# Project Work in Image Processing and Computer Vision

# 1 Introduction

In recent years all over the world wildfires lead to serious damages for forests, for the environment and thus for mankind. In particular this year in Italy about 150.000 hectares of forests burned due to wildfires[1]. This problem, just like many other affecting our planet, created the need to improve tools and IT technologies to prevent this kind of damages. The rising of Deep Learning and its success in Computer Vision can be an optimal way to tackle the Forest Fire Recognition problem.

The aim of this project work is to develop a model able to satisfy the Binary Image Classification task in Forest Fire Recognition. The idea is to follow the transfer learning method, as has already been done by [2,3], and reasoning upon the errors in order to understand why the model fails in some few cases.

# 2 Preliminaries

Let's briefly introduce some focal techniques and models exploited in this project work.

## 2.1 Transfer Learning

The transfer learning concept relies upon the idea to extend properties and ability of a neural network already trained with a dataset A for a task A, to another task B with a new dataset B.
This technique is really useful dealing with small datasets for image recognition, since in the last ten years Neural Architectures for Image Classification reached amazing results. Thus, it is convenient to apply transfer learning to those pre-trained architectures to build models able to satisfy tasks like the one proposed in this project work.
Concerning the choice of the pre-trained models(a.k.a. backbones or feature extractors), the idea was to exploit Neural Networks with a reasonable dimension (for real time usage) and strong inference time.

## 2.2 Backbones

Pre-trained models used for the experiments:

1. VGG16[4]: known for the introduction of the concept of repetition of stages (a fixed combination of layers) to increase depth while keeping regularity;

2. MobileNetV2[5]: updated version of MobileNet (already known for the exploitation of Depthwise Separable Convolutions) proposing the usage of inverted residual blocks instead of classic bottleneck residual blocks;

3. EfficientNetB0[6]: network obtained with NAS (Neural Architecture Search) and following the compund scaling method, due to the intuition about the dependence between the scaling dimensions of CNNs(width, depth and resolution).

It is important to underline that VGG16 is heavy by means of MBs with respect to the other to models. Nevertheless it is used in particular for the "EBAnO" Error Analysis part, described in the section number 5. Furthermore, all these backbones are pre-trained with ImageNet(1000 classes)[7].

# 3 First Approach

As already told in the first section of this report, the purpose of the project is to develop a neural model to solve the Binary Classification problem in recognizing forest fires. In the first place, some datasets have been found from the web and a simple baseline was tested.

## 3.1 Data

The following datasets are used in a first instance:

- FIRE dataset: outdoor fire/non fire images[8]
- "Dataset for Forest Fire Detection"[9]

These two datasets are combined in a new one with 2899 samples, 1705 images with wildfires (class "0") and 1194 without wildfires(class "1").

## 3.2 Baseline

Network Structure:

- Pre-treined Backbone ( EfficientNetB0 )
- Global Average Pooling
- Dropout layer ( dropout_rate = .2 )
- Fully Connected layer (1 unit)
- activation funciton ( sigmoid )

The combined Dataset is split in Training, Validation and Test set following the classic proportion (0.8,0.1,0.1). To conclude the description of the first model, was used Adam as optimizer, a learning rate of 0.001 and BCE (Binary Cross-Entropy) as loss function.

## 3.3 Results

From the results tables the performances look great, outperforming the LeadingIndiaAI model on their test set[Table 3.3] and achieving an outstanding accuracy score on the proposed test set[Table 3.1, Table 3.2]. But maybe is better to analyze mistakes and leaks.

Table 3.1: Accuracy scores from the first approach.

| **Set** | **Loss** | **Accuracy** |
|---|---|---|
| **Training** | 0.020 | 99.70% |
| **Validation** | 0.073 | 98.97% |
| **Test** | 0.024 | 99.65% |

Table 3.2: Report from the first approach over the test set.

| | **Precision** | **Recall** | **F1 score** | **Support** |
|---|---|---|---|---|
| **"fire"** | 0.99 | 1.00 | 1.00 | 175 |
| **"non fire"** | 1.00 | 0.99 | 1.00 | 114 |

Table 3.3: Performances of the first approach compared with the model proposed by the "LeadingIndiaAI" group on their test set:

| **Model** | **Accuracy** |
|---|---|
| **"LeadingIndiaAI"** | 93.18 |
| **"Proposed(H=W=160)"** | 93.41 |
| **"Proposed(H=W=224)"** | 95.05 |

## 3.4 Error Analysis

The only one error (Fig. 3.1) from the test set suggests the lack of variance characterizing the first approach. In a real world application it will be terrible (and unconvenient) to recognize each sunset or sunshine as a forest fire. In order to understand whether the model is recidivist with this kind of inputs, it was tested with a similar image, but it behaves in the same way (Fig. 3.2), recognizing it as fire.
Maybe the data provided to the model was too restricted and poor by means of variety of scenarios. This lead to add more and different data to the dataset.

Fig. 3.1: Only one test error. Predicted as fire

Fig. 3.2: Sunset predicted as fire. Prediction: 0.006

# 4 Proposed Model

## 4.1 Data

In the second approach the first dataset, used in the previous section, is expanded with other two datasets:

- The forest fire dataset created by the LeadingIndiaAI project[3];
- A generic fire/not fire Dataset from Kaggle [10], not closely related to outdoor fires or forests.

The final dataset has in total 4862 samples, 2089 images with wildfires (class ”0”) and 2783 without wildfires(class ”1”). As already explained in the previous section, this new dataset has more variety since it containes some ”fire/non fire” images coming from different scenarios than the forest environment.

## 4.2 Model

Different configuration were tested: the addition of more than one FC layer on the top of the network, different combinations of Data Augmentation Layers ( such as Crop,Rotation,Flip,Traslation), different Optimizers (SGD, RMSprop,Adam) and also hyperparameter tuning with Random Search for dropout and learning rates.

Final Structure:

- Data Augmentation ( Random Horizontal Flip )
- Pre-treined Backbone
- Global Average Pooling
- Dropout layer
- Fully Connected layer ( 1 unit )
- activation funciton ( sigmoid )

The dataset was split like in the first approach, while the parameters used can be found in the Table 4.1. Different backbones has been tested and EfficientNetB0 achieved the best results [Table 4.2].

Table 4.1: Parameters Table for the proposed model:

| | |
|---|---|
| **Batch size** | 64 |
| **Optimizer** | Adam |
| **Learning Rate** | 1e-3 |
| **Callbacks** | Early Stopping |
| **Dropout Rate** | .1 |
| **Loss** | BCE |

Table 4.2: Backbones comparison on the test set for the proposed model:

| **Backbone** | **Accuracy** | **F1 score** |
|---|---|---|
| **VGG16** | 96.92% | 0.974 |
| **MobileNetV2** | 97.13% | 0.976 |
| **EfficientNetB0** | 98.77% | 0.990 |

## 4.3 Results

The final model doesn't differ too much from other configurations tested, indeed in all the best cases test set accuracy always falls in the interval [98.57 +- .2 %]. From the Table 4.3 and 4.4 it is possible to see the accuracy and loss scores on the proposed dataset and the metrics on the test set.

Table 4.3: Accuracy scores from the second approach (With EfficientNetB0 as Backbone):

| **Set** | **Loss** | **Accuracy** |
|---|---|---|
| **Training** | 0.022 | 99.41 |
| **Validation** | 0.042 | 98.77 |
| **Test** | 0.043 | 98.77 |

Table 4.4: Metrics from the proposed model.

| | **Precision** | **Recall** | **F1 score** | **Support** |
|---|---|---|---|---|
| **"fire"** | 0.99 | 0.97 | 0.98 | 199 |
| **"non fire"** | 0.98 | 1.00 | 0.99 | 288 |

## 4.4 Error Analysis

The new model has not the same behaviour with sunset/sunshine images [Fig 4.1], but at same time, even if the scores are quite good, some errors are really serious[Fig 4.2,Fig 4.3].

Fig. 4.1: Sunset predicted as non fire. Prediction: 0.962

Fig. 4.2: Misclassified Fire A

Fig. 4.3: Misclassified Fire B

To deeply investigate the reasons behind those mistakes let's move to the following section.

# 5 Error Analysis - EBAnO

To investigate the reasons behind such heavy mistakes let's introduce EBAnO[11], a tool developed to produce explanations for black-box models through interpretable feature perturbations. In particular, given a CNN Image Classifier, EBAnO exploits hypercolumn representation and cluster analysis to produce a set of interpretable features for images. Therefore, through an iterative process of Gaussian blur perturbation and classification, it is able to measure the influence of input features on the final prediction of the classifier.

## 5.1 Application

In this section it's used VGG16 because of its lower number of layers with respect to other models. Practically, performing EBAnO explanation with deeper architectures (like EfficientNetB0 or many others) leads to an unsustainable RAM consumption for Colab Notebooks.

Let's focus the analysis on the misclassified wildfires A and B, since it seems to be puzzling to understand why the model doesn't recognize them. Concerning instead the remaining errors, the misclassification can be explained looking at the images: some of them have a bad quality and others contain smoke confused with clouds, features that can mislead also humans.

Therefore, A and B are classified with the VGG16 classifier: in both of the top-3 predictions is present the class "cliff" (ImageNet ID:972). Then both images, together with the "cliff" class, are provided to the LocalExplanationModel: this method from EBAnO is able to understand which parts of the input image are more correlated to the category. Looking at the images[Fig 5.1,Fig 5.2] it is evident how the flames regions are the most accountable for the "cliff" prediction, maybe due to the overlap between smoke and fire.

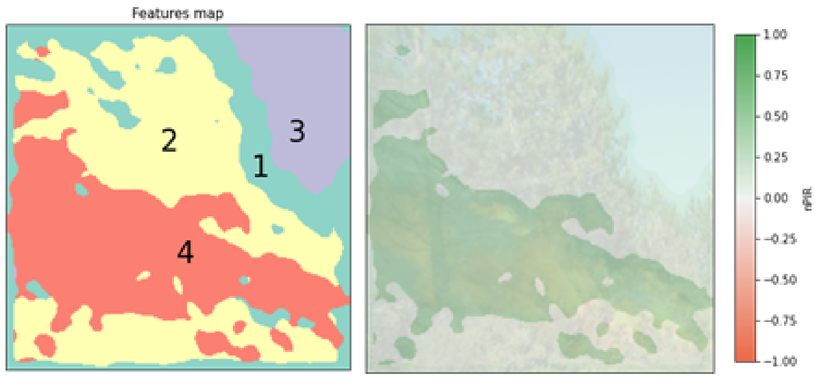


Fig. 5.1: EBAnO explanation for fire A w.r.t. the class "cliff"

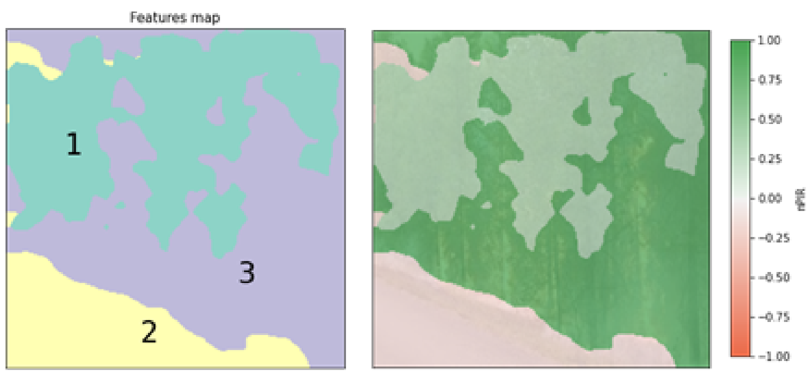


Fig. 5.2: EBAnO explanation for fire B w.r.t. the class "cliff"

Subsequently, a simple procedure was implemented to see whether the same pattern "cliff/flames" can be responsible for the msiclassification:

1. Search for other test set fire images containing ≪cliff≫ in their 5 top-predictions.
2. LocalExplanationModel to analyze possible correlations between the fire regions and the class ≪cliff≫.

Two correctly classified images, let's call them C and D, showed a similar pattern [Fig 5.3]. By analyzing the whole test set Image classification predictions, most of the samples have "volcano" (ImageNet ID:980) in their top 5 predictions. Therefore, investigating C and D with EBAnO, is possible to see that the flames already correlated with "cliff" are also correlated with "volcano"[Fig 5.4,5.5], pattern not present in the test errors A and B (as it's evident from Fig 5.6). In conclusion, the presence of the pattern "flames/cliff" and the absence of correlation between flames and "volcano" can suggest a reason behind the errors committed by the model, since A and B are the only images with this feature.

Fig. 5.3: Fires correctly classified: C and D

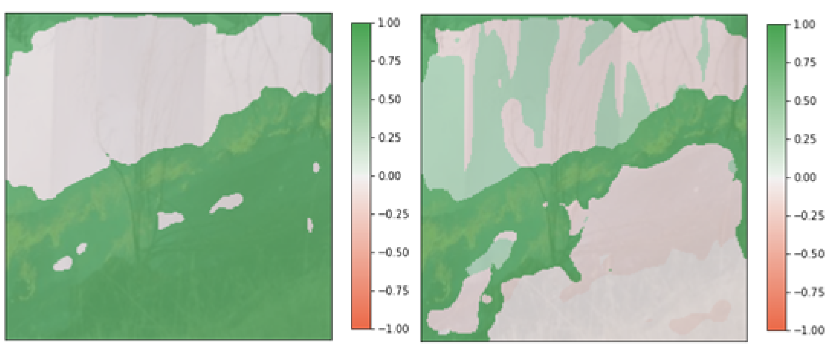


Fig. 5.4: EBAnO explanation for fire C, w.r.t. "cliff"(to the left) and "volcano"(to the right). In the computation of the feature extraction w.r.t. "cliff", the fire region is not well segmented from the rest of the image, while w.r.t. "volcano" the segmentation is clear.

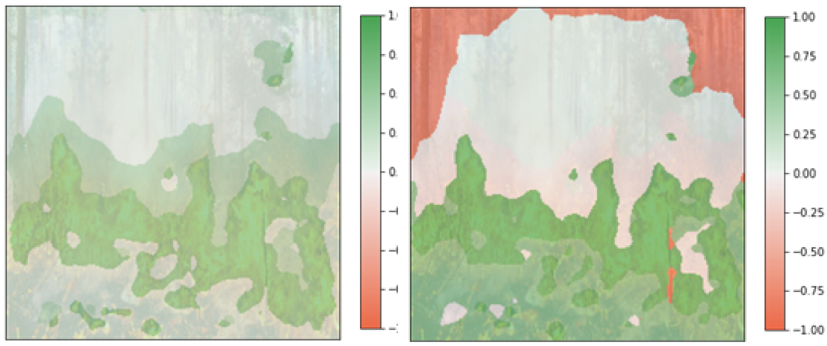


Fig. 5.5: EBAnO explanation for fire D ,w.r.t. "cliff"(to the left) and "volcano"(to the right). In the second image the fire regions are "greener" than the ones in the first image, underlying the stronger correlation with "volcano".

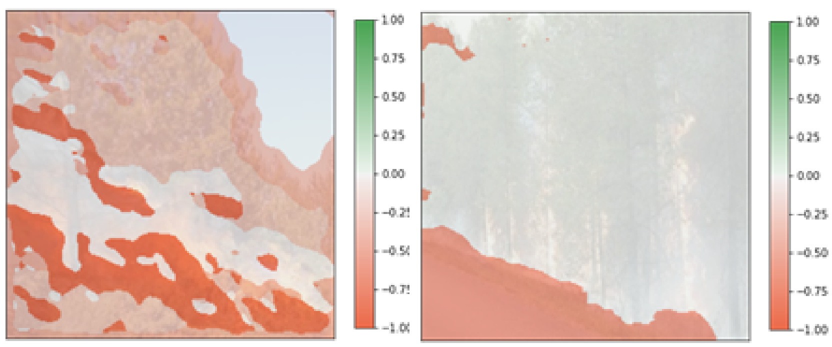


Fig. 5.6: EBAnO explanation for fires A and B ,w.r.t. "volcano". Flame regions are totally uncorrelated with the considered class.

# 6 Conclusions

Different data and approaches was used to tackle the Forest Fire Image Recognition task, reaching very high metrics scores: almost 99% of accuracy and 0.99 of f1 score. It is important to notice that despite the datasets used have many bad quality images and even some mislabelled (or ambiguous) images[Fig 6.1], the model achieves really good results. Furthermore, it was also performed an error analysis procedure to understand why the proposed model committed certain errors.

Fig. 6.1: Example of misleading samples, both labelled as ”non fire”.

## 6.1 Future Works

Concerning possible future works, they could consist in refining, balancing and extending the dataset, maybe even by using higher level Data Augmentation techniques like GANs[12] or CycleGANs[13].

Therefore, considering a real application for this model, it should be able to adapt to different environments and various situations. However, classic Deep Learning methods suffer from the so called ”catastrophic forgetting”, the tendency to forget previously learned information when learning with new data. For this reason, the Continual Learning[14] ability to adapt and learn continuously from new data could improve significantly the model.

# 7 References


1 European Forest Fire Information System https://effis.jrc.ec.europa.eu/

2 M. Rahul, K. Shiva Saketh, A. Sanjeet and N. Srinivas Naik, "Early Detection of Forest Fire using Deep Learning," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020, pp. 1136-1140, doi: 10.1109/TENCON50793.2020.9293722.

3 Forest Fire Detection through UAV imagery using CNNs https://leadingindia.ai/projectdetails/24

4 Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

5 Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4510-4520.

6 Tan, M., Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ArXiv, abs/1905.11946.

7 ImageNet https://image-net.org/

8 FIRE dataset https://www.kaggle.com/phylake1337/fire-dataset

9 Khan, Ali; Hassan, Bilal (2020), "Dataset for Forest Fire Detection", Mendeley Data, V1, doi: 10.17632/gjmr63rz2r.1

10 Fire Detection Dataset https://www.kaggle.com/atulyakumar98/test-dataset

11 Ventura, F., Cerquitelli, T. (2019). What's in the box? Explaining the black-box model through an evaluation of its interpretable features. ArXiv, abs/1908.04348.

12 Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y. (2014). Generative Adversarial Nets. NIPS.

13 Zhu, J., Park, T., Isola, P., Efros, A.A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. 2017 IEEE International Conference on Computer Vision (ICCV), 2242-2251.

14 Lesort, T. (2020). Continual Learning: Tackling Catastrophic Forgetting in Deep Neural Networks with Replay Processes. ArXiv, abs/2007.00487.