22nd International Conference on Renewable Energies and Power Quality
(ICREPQ'24) Bilbao (Spain), 26th to 28th June 2024

ISBN: 978-84-09-60656-6

Image processing to calculate flame velocity based on Deep Learning

Celso De-La-Cruz[1], Josué Torres[1] and Freddy Rojas[1]

[1] Department of Engineering,
Pontifical Catholic University of Peru,
Lima 15088, Peru,

Abstract. Flame velocity is one of the main characteristics of combustion. One of the methods to measure flame velocity is to determine the flame area by approximating the shape of the flame to a flat, conical, or truncated cone surface. To increase the precision in the calculation of the flame area and thus improve the calculation of the flame velocity, the use of semantic segmentation of the image is proposed to determine the flame image contour and then obtain the flame area with an integral calculation. Semantic segmentation was performed using the U-net structure. The results were compared with other classical image processing techniques showing the advantages and disadvantages. The results show that semantic segmentation based on Deep Learning segments the flame image with high precision even under changing conditions such as flame color variations and background imperfections.

Key words. Laminar flames, burning velocity, image processing, deep learning.

1. Introduction

Flame velocity is used as one of the most important characteristics in the study of the combustion of various fuels [1], [2], [3]. One of the methods to measure the flame speed is to determine the flame area in order to calculate the flame velocity by dividing the flow rate of the mixture by the flame area [4], [5]. For this calculation, the area is estimated by approximating a known geometric shape as a cone or truncated cone. This approximation generates error in the calculation of the flame area. A more accurate way of calculating flame speed is to get a flat flame in which the flame area matches the burner area. [6], [7]. However, getting the flame flat is very complicated.

On the other hand, image processing is used in applications such as flame lift-off detector, flame stability measurement, indoor and outdoor fire detection, and flame monitoring. In the work of Gharib, et al. [8] a convolutional autoencoder and a regression neural network was used to detect the flame lift-off in a 1 MW reaction chamber. In [9] a numerical indicator was proposed to measure flame stability in a 600 MWth coal-fired supercritical unit based on brightness, non-uniformity, average temperature and flame oscillation frequency; Image processing techniques were used to determine these parameters. Other research deals with indoor and outdoor fire detection applying the techniques: dynamic background segmentation and Extreme Learning Machine to detect the flame based on geometric characteristics of the segmented flame [10], unsupervised method of flame segmentation and detection based on image processing [11], and others. In [12] flame monitoring is carried out in a thermal power station using a Convolutional Neural Network – CNN after applying clustering of the flame image based on the K-means algorithm; this image processing was applied in order to obtain a temperature estimation of the flame combustion.

In the present work the determination of the flame area from the flame contour detected using image processing techniques is proposed. This is a more accurate method of calculating the flame area compared to those found in the literature. The use of semantic segmentation of the flame by applying a Deep Learning algorithm with the U-net structure to detect the flame contour is also proposed. The U-net algorithm was compared with classical image processing techniques. The results show that the U-net algorithm successfully segments the flame despite changes in color and brightness in the flame and background imperfections. Since the lighting and background conditions were controlled, the requirement of the number of images for training the U-net was very low compared to other applications where this algorithm is applied.

The present work is important in the field of energy efficiency because the detection of flame speed allows a more efficient combustion.

2. Image acquisition

The acquisition of images of the flame was carried out using the BFS-PGE-13Y3C-C camera from Teledyne FLIR; main characteristics and used modes are shown in Table I.

The conditions of images capture were: dark background, although with slight imperfections with shiny parts that caused small light spots; low light; camera position with minimum movements between one recording and another. This gave us controlled imaging conditions.

Table I. Teledyne Flir Blackfly S GigE BFS-PGE-13Y3C-C

Resolution	1280x1024
Megapixels	1.3
Mas Frame Rate Standard	85
Sensor Model	ON Semi PYTHON 1300
Data Interface	GigE PoE
Spectrum	Color
Acquisition mode	Continuous
Exposure time	5 ms
Image processing	None

First, many videos of the flame were recorded under different conditions; after that, a Matlab program was used to convert the videos into image sequences of the ".jpg" type. This image database was used for the two methodologies to be implemented for image processing: the first using a classical approach and the second one based on Deep Learning. Figure 1 and Figure 2 shows an examples of image acquisition results.

3. Classical Image processing

For this segmentation methodology, two methods were used: binary thresholding and the Otsu method.

A. Binary thresholding

The thresholding method consists of selecting a threshold value for a grayscale image, each pixel that is below the threshold value will be considered zero and each pixel that is above the threshold value will be considered one, because of this, red, green and blue channels of the RGB color plane; and hue, saturation and value channels from a HSV color plane were analyzed. Figure 3 shows the original RGB image that is going to be analyzed to get masks; figure 4 shown red, green and blue channels of RGB plane, and figure 5 shows hue, saturation and value channels of HSV plane.

Figure 1. Example 1 of image acquisition.

Figure 2. Example 2 of image acquisition.

Figure 3. RGB Image.

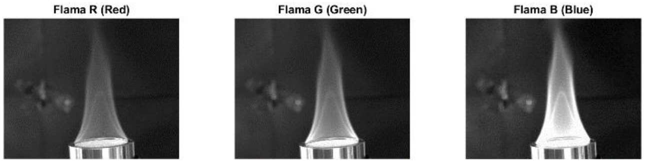

Figure 4. R, G, B channels of original flame image.

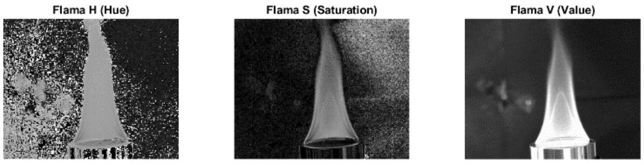

Figure 5. H, S and V channels of original flame image.

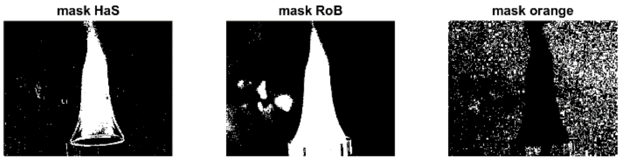

Figure 6. Masks for flame segmentation.

As seen in Figure 4, the red and blue channels provide greater contrast, which facilitates thresholding, while the green channel does not provide much information. On the other hand, it can be seen that the burner nozzle presents very similar intensity to the flame in red and blue channels.

On the other hand, as shown in Figure 5, channel H manages to relatively segment the blue part of the flame, with some background portions and without the burner nozzle. Likewise, channel S also manages to segment the blue flame with a little of the orange part and also taking into account portions of the background, and, in the same way as channel H, without the burner nozzle.

Therefore, 3 masks were generated, the first called RoB, which is the logical "or" pixel-by-pixel of a mask thresholded in the R channel and a mask thresholded in the B channel of the RGB color space, the second one was called HaS, which is the logical "and" pixel-by-pixel of mask thresholded in the H channel and a mask thresholded in the S channel, and the third one was realized to segmentate the orange part of the flame and only used a mas thresholded in the H plane. Results are shown in figure 6.

The mask with the best result is the HS mask, however, to further improve this mask, the morphological operation of image closure [13] is used. To obtain a better result in the closing operation, the image is first blurred with a 2D low-pass filter (n = 11) as in (1).

$$filtered\ image = image * filter_{lowpass}$$

$$filter_{lowpass} = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} * \left(\frac{1}{11^2}\right); \quad n = 11 \qquad (1)$$

After that, the closing operation is applied with a disk strel of r = 50 (Figure 7) to the blurred image as in equation (2).

$$filtered\ image = image\_close(image, strel) \qquad (2)$$

The result of low-pass filter and close operation is shown in Figure 8. Finally, a logical "and" of this HS processed mask with the RB mask is performed to recover details of it but without the burner nozzle, and the semantic segmentation of the flame is carried out with this final mask as shown in Figure 9.

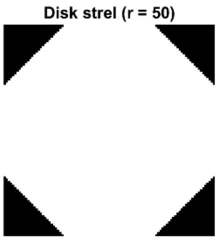

Figure 7. Disk strel (r=50)

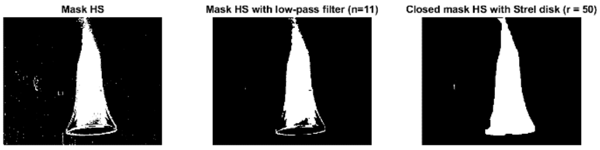

Figure 8. Low-pass filter and close operation applied to HS mask

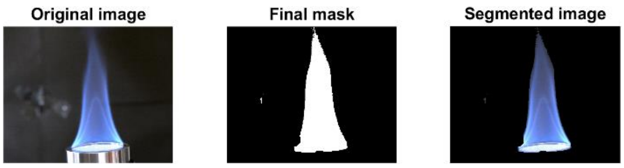

Figure 9. Flame segmentation.

Figure 10. Example of another Flame semantic segmentation

Figure 11. Example of flame with orange portion

In the examples of Figures 9-11, it is observed that when the flame is completely blue the algorithm manages to segment the flame very precisely. However, it is not able to recognize the orange portions of the flame, which are rarely segmented, because when a mask for the color orange in the HSV color plane is used, a lot of the background is segmented as a noise.

B. Otsu method

In this part, Otsu method [14], a well-known method for image thresholding is used to segmentate the flame. This method was applied to RGB plane and HSV and the results are shown in Figures 12 and 13.

After trying this method many times, the results have been very variable, sometimes it recognized the flame but not entirely and in others it recognized the entire flame but with a lot of background noise. The processed image was in completely blue flames and with portions of orange, shown in Figure 11.

The Classical Image Processing can be successfully applied when the background is completely black, requiring special conditioning of the experimental environment. But when there are some imperfections in the background, classical processing could not be suitable. The advantage of Classical Image Processing is that it does not require advanced hardware as GPU.

4. Deep learning for semantic segmentation

The Deep Learning framework used was U-net [15]. This algorithm performs convolution in the first layers (contraction path) extracting image features, meanwhile, in the last layers deconvolutions (expansion path) are used to obtain the mapping to the original image size. The detailed structure is shown in Figure 14.

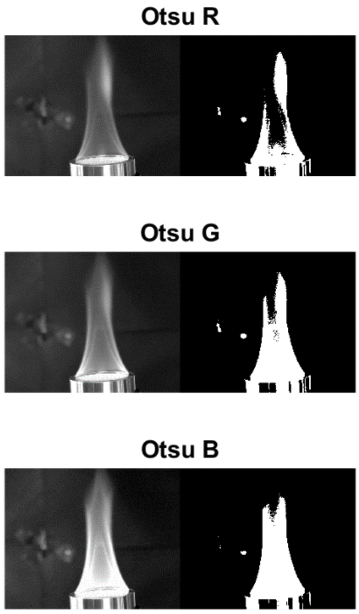

Figure 12. Otsu method to RGB image

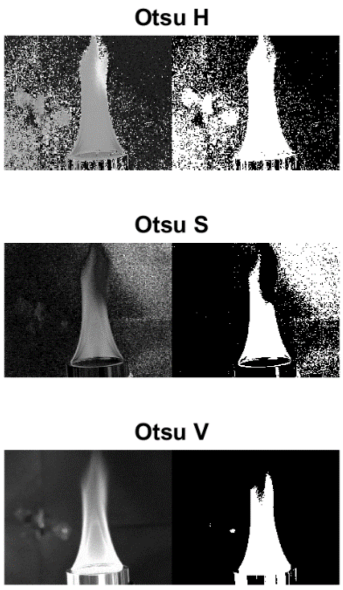

Figure 13. Otsu method to HSV image

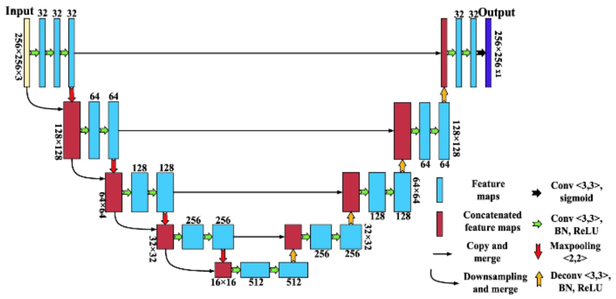

Figure 14. U-net architecture based on [15].

To train neural networks, a large number of images are needed, in the order of thousands, especially in semantic segmentation algorithms; however, being in a controlled environment with a dark background and low lighting means that the neural network does not need a large number of images to be trained and obtain good precision, because of this, 120 (256x256x3) labeled images were used as a base for which data augmentation techniques were applied such as: zoom in, zoom out, rotation (small angles) and adjustments in HSV values to simulate different light intensity of natural light (see Figure 15-16), thus resulting in a dataset of 500 images of size 256x256x3.

Figure 15. Data augmentation function applied to an image

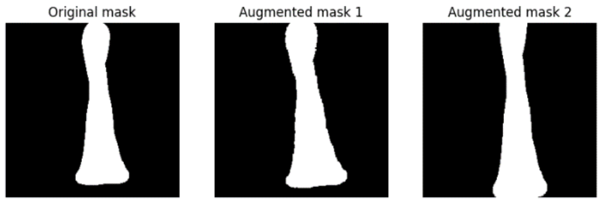

Figure 16. Data augmentation function applied to an image's mask

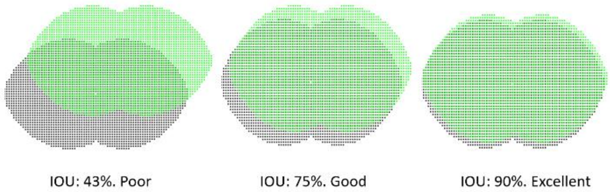

Figure 17. Example of IoU coefficients

For training, the following data distribution was made: 450 images for training (90%), 50 images for validation (10%). In addition, the Jaccard coefficient, also known as IoU (intersection over union) was used as a cost function (3).

$$\boldsymbol{J}(\boldsymbol{A},\boldsymbol{B}) = |A \cap B|/|A \cup B| \tag{3}$$

As is observed in Figure 17 a 75% value of IoU coefficient means a good prediction. After explaining this and performing different tests with different values of epochs and batch sizes, the configuration with the best result was: epochs = 25, batch size = 32, obtaining an IoU value of 90.7%. Figures 18 and 19 show the learning process of the neural network.

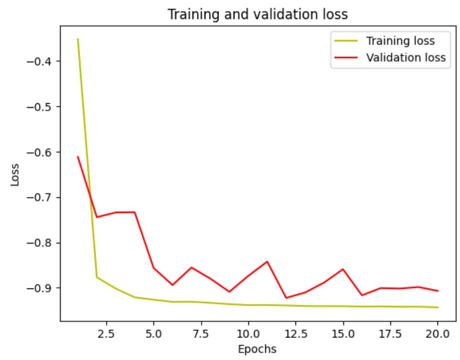

Figure 18. Training and validation loss graphic along epochs

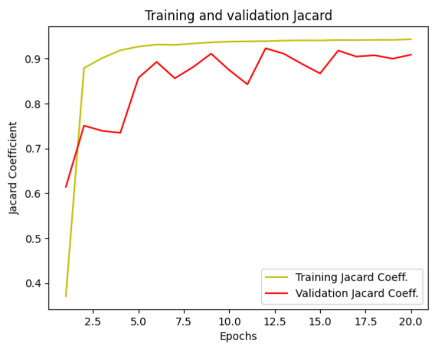

Figure 19. Training and validation IoU coefficient along epochs

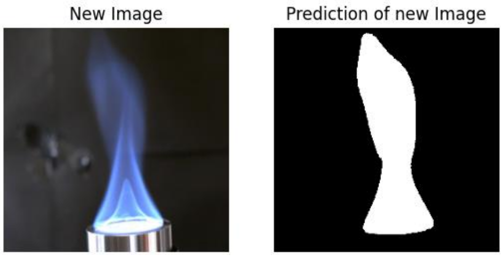

Figure 20. Example 1 of U-net prediction

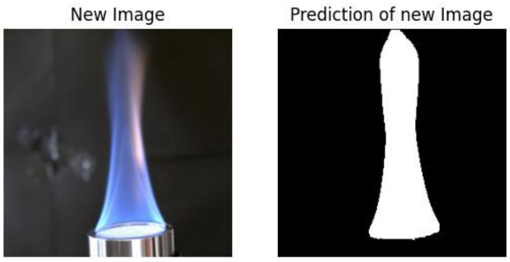

Figure 21. Example 2 of U-net prediction

Finally, some predictions made by the neural network to new images are shown in Figures 20-21.

As can be seen, the U-net neural network framework has very good predictions for flame segmentation, although the flame has orange portions, the predictions are highly effective.

5. Calculus of the lateral area of the flame

For the calculus of the lateral area of the flame, 2 methods were used, in the first one the flame is considered as a conical, so its area is calculated by equation (4) and (5).

$$A_L = \pi * g * r \tag{4}$$

Where *g* is the slant height of the cone and *r* is the radius of the cone base.
The second method should be more accurate, due to is based on calculate the partial area A_{pi} of a cone using an integral as shown in Figure 22 and equations (5-7), then, add all partial areas along the height of the flame divided in *n* portions. Numerical methods were used to calculate the integral of equation (7).

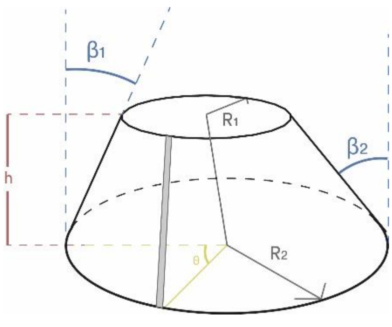

Figure 22. Partial area of a cone

$$\beta = \beta_1 + \frac{\theta}{\pi}(\beta_2 - \beta_1) \tag{5}$$

$$R_2 = R_1 - h \times \tan(\beta) \tag{6}$$

$$A_{pi} = 2\int_0^{\pi} \frac{h(R_1 + R_2)}{2\cos(\beta)} d\theta \tag{7}$$

$$A_L = \sum_i^n A_{pi} \tag{8}$$

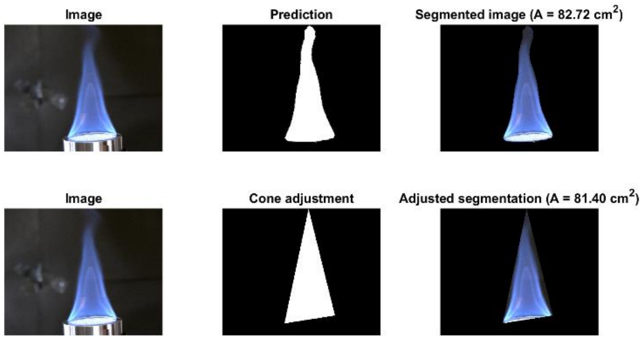

Figure 23. Lateral area of a flame

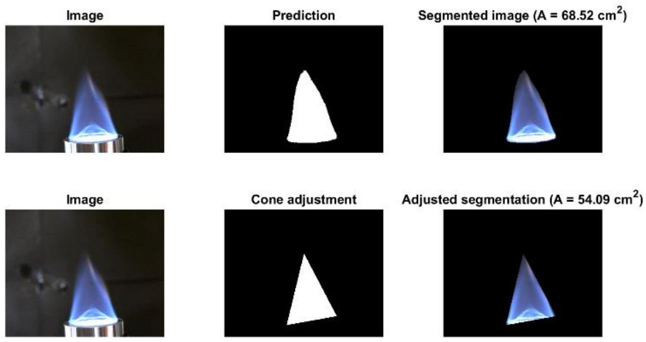

Figure 24. Lateral area of a flame

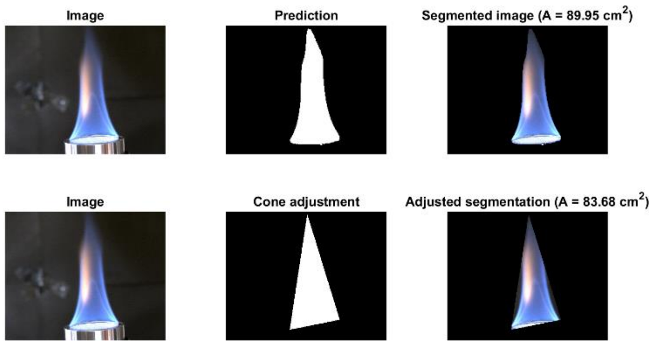

Figure 25. Lateral area of a flame

In addition, it is known that the width of the burner nozzle is 4.7 cm, which will serve to scale the measurements in pixels calculated in Matlab. Finally, some predictions of the algorithm are shown in the Figures 23-25.

As can be seen, the integral method is always larger, the reason is that this method makes a sum of the partial areas of the flame surface. Due to the flame shape, is not completely conical the integral calculus provides greater lateral area compared to the cone fit. In the same way, the cone adjustment segmentation obtains very good precision when the flame is long, just because the flame is more similar to a cone. When the flame is not as high, precision is lost with the cone approximation method.

The speed was 3 frames per second of the processing with deep learning and the calculation of the lateral area of the flame, using the MATLAB software and the hardware of intel core i5, 2.3 GHz, with an NVIDIA GeForce GTX 1050 GPU.

6. Conclusion

The U-net algorithm can segment the flame image despite variations in flame color, brightness and background imperfections, achieving an IOU metric of 90.7%.

The calculation of the flame area based on numerical methods allows calculating the area of flame shapes that are not well defined.

The number of images required for training the Deep Learning algorithm with U-net structure was 120, it is a very low number compared to other applications, the reason is that the images were taken under controlled lighting and background conditions.

Acknowledgement

Thanks are due to the DFI of the Pontificia Universidad Católica del Perú that, through the annual competition for applied research projects CAP PUCP 2021, financed research PI0735.

References

[1] D.V. Idirsov, et al., “Laminar Burning Velocity of the Surrogate of Aircraft Kerosene and its Components”, International Scientific and Technical Engine Conference (2021), doi: 10.1109/EC52789.2021.10016795.

[2] Y. Wang and S. Verhelst, “Comparative analysis and optimisation of hydrogen combustion mechanism for laminar burning velocity calculation in combustion engine modelling”, International Journal of Hydrogen Energy (2024), Vol. 56, pp. 880-893.

[3] J. Li, et al., “Laminar burning Velocities, Markstein numbers and cellular instability of spherically propagation Ethane/Hydrogen/Air premixed flames at elevated pressures”, Fuel (2024), Vol. 364.

[4] F. Rojas, F.O. Jiménez, and J. Díaz, “Determinación experimental de la velocidad de llama del gas de Camisea-Perú”, Información Tecnológica (2020), Vol. 31(4), pp. 99-112.

[5] G.P. Olivera, et al., “Experimental laminar burning velocity of syngas from fixed-bed downdraft biomass gasifiers”, Renewable Energy (2020), Vol. 153.

[6] S. Zhou, et al., “Effect of composition of biomass steam-gasified gas on laminar burning velocity and overall activation energy”, Biomass and Bioenergy (2024), Vol. 181.

[7] S. Eckart, et al., “Laminar burning velocity, emissions, and flame structure of dimethyl ether-hydrogen air mixtures”, International Journal of Hydrogen Energy (2023), Vol. 48.

[8] M. Gharib, et al., “Flame lift-off detector based on deep learning neural networks”, Combustion and Flame (2024), Vol. 260.

[9] W. Xu, et al., “Quantitative measurement of the stability of a pulverized coal fired flame through digital image processing and statistical analysis”, Measurement (2023), Vol. 206.

[10] C. Zhong, Z. Tian, H. Ding and K. Wang, "Video Flame Recognition Based on Feature Fusion and Extreme Learning Machine", 5th International Conference on Automation, Control and Robotics Engineering (2020), Dalian, China, pp. 702-705, doi: 10.1109/CACRE50138.2020.9230253.

[11] E. Buza and A. Akagic, “Unsupervised Method for Wildfire Flame Segmentation and Detection”, IEEEAcsess (2022), Vol. 10.

[12] S.M. Priyadharson and E.S. Reddy, “Lab VIEW and Deep Learning based Combustion Flame Monitoring and Fuel Flow Measurement in a Thermal Power Station”, IECON - 49th Annual Conference of the IEEE Industrial Electronics Society (2023).

[13] R. Haralick, et al., “Image Analysis Using Mathematical Morphology”, IEEE Transactions on Pattern Analysis and Machine Intelligence (1987), Vol. 9(4), pp. 541-543.

[14] P.A. Carela, “Tratamiento Digital de Imágenes en MatLab para la detección de fallas”, Final degree project. Universidad Politécnica de Cataluña (2019).

[15] O. Ronneberger, et al., “U-Net: Convolutional Networks for Biomedical Image Segmentation”, Computer Vision and Pattern Recognition (2015).