Smoke & Non-smoke Image Classification

CS484 Term Project Final Report

Melike Demirci
Dept. of Computer Science
Bilkent University
Ankara, Turkey
melike.demirci@ug.bilkent.edu.tr

Yuşa Babademez
Dept. of Computer Science
Bilkent University
Ankara,Turkey
yusa.babademez@ug.bilkent.edu

***Abstract*— Forest Fires are increasingly take place in many different countries and threating peoples and many other living creature's lives. Early detection of the fire is significant to decrease the damage. Computer vision technologies have been used widely in order to detect and classify these dangerous situations from their images. The purpose of this study is to examine the binary classification of the forest smoke images with Convolutional Neural Networks (CNN). ResNet-50 and some custom build models were trained with different parameters. The effects of different hyper parameters were examined.**

Keywords—Image Classification, Convolutional Neural Network, ResNet-50, Forest Fire Smoke

I. INTRODUCTION

Forest fire is a significant threat to forests or wooden areas. Increasing number of fire incidents leads to damaging biodiversity, disappearing habitat of animals, and losing animal or human lives. Early fire detection is very significant in terms of saving lives and habitat since it is hard to put it out when it spreads[1].

In this project, our aim was classifying images in terms of two classes such as fire and smoke by using ResNet-50 Convolutional Neural Network(CNN) architecture. By conducting many training with different parameters, the effect of different parameters on performance of the model were planned to be observed. Images were planned to be taken from Forest Fire dataset [2].

After planned experiments conducted we have observed that classifying the images in Fire dataset as fire and smoke were not a complex task for the trained CNN and accuracy of the model reached 100% very quickly. Due to this reason, we have decided to modify our dataset and class labels. Instead of classifying images as fire and smoke, we have trained and experimented models which classify images as smoke and non-smoke.

The rest of the paper is organized as follows. In the following section we present the further information about our modifications. In section 3 we present new dataset that we have used. In section 4, our study approach is given. Section 5 presents the acquired results and evaluation of them.

II. MODIFICATIONS

After examined our previously selected dataset more, we have realized that images from two classes have very different aspects in terms of RGB values, patterns, local features etc. In order to find out variance of RGB values in between fire images, smoke images and RGB variance between these two classes, we have calculated the covariance as following formula:

$$Cov(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})$$

Fig. 1. Formulation of Covariance

Covariance values of two images from same and different classes can be seen in Figure 2.

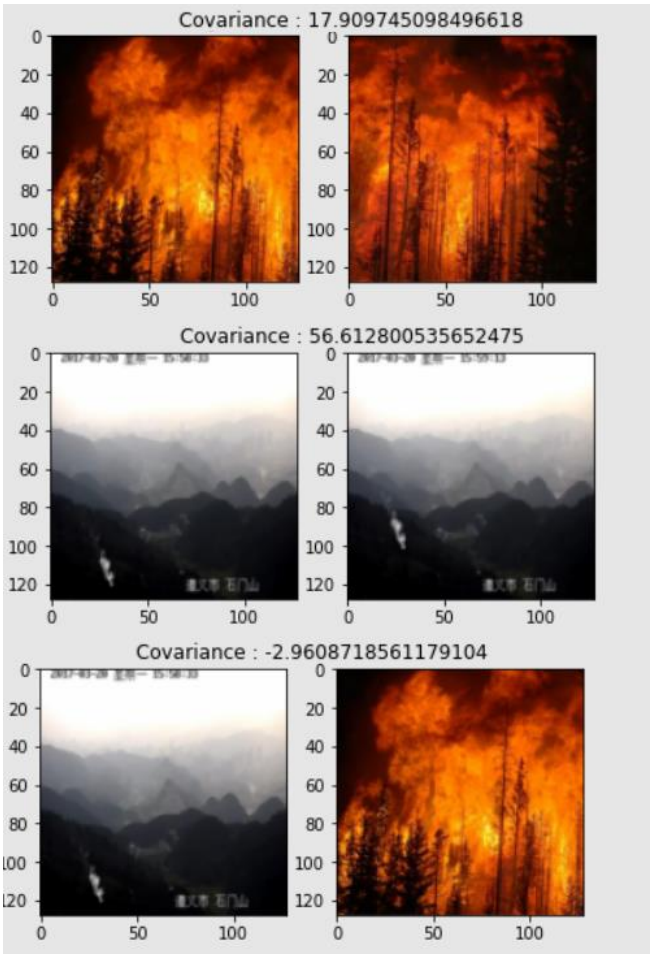

Fig. 2. Covariance of Images (fire-fire, smoke-smoke, fire-smoke)

Covariance value shows how much the variations of two variables are related. When there is a positive covariance between two variables it means that the paired values of both variables tend to increase together. On the other hand, negative covariance reveals that there is an inverse relationship between the variables [3]. As it can be seen from Figure 2, there is an inverse relationship between fire and smoke images, covariance value is -2.96. Images from the same classes have higher covariance values which shows that they are correlated in terms of RGB values.

Due to this high variance between classes, our dataset was considered to be very easy to be classified. In order to conduct experiments in hyper parameters, we needed another dataset which would not be easily classified. For that reason, we have changed our target of classifying from fire and smoke classification to smoke and non-smoke classification. Details about the new dataset is given in the next section.

III. New Dataset

Due to the change in the class labels, we needed to find a new dataset for our experiments. We already have some smoke images from our previous dataset but we needed also landscape images without smoke. There was not already prepared dataset available online so we needed to collect images from different datasets. We have collected images from 4 different sources [1][4][5][6]. Then we put them in separate folders according to their class labels.

As a result of the data collection process we have obtained 326 images for smoke class and 372 images from non-smoke class. In order to prevent imbalanced classification problem, which occurs when the distribution of examples is uneven, we have used 320 images from both of the classes. Due to the limited number of images, we have decided not to divide dataset into train-validation-test parts. Instead, we have divided our dataset only to training and validation sets as 20% validation and 80% training sets. Example images from both classes can be seen in Figure 3.

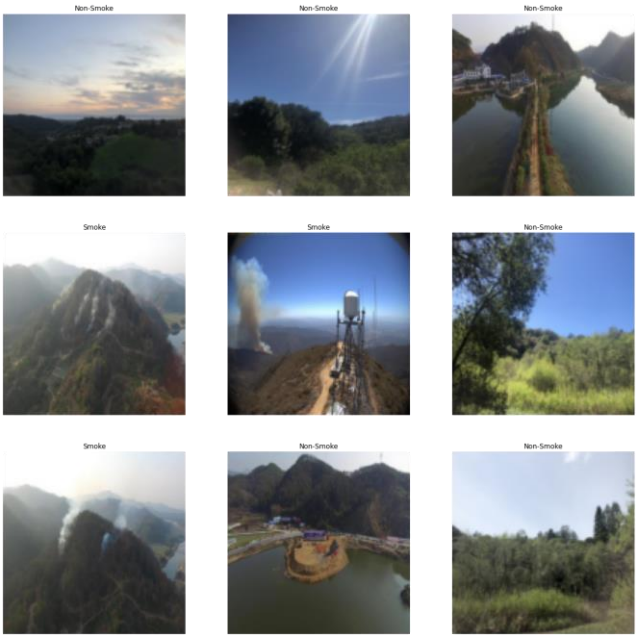

Fig. 3. Example Images from New Dataset with Their Labels

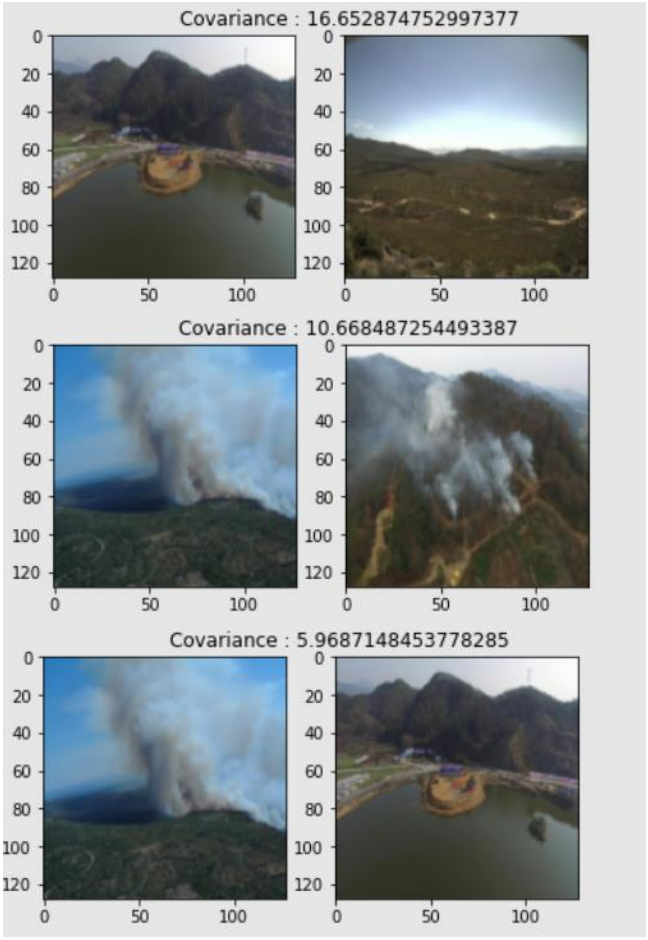

Fig. 4. Covariance of Images (non-smokes, smokes, non-smoke and smoke)

We have also checked the covariance values of our new dataset in order to prevent the problem we have faced with the previous dataset. As it can be seen from the Figure 4, there is no negative covariance value which shows that there is no inverse relationship between the images of two classes.

IV. Approach

Our first intention was to conduct experiments with the ResNet-50 models without using any pre-trained weights. However, this training process required a lot of RAM. We have used Google Colab for the trainings. Free version of the Colab did not have enough RAM for training of the model without pre-trained weights. When we tried to conduct training we encountered runtime errors. For this reason, we had to use pre-trained weights. When we checked the class labels of the pre-trained ResNet-50 weights we found that these weights are trained on Imagenet dataset [7]. There was no smoke class in the Imagenet dataset, that is why we were able to use it in our model.

A. Experiments with ResNet-50

For the training of the ResNet-50 model for our dataset, we have created a model with the ResNet-50 architecture, we did

not include the top layers. As a top layers we added dense layer with sigmoid activation function. Then we set the trainability of the ResNet-50 layers to false in order to train only the top layers. Number of trainable and non-trainable parameters can be seen in Figure 5.

```
Total params: 23,589,761
Trainable params: 2,049
Non-trainable params: 23,587,712
```

Fig. 5. Number of Parameters for ResNet-50

As a result of training and validation, we have obtained 0.8359 validation accuracy from the model.

We have trained different models with the different batch sizes such as 8,16,32,64 but we could not observe an improvement. Our first model was trained with batch size 16 and that has the best performance in terms of validation accuracy.

We tried to increase validation accuracy by changing the pooling type of the model. Our first model was having the average pooling, we also trained model with max pooling. However, maximum validation accuracy obtained did not improved, it decreased.

B. Experiments with Custom CNN's

After conducting experiments on ResNet-50 model we have decided to build our custom CNN model in order to improve performance of the model. Despite to the changes made on ResNet-50 hyper parameters, validation accuracy did not improve more than 84%.

For our custom CNN models, we have used similar structures adopted by many other CNN models. We used 3 layered models; in every layer there is a 2D convolution with kernel size (3,3) and there is a batch normalization, max pooling and dropout layers after convolution layer. Number of filters is increasing from bottom to top layers and at the end of this 3 layers there is a dense layer with sigmoid activation function.

First, we have trained a model which has filter sizes in layers as 2, 4 and 8. In this training batch size were 16, number of epochs were 500, optimizer was Adam. As a result, we have obtained 92.97% validation accuracy. That was a good improvement compared to the ResNet-50 performance.

Thus, we wanted to conduct an experiment with the size of our custom model in terms of number of filters used in layers. We have built 2 more models; one of them has 4,8,16 filters and the other has 8, 16, 32 filters. As a result of this experiment we have discovered that biggest model performs best with 96.88% validation accuracy. Medium sized model performed worst. Validation accuracies during the training can be seen in Figure 6. Also number of parameters of the models can be seen in Figure 7.

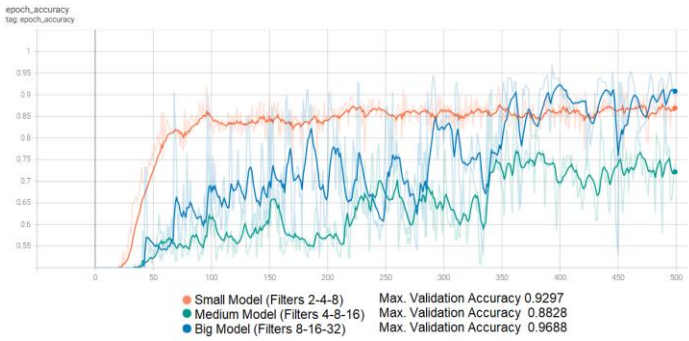

Fig. 6. Comparison on Custom Models with Different Sizes

From Figure 6, it can be seen that small model learns faster than the others. It reaches better validation accuracies in the earlier epochs. However, it does not learn more than the big model. Medium model acts more like the big model but performs worse than it. For the big model, it takes longer time to learn the data but the best performance obtained with this one.

Big	Total params: 12,529 Trainable params: 12,417 Non-trainable params: 112
Medium	Total params: 4,825 Trainable params: 4,769 Non-trainable params: 56
Small	Total params: 2,053 Trainable params: 2,025 Non-trainable params: 28

Fig. 7. Number of Parameters for Custom Models with Different Sizes

When we compared the number of trainable parameters of our custom model and ResNet-50 with pre-trained weights, we realized that our small model has very similar to ResNet-50 in terms of number of trainable parameters. In order to compare these models, we have conducted trainings.

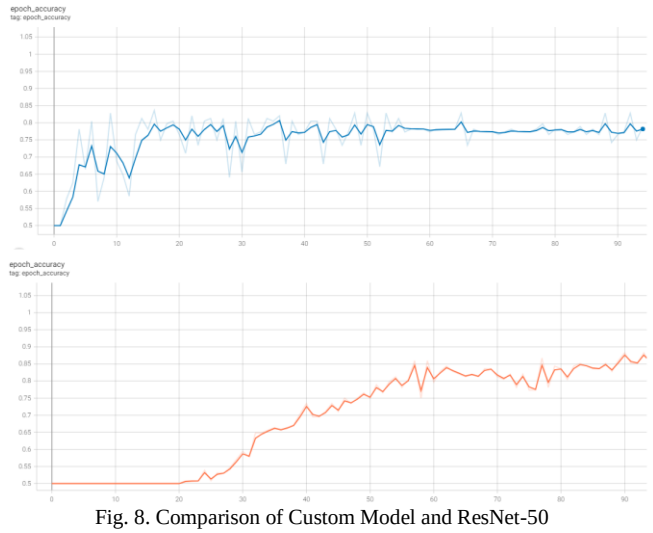

Fig. 8. Comparison of Custom Model and ResNet-50

Validation accuracies during the training of models are given in Figure 8, blue graph belongs to the ResNet-50 and

orange graph belongs to our custom model. As it can be seen, ResNet-50 model reaches higher validation accuracies in the earlier epochs compared to the custom model. This is a result of pre-trained weights. However, maximum validation accuracy obtained from the models are different. Custom model performs better with 92.97% validation accuracy compared to ResNet-50 model with 83.59% validation accuracy.

After conducting an experiment on model size, we decided to have an experiment with batch sizes. We have used big model for this purpose and did not changed any of the other parameters other than the batch sizes. We have conducted trainings with batch sizes of 16, 32, 64, 128. There was no difference in the performance of models in terms of validation accuracies. However, we observed that smaller batch size causes slower learning. From Figure 9, it can be seen that model trained with 16 batches reached its best validation accuracy in the 423rd epoch in 500 epochs. On the other hand, model trained with 32 batches reached almost same validation accuracy in the earlier epoch which was 275th epoch. Models with batch sizes 64 and 128 did not performed better than the model with 16 batches. Best choice for the batch size decided as 32.

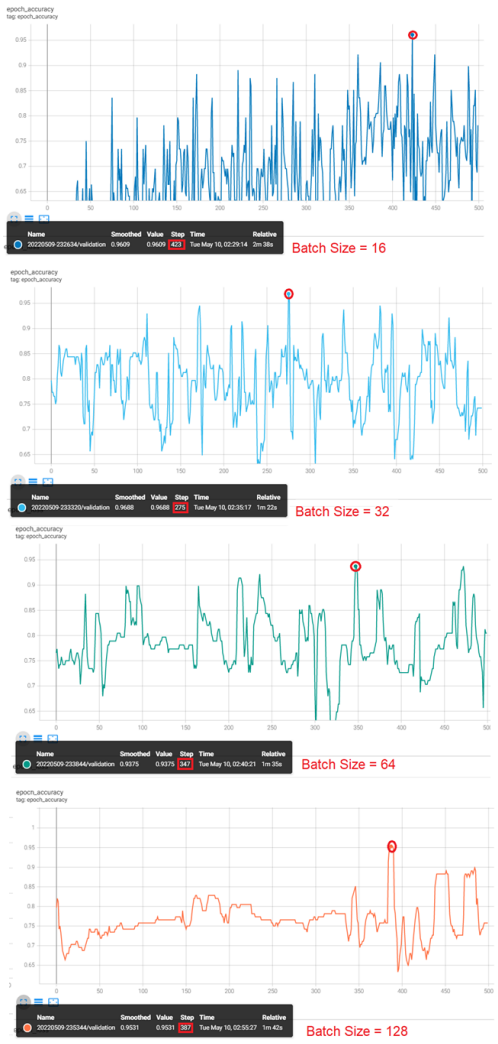

Fig. 9. Experiment on Batch Size with Custom Model

V. Results

As a result of all the experiments, best hyper parameters that we obtained as following:

- Model Architecture: Custom CNN model with 3 layers with number of filters as 8-16-32.
- Batch Size = 32
- Optimizer = Adam
- Activation Function = Sigmoid

Maximum validation accuracy obtained with this model is 96.88%. After training the model, we conducted some predictions with the images. Some images with their predicted classes can be seen in Figure 10.

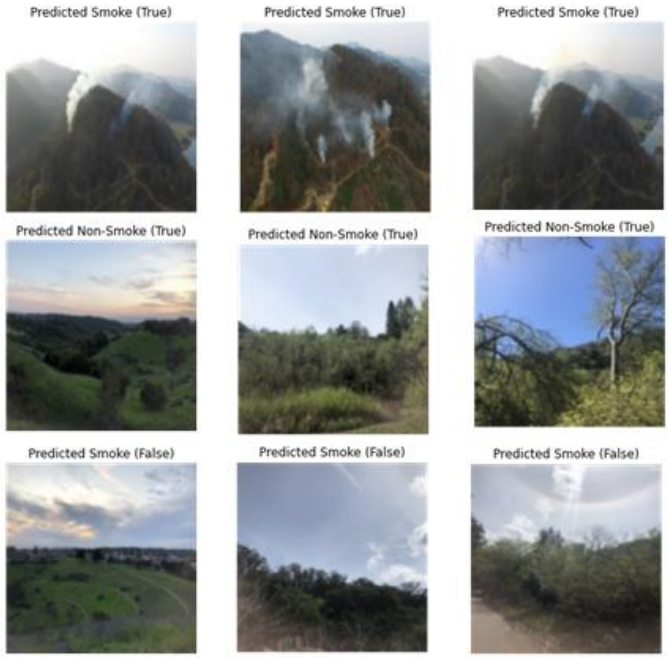

Fig. 10. Some Prediction Results with Best Model

VI. Findings

Before sharing the findings about our experiments, we should mention that we have learned a lot about dataset selection. Having a images from different classes which has significantly inverse relationship could result an very easy dataset. For these easy datasets there is no meaning to perform CNN trainings, they can be classified with simpler models.

As a result of our experiments there are some points that should be mentioned. First one is the performance of ResNet-50 model compared to the custom model. ResNet-50 model with pre-trained weights did not improved more than 82% validation accuracy despite the hyper parameter tuning while custom model could be improved up to 96% validation accuracy. We think this is due to the size of our dataset and the number of classes we have. Even though we did not train the all weights in the ResNet-50 model, while doing predictions all the weights are used. Pre-trained weights of ResNet-50 is obtained with the training conducted on Imagenet dataset which includes 14

million images with 1000 classes. Our dataset is significantly smaller than Imagenet dataset and we have only 2 classes such as smoke and non-smoke. Due to this reason, ResNet-50 model could be too complicated for out dataset. With smaller number of total parameters, we have obtained better performances.

When we observed the results we have found some false positive predictions. When we check the predictions for 20 images we saw that there was no false negative. Which means that our model does not falsely classify a smoke image as a non-smoke image. When there is a smoke in the image, it definitely classifies the image as a smoke image. However, our model classifies some non-smoke images as smoke images. When we observed these false positive results we have observed that there are clouds in the sky which might has a smoke like structure. This might be the reason behind the false positive results. In order to further improvements, this challenging problem should be resolved.

REFERENCES

[1] Zivanovic, Stanimir & Zigar, Darko & Krstić, Dejan, "The Role of Early Detection of Forest Fire in Environmental Protection, Safety Engineering," Safety Engineering, 2014.

[2] "Forest Fire Dataset," Kaggle. [Online] Available: https://www.kaggle.com/datasets/kutaykutlu/forest-fire?resource=download.[Accessed: Mar. 21, 2022].

[3] King, A.W. and Eckersley, R., 2019. Statistics for biomedical engineers and scientists: How to visualize and analyze data. Academic Press.

[4] "FIgLib Dataset," Hpwren.ucsd.edu. [Online] Available: http://hpwren.ucsd.edu/HPWREN-FIgLib/.[Accessed: Apr 16, 2022].

[5] "Open Wildfire Smoke Dataset," Github. [Online] Available:https://github.com/aiformankind/wildfire-smoke-dataset. [Accessed: Apr 16, 2022].

[6] "Wildfire Smoke Dataset," Roboflow.[Online] Available: https://public.roboflow.com/object-detection/wildfire-smoke/1. [Accessed: Apr 16, 2022].

[7] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).