WILDFIRE PREDICTION ANALYSIS

INTRODUCTION

Wildfire has emerged as one of the major causes for the destruction of human material and lives. This usually occurs when a large forested area catches fire, which becomes out of control in a few minutes. In this regard, I have analyzed a publicly available dataset from Kaggle, which consists of wildfire images into training, testing, and validation sets. A custom CNN and a pretrained VGG16 model were trained on this dataset to make predictions for the area which has a high likelihood of catching fire or not. This is the binary classification problem, and models are tested to predict the class of image as either wildfire or no wildfire.

LITERATURE REVIEW

This is one of the problem statements which I have worked on in my past experiences. Hence, there is not much research needed to understand the problem statement and its significance. However, I have reviewed some research papers to analyze the work required in processing the wildfire images and the suitable model to process these images, which you will likely see in the reference sections. The process of making CNN models efficient for image classification is explained with 3 different datasets in [1], and the idea of implementing transfer learning using pretrained CNN models can be found in [2]. This paper highlights the performance improvement of the model using pretrained weights instead of training from scratch. Different computer vision techniques are applied to overcome the challenge of facing destruction caused by the wildfire. This analysis is described in [3], in which they work on a dataset of images covering smoke from the forested area, which is likely to happen due to the cause of fire. These research studies help in designing the architecture of the CNN model and using pretrained weights to overcome the challenges of training the model.

METHODOLOGY

1. DATA STATISTICS

The wildfire prediction dataset consists of satellite images of size 350 x 350 pixels and are divided into two classes: wildfire and no wildfire. The dataset contains 22,710 images of wildfire and 20,140 images of no wildfire. The dataset is split into 70% training set, which includes 30,221 images, and 15% validation set, which includes 6,295 images. The remaining 15% is the testing set of 6,300 images.

2. DATA CLEANING

The dataset contains duplicate and corrupt/truncated images, which creates hindrance in the training of the model. Hence, during the data cleaning process, the data is processed to identify the presence of duplicate or corrupt images, if any, and they are removed from the dataset. Using duplicate data during the model training will only result in wastage of resources, and the model will not be able to generalize. So, removing duplicate images can help with increasing the model's stability and learning process.

3. DATA PREPROCESSING

Wildfire image dataset is then preprocessed to make it suitable for the model training process. The preprocessing of images is carried out using the *ImageDataGenerator* class of the TensorFlow module. The images were scaled to the same pixel sizes [0, 1], and then a few annotation techniques were applied on the training images like flipping and rotating, only for making the model generalized to unseen data. The pixel values of the images are then normalized to a scale of [-1, 1]. The images are then loaded from each directory of training, validation, and test set to use for the model training process. Random shuffling of data is allowed for training images only, so that the model can learn the features correctly. The dataset is of large size, so it is divided into batches of size 128 to process in each iteration of the training process. The target size of the images is set to 350 × 350 to help in designing the architecture of the model.

4. MODELING

- **CUSTOM CNN MODEL**

The CNN model is used for the binary classification problem, as it is one of the most efficient machine learning algorithms for image classification. The model architecture consists of three convolutional 2D layers with multiple filters of 5x5 and 3x3. The convolutional layers use the ReLU activation function at the output layer, and padding is applied to focus on the pixel values. Each convolutional layer is followed by a max pooling layer to downsample the size of the output feature map. Batch normalization is applied to the internal layers of the network after the max pooling layer to normalize the inputs of other layers, as it helps in faster model training and its stability.

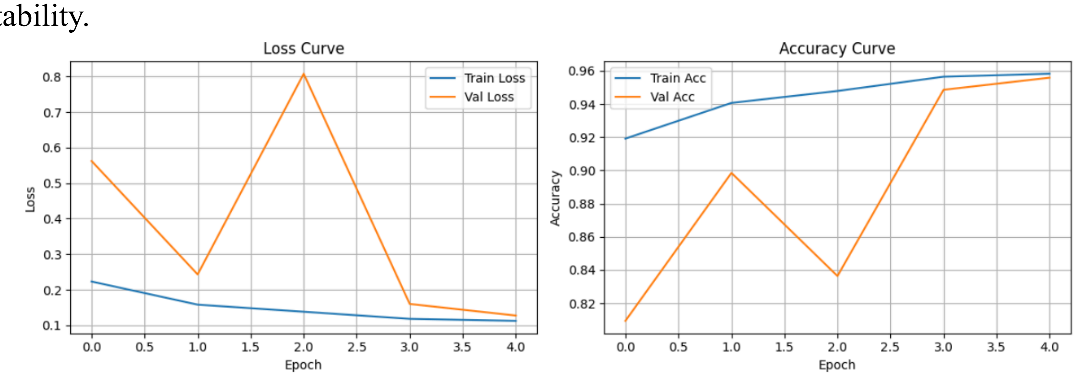

Fig 01: Training curves of CNN

The last layer of the network consists of flattened and dense layers with ReLU activation function for the non-linearity in the network. The final layer also contains a dropout layer to overcome the overfitting challenges faced during training when models try to learn the features instead of generalizing them to unseen data. The output layer is linked with a sigmoid activation function to result in 0 and 1, as the model is trained on 2 classes only. The 0 will predict no wildfire class, and 1 will result in wildfire class. The whole CNN architecture is compiled using binary cross-entropy loss efficiency for the binary classification problems, with the Adam optimizer and accuracy as the metrics to evaluate the model’s performance during training and on validation data. Due to the large number of images in the training and validation set, the model is trained for 5 epochs only due to resource and time constraints. The loss and accuracy of the model during training is shown in Fig. 01.

- **PRETRAINED VGG16 MODEL**

Transfer learning is applied on a pretrained model of CNN, which is imported from the TensorFlow library. The trained weights of the VGG16 model are imported with the include_top value set to false, as we want to avoid the softmax classification because we are working with 2 classes only. The ImageNet model's trained weights are used to avoid the training process, and a custom layer is added on the output layer with a sigmoid activation function to make it a binary classifier.

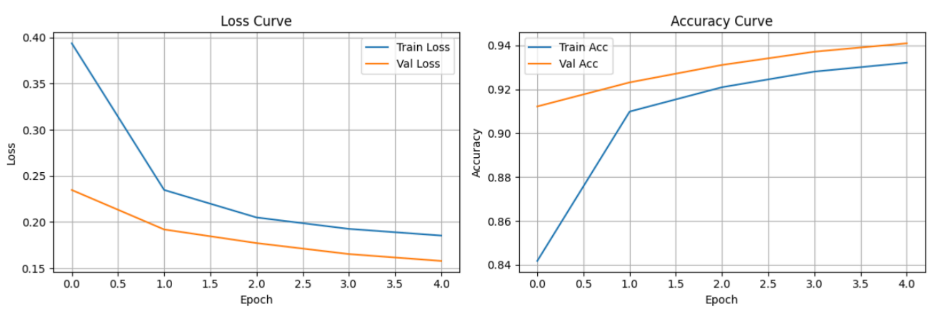

Fig 02: Training curves of pretrained VGG16

The model is then evaluated on the test data with the evaluation metrics of accuracy, precision, recall, and F1 score. The whole performance of the model is displayed through the classification report. The pretrained model is used to minimize the time spent training the model from scratch. The pretrained weights of ImageNet are introduced to extract the features of this wildfire image dataset, and then predictions were made with these learned features on the unseen test data. Overall, both the custom CNN and pretrained VGG16 model show accuracy above 90 on this wildfire image dataset. The training curves of VGG16 are shown in Fig. 02. This shows that pretrained VGG16 shows smoother results as compared to custom CNN due to pretrained weights. More discussion on results will be provided in the next section.

RESULTS

Both models show high accuracy on the wildfire image dataset, but the training curves show that the pretrained VGG16 model performs better than CNN due to pretrained weights, and it learns the features smoothly, resulting in higher accuracy. On the other hand, CNN curves show a spike, which is due to the cause of overfitting. Another major cause of performance failure is due to fewer epochs in training. The model performance can be improved by training it on a large number of epoch values or introducing the regularization parameters. Below, Table 01 shows the performance of both models using quantitative metrics for the classification problem.

Evaluation Metrics	Custom CNN	VGG16
Accuracy	0.9627	0.96
Precision	0.9619	0.95
Recall	0.9710	0.95
F1 Score	0.9664	0.96

Table 01: Performance Evaluation

Fig 03 shows the confusion matrix of CNN on the left side and the confusion matrix of VGG16 on the right side. Both models' performance shows that they are predicting the true labels correctly and false labels incorrectly. The high rate of true positive predictions indicates that the model is likely to suffer from overfitting, as the images in the dataset are noisy and corrupt. According to the quality of the images in the dataset, the model should not be able to perform well on 5 epochs of training. This shows that the model is not trained correctly, which requires introducing techniques to overcome overfitting, like increasing the dropout rate or adding regularization, etc. Further training is required to make the model perform well on the dataset.

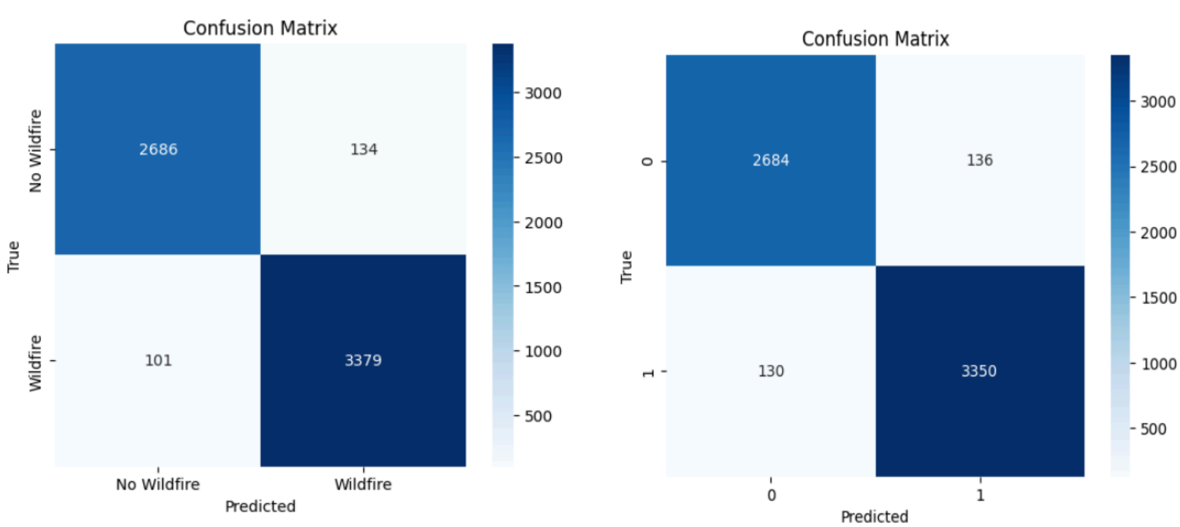

Fig 03: Performance comparison using Confusion matrix

CONCLUSION

The task of predicting the presence of wildfire in the images is performed using CNN and pretrained VGG16 model. Both models show high accuracy of above 90% on the wildfire image dataset. However, the key findings of this research show that the models are suffering from overfitting, and instead of generalizing their predictions to unseen data, the model is learning the features of the data during the training. Hence, the performance of the model can be improved by training the model on a large epoch and introducing techniques to overcome overfitting.
One of the major limitations faced during the research was the insufficient quality of images in the dataset. Images are taken from satellites, due to which most of the images appear noisy and corrupt, which creates issues during the training process. Another constraint is of the resource and time, as the image dataset is of large size, and it will require training of a day or two for large epoch values like 50 or 100. These limitations can be overcome to make the models give better results. The overall work highlights the importance of correctly detecting the presence of wildfire so that the issue can be reported timely to avoid damage on a large scale.

REFERENCES

1. A. Sharma, "Learn Image Classification with CNN in PyTorch using 3 Datasets," *Analytics Vidhya*, Feb. 24, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/
2. S. Maggioros and N. Tsalkitzis, *Wildfire danger prediction optimization with transfer learning*, arXiv preprint arXiv:2403.12871, 2024. [Online]. Available: https://arxiv.org/abs/2403.12871
3. E. R. Daniel, "Wildfire Smoke Detection with Computer Vision," *arXiv preprint arXiv:2301.05070*, 2023. [Online]. Available: https://arxiv.org/abs/2301.05070