Development of deep learning algorithms for fire detection from outdoor images

Wided Cherif ** Manel Aissaoui
Ecole Polytechnique de Tunisie, UniversitÃ© de Carthage, B.P.743, 2078. La Marsa, Tunisie
wided.cherif@insat.u-carthage.tn
manel.aissaoui@insat.u-carthage.tn
Febraury 09, 2021

***Abstract*—The technologies used in fire detection systems play an important role in delivering optimal performance in modern surveillance environments. In fact, fire can cause significant damage to lives and properties. Considering that monitoring systems using cameras become widely used in cities, this encouraged us to take advantage of the availability of these systems to develop an effective vision detection system. However, Detecting fire and smoke is a complex task from the point of view of deformations, unusual camera angles and weather changes. In order to overcome those limitations, we proposed a fire detection method based on a deep learning approach, which uses CNN architectures: RESNET50 and VGG19. We evaluated our method by training and testing it on an online dataset containing 999 images. Majoring its performance, our model indicated a high accuracy and effectiveness.**

Fire detection, Deep learning, Outdoor images, RESNET50, VGG19

I. Introduction

The power of wildfires to devastating has been tragically apparent across the world, It can cost lives and damage all kind of properties. [1] Shows that in 2020 we saw many massive wildfires and forest fires around the world, including the Australian bushfires which started last year and continued until March, while the US state of California was also hit by blazes that left many dead.
With early detection and instant response, a fire may be quickly put out. In contrast, a fire can wipe out a massive amount of forest, killing and destroying as fast as it spreads.
In this study, we revue the different methods of fire detection starting from the traditional approaches which are based on color, shapes and fire motion moving to exploring the deep learning architecture used in this context. Then we developed two models trained on a limited dataset; the first one is based on the VGG19 architecture and the second one on the Resnet50 architecture. After tuning their parameters, we compared those two algorithms in order to determine which one can be considered more accurate for this study case.

II. Related Work

A. Traditional Approaches for fire detection

Various approaches have been proposed in recent years with the aim of efficiently detecting fire in outdoor images. In most existing approaches, the color information and their linked descriptors like area size, surface coarseness, boundary roughness and skewness are the most used in classifying fire images. For example, Chen [2] used a color-based approach to detect the discrepancy among sequential images .In the other hand, Celik et al. [3] used the YCbCr color space to discriminate luminance from chrominance to identify a variety of smoke and fires in images .Using those techniques is not reliable enough to perform classification and cannot perform properly.

Other investigations focused on detecting geometrical characteristics such as edge detections. For example, Jian et al. [4] presented an enhanced edge detection operator, a canny edge detector, which uses a multi-stage algorithm. However, this method is only applicable to images of simple and steady fires.

To be properly performed, vision systems need to: handle exceptions, manage the speed-accuracy trade-off and avoid aliasing situations. However, traditional approaches are not independent from environmental factors such as lighting, shadows, and other distortions and they rely on limited characteristics of fire and smoke in images such as the motion, color, and edge of the fire or smoke. Therefore, these methods are limited and not reliable.

B. Deep Learning Approches for fire detection

In recent years, researchers have proved that among many categories of deep learning methods, CNN is the most used and employed when talking about fire detection and monitoring. Zhao proposed [5] four CNN architectures which he tested to finally determine the optimum performance of the four algorithms: Faster-RCNN [6] (one of the famous object detection architectures that uses convolution neural networks like YOLO , SSD , R-FCN and YOLO , these algorithms were trained on a large dataset that contains 13400 fire images captured in outdoor and indoor images , and 15780 of images containing no fire, the distribution was 50 percent for the training set and the rest for the test set . When the number of the images in the dataset is limited , to avoid overfitting problems , AbdelAziz and Young[7] used data augmentation techniques to increase the number of training images and the Unpaired Image to Image Translation using Cycle-Consistent Adversarial Networks [8] and the Image-to-image Translation with Conditional Adversarial Networks [9] methods to also increase the diversity of the images . After performing the tests, Faster RCNN has proved to be the most performing with an accuracy reaching 99.43 percent [5]. Furthermore, AbdelAziz and Young [7] created a CNN model to classify fire and smoke images inspired from VGG-net architecture but not as deep as the original VGG-net , the CNN model contains 12 layers : 6 Conv layers and 3 Pooling layers to extract features and finally three fully connected layers for the classification.The used activation function was the adaptive piecewise linear activation function instead of Relu ,which improves the performance of the network.[10] also proposed a deep VGG-Net classifier to automatically detect video smoke in order to provide better statistical stability and performance.A very good accuracy and precision were obtained and proved that this algorithm can be used as real-time model to detect fire.

III. Dataset

A. Data Presentation

One of the main limitations of vision-related tasks is the insufficiency of robust data for training and evaluating the proposed method. To find a suitable dataset, we examined datasets that were used in prior studies. The choosen dataset was created during the NASA Space Apps Challenge in 2018.It is divided into two classes: fire images class containing 755 outdoor-fire images, the other one is non-fire images which contain 244 nature images like forest , rivers ,waterfall etc

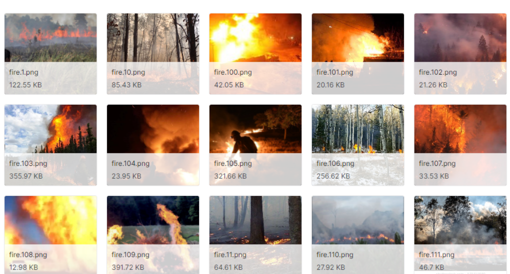

Fig. 1. A sample of the dataset

B. Data Pre-Processing

To proceed in our work, we first have to pre-process our data because the quality of the data we are going to use affects the ability of our model to learn and its performance. If we have high quality input data, then we expect high quality results. First, we started by filtering the data by deleting all the images that can affect and disturb the training process.

We noticed that the diversity of this image data is insufficient to be suitable for training:the number of images containing no fire is very much higher than the other categories which can prevent the model from predicting correctly. To overcome the problem of unbalanced data the next step consists in data stabilization in order to end up with two balanced categories with 730 images containing no fire and 745 images with fire

TABLE I
IMAGE DISTRIBUTION IN THE DATASET

Dataset	Fire Images	Non-Fire Images	Total
Our Dataset	745	730	1475

and smokes.

Then the data-set should be split into:
Training Data: A set of examples used for learning that is to fit the parameters of the classifier.
Validation Data: Is a sample of data held back from training our model that is used to give an estimate of model skill while tuning model's hyper-parameters.
Test Data: A set of examples used only to assess the performance of a fully-specified classifier.

IV. Proposed Method

A. VGG19 Architecture

VGG is a deep CNN used to classify images. It has 19 weight layers consisting of 16 convolutional layers with 3 fully connected layers. The VGG-net19 ranked second in 2014 challenge with a top 5 classification error of 0.0732 [11] and achieves the top-5 accuracy of 92.3 percent on ImageNet.
VGG architecture: The input to VGG based convNet is a

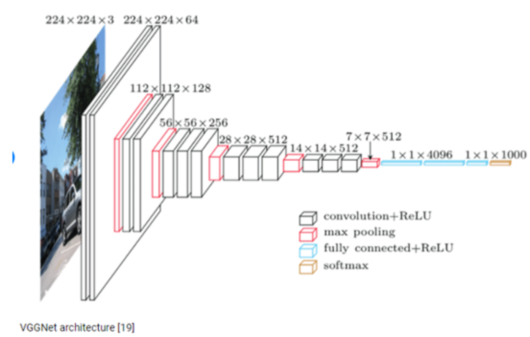

Fig. 2. The VGG-19 Architecture

224*224 RGB image .The first two layers are convolutional layers with 3*3 filters, Then a pooling layer was used with max-pool of 2*2 size and stride 2 which reduces height and width of a volume from 224*224*64 to 112*112*64. This layer is followed by 2 more convolution layers with 128 filters. This results in the new dimension of 112*112*128. After using the pooling layer, the volume is reduced to 56*56*128. Then, two more convolution layers are added with 256 filters each followed by down sampling layer that reduces the size to 28*28*256. Two more stacks, each with 3 convolution layer is separated by a max-pool layer. After the final pooling layer, 7*7*512 volume is flattened into Fully Connected (FC) layer with 4096 channels and softmax output of 1000 classes meaning that the pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

B. Resnet50 Architecture

Unlike traditional sequential network architectures such VGG, ResNet is instead a form of exotic architecture that relies on residual blocks that allow stacking to obtain a deep neural network. The resulting network can solve many recognition problems with high quality and is very scalable to keep the desired balance between inference speed and accuracy[12].

We have chosen the ResNet-50 architecture since it deals with the Vanishing Gradient Problem. The core idea of ResNet is introducing a so-called identity shortcut connection that skips one or more layers to reduce the information loss.

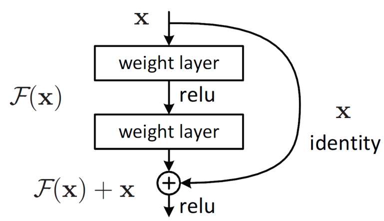

Fig. 3. The Resnet Architecture

ResNet-34 achieved a top-5 validation error of 5.71 percent better than BN-inception and VGG. ResNet-152 achieves a top-5 validation error of 4.49 percent. An ensemble of 6 models with different depths achieves a top-5 validation error of 3.57 percent. Winning the 1st place in ILSVRC-2015

C. Model Configuration

In this step we specify all the parts of our model:
Loss-function : In machine learning, one of the most commonly used loss functions for classification problems is the cross-entropy loss. For our case, we used binary-crossentropy that is used in binary classification tasks.
The optimizer : In order to determine the optimal weights for the model, Neural network uses an algorithm called an optimizer.For our model, we used Adam optimizer [13] to update networks weights iterative based in training data.
Activation function: is used in the dense layers to determine the output of the neural network using mathematical equations .In our case we used the ReLU function (rectified linear activation function) that proved its efficiency to overcome the vanishing gradient problem to make the model learn faster and perform better.
Parameters initializations: Using the Resnet architecture or the VGG architecture, we have directly retrieved the weights from training on ImageNet (weights = imagenet) instead of randomly initializing the weights.
Epochs: One epoch is when an entire data-set is passed forward and backward through a neural network only once. Since one epoch is too big to feed to the model at once we divide it into several smaller batches.
Batch: Total number of training examples present in a single batch. It impacts learning significantly.
Learning Rate: Learning Rate determines how fast or slow we will move towards the optimal weights -A low learning rate means more training time and more time results in increased GPU cost -A high learning rate could result in a model that might not predict anything accurate. And that's why after testing different values we chose to fix the value of the learning rate on 0.0001
Regularization techniques: In order to overcome the overfiting problem we have used some regularization techniques:

1) Data augmentation: When dealing with a small dataset the model risks to have overfitting issues.To overcome this problem and improve the performance of the model it has to be fed larger dataset. For that reason we used data augmentation technique, a strategy that consists on increasing the number of images in the dataset as well as its diversity without the need to collect new data. In fact, it is based on generating more training data from the existing one [7]. To apply this technique we used the ImageDatagenerator function that have several parameters :
Rotation range : a value between 0 and 180 degrees to rotate pictures.
Width-shift and height-shift: to translate picture vertically or horizontally.
Shear-range : to apply shearing transformation.
Zoom-range : to zoom randomly inside pictures.
Horizontal-flip for flipping half the image horizontally.
Fil-mode to fill the new created picture obtained after a rotation or a width/height shift).

2) Dropout Technique: Another solution we adopted to solve the overfitting problem is the dropout function, one of the most popular techniques to prevent neural networks from overfitting. It consists on randomly droping out a number of output features of the layer during the training phase[14].

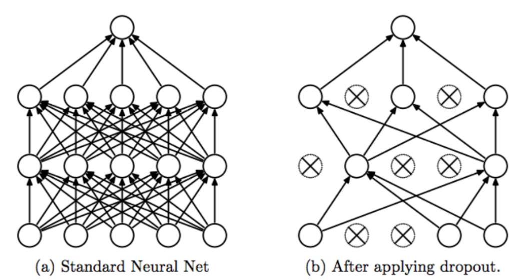

Fig. 4. The Dropout Technique

V. Experimental results and discussion

After training our model for 100 epochs with a batch size equal to 16 and fixing the learning rate to 0.0001 we have obtained the following confusing matrix and plots.

A. VGG19 exprimental results

1) Confusion Matrix: - An image from class 0 (No Fire) is predicted 99 percent to be 0 (No Fire) and 1 percent to be 1 (Fire).
- An image from class 1 (Fire) is predicted 66 percent to be

1 (Fire) and 34 percent to be 0 (No Fire).

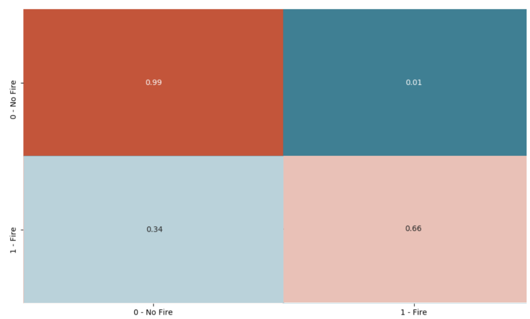

Fig. 5. The confusion matix for VGG19

2) Accuracy and Loss plots: Using the regularization techniques, we have managed to get rid of the overfitting, the train loss and validation loss are then synchronized and the performance on validation set is almost the same as the performance on training set.

Fig. 6. The accuracy and Loss plots for VGG19

B. Resnet50 exprimental results

1) Confusion Matrix: - An image from class 0 (No Fire) is predicted 98 percent to be 0 (No Fire) and 2 percent to be 1 (Fire).
- An image from class 1 (Fire) is predicted 86 percent to be 1 (Fire) and 14 percent to be 0 (No Fire).

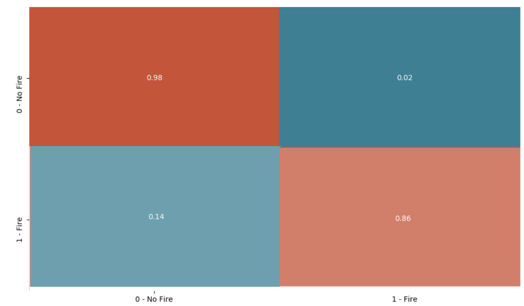

Fig. 7. The confusion matix for Resnet50

2) Accuracy and Loss plots: As we can see, The performance on validation set is almost the same as the performance on training set.
We can notice also that we have better results using the Resnet architecture with an elimination of the overfitting problem.

Fig. 8. The accuracy and Loss plots for Resnet50

C. Discussion

Our experiments mainly aimed to evaluate the performance of both the VGG-Net CNN classifier and ResNet 50 classifier in order to determine the one who presents more accuracy and who is capable to deliver better results .
Even though the two models don't suffer from overfitting , The accuracy plot and the confusion matrix shows that Resnet50 has reached a better accuracy with the value of 98 percent. Also the experimental results demonstrate that ResNet50 consumes less time for per training epoch comparing to VGG-19.

VI. Conclusion

In this paper, We presented different robust deep learning model architectures for classifying fire and smoke images captured by a camera or nearby surveillance systems.
During the experiments, we assessed the performances and generalizing abilities of well-known CNN architectures.We focused on comparing the VGG19 architecture and the Resnet architecture using different kind of evaluation metrics.
As a result, Resnet50 gave a better result with an accuracy of 98 percent.
Our future projection is to build a lightweight model with robust detection performance that would allow us to set up embedded devices, which have low computational capabilities.

References

1.Wildfires, forest fires around world in 2020.
2. Chen, T.H.; Wu, P.H.; Chiou, Y.C. An early fire-detection method based on image processing. In Proceedings of the International Conference on Image Processing (ICIP), Singapore, 24â€“27 October 2004; pp. 1707â€“1710.
3. Celik, T.; Demirel, H.; Ozkaramanli, H.; Uyguroglu, M. Fire detection using statistical color model in video sequences. J. Vis. Commun. Image Represent. 2007, 18, 176â€“185.
4. Jiang, Q.; Wang, Q. Large space fire image processing of improving canny edge detector based on adaptive smoothing. Proc. Int. CICC-ITOE 2010, 1, 264â€“267.
5. Image fire detection algorithms based on convolutional neural networks
6. Den Breejen, E., Breuers, M., Cremer, F., Kemp, R., Roos, M., Schutte, K., and De Vries,J. S..: Autonomous forest fire detection. In: Proc. 3rd Int. Conf. on Forest Fire Research. p. 2003-2012 (1998).
7. An Efficient Deep Learning Algorithm for Fire and Smoke Detection with Limited Data
8. Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. ”Unpaired image-to-image translation using cycle-consistent adversarial networks.” arXiv preprint arXiv:1703.10593, 2017. doi: 10.1109/ICCV.2017.244.
9. Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. ”Image-to-image translation with conditional adversarial networks.” arXiv preprint, 2017. doi: 10.1109/CVPR.2017.632.
10. Hybrid Deep VGG-NET Convolutional Classifier for Video Smoke Detection.
11. Automatic Fire and Smoke Detection Method for Surveillance Systems Based on Dilated CNNs
12.ResNet-like Architecture with Low Hardware Requirements.
13.Gentle Introduction to the Adam Optimization Algorithm for Deep Learning
14.dropout : a simple way to prevent neural networks from overfitting.