Superpixels Segmentation and Interpretable Fuzzy Models for Fire Data Annotation

Pedro Manuel Serro Messias
pedro.messias@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

January 2021

Abstract

Wildfires are natural disasters that can be quite unpredictable, burning large areas of forests and destroying properties. Fire detection and early prevention enable a faster reaction from emergency teams and to decrease the possibility of fire damage. Therefore, automatic systems capable of detecting fires are increasingly important. Their development requires a high number of data in order to guarantee good performances and be reliable in real scenarios. However, the low number and poor quality of available datasets in the literature, and the lack of annotations hamper the development of such automatic techniques. The objective of this work, developed in the framework of project Eye in the Sky (https://adai.pt/eyeinthesky/), is to propose an architecture based on segmentation and interpretable linguistic models capable of generating wildfire annotations. The proposed approach takes advantage of rich color features representative of fire in two different stages, segmentation and classification. The first one is related to generating superpixels and aggregating these into regions based on the representation of fire colors in the YCbCr color space. Subsequently, the classification of each region is achieved using interpretable rule-base models based on the HSL and YCbCr color spaces, which generates a pixel-wise fire segmentation and the semantic annotations of the fire colors. Furthermore, this method allows certain fine-tunable parameters in order to improve its overall results. The proposed approach is evaluated in different real contexts using a publicly available database.
Keywords: Fire detection, HSL color space, YCbCr color space, Superpixel, Interpretable Linguistic Models, Fire Data Annotations

1. Introduction

Natural disasters like floods, earthquakes and wildfires have a considerable environmental and economic impact. Every year, wildfires destroy hectares of lands, burning forests and villages, leading to the loss of material goods and possibly lives. Over the years, the increased number of wildfires has raised a need in developing automatic techniques capable of helping in wildfire fighting. Fire detection approaches able to provide earlier fire alerts would enable a faster response from emergency teams and reduce the possibility of fire destruction in large areas and, potentially, material goods.

1.1. Fire detection topic overview

Conventional fire detection approaches are usually based on data collected from ultraviolet or infrared fire sensors. In addition, wireless sensor networks (WSNs) emerge as an alternative to conventional techniques. These consist on a large number of cheaper and smaller sensors able to collect different types of data (e.g., temperature or carbon monoxide density) and send an alarm whenever a fire is detected. However, those sensors are required to be in close proximity of a fire in order to register reliable data, which can be quite expensive in terms of their deployment and maintenance, specially in large fields [2]. For fire detection, several computer vision-based approaches utilize the visual features encountered in images and videos, specially color since fires are usually representative of bright and warm colors. Such approaches are based on color spaces, which are associated with a crisp representation of colors. The most well-known is the RGB color space, as it is related to how the human eye perceives color. Moreover, other approaches exploit the HSV color space [5] or the YCbCr [10]. Several classical computer vision techniques utilize visual and motion features of fire and smoke areas to develop better and more reliable classifiers. However, since classifying a fire pixel in terms of colors can be a subjective task, these techniques using rules with fixed fine-tuned values have certain limitations when applied to different scenarios or datasets. Considering these limitations, the scientific community has been applying visual fire attributes (e.g., color or motion features) with intelligent systems methods. Deep learning methods have been widely used as these are able to handle high-dimensional data like images or videos [6]. Many fire detection techniques are hampered by the overall quality of datasets and the lack of annotations. These limitations are usually associated with the low number and poor quality of samples, and the representation of different contexts than nature scenery, e.g., fires in buildings or indoor. Moreover, several images come from videos, increasing the number of samples but not its significance as these are very similar to each other. Finally, the lack of annotations can be a hindrance for a better comprehension of the existent models.

1.2. Proposed approach

To address such limitations, this work tackles the development of a method for fire data annotations through semantic segmentation, instead of the more common fire detection approaches. In fact, the proposed approach presents a method to generate fire segmentations and labels describing fire colors, which can then be validated by experts related to wildfires to create reliable ground truth data. At a first stage, the proposed architecture relies on the rich color features representative of fire, namely in the HSL and YCbCr color spaces, which allow an insightful interpretation that, in return, enables the color-based superpixel segmentation. Afterwards, interpretable rule-based linguistic models are employed to classify superpixels in terms of their color attributes to infer which correspond to the fire or non-fire classes. Moreover, these models allow to generate semantic labels describing the fire colors present in the image [8]. The proposed approach is evaluated against a subset of the Corsican Fire Database, demonstrating excellent segmentation results and their ability to handle a variety of real-contexts, e.g., with fire at long-distances, with firefighters or firetrucks, and in smoke situations. Finally, we demonstrate the different limitations that the several proposed models face and this approach's ability to allow experts to intuitively fine-tune the model output or adjust some parameters, e.g., threshold, in order to improve the final segmentation and ensure the expert confidence during the annotation process.

2. Background

2.1. Color Spaces

As humans and computers have different perceptions of color, color spaces allow the latter to represent colors in different geometric representations. There are many different color spaces defined in $\mathbb{R}^c$, where c represents the number of channels and its usually three. The two color spaces used throughout this work are the HSL and YCbCr color spaces. The HSL color space represents colors in terms of hue (H), saturation (S) and lightness (L), which allows an easier interpretation in describing colors. The YCbCr color space is characterized by the luminance (Y) and the chrominances blue (Cb) and red (Cr), and enables the separation between the luminosity (luminance) and the color information (chrominances).

2.2. Superpixels

Superpixel algorithms are considered an image segmentation technique as their objective is to group pixels together based on their characteristics, thus resulting in an image oversegmentation. This technique is quite interesting as superpixels allows to decrease the difficulty in later operations. In addition, it also reduces the processing time since the number of superpixels is much lower than the number of pixels.

2.2.1 Simple Linear Iterative Clustering

The simple linear iterative clustering (SLIC) [1] is a segmentation algorithm based on K-means clustering that creates superpixels within a five-dimensional $[labxy]$ space defined by the L, a, b values of CIELAB color space and the x, y pixel coordinates. In essence, the superpixels are created by grouping pixels that are similar in color and fairly close to each other.

We chose this segmentation algorithm because it is relatively fast to compute, memory efficient and simple to use when compared to other methods [1].

Algorithm description. This approach can be divided into four steps: initialization, assignment, update and post-processing. The algorithm begins by initializing K cluster centers that are spaced S pixels apart, where K is the desired number of superpixels and S the distance between each cluster center. The clusters centers are then moved to the lowest gradient position in a 3x3 area to prevent creating a center on an image boundary and decrease the chances of choosing a pixel with noise.

In the assignment step, each pixel is linked to the closest cluster center in a search region of $2S$x$2S$. In other words, it calculates the distances between a pixel and every cluster center in a $2S$x$2S$ area, as supposed to the entire image, assigning the pixel to the nearest cluster center. In fact, this means that the number of distance calculations is substantially lower when compared to conventional k-means clustering where a pixel is compared to all cluster centers within the image, which results in a faster algorithm. For these reasons, this approach requires a distance measure, D, to calculate the closest cluster center for each pixel.

Considering an image with N pixels, each superpixel size is approximately N/K pixels. In order to create fairly identical superpixels, each cluster center is defined at $S = \sqrt{\frac{N}{K}}$. The euclidean distance can not be used within pixels represented in the 5-D $[labxy]$ as it would be dependent on the number of superpixels. A lower number results in larger superpixels where the distance between their centers would outweigh their color similarity, resulting in superpixels that do not retain the image boundaries. Therefore, [1] use a new distance measure D (equation (3)) to allow a nearly equal weigh between the color similarity and distance proximity.

This distance D combines both the color distance (lab) and the spatial distance (xy), where N_{lab} and N_{xy} are their maximum distances in a cluster, respectively, thus resulting in:

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

$$D' = \sqrt{\left(\frac{d_{lab}}{N_{lab}}\right)^2 + \left(\frac{d_{xy}}{N_{xy}}\right)^2} \tag{1}$$

where $[l_i, a_i, b_i, x_i, y_i]^\top$ is the pixel i represented in the 5-D space and $C_k = [l_k, a_k, b_k, x_k, y_k]^\top$ the cluster center.

Then, the maximum spatial distance in a cluster N_{xy} is considered equal to S, resulting in $N_{xy} = \sqrt{N/K}$ and, to simplify the process as color distances can be very diverse, N_{lab} is considered equal to a constant m, resulting in the following equation:

$$D' = \sqrt{\left(\frac{d_{lab}}{m}\right)^2 + \left(\frac{d_{xy}}{S}\right)^2} \tag{2}$$

Finally, the distance measure D is defined as:

$$D = \sqrt{{d_{lab}}^2 + \left(\frac{d_{xy}}{S}\right)^2 m^2} \tag{3}$$

This variable m allows us to control the compactness of a superpixel, i.e., control the weigh between color similarity and spatial proximity. For a large m value, spatial proximity outweighs color similarity, where the superpixel is more regularly shaped. This variable may take values in the range [1, 20].

After all pixels have been assigned to a cluster center, the update step allocates the mean value of all pixels within a cluster to its center. Finally, the assignment and update steps are repeated iteratively until the residual error E stabilizes, where this error E is associated with the distance between previous cluster centers and new centers. After the clustering procedure, some pixels may not belong to the same superpixel as their cluster center because this algorithm does not enforce connectivity. Therefore, a post-processing step enforces connectivity and reassigns those pixels to the nearest cluster center using a connected components algorithm.

2.3. Fuzzy Theory

Fuzzy logic is a soft-computing approach as it allows to connect the human ability to learn from previous mistakes with complex computational problems using mathematical knowledge. In fact, fuzzy modeling is considered interpretable and transparent when compared to other techniques, as it can often use natural language to describe the inputs, outputs and the relationships between them to build fuzzy models.

2.3.1 Fuzzy logic and Fuzzy Inference

Traditional logic usually states that a variable can only belong exclusively to one class by either taking a true (1) or false value (0). In contrast, fuzzy logic allows variables to have values between 0 and 1 that represents the degree of membership. A fuzzy inference system can be represented by the procedure of taking input variables though a fuzzy model and reaching an output. This process involves memberships functions to define the degrees of membership, If-Then rules to establish relationships between the inputs and outputs, using fuzzy logic operators.

Fuzzy sets. A fuzzy set is set composed of variables having a degree of membership that measures their degree to belonging to a set. As fuzzy sets do not have well-defined boundaries, they are usually described using membership functions [3].

Membership functions. A membership function, as the name implies, is a function that allows to assign membership values (or degree of membership) between 0 and 1 to each point of the fuzzy set [3]. Some common types of membership functions (MF) are: Triangular MF, Trapezoidal MF, Gaussian MF and Sigmoidal MF.

If-Then rules. Fuzzy models use logic relations and "If-Then" rules to establish relationships among the variables defined in the model. The following is the general form of a "If-Then" rule:

If antecedent proposition,
then consequent proposition.

Generally, these rules relate linguistic terms of input variables (antecedent proposition) to linguistic terms of output variables (consequent proposition), where these linguistic terms can be defined by choosing suitable fuzzy sets [3].

Logical operators. In fuzzy logic, the result of the some logical operation is a value between 0 and 1, therefore it is necessary to use functions in order to maintain the truth table of those logical operators. The logical operators used are AND, OR and NOT. In fuzzy logic, the operator AND is represented using the function min, i.e., A AND B results in $min(A, B)$. The operator OR is associated with the function max, where A OR B is equal to $max(A, B)$. Finally, the operator NOT as in NOT A becomes $1 - A$ (1 minus A).

2.3.2 Fuzzy Inference System

There can be different fuzzy inference systems depending on the structure of the consequent proposition. The most common ones are the Mamdani (Linguistic) fuzzy model, where both the antecedent and consequent are fuzzy linguistic terms, and the Takagi-Sugeno (TS) fuzzy model, where the consequent is a polynomial function in respect to the antecedent variables. This work uses the Mamdani-type fuzzy model as it is generally more interpretable and well-suited to human input. A fuzzy inference system is comprised of five parts: fuzzification of the input variables, application of the fuzzy operator, implication from antecedent to the consequent, aggregation of the consequents across the rules and defuzzification. An example called "Movie Ratings" is create to illustrate the different parts of a fuzzy inference system. The overall procedure is outlined in Fig. 1. This example has two inputs, plot and actors, three If-Then rules and one output, rating.

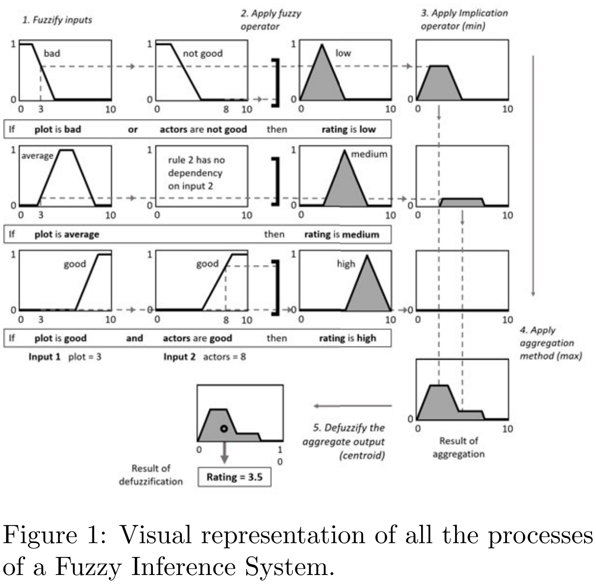


Figure 1: Visual representation of all the processes of a Fuzzy Inference System.

Considering the inputs, plot and actors, with values equal to three and eight, respectively, the fuzzification of the input variables step takes the numerical values and determines their equivalent degree of membership using membership functions. The second step, application of the fuzzy operators, takes the two membership values and selects one, depending on the fuzzy operator. Note that the first and third rule have different fuzzy operators. Subsequently, the third step , applying the implication method, is related to reshaping the output fuzzy set from the previous step. In this case, the implication operator min truncates the fuzzy set. The fourth step is to aggregate all the output fuzzy sets from the implication method, as it is one fuzzy set per rule, using the max operator, which results in a final fuzzy set. Finally, the defuzzification method transforms the output fuzzy set of the aggreation method to a single crisp value, using the centroid method that returns the center of area under the curve.

3. Database

The creation of a meaningful and relevant dataset for wildfire detection and segmentation can be quite the challenge. There are a lot of different real-world scenarios that need to be considered, and creating annotations and ground truth data can usually take long hours. Images with fire and smoke are important but do not encompass all the real-world situations. For a balanced dataset, it is imperative to include samples with firefighters or firetrucks in the field of operations, with sunsets or clouds, as these display similar colors to fire and smoke, and with areas of interest (flame or smoke) at long distances. In terms of annotations, there is a significant value for researchers having further information about a sample. Having prior knowledge about the characteristics of a dataset can help researchers improve fire detection approaches and embed relevant information. Considering how similar sunsets and clouds can be to flame and smoke colors, it can be expected that some algorithms will sometimes misclassify these situations as fire or smoke, respectively. However, having annotations indicating the presence of a sunset or clouds can help the user understand those misclassifications and the reasons behind it, thus allowing to create a more robust algorithm to succeed in these scenarios.

3.1. Corsican Fire Database

The Corsican Fire Database (CFDB) is an online database of wildfire images and sequences that allows the evaluation and comparison of computer vision algorithms related to wildfire detection [9]. This database enables users to upload their own images and image sequences, that can be categorized with different annotations, in order to create an evolving dataset. The dataset contains 500 images in the visible spectrum, 100 pairs of visible and near infrared images, and 5 multi-modal sequences in the visible and near infrared areas.

Fire Image Dataset. The development and testing of the proposed approach was achieved using a smaller dataset (Fig. 2) of the CFDB. This dataset is composed of 207 images from the 500 samples in the visible region. We removed part of the images due to samples having low resolution or noise, decreasing the overall quality of the dataset, or samples having fires created in controlled environments, as these do not represent wildfire scenarios (forests). The performance of the algorithm was fully tested using the 207 images, whilst the development of each model was attained using 50 images for an easier and faster evaluation analysis. Furthermore, the ground truth data considered for this work is the ground truth (binary) images available in the Corsican Fire Database.

Figure 2: Fire Image Dataset. Side-by-side samples of fire images (colored) and respective ground truths (binary), including firefighters and firetrucks, and varying visibility conditions, e.g., day, sunset, night and with smoke.

4. Methodology

Problem Description. The problem tackled by the proposed architecture in terms of data annotation can be under a two-stage approach: *i) segmentation* and *ii) classification*. The first one is relative to the segmentation of an image that is partitioning into several superpixels and, subsequently, regions (see section 4.2.1). This is achieved using the SLIC algorithm and taking advantage of the HSL and YCbCr color features. The second addresses the classification of the different segmented regions by assigning different categories accordingly to their similarity to fire color attributes [8]. In essence, the objectives of this work are twofold: *i)* pixel-wise segmentation of fire and *ii)* description of the fire color category.

Consider a preset color space, $\mathcal{D}$, that can be defined as $\mathcal{D} \in \mathbb{R}^c$, where c represents the number of channels. A sample image, I, encoded in a $\mathcal{D}$ color space domain, is composed of multiples pixels with every pixel x also being defined as vectors in $\mathbb{R}^c$. The first objective addresses the pixel-wise segmentation of the flame in an image by representing every pixel as part of two different classes, $\mathcal{F}$ and $\mathcal{N}$. Let $\mathcal{F}$ represent the set of pixels associated to the fire class, i.e., pixels exhibiting fire colors, and $\mathcal{N}$ the pixels belonging to the non-fire class, in other words, pixels that do not display fire colors [8]. Accordingly, the *ground truth* defines the expert validated data, where both classes are, in this case, binary and mutually exclusive, i.e., $x \in \mathcal{F}$ or $x \in \mathcal{N}$. Furthermore, the second objective tackles the description of fire colors where four different categories, namely red, orange, yellow and other, were modeled in order to create annotations relatively to both the number of fire pixels and the number of pixels belonging to each category, enabling further improvements in the overall dataset. Note that only pixels belonging to the fire class, $x \in \mathcal{F}$, are subsequently part of one of the four categories $\{\mathcal{C}_{\text{red}}, \mathcal{C}_{\text{orange}}, \mathcal{C}_{\text{yellow}}, \mathcal{C}_{\text{other}}\}$. In contrast to the first objective, defining pixels to a color category in a crisp way can be very challenging.

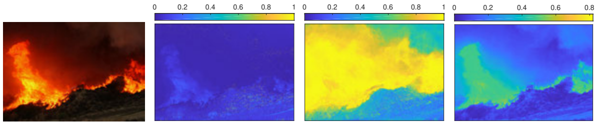
4.1. Seeing Fire across Color Spaces

As the objective is to segment the flame based on color, choosing relevant color spaces can be very helpful when defining parameters that correspond to fire colors, thus, leading to better results. This is a particularly important step for images with similar fire colors in non-fire regions and when there is smoke over the flame, decreasing the perception of the fire colors even for human annotation.

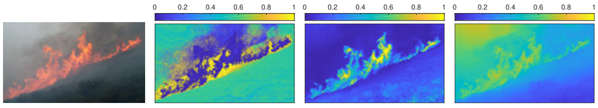
For these reasons, the color spaces used are the HSL and the YCbCr. The HSL color space is easy to use and works well in scenarios with a high contrast between the flame and the background. The saturation and lightness channels are more intuitive in defining fire colors and separating them from dark smoke (high saturation) and clouds (low lightness). Moreover, the hue channel makes it easier to specify the range of colors for the linguistic terms (e.g., red, orange, yellow). However, this color space is more challenging when it comes to images with smoke in the scene or regions with similar colors to fire colors (Fig. 3). In these situations, the YCbCr color space allows for an easier flame segmentation (Fig. 4) because of its ability to separate the colors, but it is less interpretable than the HSL making its development more complex. Color-based features derived from these datasets are employed in the two stages of the proposed approach, namely in the segmentation and classification parts.

4.2. Color-based Superpixel Segmentation

The use of superpixels allows to segment an image by clustering pixels based on their color and proximity. This technique can be very useful since superpixels adhere better to the image boundaries and

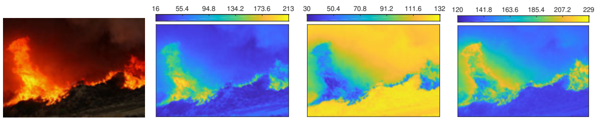


(a) Original Image (b) Hue (c) Saturation (d) Lightness

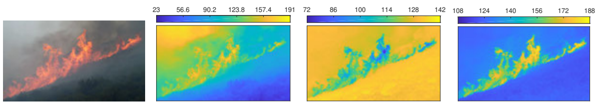


(e) Original Image (f) Hue (g) Saturation (h) Lightness

Figure 3: Fire Features in HSL. Visual display of each HSL color channel for two different images.



(a) Original Image (b) Luminance (c) Chrominance Blue (d) Chrominance Red



(e) Original Image (f) Luminance (g) Chrominance Blue (h) Chrominance Red

Figure 4: Fire Features in YCbCr. Visual display of each YCbCr color channel for two different images.

reduce the complexity of several image processing operations [1], thus decreasing processing times.

Superpixel Algorithm. The superpixels are generated using the simple linear iterative clustering (SLIC) algorithm (see section 2.2.1) that allows the specification of both the desired *number of superpixels*, N_{sp}, and their *compactness*, m.

Number of superpixels, N_{sp} - This number must ensure that the algorithm can achieve a fine-grained segmentation, which drives the quality of the segmentation. Since flame shapes are very irregular and the image data might contain regions of interest captured at long distances, if N_{sp} is too small the image partitioning results in larger superpixels that do not adhere exclusively to the flames. This behavior is illustrated in Fig. 5, where in the lower left corner of the samples presented we can distinctly observe that the superpixels can capture the flames but also aggregate other information nearby. This would inherently degrade the quality of the segmentation, but more importantly, it could prevent an accurate semantic segmentation because a misleading mean color value of the superpixel could result in its misclassification. However, selecting higher values of N_{sp} results in a larger number of increasingly small superpixels, as depicted in Fig. 5, which are harder to merge using the mean color statistics as these capture less context information. The value established by default in our algorithm is 1000 as it is considered an adequate trade-off between these factors.

Compactness, m - Since this parameter controls the shape of the superpixels, it is particularly relevant when segmenting irregular shapes like fire. The influence of varying this parameter can be observed notably in Fig. 5, by comparing the samples on the upper right corner of top and bottom row images. The effect of enforcing a higher compactness (depicted on the bottom row) could result in less fine-grained semantic segmentation for both fire and fire colors. For this reason, the value of m was established as 1, because it is the lowest value possible, making superpixels adhere better to irregular boundaries.

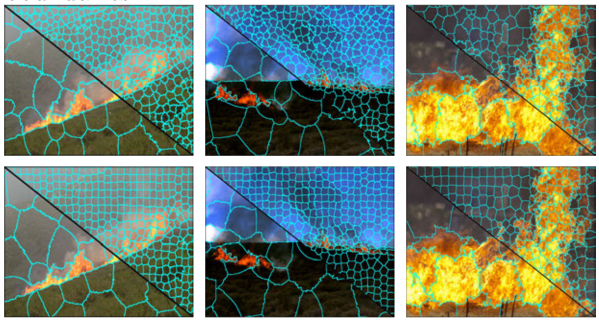


Figure 5: Visual comparison between different N_{sp} and C. The m in the lower left corner of each image is 100 and 2000 in the upper right corner. Images in the top row have a m equal to 1 and images in the bottom row to 20.

Furthermore, the superpixels are defined as vectors in $\mathbb{R}^3$ in both HSL and YCbCr color spaces, where each entry corresponds to the mean color of each channel (H_j^{sp}, S_j^{sp}, L_j^{sp} or Y_j^{sp}, Cb_j^{sp}, Cr_j^{sp}) of the superpixel sp, with $j \in [1, N_{\text{sp}}]$.

4.2.1 Merging superpixels

This method combines neighboring superpixels that register a similar color shade. This is performed using the YCbCr color features as these allow the separation of fire from other instances like smoke. Adjacent superpixels, j and i, are compared based on their mean color ($Y_{j,i}^{\text{sp}}$, $Cb_{j,i}^{\text{sp}}$, $Cr_{j,i}^{\text{sp}}$) and merged if each entry of the pairwise difference is lower or equal to a threshold $\{0.034, 0.1, 0.03\}$, as follows:

$$\begin{aligned} \mid Y_j^{\text{sp}} - Y_i^{\text{sp}} \mid &\leq 0.034 \\ \mid Cb_j^{\text{sp}} - Cb_i^{\text{sp}} \mid &\leq 0.1 \\ \mid Cr_j^{\text{sp}} - Cr_i^{\text{sp}} \mid &\leq 0.03 \end{aligned} \quad (4)$$

Figure 6 demonstrates how adjacent superpixels would merge together if all the conditions are met. Superpixel 16 is compared with superpixels 1 and 27, and, since all the three conditions in 4 are satisfied, the three superpixels are merged together creating region number 1. In the end, this process results in an image similar to the far right one. The threshold values used to merge the superpixels were fine-tuned until the overall flame-only segmentation was fairly good.

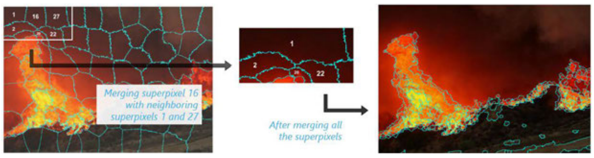


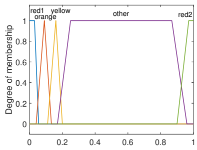
Figure 6: Zoomed area shows how similar superpixels 1, 16 and 27 merge, creating region number 1. Merging process results in a region-defined image.

4.3. Interpretable Rule-based Models

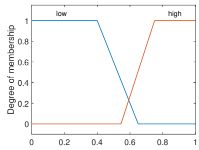
Humans describe colors using linguistic terms like red, green or pink, but color perception might differ from person to person [4]. Likewise, image retrieval for search-based analyses is also based on key categorical terms, namely color. Considering that relevant fire characteristics are related to their color, the annotation of these attributes is fundamental towards creating large-scale datasets with relevant information that can be used in a wide range of fire detection scenarios.

This work proposes interpretable linguistic models designed for fire segmentation and classification of the regions obtained from the previous step. The rule-based architecture is built with Mamdani-type fuzzy inference systems [7] that describe the rules of the knowledge base with linguistic terms. The concept of the rules relies on the association of the linguistic terms between both the modeling of the color-based features and the categorization of colors, with the underlying range of values. Our approach leverages two complementary models, developed for the HSL and YCbCr color spaces and outlined in Table 1. Both models are defined with three inputs, corresponding to the mean color of each channel for every region. The two models output a fire possibility per region, that is leveraged to perform the classification of the merged superpixels and achieve the pixel-wise segmentation of fire in the images. In addition, the HSL model is able to describe fire color categories to perform semantic segmentation of the colors in the image. The proposed architecture may integrate both models in the data annotation approach combining the HSL and YCbCr using a weighted average or maximum operators, to generate a segmentation of fire and the fire color categories [8].

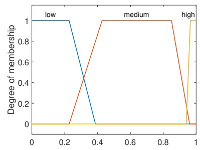
HSL model. The HSL fuzzy model uses linguistic terms to describe levels of hue, saturation and lightness to model the fire possibility as low or high, and the fire color category, which classifies a region to a corresponding color subset $\{\mathcal{C}_{\text{red}}, \mathcal{C}_{\text{orange}}, \mathcal{C}_{\text{yellow}}, \mathcal{C}_{\text{other}}\}$. This model uses nine If-Then rules to join the inputs variables with the output variables, which are defined using membership functions (Fig. 7).



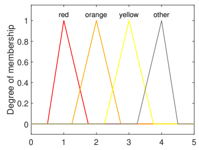
(a) Hue input



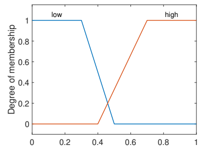
(b) Saturation input



(c) Lightness input



(d) Fire color category output



(e) Fire color possibility output

Figure 7: Visual representation of the membership functions for the inputs and outputs of the HSL model.

YCbCr model. The YCbCr fuzzy model employs terms describing degrees of luminance, chrominance blue and chrominance red to model the fire color possibility in terms of low, medium, medium high and high. Similarly, this model uses fourteen rules to establish relationships between the inputs and outputs variables, which are defined using membership functions (Fig. 8).

5. Results

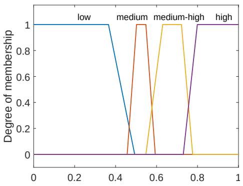
Considering the objectives of the proposed fire annotation approach, the performance analysis focus in evaluating the developed architecture with several baseline techniques. This work presents the assessment and comparison of four different methods to evaluate the proposed architecture. The first two techniques are based solely on the HSL and YCbCr models. Then, the other two combine both models to achieve a multi-purpose model capable of utilizing their qualities and complement each other. The Weighted model is based on a weighted average, considering several weighted values. Finally, the Max Value model, as the name implies, selects the higher output between the HSL and YCbCr models. All the following results are obtained using this last model, since it registered the best overall results.

5.1. Performance metrics

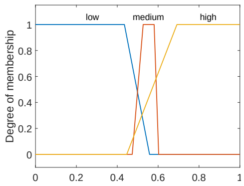
The following metrics allow the assessment of the different models in several important aspects. The

Table 1: Description of the interpretable ruled-based linguistic models in terms of inputs, outputs, membership functions and their respective parameters.

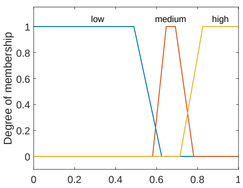
Model	Input			Output		
	Variable	Linguistic terms	Parameters	Variable	Linguistic terms	Output parameters
HSL	Hue	red1, orange, yellow,	[0, 0, 0.03, 0.055]; [0.04, 0.09, 0.133]; [0.11, 0.16, 0.2];	fire possibility	low, high	[0, 0, 0.3, 0.5]; [0.4, 0.7, 1, 1];
		other, red2	[0.17, 0.25, 0.87, 0.96]; [0.9, 0.97, 1, 1];	fire color	red, orange,	[0.5, 1, 1.75]; [1.25, 2, 2.75];
	Saturation	low, high	[0, 0, 0.4, 0.65]; [0.545, 0.75, 1, 1];		yellow, other	[2.25, 3, 3.75]; [3.25, 4, 4.5];
	Lightness	low, medium, high	[0, 0, 0.23, 0.39]; [0.23, 0.427, 0.85, 0.96]; [0.94, 0.965, 1, 1];			
YCbCr	Luminance	low, medium,	[0, 0, 0.365, 0.49]; [0.457, 0.5, 0.548, 0.594];	fire possibility	low, medium,	[0, 0, 0.23, 0.33]; [0.27, 0.35, 0.53, 0.65];
		medium high, high	[0.548, 0.63, 0.72, 0.776]; [0.73, 0.8, 1, 1];		medium high, high	[0.6, 0.65, 0.75, 0.8]; [0.75, 0.83, 1, 1];
	Chrominance Blue	low, medium, high	[0, 0, 0.435, 0.56]; [0.47, 0.527, 0.58, 0.6]; [0.446, 0.69, 1, 1];			
	Chrominance Red	low, medium, high	[0, 0, 0.49, 0.625]; [0.58, 0.647, 0.69, 0.78]; [0.71, 0.826, 1, 1];			



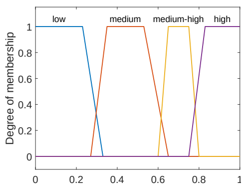
(a) Luminance input



(b) Chrominance Blue input



(c) Chrominance Red input



(d) Fire color possibility output

Figure 8: Visual representation of the membership functions for the inputs and outputs of the YCbCr model.

Table 2: Performance measures for segmentation.

Accuracy	IoU	Precision	Recall
$\frac{TP+TN}{TP+FN+FP+TN}$	$\frac{TP}{TP+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$

metrics used, outlined in Table 2, are: Accuracy, Intersection over Union (IoU), Precision and recall. These are defined based on true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The terms ”true” and ”false” are associated to whether the prediction model was able to correctly classify the positives and negatives, respectively. Considering that each proposed model generates output values for each region that are between 0 and 1, representative of the degree of membership. These are then converted to classes by applying a rounding threshold. To assign each region to the fire and non-fire classes, the threshold δ use is 0.5 and is later on changed to show the capabilities of the proposed approach.

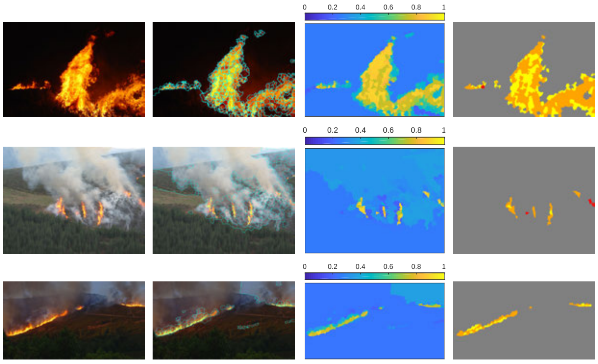
5.2. Results evaluation

The pixel-wise segmentation is defined by classifying each region with the fire color possibility output from the HSL and YCbCr models, which regards the possibility of a region being fire-colored. The description of the fire color category is achieved by assigning different semantic labels to the segmentation regions belonging to the fire class. The three samples depicted in Fig. 9 and Fig. 10 represent different scenarios that the Fire Image Dataset (section 3.1) encompasses and showcase all the intermediate steps of the proposed architecture. In Fig. 9, the first image represents the original sample in the common RGB color space. The second one exhibits the result after merging all the similar superpixels together to achieve a separation between the flame and the surrounding scenery. Subsequently, the fuzzy models generate the classification of all regions according to the fire color possibility output (Fig. 9c), with warm colors representing higher values that will be assigned to the fire class. Moreover, the HSL linguistic model classifies the fire colors according to the color categories (Fig. 9d). As one can see, the approach is capable of identifying all fire colors in the first two samples and assigns the other color label to the remaining colors of the scene. Finally, Fig. 10c outlines the intersection between the fire classification (Fig. 10b) and the color classification (Fig. 9d), which allows to automatically generate fire data annotations close to expert

Table 3: Overall results of all the models.

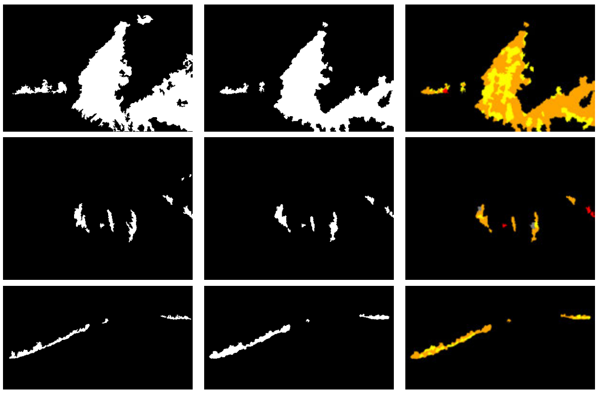
Model		δ	Accuracy	IoU	Precision	Recall
HSL	HSL model	0.5	93.39	62.49	92.50	65.77
YCbCr	YCbCr model	0.5	91.58	56.68	93.50	58.97
Weighted	0.4 HSL + 0.6 YCbCr	0.5	93.16	61.52	94.24	63.83
	0.3 HSL + 0.7 YCbCr	0.5	93.01	61.00	94.27	63.19
	0.2 HSL + 0.8 YCbCr	0.5	92.81	60.13	94.62	62.32
Max Value	max(HSL,YCbCr)	0.5	93.47	66.51	**92.74**	71.23
		0.4	**94.04**	**73.53**	88.57	**83.14**

annotations.



(a) Original Image (b) Final Merging (c) Fire Possibility (d) Fire Colors

Figure 9: Samples representing real-world scenarios to exemplify the results at each stage of the proposed approach.



(a) Ground truth (b) Binary Output $\delta = 0.5$ (c) Output

Figure 10: Display of the binary output image and output generated by the Max Value model.

The overall results of all the proposed methods are outlined in Table 3. Comparing to the other models, the Max Value model registers the best results in accuracy, IoU and Recall in both thresholds. The precision metric decreases for a threshold of 0.4, meaning that the number of false positives increased.

Limitations. In a relevant and complete fire image database, it is imperative to include scenery or objects that resemble fire colors as these represent real-world scenarios. Although the proposed architecture registers significant results and allows to create rich annotations, the approach still faces some limitations especially in situations with similar fire colors in non-fire regions, e.g., sunsets, firefighters or firetrucks, as outlined in Fig. 11. Such limitations can be improved by either some type of expert input or with algorithms able to identify this kind of scenes and objects.

(a) Original Image (b) Ground truth (c) Output $\delta = 0.5$

Figure 11: Samples where the proposed approaches registers some limitations in the classification.

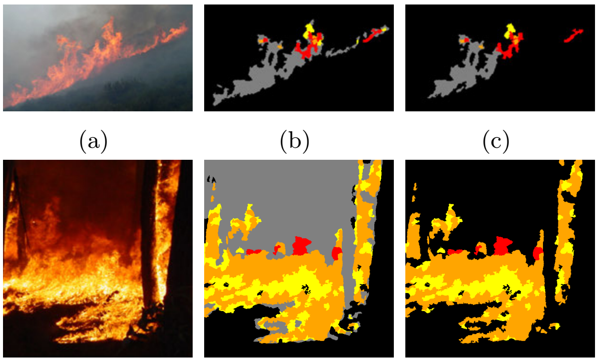
5.3. Adaptability of the proposed approach

The proposed approach enables easy adjustments of some key parameters associated with the intermediate steps from the architecture. Several parameters are the number of superpixels, which can affect the overall result of merging all the superpixels, the threshold, allowing regions with a lower fire possibility to be included, and the fuzzy model parameters that represent fire colors in each color space.

Threshold. The threshold value can change the overall segmentation. For images with high levels of smoke, this parameter usually enhances the final semantic segmentation (Fig. 12b). However, since these images usually register an overall high saturation, the proposed approach classifies most of the improvements to the other class (grey color). Moreover, the same observations are registered in regions similar to fire colors, where using a threshold of 0.4 results in a worse segmented image (Fig. 12e).

6. Conclusions

The two objectives outlined for this work were successfully achieved. Regarding the first objective, the pixel-wise segmentation of the fire can be obtained by specifying a threshold to the image containing the fire color possibilities of each region. The second objective, description of the fire color



(a) (b) (c)

(d) Original Image (e) Output $\delta = 0.4$ (f) Output $\delta = 0.5$

Figure 12: Samples displaying how the threshold affects the final semantic segmentation.

category, can then be generated by intersecting the pixel-wise segmentation of the fire with the output fire color category of the HSL fuzzy model. The proposed method, the Max Value model, gives the best overall performance of all the baseline approaches. It can cope with a series of different scenarios as it was previously demonstrated. Where previously baseline models had some step backs, the Max Value model was able to overcome those by combining the two fuzzy models. Considering that the proposed approach detects fire colors, it is to be expected that in certain situations it would detect similar fire colors in non-fire objects. However, the interpretability and adaptability of the proposed architecture comes into play, allowing the expert to adjust some parameters to validate and generate reliable ground truth data and respective annotations. The most common parameter that can usually be changed in other fire detection techniques is the classification threshold. As one could see in section 5.3, changing the threshold value resulted in better overall results. This parameter is particularly interesting in regions where the fire has smoke over it, since these areas usually output a lower possibility of being fire-colored as the smoke decreases the pixel's saturation. Therefore, decreasing the threshold value enables these regions to be considered. Another parameter, particular to the proposed approach, is the number of superpixels. As it was demonstrated, the number of superpixels can sometimes increase or decrease the overall performance. Finally, the fuzzy model parameters could also be changed in the situation that the previous ones mentioned would not increase the results. However, these are not very straighforward like the threshold or number of superpixels, since the fuzzy model parameters are related to how the HSL and YCbCr color spaces perceive fire colors.

In conclusion, the proposed architecture contributes ot fire detection research, as even though it is not a fire detection approach, it allows to propose automatic fire data annotations for expert validation towards the creation of high-quality large-scale datasets.

Acknowledgements

The author would like to thank professor Alexandra Moutinho, engineer Maria João Sousa and his family and friends.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 2002.

[3] R. Babuška. *Fuzzy Modeling for Control.* Springer, 1998.

[4] J. Chamorro-Martínez and J. M. Keller. Granular modelling of fuzzy color categories. *IEEE Transactions on Fuzzy Systems*, 2019.

[5] A. Chenebert, T. P. Breckon, and A. Gaszczak. A non-temporal texture driven approach to real-time fire detection. In *18th IEEE International Conference on Image Processing*, 2011.

[6] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

[7] E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, 1977.

[8] P. Messias, M. J. Sousa, and A. Moutinho. Color-based superpixel semantic segmentation for fire data annotation. pages 1–8, 2021. Submitted to CVPR on 16th November 2020.

[9] T. Toulouse, L. Rossi, A. Campana, T. Celik, and M. A. Akhloufi. Computer vision for wildfire research: An evolving image dataset for processing and analysis. *Fire Safety Journal*, 2017.

[10] T. Çelik and H. Demirel. Fire detection in video sequences using a generic color model. *Fire Safety Journal*, 44:147–158, 2009.