Deep learning project report

Fire-Detection Using Aerial Images From Drones

DSTI – Arrabi Teffahi

INTRODUCTION

Wildfires have become a major global challenge, not only due to the devastating impact they have on ecological resources but also because of the significant economic losses they can cause. In some countries, wildfires pose a direct threat to human life. This highlights the critical need for effective solutions to detect and control wildfires early.

In this project, we developed a deep learning model designed to detect wildfires using aerial imagery captured by drones. The dataset used for this task is publicly available and known as the FLAME dataset. Our model is based on a Convolutional Neural Network (CNN) architecture, inspired by Google's Xception model, with certain modifications to suit the specific needs of this application.

To evaluate our model's performance, we used precision and recall metrics, which are well-suited for this type of classification task. The entire training process was tracked using Microsoft Azure Machine Learning Studio, where we utilized a local machine equipped with an RTX 3060 GPU and CUDA for accelerated training. A custom dashboard was created in Azure ML Studio to monitor and manage the model's versions, architecture, data, code, and performance metrics over time.

Once trained, the model was deployed to an Azure ML Studio cluster endpoint and integrated into a Streamlit web application for real-time wildfire detection. All the code and resources related to this project are available on our GitHub repository.

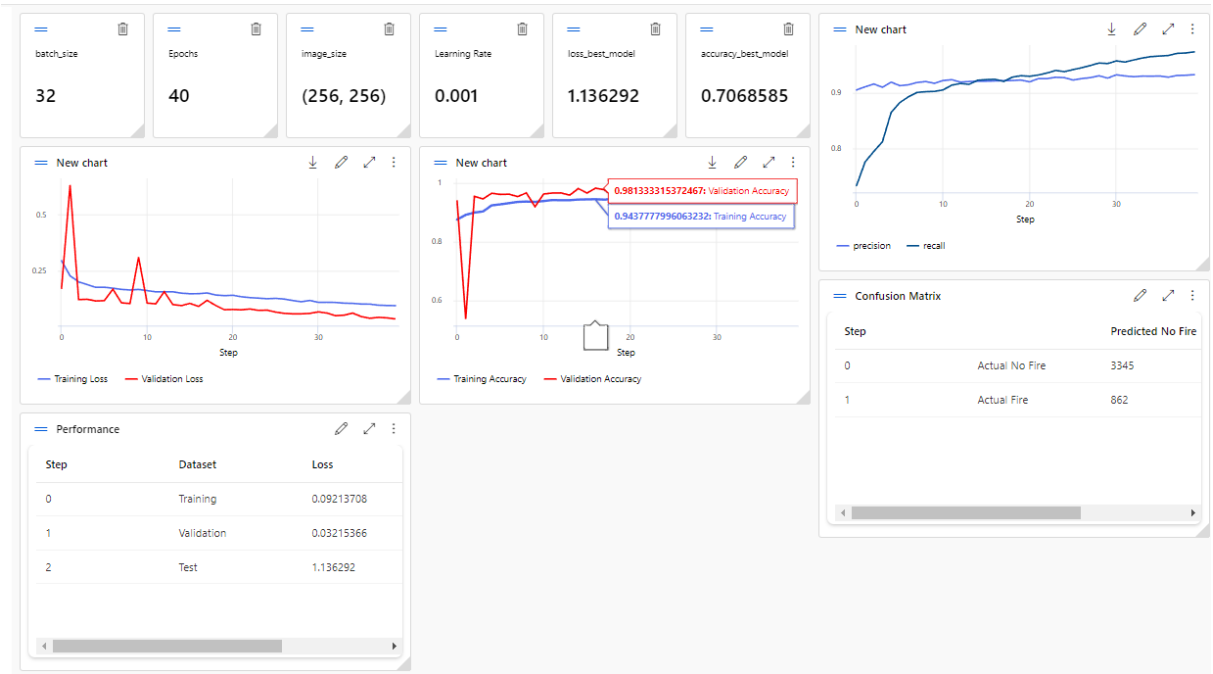
Our report is organized as follows: in the first section, we discuss how the model's evaluation was monitored using Azure Machine Learning Studio. The second section details the model architecture, followed by the third section, which presents the results. Finally, we describe the deployment method in the last section, concluding with a summary of our findings.

MODEL MONITORING

In this section, as mentioned in the introduction, we will discuss how we monitored our deep learning model.

To begin with the technical setup, we first created an Azure Machine Learning Studio resource on Microsoft Azure. We then established a dedicated workspace for our project. The next step involved integrating the Azure ML Studio SDK into our code, allowing us to track and log various aspects of the model directly into Azure ML Studio. By utilizing the *Experiments* feature in Azure ML Studio, we were able to monitor our model's progress while using local computation (local cluster) for the training phase, which helped us avoid additional Azure costs.

For model evaluation, we selected recall and precision as the primary metrics. Given that the classification problem is binary (fire vs no fire), we prioritized a high recall and slightly lower precision to ensure the model performs well in detecting wildfires. In addition to these metrics, we also tracked accuracy for both the training and test sets. For each experiment, we recorded the accuracy corresponding to the best model epoch. Figure 1 shows an example of the monitoring dashboard we created to track model performance.



THE MODEL ARCHITECTURE

Image classification is a challenging task in the field of image processing. Historically, traditional image processing techniques relied on RGB channel comparison to detect objects like fire in images or videos. However, these methods are often prone to errors and false positives. For instance, methods that compare RGB values based on a threshold to detect fire may incorrectly identify scenes such as sunsets or sunrises as fire. Deep neural networks (DNNs), on the other hand, are able to learn complex features and patterns that are not directly related to fire, improving accuracy in classification tasks.

In some studies, pixel-based classification and segmentation are performed using the HSV (Hue, Saturation, Value) format. In our study, we adopted a supervised machine learning approach to classify frames captured by drones. When fire and non-fire regions coexist in a single frame, the entire frame is labeled as "fire," whereas frames without any fire are labeled as "non-fire." Instead of using green or fusion heat maps, we used the aerial images from the FLAME data set

The binary classification model used in this study is based on the Xception network, which was proposed by Google-Keras. Xception is a deep Convolutional Neural Network (DCNN) that replaces the standard Inception modules of the Inception architecture with depth-wise separable convolutions. This modification results in improved performance and efficiency. Figure 2 provides an overview of the Xception model structure.

The Xception model consists of three main blocks:

1. **Input Layer**: The input layer size is determined by the image dimensions and the number of channels, which in our case is (254 × 254 × 3). The RGB values in each channel are scaled to float values between 0 and 1.

2. **Hidden Layers**: The hidden layers are based on depth-wise separable convolutions with shortcut connections between convolution blocks (similar to the ResNet architecture). The entry flow begins with two 2D convolutional blocks, each with a filter size of 8 and a stride of 2 × 2, followed by batch normalization and a Rectified Linear Unit (ReLU) activation function. Batch normalization helps speed up training and introduces randomness by reducing the reliance on initial weights, regularizing the model. After this, the model progresses through two separable 2D convolutional blocks. The final block in the hidden layers is another separable 2D convolutional layer, followed by batch normalization and ReLU.

3. **Output Layer**: As fire detection is a binary classification task (Fire/No Fire), the output layer uses a Sigmoid activation function to produce probabilities for each

class. The loss function for this binary classification problem is binary cross-entropy.

RESULTS

The total dataset for this project consists of 39,375 frames, with 25,018 labeled as "fire" and 14,357 labeled as "non-fire." The dataset was split into 80% for training and 20% for validation, with all frames shuffled prior to being fed into the network. To address the imbalance between the two classes, data augmentation techniques, such as horizontal flipping and random rotation, were applied to generate new frames and reduce bias.

The model was trained over 40 epochs, with the learning rate for the Adam optimizer set to 0.001, which remained constant throughout the training phase. A batch size of 32 was used to fit the model during training. For evaluation, a separate test dataset of 8,617 frames was used, comprising 5,137 fire-labeled frames and 3,480 non-fire-labeled frames. These frames were fed into the pre-trained network to assess the model's performance. Figure 3 reports the loss and accuracy for the training, validation, and test sets.

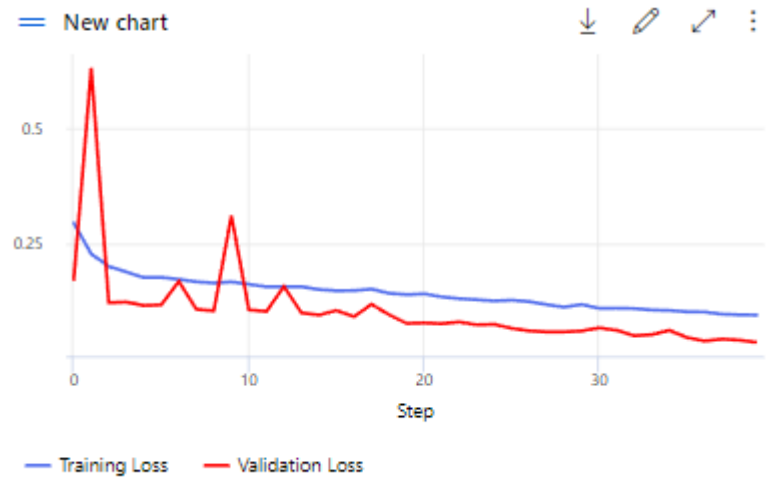
Performance

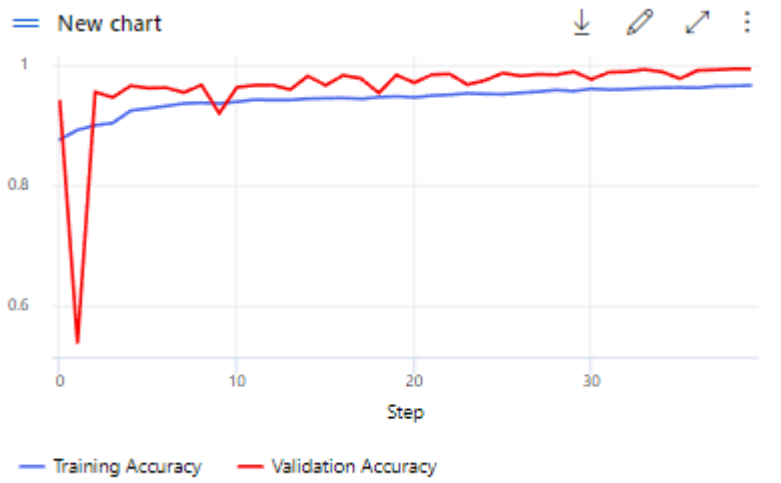
Dataset	Loss	Accuracy
Training	0.09213708	0.9647301
Validation	0.03215366	0.9920000
Test	1.136292	0.7068585

It is important to note that the training frames were collected using a equiped drone 1, while the test frames were captured using a equiped drone 2. This ensured no overlap between the training and test sets, confirming that the model's performance was not biased by the imaging equipment. Therefore, we can expect the model's accuracy to be even higher if the same imaging conditions were used for both training and testing. The achieved accuracy for the "Fire vs No-Fire" classification task was 70.23%.

Figure 4 and figure 5 illustrates the training and validation loss and accuracy throughout the training process, while Figure 6 presents the confusion matrix for the test set. The vertical axis of the confusion matrix represents the true labels of the frames, and the horizontal axis shows the predicted labels. Since the dataset was imbalanced during

training, the model exhibits a higher false positive rate (classifying a non-fire frame as fire) compared to the false negative rate (classifying a fire frame as non-fire).





Confusion Matrix

	Predicted No Fire	Predicted Fire
Actual No Fire	3345	1792
Actual Fire	862	2618