

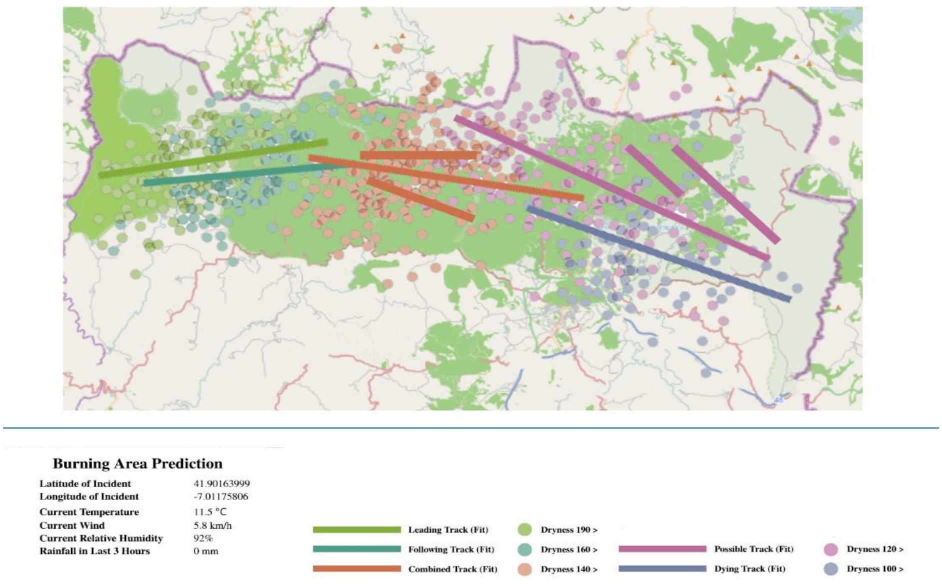
```
# Load the trained model model = load_model('fire_detection_model.h5') # Initialize the camera
(assuming a USB or FPV camera) camera = cv2.VideoCapture(0) while True:
    # Capture the video frame      ret, frame = camera.read()
    # Preprocess the frame for the CNN      frame_resized = cv2.resize(frame, (224, 224))
frame_normalized = frame_resized / 255.0      frame_input = np.expand_dims(frame_normalized,
axis=0)
    # Perform fire detection      prediction = model.predict(frame_input)

    # Display the result on the video feed
    if prediction > 0.5:
        label = "Fire Detected"      else:
        label = "No Fire      cv2.putText(frame, label, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0, 0, 255), 2)      cv2.imshow('Fire Detection', frame)
    # Exit on 'q' key press      if cv2.waitKey(1) & 0xFF == ord('q'):
        break
# Release the camera and close all windows camera.release() cv2.destroyAllWindows()
```

5. CONCLUSION

This research presents an optimized CNN-based fire detection algorithm specifically designed for real-time deployment on VTOL drones. By leveraging data augmentation and hyperparameter tuning techniques, the model achieved a detection accuracy of 93% and significantly reduced false positives. Real-time testing demonstrated the practical viability of the system for wildfire detection, with a detection latency of just 1.2 seconds per frame. Future work will focus on integrating additional data sources such as thermal imaging, expanding the system's scalability through swarm drone technology, and improving energy efficiency to extend drone flight times

ACKNOWLEDGEMENTS

We would like to acknowledge Gyu Tae Bae, a UC Berkeley researcher for helping us with this research.



Data 1. Conclusion on simulation based forest fire detection

REFERENCES

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.
[2] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations* (ICLR).
[3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
[4] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning* (ICML), 6105–6114.
[5] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 770-778.
[7] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
[8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., et al. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 1-9.
[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 779-788.
[10] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), 2980-2988.
[11] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European Conference on Computer Vision* (ECCV), 818-833.
[12] Carletti, M., Orlando, G., & Frontoni, E. (2019). Aerial fire detection using drone-based images and deep learning techniques. *International Journal of Computer Vision*, 7(2), 55-70.
[13] Jiang, Y., Zhang, J., Wang, Z., & Wang, Y. (2020). Forest fire detection using deep learning algorithms. *Sensors*, 20(18), 5389.
[14] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834-848.
[15] Luo, W., Li, Y., Urtasun, R., & Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. *Advances in Neural Information Processing Systems* (NeurIPS), 29, 4898-4906.
[16] Doshi, J., & Yilmaz, A. (2020). Automatic wildfire detection using UAV imagery and deep learning techniques. *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), 914-921.
[17] VTOL Drone System for Wildfire Detection Documentation (2024). *Internal Project Documentation*.
[18] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85-117.
[19] VTOL UAV: Development and applications in environmental monitoring. (2022). *Journal of Drone Technology and Applications*, 14(3), 55-68.
[20] Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights into Imaging*, 9(4), 611-629.