The Monitoring of Burning Buildings with Convolutional Neural Network

Florin Batog
Department of Computer Science and Information Technology, Faculty of Automation, Computers, Electrical Engineering and Electronics, Dunarea de Jos University of Galati
Galati, Romania
batogflorin1999@gmail.com

Simona Moldovanu
Department of Computer Science and Information Technology, Faculty of Automation, Computers, Electrical Engineering and Electronics, Dunarea de Jos University of Galati
Galati, Romania
simona.moldovanu@ugal.ro

***Abstract*— Artificial intelligence is constantly expanding. This is used in many domains, from agriculture to the medical field. Lately, the focus has been on image-based learning (deep learning) because it has greater applicability in reality. An example is the identification of buildings that burn in real-time, the purpose being the saving as many lives as possible by the firefighters. This paper aims to identify the optimal color space for the identification of fire in images. To achieve this the images were selected from two databases taken from the Kaggle platform, and the images were processed with six color spaces: RGB, YCbCr, HSV, HLS, L*a*b, and L*u*v. In this form the images fed convolutional neural networks (CNNs). After that, the models were trained on six datasets, a dataset for each color space, and after compiling of CNNs, a testing set for the prediction model was proposed. The results of the model were analyzed and interpreted according to accuracy and loss function and the space YCbCr identified the fire from the images with an accuracy of 100% and loss function of $4.02 \cdot 10^{-05}$.**

***Keywords*— color space, neural network, model accuracy**

I. INTRODUCTION

Different abnormal events can occur in a city area such as fire, accidents, disaster, medical emergencies, fight, and flood for which getting early information is important. Identifying the fire with surveillance cameras could minimize the chances of big disasters and can control an abnormal event on time. Among such abnormal events, fire is one of the commonly happening events, whose detection at early stages during surveillance can avoid home fires or fire disasters [1].

The main causes of arsons include human error, nearly 85% of wildland fires in the United States are caused by humans [1]. Most common causes of commercial fires are cooking equipment (65% of fires in healthcare facilities), heating equipment (14% of fires in industrial or manufacturing properties), electrical and lighting equipment (12% of fires in office properties), smoking materials (9% of fires in office properties) [2]. According to the Romanian General Inspectorate for Emergency Situations (IGSU) in our country approximately 6,500 house fires occur annually, the main causes are defective or unclean chimneys (31%), defective or improvised electrical installations (23%), or unsupervised means of heating (12%) [3].

In scientific literature exist a few algorithms which deal with flame pixel classification, these are Moderate Resolution Imaging Spectroradiometer (MODIS) [2], Advanced Very High-Resolution Radiometer (AVHRR) [3], and Visible Infrared Scanner (VIRS) [4], algorithms are used for detect the brightness temperatures over the years, and constantly were improved. Nowadays networks hold supremacy in classifying the images that contain fire. Li and Zhao [5] proposed aster-RCNN, R–FCN, SSD, and YOLO V3 CNNs, and the accuracy of fire detection algorithms based on object detection reached to 83.7% with YOLO V3 CNN, Majid *et al.* [6] used the Grad-CAM method for the visualization and localization of fire in the image and EfficientNetB0 CNN for image classification. For the Fire Detection Dataset and DeepQuestAI image datasets, a test accuracy of 95.40% was obtained.

The fire detection performance depends on several factors: preprocessing image methods, pixel classifier or image artifacts, and image quality.

Çelik and Demirel [7] used YCbCr (Y is luminance, Cb and Cr are Chrominance Blue) RGB (red, green, blue) color spaces for separating the flame from the video, they segmented the video and a chrominance model for flame pixel classification was used, the results provided an accuracy of 99%, the statistical method used for classification was receiver operating characteristics (ROC). Muhammad *et al.* [8] proposed ImageNet and SqueezeNet CNNs, RGB color corroborated with the binary mask, accuracy of 92.59% was obtained for the test set.

The proposed study is devised in four parts. In first section color spaces are described. In the second section, dataset, programming languages and hardware resources are added. In order to the paper to be easier to follow a flowchart is proposed in Proposed Methodology and CNNs architecture. The section V contains the results and discussions, and in the last part of paper the conclusions are drown.

II. COLOR SPACES

Color space is the range of colors that an image-editing program can display. In scientific literature exists more color spaces, but in this paper will study the RGB, YCbCr, HSV, HLS, L*a*b and L*u*v color spaces. In the following, the mathematical description of each color channel is shown.

1. The well-known RGB color space is used to describing the electronic devices what have a display, such as phone or computer monitors and color television. A color form this space is obtained mixing of the three primaries value from [1, 256] range in different proportions. The next color spaces can be obtained from RGB or vice versa, the conversion methods were chosen from OpenCv library.

Cite as: F. Batog and S. Moldovanu, "The monitoring of burning buildings with convolutional neural network", *Syst. Theor. Control Comput. J.*, vol. 3, no. 2, pp. 1–8, Dec. 2023.
DOI: 10.52846/stccj.2023.3.2.50

2. YCbCr color space is composed of Y is luminance, Cb and Cr are Chrominance Blue [9], the mathematical interpretation of YCbCr space by equations (1)-(7) is given.

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (1)$$

$$Cr \leftarrow (R - Y) \cdot 0.713 + delta \quad (2)$$

$$Cb \leftarrow (B - Y) \cdot 0.504 + delta \quad (3)$$

$$R \leftarrow Y + 0.713 \cdot (Cr - delta) \quad (4)$$

$$G \leftarrow Y - 0.714 \cdot (Cr - delta) - 0.344 \cdot (Cb - delta) \quad (5)$$

$$B \leftarrow Y + 1.773 \cdot (Cb - delta) \quad (6)$$

$$delta = \begin{cases} 128 \; for \; 8 - bit \; images \\ 32768 \; for \; 16 - bit \; images \\ 0.5 \; for \; floating - points \; images \end{cases} \quad (7)$$

3. HSV (Hue, Saturation, Value) color space is a conic or cylinder geometrical space. It is an alternative for RGB color space, their target being to come closer to color human perception. The equations (8)-(10) express the transformations form RGB to HSV color space.

$$Y \leftarrow (R, G, B) \quad (8)$$

$$S \leftarrow \begin{cases} \frac{V - (R,G,B)}{V}, \; if \; V \neq 0 \\ 0, \; otherwise \end{cases} \quad (9)$$

$$H \leftarrow \begin{cases} \frac{60(G - B)}{V - min(R, G, B)} & if \; V = R \\ 120 + \frac{60(B - R)}{V - min(R, G, B)} & if \; V = G \\ 120 + \frac{60(R - G)}{V - min(R, G, B)} & if \; V = B \\ 0 & if \; R = G = B \end{cases} \quad (10)$$

4. HLS (Hue, Lightness, Saturation) color space looks like HSV, the difference between them is the information about the lightness [1]. For obtaining of HSL from RGB color space the equations (11)-(15) were used.

$$Y_{max} \leftarrow (R, G, B) \quad (11)$$

$$Y_{min} \leftarrow (R, G, B) \quad (12)$$

$$L \leftarrow \frac{Y_{max} + Y_{min}}{2} \quad (13)$$

$$S \leftarrow \begin{cases} \frac{Y_{max} - Y_{min}}{Y_{max} + Y_{min}} & if \; L < 0.5 \\ \frac{Y_{max} - Y_{min}}{2 - (Y_{max} + Y_{min})} & if \; L \geq 0.5 \end{cases} \quad (14)$$

$$H \leftarrow \begin{cases} \frac{60(G - B)}{Y_{max} - Y_{min}} & if \; Y_{max} = R \\ 120 + \frac{60(B - R)}{Y_{max} - Y_{min}} & if \; Y_{max} = G \\ 120 + \frac{60(R - G)}{Y_{max} - Y_{min}} & if \; Y_{max} = B \\ 0 & if \; R = G = B \end{cases} \quad (15)$$

5. The L*a*b color space is derived from the CIE XYZ (International Commission on Illumination) tristimulus values. It is a uniform system in terms of human perception. It means that the mathematical distance between two colors is directly proportional with the perceptual distance between them, where L* is perceptual lightness and a* and b* for the four unique colors of human vision: red, green, blue and yellow [9].The relations (16)-(23) show the content on each channel color of L*a*b color space.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (16)$$

$$X \leftarrow \frac{X}{X_n}, where \; X_n = 0.950456 \quad (17)$$

$$Z \leftarrow \frac{Z}{Z_n}, where \; Z_n = 1.088754 \quad (18)$$

$$L \leftarrow \begin{cases} 116 * Y^{\frac{1}{3}} - 16 & for \; Y > 0.008856 \\ 903.3 \cdot Y \; for \; Y \leq 0.008856 \end{cases} \quad (19)$$

$$a \leftarrow 500\big(f(X) - f(Y)\big) + delta \quad (20)$$

$$b \leftarrow 200\big(f(Y) - f(Z)\big) + delta \quad (21)$$

$$f(t) = \begin{cases} t^{\frac{1}{3}} & for \; t > 0.008856 \\ 7.787t + \frac{16}{116} & for \; t \leq 0.008856 \end{cases} \quad (22)$$

$$delta = \begin{cases} 128 \; for \; 8 - bit \; images \\ 0 \; for \; floating - points \; images \end{cases} \quad (23)$$

6. L*u*v* color space, where L, U and V values give the coordinates of the colors in the CIE system, L* for perceptual lightness and the u* and v* coordinates measure positions on green/red and blue/yellow axes [9]. The X, Y and Z valuables shown by equation (16) corroborated with equations (24)-(28) contribute to the composition of the L*u*v* color space.

$$L \leftarrow \begin{cases} 116 * Y^{\frac{1}{3}} - 16 & for \; Y > 0.008856 \\ 903.3 \cdot Y \; for \; Y \leq 0.008856 \end{cases} \quad (24)$$

$$u' \leftarrow 4 * \frac{X}{(X + 15 * Y + 3Z)} \quad (25)$$

$$v' \leftarrow 9 * Y / (X + 15 * Y + 3Z) \quad (26)$$

$$u' \leftarrow 13 * L * (u' - u_n) \; where \; u_n = 0.19793943 \quad (27)$$

[1] "HLS and HSV Color Representations," accessed May 30, 2023, https://support.ptc.com/help/mathcad/r8.0/en/index.html#page/PTC_Mathcad_Help/hls_and_hsv_color_representations.html.

$$v \leftarrow 13 * L * (v' - v_n) \; where \; v_n = 0.46831096 \quad (28)$$

III. Dataset Description and Programming Languages

To realize the dataset used in this paper two datasets from the Kaggle platform were proposed, the images selected were grouped into two classes: fire and nonfire. The reason for collecting images from more datasets was to feed the CNNs with which contain fire or nonfire. The first dataset from Kaggle was made for NASA Space Apps Challenge competition in 2018. The target was to develop a model which identifies the images with fire. This dataset contains 999 images organized in two folders: a folder with 755 images that capture the fire and a folder with 244 images that do not capture the fire[2]. The second dataset from Kaggle contains 100.000 images of public and private buildings, both from outside and inside of the buildings[3]. After the images were selected from both datasets mentioned earlier, it was gained 345 images where 165 images captured fired buildings and 180 images that did not capture fired buildings.

Within this project, it was used as the programming language Python because it has a lot of libraries dedicated to the Data Science domain. From these libraries, it was used Numpy to work with arrays, multidimensional arrays and matrices[4]; Matplotlib for graphic representation of the model results [5], OpenCV for image processing [6] and Tensorflow to create the CNNs[7]. Also, it was used Google Colaboratory [8] for the easy image processing. To obtain the results from this paper, the tools mentioned earlier were used on a laptop with Intel Celeron 1.10 GHz processor and 4 GB RAM on 64-bit operating system.

IV. Proposed Methodology and CNN Architecture

The study reported in this paper has outlined a methodology for the researching in four stages (see Fig. 1). The main stage was collecting of original images from two datasets and uploading them on Google Drive (see Fig. 2). The second stage consisted of preprocessing of images in six color spaces (see Fig. 3 and 4) and the third stage was projecting a CNN so that to be earned the best accuracy in classification process.

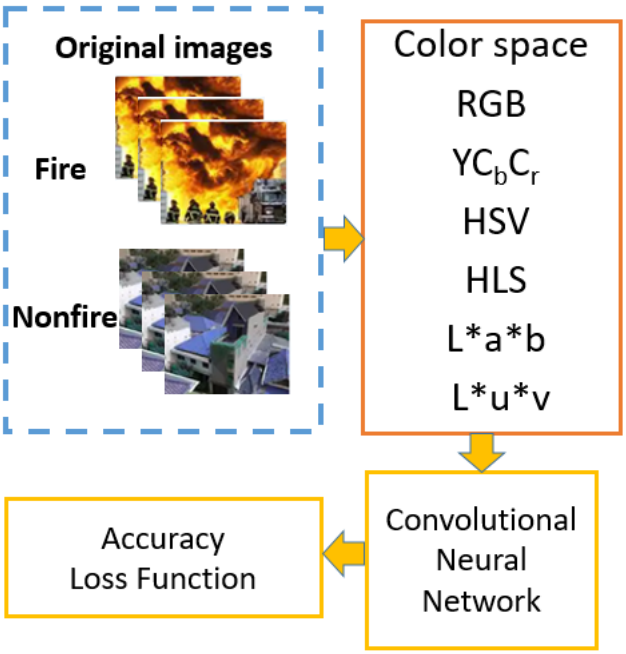


Fig. 1. The flowchart of the proposed study.

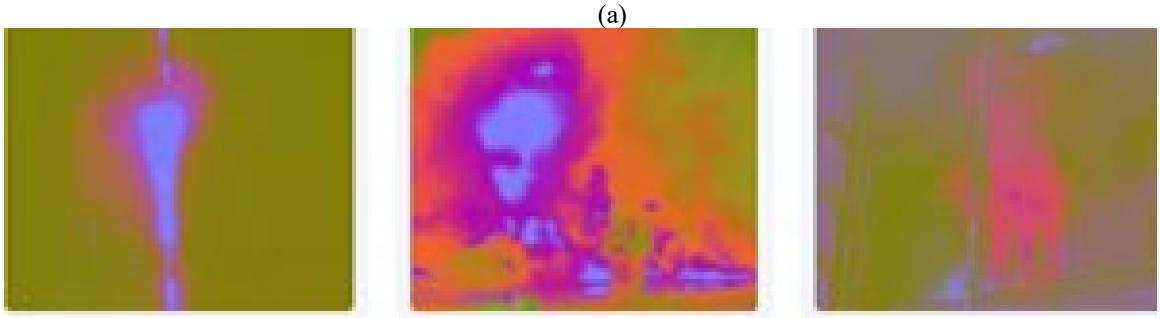
[2] "FIRE Dataset | Kaggle," accessed July 18, 2023, https://www.kaggle.com/datasets/phylake1337/fire-dataset.
[3] "Modern Architecture (100k Images) | Kaggle," accessed July 18, 2023, https://www.kaggle.com/datasets/tompaulat/modernarchitecture.
[4] "NumPy Documentation — NumPy v1.25 Manual," accessed June 26, 2023, https://numpy.org/doc/stable/.
[5] "Matplotlib — Visualization with Python," accessed June 26, 2023, https://matplotlib.org/.
[6] "OpenCV - Open Computer Vision Library," accessed June 26, 2023, https://opencv.org/.
[7] "Introduction to TensorFlow," accessed June 26, 2023, https://www.tensorflow.org/learn.
[8] "Colab.Google," accessed June 26, 2023, https://colab.google/.

(a)

(b)

Fig. 2. The original images from dataset; (a) images from the class fire; (b) images from the class nonfire

(a)



(b)

(c)

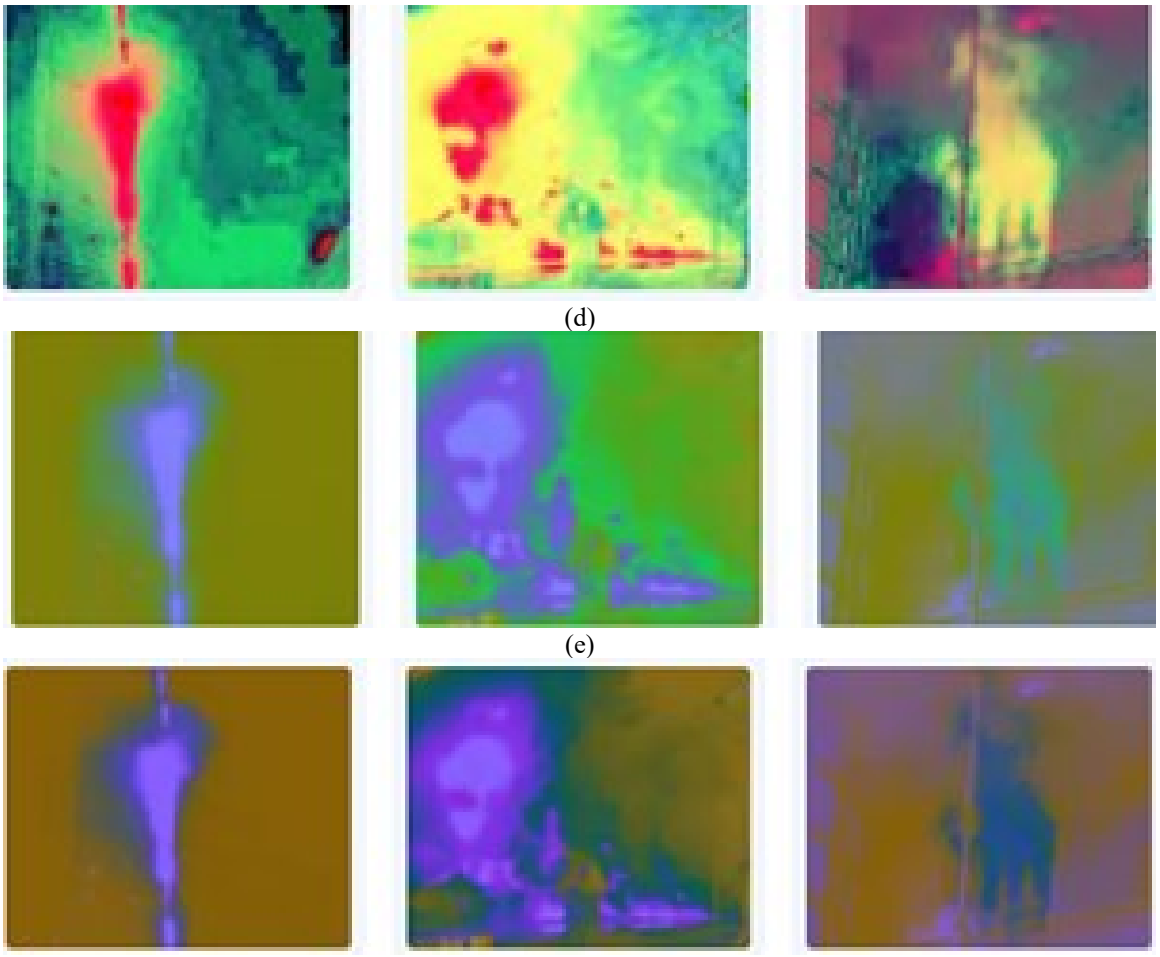
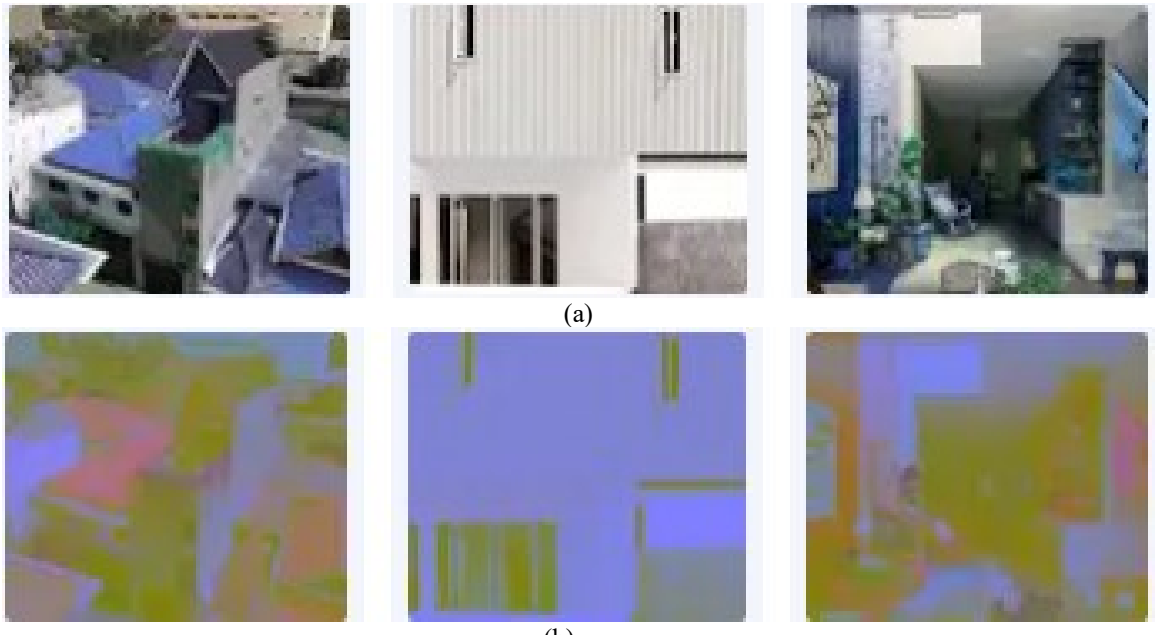


Fig. 3. Images with fire converted with color spaces; (a) RGB; (b) YCbCr; (c) HSL; (d) HSV; (e) L*a*b; (f) L*u*v



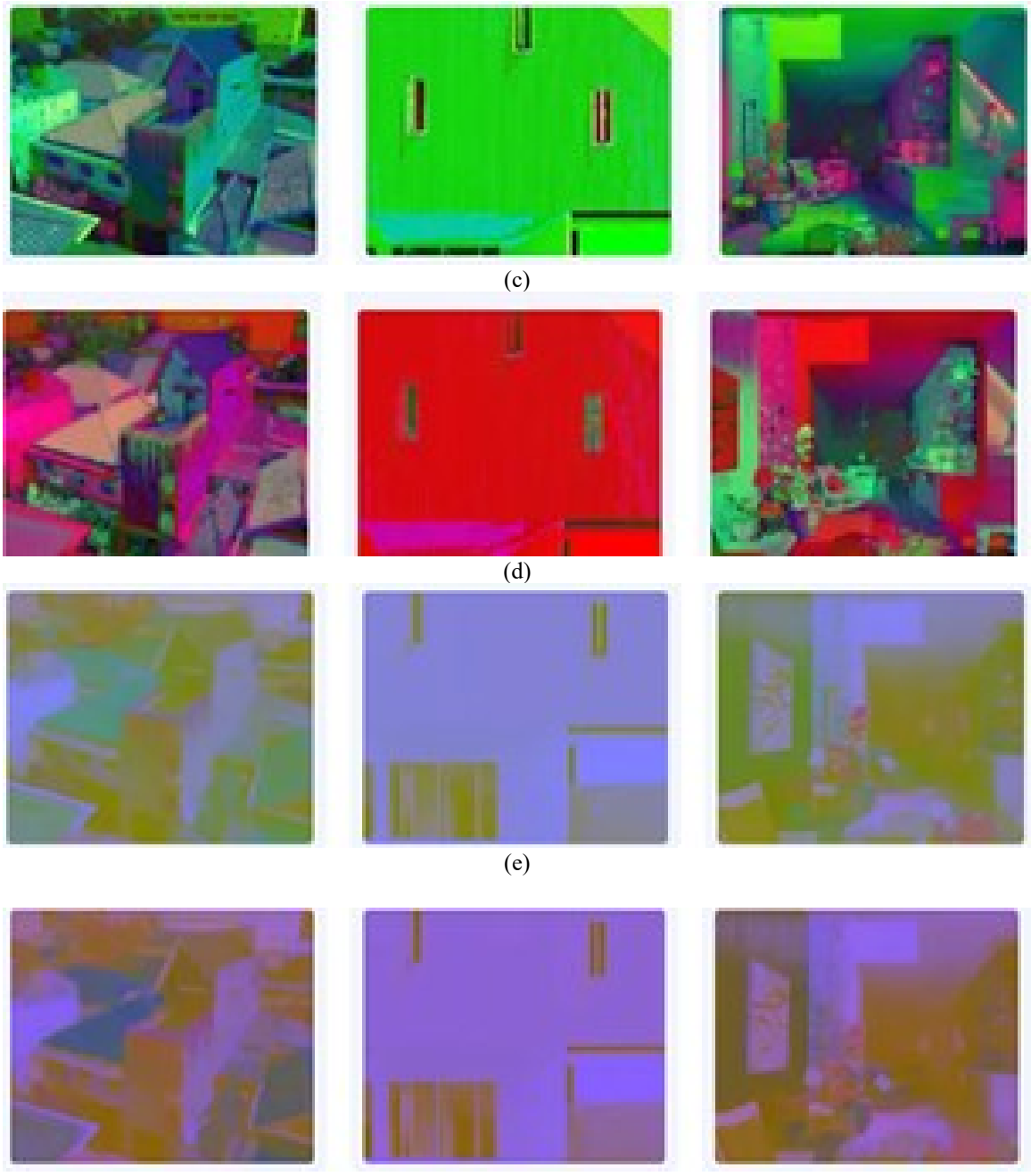


Fig. 4. Images without fire converted with color spaces; (a) RGB; (b) YCbCr; (c) HSL; (d) HSV; (e) L*a*b; (f) L*u*v

After visualization of the images from both classes passed through the color spaces we cannot choose the best color spaces used to identify the fire from images. So, we will use the images processed in all color spaces in the CNNs, followed by checking the model accuracy for every case. The next step was to move the images to different folders, a folder for the images processed with color spaces where we also create two folders: one with images for training of CNN and another folder with images for testing of CNN, which in their turn contains two folders, one that contains images that capture burning buildings and one that contains images that do not illustrate buildings set on fire. The proportion is 80% of the total number of images for the training folder and 20% of the total number of images for the CNN test folder.

The next step is to train the neural network based on the images resulting from their processing in the color spaces. To train the CNN it is necessary to divide the images set into a training set, into a test set, and to create the validation set from the training set. Also, the width, height and the batch sizes must be set because the images used as input for the CNN may have different sizes and this would negatively affect the CNN performance. In this paper, the images width and height were set to 80 and the batch sizes to 10 and 20. Next, in the CNN architecture the size of images was decreased using the Sequential class from the Keras. Next, the images were augmented and for each layer the ReLu function was activated.

After creating the CNNs, the architecture and hyperparameters are shown in Table I, these were compiled and trained for 10, 20, 30, 40 and 50 epochs. Also, to update CNN weights in iterative process it was proposed the optimizer Adam and the obtained accuracy on testing was kept

for each running. After that, it was analyzed the accuracy and loss function after the CNN compilations and it was kept the number of epochs with the highest accuracy result which will be used with the batch size of 10 and 20.

TABLE I. Hyperparameters of the deep CNN architecture

Layer (type)	**Output shape**
rescaling (Rescaling)	(None, 80, 80, 3)
conv2d (Conv2D)	(None, 80, 80, 32)
conv2d_1 (Conv2D)	(None, 80, 80, 32)
max_pooling2d (MaxPooling2D)	(None, 40, 40, 32)
conv2d_3 (Conv2D)	(None, 40, 40, 64)
max_pooling2d_1 (MaxPooling2D)	(None, 40, 40, 64)
conv2d_4 (Conv2D)	(None, 20, 20, 128)
conv2d_5 (Conv2D)	(None, 20, 20, 128)
max_pooling2d(MaxPooling2D)	(None, 20, 20, 128)
flatten (Flatten)	(None, 12800)
dense (Dense)	(None, 128)
dense_1 (Dense)	(None, 128)
Batch size	10, 20
Learning rate	0.001
Number of epochs	10, 20, 30, 40, 50

The CNN architecture used in this paper contains six convolutional layers and three pool layers, one flatten layer and two dense layers. First of all the images from input are rescaling at 80 x80 pixels. Based on a kernel of 3x3, the first two convolutional layers help us to extract the basic features like borders and the next convolutional layers help us to extract more complex features like texture, objects and patterns. The pool layers take all features extracted with the convolutional layers and with the max() function take the maximum value from the filter. The pool layers help us to reduce the size of the input image and to avoid the overfitting. After applying this function we can see that the width and the height of the image in pixels reduced from 80 pixels to 40 pixels after first pool layer, from 40x40 pixels to 20x20 pixels after the second pool layer and from 20x20 pixels to 10x10 pixels after the third pool layer. As we go in depth, the number of output channels grow, from 3 to 32, from 32 to 64 and from 64 to 128. The flatten layer has the role to convert the feature map what it received from the max-pooling layer into a format so that the next layer called dense layer be able to understand the data. The dense layer cannot miss from classification process, because it is used to binary classify images based on output from convolutional layers.

The performance of the binary classification is evaluated by accuracy gives by equation 1, and loss function shown by equation 2, the accuracy was computed from test dataset form 2x2 confusion matrix.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{29}$$

where TP (true positives), TN (true negatives), FP (false positives), FN (false negatives) of the samples [10].

The proposed loss function was mean squared error, (MSE) it is the most popular loss function that finds the average of the squared differences between the target value and the predicted values outputs

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \widehat{y^{(i)}})^2 \tag{30}$$

where $y^{(i)}$ are target values and $\hat{y}^{(i)}$ are values outputs [11], [12].

V. Results and Discussion

This section presents the testing results of each color spaces. It is proposed the ablation process which contains the number of epochs and the batch size. After powering the CNNs with images passed through color spaces and its training for 10, 20, 30, 40 and 50 epochs the results were stored in Table II. Also, the Table II stored the accuracies and the losses of each model applied on images passed through all space color.

TABLE II. The accuracy and loss scores for each color space for 10, 20, 30, 40 and 50 epochs

Color space	**Accuracy/epochs**				
	10	***20***	***30***	***40***	***50***
RGB	0.971	0.971	0.985	**0.985**	0.985
YCbCr	1	**1**	0.985	**1**	**1**
HSV	1	0.942	**1**	0.971	0.956
HLS	1	**1**	0.782	0.985	0.956
L*a*b	0.971	0.985	0.985	0.985	**1**
L*u*v*	0.985	0.971	**1**	**1**	**1**
	Loss function				
	10	***20***	***30***	***40***	***50***
RGB	0.110	0.055	0.021	**0.027**	0.091
YCbCr	0.127	0.006	0.013	**$1.2\cdot 10^{-04}$**	0.001
HSV	0.062	0.219	**0.007**	0.073	0.248
HLS	0.006	**0.002**	0.626	0.111	0.435
L*a*b	0.092	0.022	0.026	0.020	**0.001**
L*u*v	0.097	0.113	0.014	0.005	**$2.7\cdot 10^{-04}$**

As we can see, for RGB and YCbCr the best model performed on 40 epochs, for HSV the best model performed on 30 epochs, for HLS the best model performed on 20 epochs, for L*a*b and L*u*v the best model performed on 40 epochs.

In CNNs is important the number of samples used in one forward and backward pass through the layers, so that the study continued with 10 and 20 batch size. So that, the next step is to take the best model for each space color based on the best accuracy score and the lowest loss function, these values are putted in highlight in Table I, and in ablation process was chosen for each color the emphasis values and the CNNs were run for 10 and 20 batch size. The results are stored in Table III.

TABLE III. The accuracy and loss scores for each color space for batch size 10 and 20

Color Space	**Accuracy/ Batch size**		**Loss function/ Batch size**	
	10	*20*	*10*	*20*
RGB	0.985	0.971	0.021	0.084
YCbCr	**1**	1	**$4.02\cdot 10^{-05}$**	0.009
HSV	1	0.956	$9.54\cdot 10^{-04}$	0.163
HLS	0.956	0.956	0.159	0.186
L*a*b	1	0.956	$6.07\cdot 10^{-04}$	0.194
L*u*v*	1	1	$1.74e^{-04}$	$3.9\cdot 10^{-04}$

Corroborating the results obtained in ablation process, and values stored in table 2 and 3 for each studied space color, it seen that for YCbCr, HSV, L*a*b, and L*u*v* the accuracy is 100% percentage, but the difference is given by loss function.

VI. CONCLUSIONS

This paper has proposed to find the best color space to identify the fire from images. To achieve this, the images from the dataset were passed through more color spaces such as RGB, YCbCr, HSV, HLS, L*a*b and L*u*v. The ablation process consisted of 10, 20, 30, 40 and 50 epochs, and 10 and 20 batch size for each space color. The results provided by this process shew that YCbCr is the color space what classifies the best the images with fire and without fire. In this case the accuracy is of 100% and loss function by $4.02\cdot 10^{-05}$, these results are obtained for 40 epochs and 10 batch size. In the future research, the study will be performed on each color channel of color spaces.

REFERENCES

[1] K. Muhammad, J. Ahmad, and S. W. Baik, “Early fire detection using convolutional neural networks during surveillance for effective disaster management,” Neurocomputing, 2018, vol. 288, pp. 30–42.

[2] K Y. J. Kaufman, D. D. Herring, K. J. Ranson and G. J. Collatz, "Earth Observing System AM1 mission to Earth," in IEEE Transactions on Geoscience and Remote Sensing, 1998, vol. 36, no. 4, pp. 1045-1055.

[3] L. Giglio, J. D. Kendall, and C. O. Justice, “Evaluation of global fire detection algorithms using simulated AVHRR infrared data,” International Journal of Remote Sensing, 1999, 20, pp. 1947 – 1985.

[4] L. Giglio, J. D. Kendall and R. Mack, “A multi-year fire data set for the tropics derived from the TRMM VIRS,” International Journal of Remote Sensing, 2003, pp.1-10.

[5] Li and Zhao, P. Li, W. Zhao, Image fire detection algorithms based on convolutional neural networks, Case Stud. Thermal Eng., 2020, vol. 19, pp. 100625.

[6] Majid S., Alenezi F., Masood S., Ahmad M., Gündüz E.S., Polat K., Attention based CNN model for fire detection and localization in real-world imagesExpert Syst. Appl., 2022, vol. 189, pp. 116114.

[7] Celik T., Demirel H., Fire detection in video sequences using a generic color model. Fire Saf. J., 2009, vol. 44, pp. 147–158.

[8] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional Neural Networks Based Fire Detection in Surveillance Videos," in IEEE Access, 2018 vol. 6, pp. 18174-18183.

[9] R. Szeliski, "Computer Vision: Algorithms and Applications", Springer Nature, 2022, pp. 72-79.

[10] S. Moldovanu, C.D. Obreja, K. Biswas and L. Moraru, "Towards accurate diagnosis of skin lesions using feedforward back propagation neural networks," Diagnostics, 2021, vol. 11, pp. 936.

[11] L.Chen, H. Qu, J. Zhao, B. Chen and J. C. Principe,”Efficient and robust deep learning with correntropy-induced loss function. Neural Comput. & Appl., 2016, vol. 27, pp. 1019–1031.

[12] M. Mihaela, A.L. Culea-Florescu and S. Moldovanu, “A Convolutional Neural Network for skin lesion classification in Dermoscopic Images using discrete wavelet transform,” IEEE European Control Conference (ECC), Bucharest, Romania, pp. 1-5, 13-16, June 2023.