Fire Detection using Residual Deeplabv3+ Model

Houda Harkat*, Jose M.P. Nascimento†, and Alexandre Bernardino‡,
* Instituto Superior Técnico
Av. Rovisco Pais 1, 1049-001, Lisbon, Portugal
University of Sidi Mohamed Ben Abdellah
Route Imouzzer, BP 2626, FES 30000, Morocco
houda.harkat@usmba.ac.ma
† Instituto de Telecomunicações and Instituto Superior de Engenharia de Lisboa, IPL
Lisbon, Portugal
‡ ISR - Instituto de Sistemas e Robótica
Av. Rovisco Pais 1, 1049-001, Lisbon, Portugal

***Abstract*—Fire detection and prevention had become a high priority task, in the last decade, due to the higher number of forest fires. Automatic detection systems facilitate the intervention and reduce the cost of firefighters travel in case of false occurrences. Deep learning based systems has drawn promising high results in the field, in particular, Deeplabv3+ which is an architecture based on the so called Atrous Spatial Pyramid Pooling that enhances the segmentation results.**

This paper present the results of Deeplabv3+ applied over the french Corsican dataset with an Xception backbone. The model had been trained over the RGB collection of pictures of the dataset and over the whole dataset that englobes RGB and infrared (IR) pictures. Several experiments using Dice and Tversky loss functions are conducted in order to further reduce the problems induced by unbalanced datasets. The performance is measured with some extended metrics namely: mean Intersection over Union (IoU), mean Boundary F1 (BF) Score, and mean Dice similarity measure in addition to the standard accuracy metric. The achieved results demonstrate that the Deeplabv3+ with Xception gives very encouraging results for fire detection over RGB and IR images.

***Index Terms*—Fire detection, deep learning, semantic segmentation, Deeplabv3+, Xception**

I. Introduction

In the last decades, due to the huge number of forest and urban fires recorded, more resources have been allocated to accurate detection of fires and faster intervention. In fact, the statistics in Portugal show that 7.5 million hectares of forests had been burnt between 2003 and 2005 [1], which is more than 25% of the country area. It is well known that to minimize the occurrence of large wildfire, it is necessary to have a fully prevention strategy, and to assure fast response of firefighters. Nevertheless, to assure a fast response it is fundamental to have an accurate detection system, which will allows to reduce the cost of firefighters travel. In this context, authors had proposed several intelligent models for fire detection based on digital content (images and videos) captured during inspection flights. The work had been concentrated on two aspects: early fire detection (smoke) [2], [3], and flame detection [4], [5]. Deriving, the critical situation of fast propagation of fire, more emphasis is given to fire detection. The implemented works are subclassified into two main groups: decision-based systems and segmentation methods. The first category aims to localize the region of fire or smoke with a lower accuracy (windowing techniques) [6], while the second apply a pixel by pixel logic (semantic segmentation). Most of segmentation techniques are based on deep learning and on extended architectures of convolutional neural networks (CNN). A joint detection framework based on faster RCNN and 3D CNN was proposed for smoke detection in [3]. An interesting decision-based approach [7], which introduces a fire detection system in an urban area deploying a static ELASTIC-YOLOv3 model. In [9], authors had implemented a Deeplabv3 [10] model for fire segmentation using three types of backbones: ResNet-50, ResNet-101, and ResNet-105, which had reached a mean Intersection over Union (IoU) of 70.51% and an accuracy of 98.78% on the evaluation set. Deeplabv3+ [11], extends Deeplabv3 [9] by placing an effective decoding module to refine the segmentation along objects boundaries. This paper assess the performance of the Deeplabv3+ [11] within the Xception [11], [12] backbone. The experiments are conducted on two custom sets of the Corsican dataset, highly unbalanced. On these experiments two different loss functions are used, namely the Dice and Tversky and the Cross entropy loss functions. The results are very encouraging, the best model trained over an RGB dataset, draws an overall accuracy of 98.48%, mean IoU of 93.27%, mean BF score of 92.91% and mean Dice similarity measure of 91.64%. Another model able to manage infrared pictures as well, draws an overall accuracy of 96.11%, mean IoU of 80.29%, mean BF score of 89.07% and mean Dice similarity measure of 87.86%. The rest of this paper is organized as follows: Section 2 presents the Deeplav3+ model and the Xception architecture. In Section 3, the Corsican dataset, the problem formulation, and the loss function deployed are presented. In Section 4, the experimental prototypes and their results are presented and compared. Finally, section 5 concludes this paper and highlights some lines for future work deployment.

II. Deeplabv3+ Architecture

The encoder-decoder networks, widely used, implement an encoder block which gradually reduces the feature allowing to

2021 Telecoms Conference (ConfTELE) | 978-1-6654-1588-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CONFTELE50222.2021.9435459

gather a rich information. Moreover, the decoder block which gradually recovers the information permits to attend higher accuracy for semantic segmentation application.

A. Type of Convolutions

In deep learning field, it exists several types of Convolutions. In the following, we will give an overview of different types of convolutions used within the Deeplabv3+ model to deploy.

1) Dilated Convolutions: Dilated convolutions allow to introduce a kind of space between the values of the kernel. This space is defined following the dilation rate parameter. Its allows augmentation of the receptive field to have a wider field of view at the same feature map size. The mathematical formulation is given in expression (3), where $Y(m,n)$ is the output of a 2D Atrous convolution from input $X(m,n)$ through a convolution filter $W(i,j)$, adopting a rate factor r [12]:

$$Y(m,n) = \sum_{i=1}^{a}\sum_{j=1}^{b} X(m+ri, n+rj)W(i,j) \qquad (1)$$

2) Transposed Convolutions: A transposed convolution is like a deconvolutional layer, with the difference in the mathematical operation carried out. The layer performs a regular convolution with a reverted spatial transformation. A transposed convolution permits to combine the upscaling of the input with a convolution, instead of doing the two things separately.

3) Separable Convolutions and Depthwise Separable Convolutions: A separable convolution is designated to split the kernel operation into several arithmetic. First it is the convolution expressed as:

$$Y = conv(X,k) \qquad (2)$$

X, and Y are the input and output images respectively, and k is the kernel. Next, its the computation k can be calculated by the following equation:

$$k = k_1.dot(k_2) \qquad (3)$$

The dot product consists of multiplying and summing aligning entries. In fact, it is named separable convolution because it is computed in the form of two 1D convolutions with kernels k_1 and k_2. Neverthless, depthwise separable convolutions opts for kernels that could not be factored into two smaller values. The layer deals with both spatial and depth (number of channels) dimensions. This layer type, drastically diminishes the computation complexity of the model. The depthwise convolution compute the spatial convolution for each channel in an independent manner, and the output is combined through the pointwise convolution.

4) Atrous Spatial Pyramid Pooling (ASPP): ASPP architecture resample features at four different atrous rate, to capture a diversified feature map. Four parallel atrous convolutions are used with graduating rates and their outputs are concatenated.

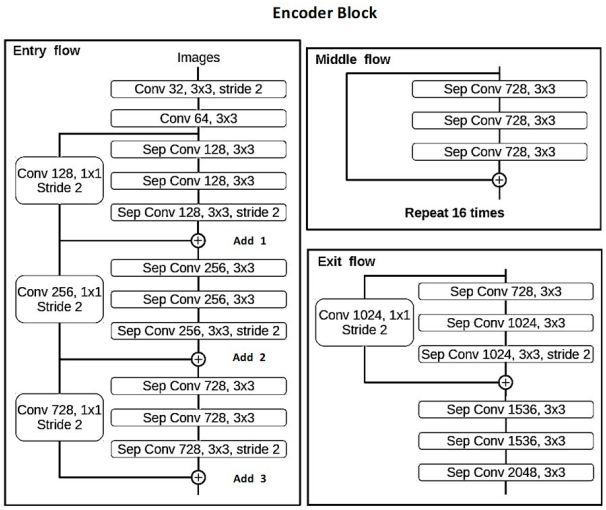

Fig. 1. The modified Xception encoder implemented within Deeplabv3+

B. Modified Xception

Deeplabv3+ is used with a modified Xception model as backbone, presented in Fig. 1. Some minor changes [11] had been proposed to the original Xception model [12]:

- The entry flow past is slightly modified to allow a fast computation and a high memory efficiency.
- All the max pooling layers are replaced by depthwise separable convolutions with striding to be able to connect to an atrous separable convolution for feature extraction at random resolution.
- Batch normalization and ReLU layers are introduced after each 3×3 depthwise convolution operation.

C. Atrous Separable Convolution (ASPP)

The Deeplabv3+ implemented deploys atrous rates of 6, 12 and 18 with depthwise separable convolution instead of the original convolution layers. The complete architecture is illustrated in Fig. 2,

D. Deeplabv3+ Decoder

The used decoder module, illustrated in Fig. 3, is characterized with a very simple architecture. A bilinear upsampling arithmetic with a factor of 4 is computed and then the result is concatenated with the features, from the encoder, with an identical spatial resolution. The 1×1 convolution is introduced with the aim to reduce the number of channels. In fact, the incoming features have many channels which may outweigh the importance of the features and could induce overfitting of the model. Afterwards, a few 3×3 convolutions followed by another bilinear upsampling (factor of 4) operation are applied to additionally refine the feature map.

III. Model Objectives

This section presents an overview of the different loss functions used in simulation. It is also presented the two different set of data of the Corsican french dataset [13].

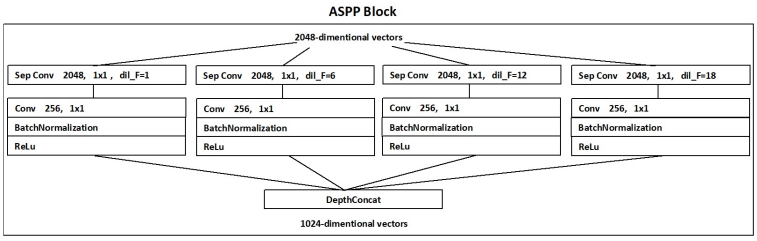

Fig. 2. The ASPP module implemented within the Deeplabv3+

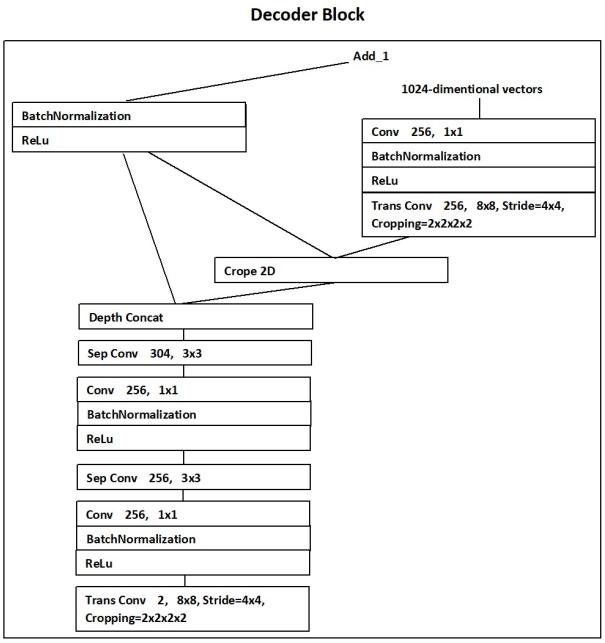

Fig. 3. The decoder implemented within Deeplabv3+. 'Trans Conv' refers to a transposed convolution layer

A. Loss Functions

Besides the Cross Entropy loss function, two other loss functions, Generalized Dice and Tversky loss, are considered in the performed experiments with the aim to correct the impact problems induced by unbalanced datasets.

Let p_{0i} be the probability of the i-th voxel to be a fire and p_{1i} to be the probability of the i-th voxel to be a non-fire. The ground truth is denoted as g, g_{0i} is 1 for a fire voxel and 0 for a non-fire voxel and vice versa for the g_{1i}. The number of classes is denoted by N.

1) Cross Entropy Loss: The Cross Entropy loss function is defined as follows [14]:

$$T_{ce} = -\left(\sum_{i=1}^{N} y_{0i} \log p_{0i} + \sum_{i=1}^{N} y_{1i} \log p_{1i}\right). \tag{4}$$

This loss function does not take into account correcting the issues induced by unbalance data.

2) Generalized Dice Loss: The Dice loss function [15], is defined as follows:

$$T_{gd} = 1 - 2\frac{\sum_{i=1}^{N} p_{0i} g_{0i}}{\sum_{i=1}^{N} p_{0i} + \sum_{i=1}^{N} g_{0i}}. \tag{5}$$

This formula aims to maximize the overlap between calculated maps and reference segmentation maps. Usually, a small value is added to the denominator and the numerator to avoid invalid division by zero.

3) Tversky Loss: The Tversky loss function is defined as follows [16]:

$$T_{tv}(\alpha, \beta) = \frac{\sum_{i=1}^{N} p_{0i} g_{0i}}{\sum_{i=1}^{N} p_{0i} g_{0i} + \alpha \sum_{i=1}^{N} p_{0i} g_{1i} + \beta \sum_{i=1}^{N} p_{1i} g_{0i}}. \tag{6}$$

The parameters α and β allows weights adjustment of false positives (FPs) and false negatives (FNs). For larger β weighs boosts the recall factor more than the precision since the FPs are more emphasized in this case. When the value of β is higher than α, the training performance of imbalanced problem are basically improved, the FN rate over test set is reduced and the recall factor is boosted.

B. Corsican dataset

The Corsican dataset [13] is composed of multiple fire scenes collected in RGB and infrared (IR) pictures. Basically there are scenes composed by RGB pictures with the corresponding infrared (IR) images, scenes represented only by RGB pictures, and scenes represented with images on the infrared band.

Two personalized datasets are builded within IR and RGB pictures of original Corsican dataset, to assess the quality of information of the features depending on image types. CCorsican-RGB, the first dataset, encloses all the 1135 RGB images within the dataset. CCorsican-RGB_R+IR, the second dataset, incorporates all the images of the original Corsican dataset (RGB and IR images), resulting in 1775 pictures.

The fire pixels are less than 20% of the entire set of data for the two custom datasets.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experimental Setup

The Deeplabv3+ architecture is tested in nine different configurations: within three different loss function (Cross Entropy, Dice and Tversky), and for each, three different

initial learning rate value (10^{-1}, 10^{-2}, and 10^{-3}). Different scenarios are considered for CCorsican-RGB and CCorsican-RGB_R+IR datasets, using two different input image sizes: 300×300 and 600×600. The CCorsican-RGB_R+IR set contains duplicate profiles, i.e. IR and RGB images of the same view. However, this is not applicable to all the views, some views are represented with a single profile. The datasets are randomly divided as follow: 60% for training, 20% for validation and 20% for test. The dataset partition, for CCorsican-RGB and CCorsican-RGB_R+IR, is given in Table I. The grayscale pictures are converted to RGB format by duplicating the same information over the two missing channels, since the architecture is built for 3D entries.

TABLE I
THE DATASETS PARTITION IN TERM OF NUMBER OF PICTURES

Set \ Dataset	CCorsican-RGB	CCorsican-RGB_R+IR
Training	681	1065
Validation	227	355
Test	227	355

Since the network training is time consuming, for each run it can take between 180 to 450 minutes, any cross-validation strategy was adopted. The models are parameterized and trained on MATLAB environment over GPU, on a server equipped with a Nvidia tesla K40c card in the facilities of the Instituto de Telecomunicações of Coimbra. The models are trained with a stochastic gradient descent algorithm with momentum term equal to 0.9 and learning rate schedule with a piecewise approach [17]. For the learning rate schedule, the drop period and factor are set to 10 and 0.3, respectively. The chosen setting helps to increase the accuracy and reduce the loss during training. The number of training epochs is set to 100, and the earlier stopping option is activated to avoid overfitting of the models. In the models architecture, every convolution layer is succeed by a Batch Normalization to further reinforces the training stability, augments the performance, and have a faster converge.

B. Results and Discussion

In this subsection the results of the experiments conducted over the two datasets separately, for the two image sizes 300×300 and 600×600 are presented.

1) Experiments with CCorsican-RGB set: The model is first trained over CCorsican-RGB set, with an input size 300×300 and a batch size equal to 22. The performance in terms of overall accuracy (Acc), IoU, BF Score and Dice score (Dsc) are presented in Table II, where the M-IoU, M-BF and M-Dsc refer to the mean values of IoU, BF score and Dice score over all classes over all images of the corresponding dataset.

The best results are achieved for a training with an initial learning rate of 10^{-2}. Moreover, one could remark that the Generalized Dice and the Tversky losses, highly correct the BF Score and the Dice metric. The results, for the 600×600 image size and a batch size equal to 8, are presented in Table III. In

TABLE II
DEEPLABV3+ MODEL RESULTS FOR DIFFERENT LOSS FUNCTIONS AT DIFFERENT LEARNING RATE VALUE OVER CCORSICAN-RGB SET, THE INPUT SIZE IS 300X300 AND THE BATCH SIZE IS SET TO 22.

Learning rate	Metrics	Cross Entropy	Generalized Dice	Tversky
10^{-3}	Acc	97.54	96.94	96.87
	M-IoU	89.48	86.68	86.72
	M-BF	80.58	87.46	85.31
	M-Dsc	84.59	88.76	87.94
10^{-2}	Acc	**98.48**	97.72	97.57
	M-IoU	**93.27**	89.83	89.36
	M-BF	**92.91**	94	92.44
	M-Dsc	**91.64**	92.60	91.72
10^{-1}	Acc	96.55	96.78	95.59
	M-IoU	85.36	86.21	82.39
	M-BF	82.95	87.72	77.74
	M-Dsc	85.69	88.28	83.98

general, the results are better with an input size of 300×300 and a batch size of 22. For comparison propose, for every case

TABLE III
DEEPLABV3+ MODEL RESULTS FOR DIFFERENT LOSS FUNCTIONS AT DIFFERENT LEARNING RATE VALUE OVER CCORSICAN-RGB SET, THE INPUT SIZE IS 600×600 AND THE BATCH SIZE IS SET TO 8.

Learning rate	Metrics	Cross Entropy	Generalized Dice	Tversky
10^{-3}	Acc	97.83	97.19	97.32
	M-IoU	90.70	87.64	88.40
	M-BF	86.84	89.79	90.06
	M-Dsc	88.69	90.50	91.03
10^{-2}	Acc	97.15	97.36	**97.46**
	M-IoU	87.69	88.40	**88.88**
	M-BF	86.01	91.03	**91.13**
	M-Dsc	87.12	91.38	**91.29**
10^{-1}	Acc	93.89	95.43	95.63
	M-IoU	76.89	81.93	82.29
	M-BF	61.40	78.74	78.58
	M-Dsc	61.36	84.59	83.78

(depending on input size and dataset), a best model is selected. The model is highlighted in grey in every table. In Table II, the chosen model is the one trained with Cross Entropy loss at an initial learning rate of 10^{-2} (please refer to Model_1 in Table VI). However, in Table III, the best model (please refer to Model_2 in Table VI) is the one trained with Tversky loss at an initial learning rate of 10^{-2}.

2) Experiments with CCorsican-RGB+IR set: This time, the model is trained over CCorsican-RGB+IR set, with an input size 300×300 and a batch size equal to 22. The performance are given in Table IV.

While, for the case of an input size of 600×600 and a batch size equal to 8, the results are grouped in Table V for different cases of simulation.

In Table IV, the chosen model is the one trained with Generalized Dice loss at an initial learning rate of 10^{-2} (please refer to Model_3 in Table VI). However, in Table V, the best

TABLE IV
DEEPLABV3+ MODEL RESULTS FOR DIFFERENT LOSS FUNCTIONS AT DIFFERENT LEARNING RATE VALUE OVER CCORSICAN-RGB+IR SET, THE INPUT SIZE IS 300X300 AND THE BATCH SIZE IS SET TO 22.

Learning rate	Metrics	Cross Entropy	Generalize Dice	Tversky
10^{-3}	Acc	96.49	95.11	95.09
	M-IoU	82.48	76.29	76.52
	M-BF	77.54	85.43	85.15
	M-Dsc	80.98	83.50	83.41
10^{-2}	Acc	93.82	**96.11**	95.27
	M-IoU	75.21	**80.29**	76.15
	M-BF	80.80	**89.07**	86.58
	M-Dsc	83.83	**87.86**	83.63
10^{-1}	Acc	93.44	91.18	92.47
	M-IoU	68.72	66.85	69.89
	M-BF	67.59	68.55	71.52
	M-Dsc	56.82	69.83	74.44

TABLE V
DEEPLABV3+ MODEL RESULTS FOR DIFFERENT LOSS FUNCTIONS AT DIFFERENT LEARNING RATE VALUE OVER CCORSICAN-RGB+IR SET, THE INPUT SIZE IS 600X600 AND THE BATCH SIZE IS SET TO 8.

Learning rate	Metrics	Cross Entropy	Generalize Dice	Tversky
10^{-3}	Acc	93.94	94.11	94.07
	M-IoU	69.77	70.94	71.29
	M-BF	78.39	82.51	82.21
	M-Dsc	49.60	52.47	52.52
10^{-2}	Acc	94.44	**94.22**	93.96
	M-IoU	72.86	**71.56**	71.17
	M-BF	80.81	**83.38**	82.23
	M-Dsc	51.66	**53.11**	52.43
10^{-1}	Acc	92.63	92.44	91.36
	M-IoU	65.49	67.65	67.71
	M-BF	66.71	75.39	65.53
	M-Dsc	36.76	47.78	70.93

model (please refer to Model_4 in Table VI) is the Generalized Dice loss as well with a loss at an initial learning rate of 10^{-2}.

C. Comparison

Table VI presents a comparison of the four selected models. Furthermore, Fig. 4.(a) shows an example of an image of the dataset and Fig. 4.(b) its ground truth. The segmentation mask results are highlighted for the four models: model_1 (Fig. 4.(c)) model_2 (Fig. 4.(d)) , model_3 (Fig. 4.(e)) and model_4 (Fig. 4.(f)). The corresponding BF score and Dice metric values, for this example, are presented in Table VII.

It is visually clear that the model_2 and model_4, trained with an input size of 600x600 and a batch size equal to 8, generate a lot of false detections (the green colour) in comparison with the two other models. The results of tab VII demonstrate that the model_1 and model_3 gives the best results in terms of BF score and Dice coefficient value.

An example that illustrates the performance of the model_3 over IR pictures is presented in Fig. 5.

Our results are compared to the models trained over Corsican dataset, in the literature. An Affine comparison is not

TABLE VI
THE FOUR BEST MODELS CHOSEN FROM THE PREVIOUS SIMULATIONS

Model name	Metrics	Values	Loss+Rate	Further details
Model_1	Acc	**98.48**	**Cross Entropy $+10^{-2}$**	**CCorsican -RGB, 300x300**
	M-IoU	**93.27**		
	M-BF	**92.91**		
	M-Dsc	**91.64**		
Model_2	Acc	97.46	Tversky $+10^{-2}$	CCorsican -RGB, 600x600
	M-IoU	88.88		
	M-BF	91.13		
	M-Dsc	91.29		
Model_3	Acc	**96.11**	Generalized Dice $+10^{-2}$	CCorsican -RGB+IR, 300x300
	M-IoU	**80.29**		
	M-BF	**89.07**		
	M-Dsc	**87.86**		
Model_4	Acc	94.22	Generalized Dice $+10^{-2}$	CCorsican -RGB+IR, 600x600
	M-IoU	71.56		
	M-BF	83.38		
	M-Dsc	53.11		

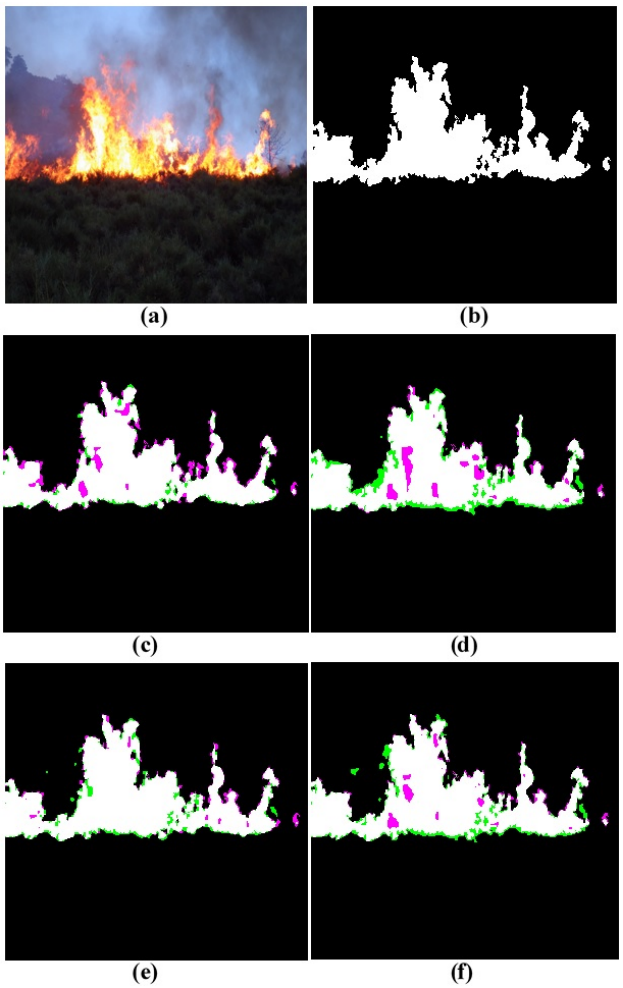

Fig. 4. Example of segmentation of an RGB picture: (a) The picture, (b) the ground truth; results using (c) model_1, (d) model_2, (e) model_3, and (f) model_4. Green color represents the pixels that must be recognized as non-fire, but the selected model had designed them as fire pixels (FP). Pink pixels should be classified as fire, but the selected model had recognized them as non-fire pixels (FN).

TABLE VII
THE EXAMPLE OF FIG 4 PERFORMANCE

Metrics	Model_1	Model_2	Model_3	Model_4
BF	**90.71**	76.21	**91.31**	84.77
Dsc	**92.38**	89.60	**93.92**	92.13

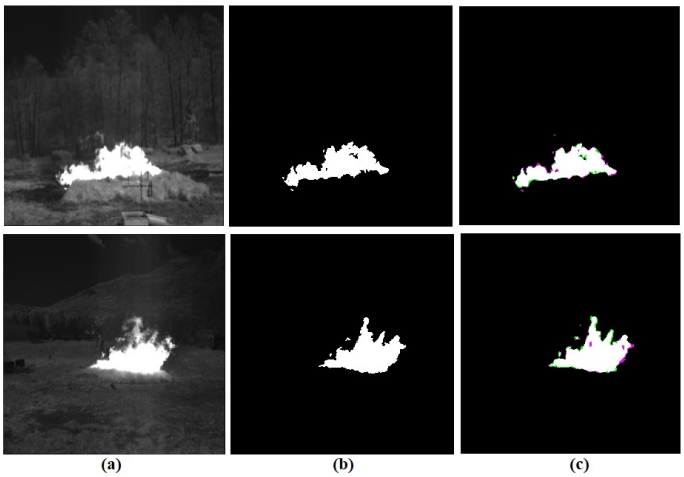

Fig. 5. Example of segmentation of IR pictures using model_3 (a) The picture, (b) the ground truth, and (c) the obtained segmentation mask. The BF score is equal to 98.17%and 94.61% for the first and the second example respectively.

possible since the metrics used are different. The model described in work [18] draws an accuracy higher than our selected models, but the authors didn't analyse the IoU or the BF score performance. The accuracy is not enough to judge, because it is a global metric that do not reflects the real situation in case of unbalance class problem. Analysing the above metrics is essentiel in the case of pixel segmentation, since the global accuracy gives the performances related to all the pixels over all images and do not gives a wider idea about the matching related of every picture. Nevertheless, our selected models draws a higher Jaccard and Dice index values than the model computed in [19].

V. CONCLUSION

Fire detection using deep learning model is a very challenging task. Unbalanced datasets generate a lot of issues that could be avoided by using advanced loss formulas that correct the problem and gives better results. In this paper, the performance of a Deeplabv3+ architecture with Xception backbone is assessed for different loss function over two different sets of data, that contains RGB and IR pictures. The given results are very promising for further deployment of enhanced similar architectures.

To conclude, in the future we will focus on the amelioration of the model architecture, and we will use the selected models to label a new dataset recently captured for wildfire detection.

ACKNOWLEDGMENT

This work was supported by Instituto de Telecomunicações and Fundação para a Ciência e a Tecnologia (FCT) under Project UIDB/50008/2020 and Project FIREFRONT with reference PCIF/SSI/0096/2017. Authors are grateful to collegues from Instituto de Telecomunicações de Coimbra for providing access to a server doted with a Nvidia tesla K40c GPU.

REFERENCES

[1] Florestais, Direcção Geral Dos Recursos. "Incêndios Florestais–Relatório de 2005." Divisão de Defesa da Floresta Contra Incêndios, DGRF. Lisboa (2006).
[2] Wu, H., Wu, D., & Zhao, J. (2019). An intelligent fire detection approach through cameras based on computer vision methods. Process Safety and Environmental Protection, 127, 245-256.
[3] Lin, G., Zhang, Y., Xu, G., & Zhang, Q. (2019). Smoke detection on video sequences using 3D convolutional neural networks. Fire Technology, 55(5), 1827-1847.
[4] Barmpoutis, P., Dimitropoulos, K., Kaza, K.,& Grammalidis, N. (2019, May). Fire detection from images using faster R-CNN and multidimensional texture analysis. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 8301-8305). IEEE.
[5] Saeed, F., Paul, A., Karthigaikumar, P., & Nayyar, A. (2019). Convolutional neural network based early fire detection. Multimedia Tools and Applications, 1-17.
[6] Pan, H., Badawi, D., Zhang, X., & Cetin, A. E. (2019). Additive neural network for forest fire detection. Signal, Image and Video Processing, 1-8.
[7] Park, M., & Ko, B. C. (2020). Two-step real-time night-time fire detection in an urban environment using Static ELASTIC-YOLOv3 and Temporal Fire-Tube. Sensors, 20(8), 2202.
[8] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), 834-848.
[9] Mlích, J., Koplík, K., Hradiš, M.,& Zemčík, P. (2020, February). Fire Segmentation in Still Images. In International Conference on Advanced Concepts for Intelligent Vision Systems (pp. 27-37). Springer, Cham.
[10] Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.
[11] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F.,& Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV) (pp. 801-818).
[12] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).
[13] Toulouse, T., Rossi, L., Campana, A., Celik, T., & Akhloufi, M. A. (2017). Computer vision for wildfire research: An evolving image dataset for processing and analysis. Fire Safety Journal, 92, 188-194.
[14] Crum, W. R., Camara, O., & Hill, D. L. (2006). Generalized overlap measures for evaluation and validation in medical image analysis. IEEE transactions on medical imaging, 25(11), 1451-1461.
[15] Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Cardoso, M. J. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In Deep learning in medical image analysis and multimodal learning for clinical decision support (pp. 240-248). Springer, Cham.
[16] Salehi, S. S. M., Erdogmus, D., & Gholipour, A. (2017, September). Tversky loss function for image segmentation using 3D fully convolutional deep networks. In International Workshop on Machine Learning in Medical Imaging (pp. 379-387). Springer, Cham.
[17] Yang, H., Liu, J., Sun, H., & Zhang, H. (2020). PACL: Piecewise Arc Cotangent Decay Learning Rate for Deep Neural Network Training. IEEE Access, 8, 112805-112813.
[18] Chaoxia, C., Shang, W.,& Zhang, F. (2020). Information-Guided Flame Detection based on Faster R-CNN. IEEE Access, 8, 58923-58932.
[19] Toptaş, B., & Hanbay, D. (2020). A new artificial bee colony algorithm-based color space for fire/flame detection. Soft Computing, 24(14), 10481-10492.