VOL. 19, NO. 3, FEBRUARY 2024 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

BM-UNET: LIGHTWEIGHT MULTICLASS-FIRE SEGMENTATION NETWORK AND ROBUST METHODS OF AIMING THE FIRE BASED ON THE STATE-MACHINE OF MEAN SEGMENTATION MASK AND CONTOUR CENTER-MASS

Vladimir Bochkov[1], Liliya Kataeva[2] and Evgeniy Linev[1]
[1]Alexeev State Technical University, Russia
[2]Alexeev State Technical University, Russia
E-Mail: vladimir2612@bk.ru

ABSTRACT

The article presents the BM-UNet neural network architecture optimized for mobile devices. The article focuses on the possibility of real-time operation, takes into account various techniques for lightening the model, and provides a comparison with UNet-half. An improved model in terms of performance/accuracy ratio is applied in the algorithm of frame-by-frame segmentation of the flame on video, the result of which is averaged, and the optimal extinguishing point is found. For the latter, an approach to organizing a finite state machine is presented for switching between time-averaging windows for the possibility of timely response and minimization of mechanical losses of targeting. Combined, the method of frame-by-frame segmentation of flame contours and the algorithm for finding the center-mass point can be used in robotic flame extinguishing systems.

Keywords: image fire segmentation, BM-UNet, state machine.

Manuscript Received 13 November 2023; Revised 2 March 2024; Published 30 March 2024

1. INTRODUCTION

Fires are the most frequent natural and man-made accidents. The creation of means of robotic monitoring of the environment and timely prevention of fires is an important and urgent task today. Modern tools use expensive thermal imagers to detect flames. The disadvantage of the high price of the end device leads to the limitation of serial production of robots. A cheaper alternative is to use a standard video stream and advanced neural networks performing flame configuration extraction and targeting tasks. It is important to note that neural network algorithms contain a large number of matrix calculations. The development of modern mobile devices and the use of mobile GPU/NPU chips make it possible to carry out such calculations in real-time. This makes it possible to focus computing on compact mobile devices as the core elements of robotic tools, which is much better for environmental protection than running algorithms in cloud-based systems that are not available everywhere in forest areas.

The article discusses the process of optimization of the existing method of multi-class flame segmentation based on the neural network model wUUNet [1], presented in Figure-1a. The main nuance of the application of this model is its high computational cost since it consists of two UNet blocks that perform segmentation of the binary flame signal (there is / no flame in a pixel) and a clarifying multi-class segmentation by color: red, orange, yellow flame. Multi-class segmentation is necessary to extract the hottest areas of the flame for optimal extinguishing [2-4], expressed through the color spectrum of combustion. The article presents an optimized version of the wUUNet model, called BM-UNet, shown in Figure-1b, in which the clarifying multi-class part of the neural network is represented by a single convolutional layer instead of the UNet model. The model undergoes further optimizations, in particular, a multiple reduction in the dimensionality of convolutional layers, and the use of addition operations instead of concatenation for the pass links of the UNet model. The comparison is made with lightweight UNet-Half nets [5].

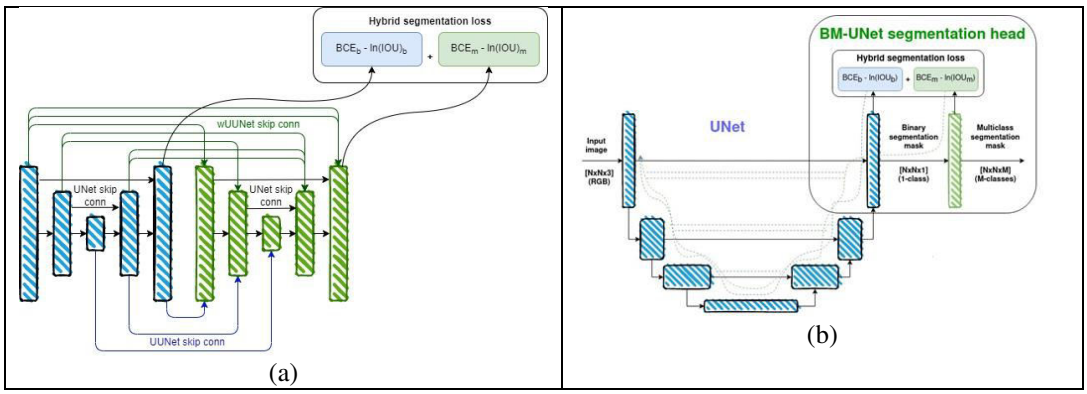


Figure-1. The wUUNet architecture and its optimal analogue, are presented in the article BM-UNet. wUUNet consists of combinatorially coupled blocks of binary (highlighted in blue) and multiclass (green) UNet, whereas in the BM-UNet model, one additional layer to the binary UNet model is allocated behind the multiclass part of the model.

The result of the segmentation neural network is fed into an algorithm that searches for the extinguishing point as the center of mass along the largest contour of the average signal mask. For optimal operation of a robot that extinguishes by applying a jet of water to the source of flame, the targeting algorithm must minimize the detection of the target, which means giving the aiming point without delay, minimizing mechanical losses for aiming, and that's mean stabilize the targeting point and respond to changes in the scene (movement of the video camera). These conditions are met by the finite state machine method of switching the state between the windows of averaging the flame signal over time.

2. MATERIALS AND METHODS

The problem of finding the contours of the flame in the video has been solved in the scientific community in various ways. In [6] an attempt was made to extract super-pixel areas (by searching for zones of similar color) and then classify them. A visualization of this algorithm is shown in Figure-2a. Papers [7-10] are devoted to the search for framing rectangular areas of flame in video, i.e. the standard problem of detecting objects in an image is shown in Figure-2b. It is important to note that the flame object in the image has a non-convex contour, so the approximation of the contour with a framing rectangle is a rough approximation.

The main type of task in the field of obtaining flame contours in an image is the segmentation problem, in which the image is classified pixel-by-pixel for flame detection.

There are various modern neural network algorithms for segmenting objects in an image. The main classes of methods are UNet [11], and Deeplab [12, 13]. Advanced models in the field of segmentation in terms of accuracy are UNet++ [14], and UNet-FPN [15]. Among the lightweight variants is the UNet-half [1] approach to reducing the decoded level, which is also the basis of the BM-UNet model presented in the article. All of the above models are applied in comparison to the problem of multiclass flame segmentation.

Papers [16-17] deal with the problems of binary segmentation of a flame, in which there is a binary contour of a fire without specifying its structure. The paper [2] is devoted to multi-class segmentation, in which the models of double UNet - UUNet and wUUNet - are presented. The result of this algorithm is shown in Figure-2c.

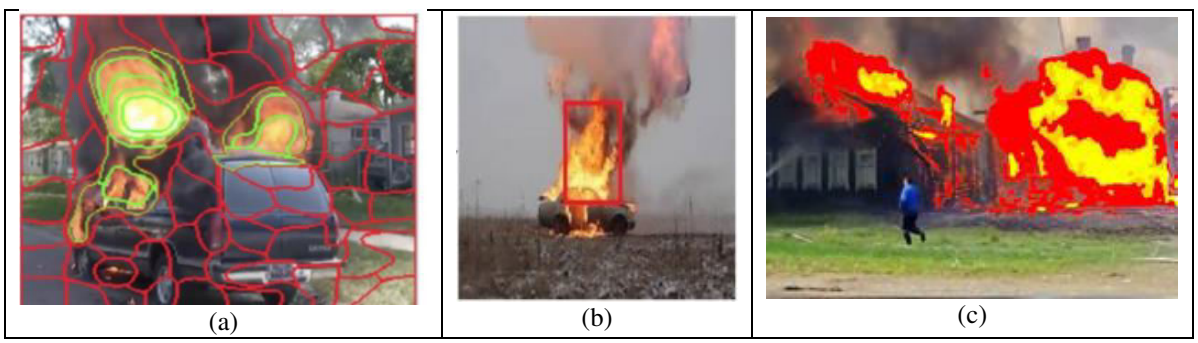


Figure-2. Types of machine vision tasks for solving flame detection problems in an image: a) extraction and classification of a super-pixel region, b) flame detection and localization by a framing rectangular area, c) pixel-by-pixel multi-class segmentation of flame contours in the video.

The BM-UNet architecture is a continuation of the idea of binary-multiclass flame segmentation by color with a bias towards computational optimization. Also in the paper [2] is a dataset consisting of red, orange, and yellow flame outlines marked by classes in images of real fires in forest and urban areas. It is used to train and compare the models presented in the article. For the training procedure of all models, the Adam optimization algorithm [18] and the method of stochastic gradient descent with restarts (SGDR) [19] are used every 250 epochs. The total number of epochs is 2000.

2.1 Mathematical Formulation of the Problem of Multiclass Flame Segmentation in the Image

Image segmentation is a classification task for each pixel of the image, forming a response matrix with the exact contours of the desired objects:

$$A(I) = S \tag{1}$$

$$S = a_{i,j} \mid a_{i,j} \in [0, n] \tag{2}$$

$$DIM\,(I) = DIM\,(S) \tag{3}$$

Algorithm A is selected for the image matrix I, which calculates the matrix characteristic S, each element of which is equal to either 0 (no object) or a positive value denoting the number of the object class in the task. The task searches for three classes of objects: red (1), orange (2), and yellow flame (3).

The signal of the matrix S is decomposed into a basis-vector space of dimension n, in which 1 is placed in the component corresponding to the flame class (0 in all others):

$$S = a_{i,j,k} \mid a_{i,j} = \sum_{k-1}^{n} a_{i,j,k} \overline{e_k} \tag{4}$$

To check the accuracy, the Jaccard metric [20] and its average variant for all n classes in the problem are used:

$$J_n(p_n, y_n) = \frac{P_n \cap Y_n}{P_n \cup Y_n} = \frac{|P_n \cap Y_n|}{|P_n| + |Y_n| - |P_n \cap Y_n|} = \frac{\sum_{i,j} p_n y_n(i,j)}{\sum_{i,j} p_n(i,j) + \sum_{i,j} y_n(i,j) - \sum_{i,j} p_n y_n(i,j)} \tag{5}$$

$$J(p, y) = \sum_{i=1}^{n} J_n(p_n, y_n) \Big/ n \tag{6}$$

Here $p_n(i, j)$, is the probability of predicting an object n in the pixel [i, j], $y_n(i, j)$ - is the labeled values of the actual object in the pixel.

The best models are further optimized using the TF-Lite framework for better performance on mobile devices. The article presents optimization techniques such as:

a) Use of sparse bundles [21] (Figure-3a)
b) Constant value and multiple decrease in the dimensionality of convolutional layers (Figure-3b)
c) Replacement of the concatenation operations of the model states in the throughput links of the UNet model with the addition operation (Figure-3c)

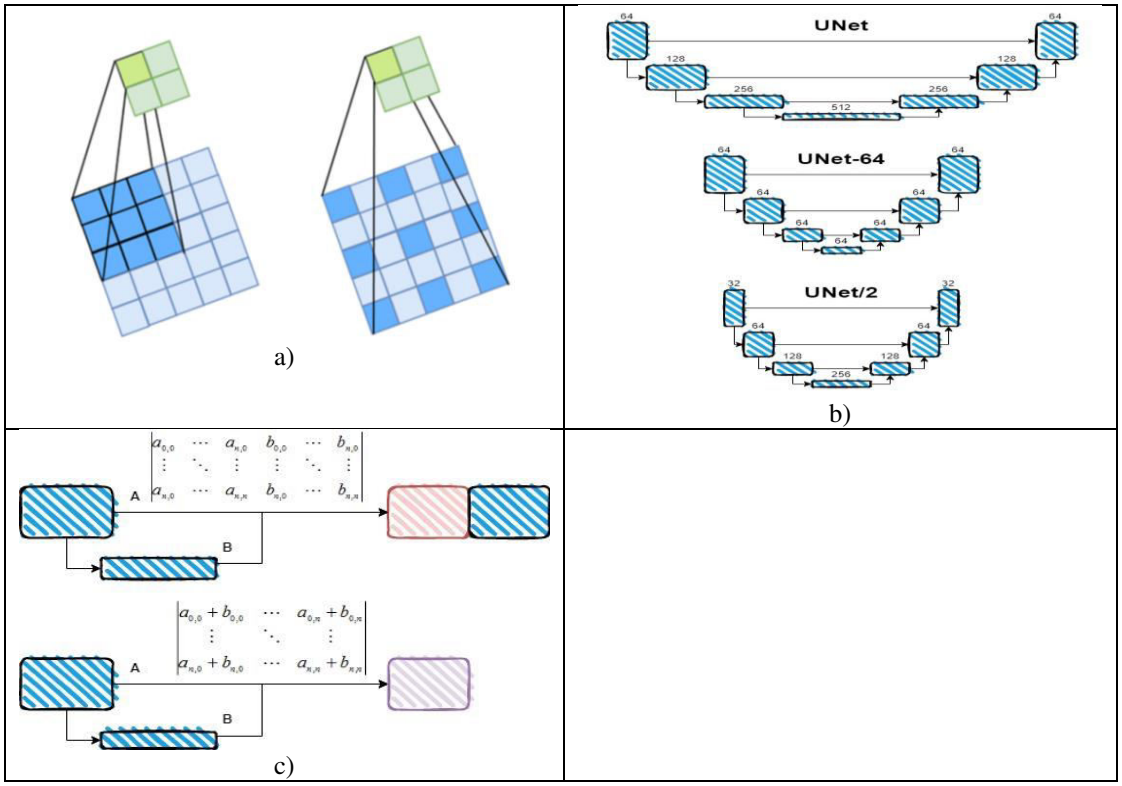


Figure-3. Ways to optimize the computational complexity of the UNet neural network model: a) the use of sparse convolutions, b) reducing the dimensionality of convolutional layers, c) replacing the concatenation operation with the addition of matrices in the neural network throughputs.

Graphs of algorithm performance (frame calculation time) are demonstrated. The target is real-time performance at 25 frames per second (FPS) of video, which equates to 40ms per frame.

The performance-optimal model is used to produce the outlines of the flame in the video stream, that is, the sequence of images that form an animated fragment. The S matrix as the output of the segmentation algorithm is used to compute the point of aiming at the source of the flame. In addition to the extinguishing point p, it is advisable to calculate the contour of the source of fire C in which it is located.

To assess the quality of the selection of extinguishing points, average metrics are used in the video:

$$\bar{q} = \sum_{t=1}^{n} q(p(t)) \Big/ n \tag{7}$$

The article presents the following metrics:

1. Presence of a flame signal in the frame:

$$q_1 = \begin{cases} 1, \exists s' \neq 0 \in S' \\ 0, \overline{\exists s' \neq 0} \in S' \end{cases} \tag{8}$$

2. Entry of the extinguishing point p into the target flame circuit C:

$$q_2 = \begin{cases} 1, p \in C \\ 0, p \notin C \end{cases} \tag{9}$$

3. Flame class at the specified point, as an element of the matrix S':

$$q_3 = s'_{p=i,j} \in S' \tag{10}$$

4. Displacement difference relative to the point in the previous frame:

$$q_4 = \left\| p_t - p_{t-1} \right\|, q_4 \big|_{t=0} = 0 \tag{11}$$

The last metric is responsible for the stability of the extinguishing point *p*, relative to the frames of the video fragment. A high-quality point search algorithm corresponds to the maximum value of the metrics q_1, q_2, q_3, and the minimum q_4.

From the response matrix of the flame signal S in the image, it is possible to form the contours of the fire using a recursive algorithm of 8-connectivity [22] of identical neighboring pixels around the current one. As a result of the algorithm's operation, a set of closed contours is obtained $\{C_k\}, C = \{[x, y]\}$, each of which consists of many points on the boundary that uniquely describe the geometric representation of the contour.

On the contour C, the calculation of moments of order 0 and 1 is applied:

$$\mu_{ij}(C) = \sum_{x,y} x^i y^j \quad (12)$$

Based on the moments, you can determine the center of mass of the contour:

$$\bar{x} = \frac{\mu_{10}}{\mu_{00}}, \bar{y} = \frac{\mu_{01}}{\mu_{00}} \quad (13)$$

The area of the contour is its zero-order moment:

$$S(C) = \mu_{00} \quad (14)$$

Thus, to find the optimal point of fire extinguishing, the contour of the maximum burning area is searched, and its center of mass is calculated:

$$p(C) = \{\bar{x}, \bar{y}\}, C = \max_k [\mu_{00}(C_k)] \quad (15)$$

A stable position of the target designation point is necessary for correct aiming at the source of the fire without additional changes in the mechanical movement of the muzzle of the water cannon and the associated energy costs. Since the flame is an object with a pronounced contour change in the video, it makes sense to consider the contour calculations over the averaged exponential moving average output of the S signal:

$$EMA_s(t) = (1-\lambda) \cdot EMA_s(t-1) + \lambda \cdot S(t) \quad (16)$$

where λ - is the size of the averaging window depending on the characteristic of the number of frames per second and the specified averaging time τ in seconds:

$$\lambda = \frac{1}{FPS \cdot \tau} \quad (17)$$

On the averaged signal, a binarization (clipping) operation is performed according to the threshold $\beta = 0.9$:

$$\overline{S(t)} = \begin{cases} 1, EMA_s(t) > \beta \\ 0, EMA_s(t) \le \beta \end{cases} \quad (18)$$

Then calculations are made over $\overline{S(t)}$ (12-15) to find the contour of the fire and the extinguishing point. The use of averaging reduces the fluctuation of the extinguishing point and reduces the value of the q_4 metric depending on the size of the averaging window T.

However, by increasing the averaging window T, the activation time (finding the extinguishing point) of the signal also increases.

To eliminate this drawback, the finite state machine algorithm is used to switch the state between averaging schemes with different window values:

$$\{\tau_i\} = \{0.1c, 0.25c, 0.5c, 1c, 2.5c, 5c\} \quad (19)$$

$$\bar{S} = \begin{cases} \bar{S}(\tau_{i+1}) \mid EMA_s(\tau_{i+1}) > \beta \\ \bar{S}(\tau_i) \mid EMA_s(\tau_i) > \beta \\ \bar{S}(\tau_{i-1}) \mid EMA_s(\tau_i) < \beta \end{cases} \quad (20)$$

The state machine has the effect of switching to the next state of the average signal matrix (with a large window) in case the latter is activated, and resetting to a smaller averaging window in case of deactivation of the current one (if the source of fire disappears or the scene changes as a result of camera movement). A detailed comparison of the results of the methods is described in the next chapter.

3. RESULTS

3.1 Optimization of the Execution Time of Neural Network Segmentation Models

The section presents a comparison of modern methods for reducing the computational complexity of neural network models and their application to the problem of multi-class segmentation on a modern smartphone based on the Qualcomm Snapdragon 855 processor.

The first procedure for reducing complexity is the use of dilated convolutions [21], schematically shown in Fgure-3 a. This type of matrix layer produces a matrix that is 2k smaller than standard convolutional kernels. The model changes in structure and is subject to repeated training, the graphs of which are shown in Figure-4. The sparse convolution model (orange graph) shows lower accuracy than the standard model, which is also shown in Table-4.

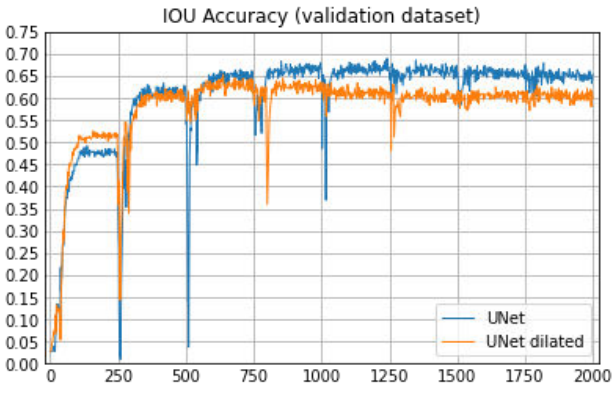
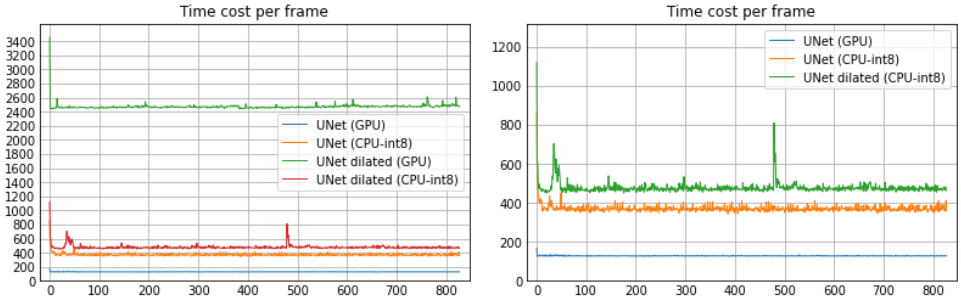


Figure-4. Comparison of convergence of training schedules of UNet models using standard and sparse convolutions.

Figure-5 shows the processing time of the frame by the model with standard and sparse convolutions. The sparse convolution model runs much slower on a mobile GPU than the standard ones. This is because the memory model of massively parallel systems provides block computing without sparse.



a) b)

Figure-5. Performance comparison of models using ordinary and sparse convolutions.

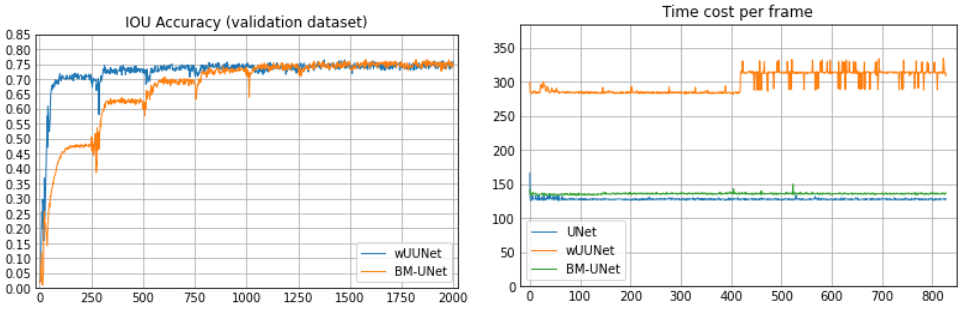
Table-1 shows the accuracy and speed of the models. Sparse convolutions are not suitable for optimizing the performance of segmentation neural networks of the UNet class.

Table-1. Comparison of the accuracy of segmentation of UNet models.

Model	Accuracy IOU	GPU time	CPU time
UNet	74.71%	128 ms	370 ms
UNet-dilated	70.46%	2576 ms	477 ms

In [2] it was found that the wUUNet neural network model shows the best results in terms of segmentation accuracy. This model consists of two UNet models and, accordingly, has a twofold increase in the number of operations, as indicated in Figure-1a.

For the architecture, the optimal variant of binary-multiclass segmentation BM-UNet is presented, the clarifying multiclass part of which is reduced from the whole UNet model to one convolutional layer, which is demonstrated in Figure-1b. A comparison of models in the training procedure is shown in Figure-6a, and the execution time in the inference mode is in Figure-6b.



a) b)

Figure-6. Comparison of accuracy (a) and performance (b) of UNet, wUUNet, and BM-UNet.

The wUUNet model achieves high accuracy faster, but BM-UNet also converges to 75% accuracy. It is worth noting that with similar characteristics in terms of accuracy in the later epochs of training, the BM-UNet model works significantly faster than wUUNet and slightly slower (but much more accurately) than UNet, which is also demonstrated in Table-2.

Table-2. Comparison of the segmentation accuracy of the UNet, BM-UNet, and wUUNet models.

Model	IOU (Multi-Class)	IOU (Binary)	Uptime
UNet	74.71%	84.69%	128 ms
wUUNet	76.14%	88.53%	299 ms
BM-UNet	75.83%	88%	136 ms

The paper [1] presents the UNet-half architecture (Figure-7). This model applies a constant number of convolutional layer filters, applies matrix addition instead of concatenation, and reduces the decoding procedure to a single layer by scaling from all encoding levels to the size of the input image. Applying these features to the BM-UNet model, we get the BM-UNet-half variant.

Variants of models with a constant number of layers BM-UNet-64 and replacement of concatenation with the addition operation BM-UNet-64+ are taken into account. Training graphs of all models in comparison with the standard BM-UNet are shown in Figure-8. The convergence graph of variant 64+ is almost the same as that of the original model, which means that this variant is highly efficient and, in particular, the replacement of matrix concatenation operations with addition.

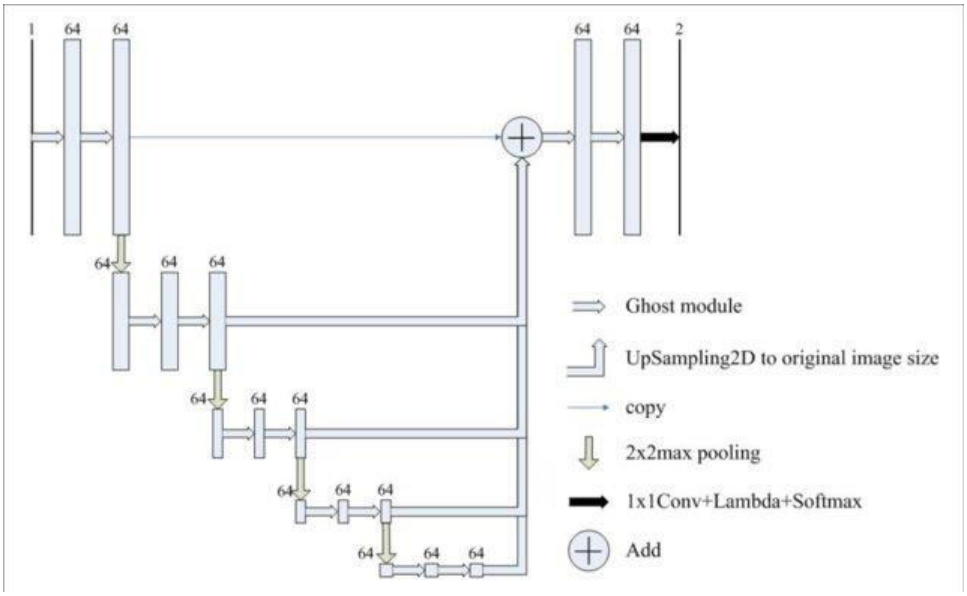


Figure-7. The architecture of the neural network UNet-half.

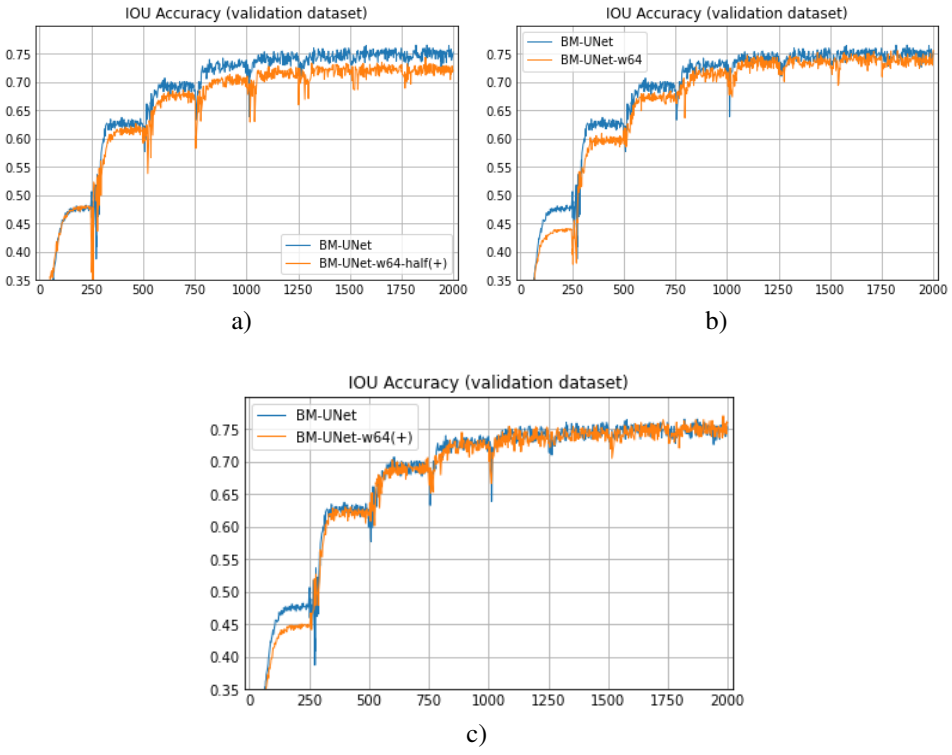
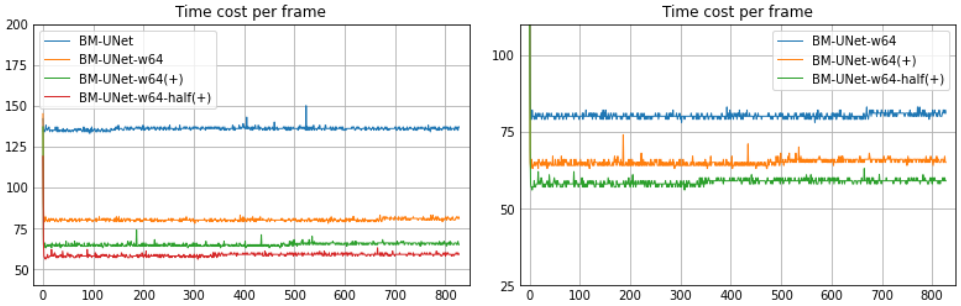


Figure-8. Comparison of the convergence of BM-UNet training schedules with optimization options.

Figure-9 illustrates a significant reduction in frame processing time by applying a constant small number of convolutions in the W64 model layers. The half model is the fastest, but the w64+ model is not significantly inferior to the leader and is much faster than the model using w64 concatenation.



a) b)

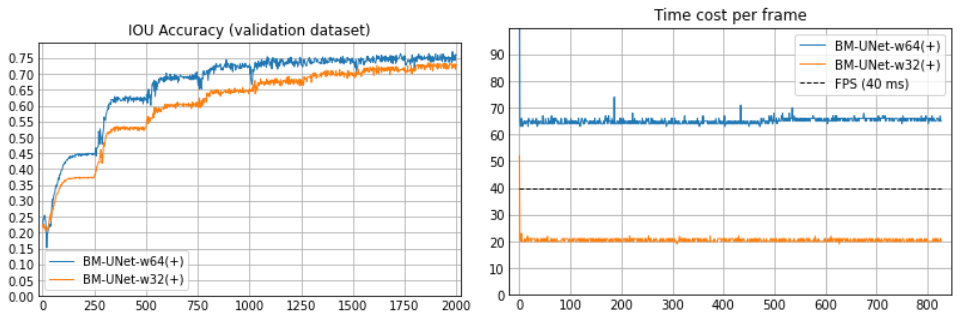
Figure-9. Comparison of the performance of BM-UNet methods and its optimized analogues.

In Table-3 accuracy and speed characteristics are indicated. The fastest half model is the least accurate, while the w64+ model exhibits close accuracy characteristics to the original BM-UNet model, significantly outperforming it in speed.

Table-3. Comparison of the segmentation accuracy of BM-UNet models.

Model	IOU (Multi-class)	IOU (Binary)	Uptime
BM-UNet	75.83%	88%	136 ms
BM-UNet-half	71.41%	83.97%	58 ms
BM-UNet-64	75.14%	88.50%	80 ms
BM-UNet-64+	75.82%	86.50%	65 ms

An additional step in optimizing the time of execution of the BM-UNet-64+ model is to halve the number of convolution filters in the layers, obtaining the BM-UNet-32+ model. Comparison graphs of the BM-UNet-64(32)+ models are shown in Figure-10. The accuracy of the 64+ is better than that of the 32+, but at the end of the workout, the gap is significantly reduced by using the SGDR gradient descent restart method [19]. Figure-10b shows a three-fold difference in image processing time between the models. The dotted line at 40 ms in Figure-15b shows the frame lifetime for a 25 FPS refresh rate frequently used in machine vision applications. Values below the dotted line correspond to the execution of the algorithm in real-time, which the smartphone copes with by segmenting the BM-UNet-32+ model. The results of the comparison of the models are shown in Table-4, which confirms the facts.



a) b)

Figure-10. Comparison of training graphs (a) and execution time (b) of BM-UNet methods of dimension 64 and 32-layer convolutions.

Table-4. Comparison of segmentation accuracy of BM-UNet-64(32)+ models.

Model	IOU (Multi-class)	IOU (Binary)	Uptime
BM-UNet-64+	75.82%	86.50%	65 ms
BM-UNet-32+	73.26%	86.61%	20.5 ms

Thus, within the framework of the procedure for optimizing neural network segmentation models, a recommendation for the use of the BM-UNet-32+ model in smartphones is indicated.

3.2 Determining the Extinguishing Point Based on the Segmentation of the Flame in the Video

The output of the segmentation algorithm above the frame (the input image matrix) is a matrix of identical size, in which each element corresponds to the value of the presence or absence of a flame in it. To find the fire extinguishing point, the method of extracting the signal contours and the formula for finding the center of mass above the contour (12-15) are used. This approach is used both for the binary mask of the presence/absence of fire B and for the multi-class mask M, which specifies a specific class.

For a binary signal, the search for the contour of the maximum size and its center is used. In the case of a multi-class signal, the prioritization of the choice of the flame contour on which the extinguishing point is located according to the flame class from red (lowest priority) to yellow (highest priority) is also taken into account, since the brighter the flame, the higher the temperature index it has, and therefore the source of energy.

In Chapter 2, the q_1,.., *and* q_4 quality metrics for the selection of the extinguishing point are indicated, and for these search schemes, the indicators are indicated in Table-5.

Table-5. Comparison of quenching point selection quality metrics.

Metric	Multi-class. Max. contour	Binary. Max. contour
Presence of a quenching point (q_1)	99.98%	99.98%
Getting the extinguishing point into the fire contour (q_2)	99.00%	91.93%
Flame class at the extinguishing point (q_3)	2.65	1.64
The offset of the current point relative to the previous one (position stability, q_4)	17.31px	7.98px

There is a two-fold improvement in the q_4 metric in binary mode. This is due to two factors:

Choosing a contour according to the flame class in a "greedy" way leads to the finding of "volatile" areas of yellow flame, and not the hearth, as shown in Figure-11.

Changing the selection of the extinguishing contour too often, due to the short time for the volatile area of yellow or orange flame to appear on the frame. The binary scheme takes into account the entire fire contour and the center of mass of the contour indicates the constant zone of the fire. This effect is also demonstrated in Figure-11.

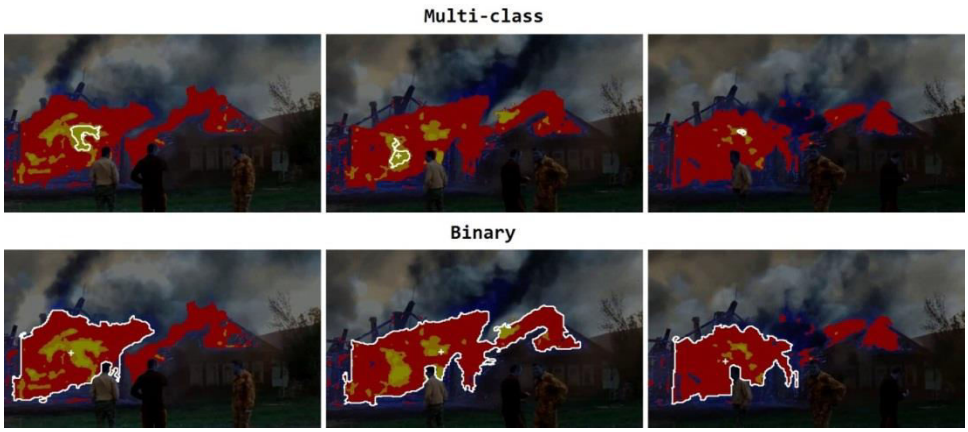


Figure-11. Visualization of contour selection (highlighted in white) and extinguishing point (white cross) in binary and multi-class approaches

A hybrid binary-multi-class scheme for selecting the extinguishing point eliminates the "jump" effect. The maximum fire contour is selected by a binary signal for the presence of flames. The search for flames in the selected binary path is carried out by a multiclass mask with priority toward the yellow and orange flame zones. For this purpose, formula (12) is used in its generalized description with the introduction of the physical concept of density ρ:

$$\mu_{ij}(C) = \sum_{x,y} \rho(x,y) x^i y^j \quad (21)$$

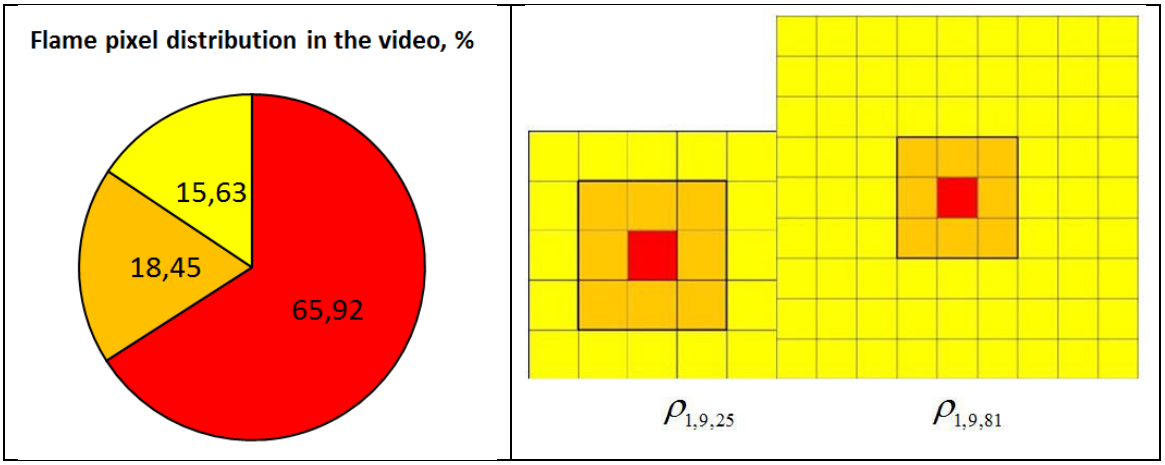
In the first case, ρ is the scalar value of the multi-class signal matrix M in the region of the circuit C:

$$\rho(x,y) = m_{y,x} \in M \cap C \mid m_{y,x} = \{0,1,2,3\} \quad (22)$$

Given the uneven distribution of the flame pixels of the different colors indicated in Figure-12. a and the importance of determining the hottest (yellow) section of the flame as the target of the gun, the density of orange and yellow flames should be much higher than that of red.

$$\rho_{r,o,y}(x,y) = \begin{cases} r \mid m_{y,x} = 1 \\ o \mid m_{y,x} = 2 \\ y \mid m_{y,x} = 3 \end{cases} \quad (23)$$

To define the functions $\rho_{1,9,81}$, a geometric representation of the pixel matrix was used, shown in Figure-12b. On the left for a, the density corresponding to the germination area of the pixel boundaries is visualized (1x1 for red, 3x3 for orange, 5x5 for yellow flame). On the right for b is the density according to germination for orange flame, and squaring for yellow. Thus, the displacement of the geometric center of mass of the binary contour towards the zone of a hotter flame is achieved, as shown in Figure-13.



a) b)

Figure-12. The average distribution of flame pixels on Gorenje video (a) and visualization of the signal density function p for pixels of red, yellow, and orange flames.

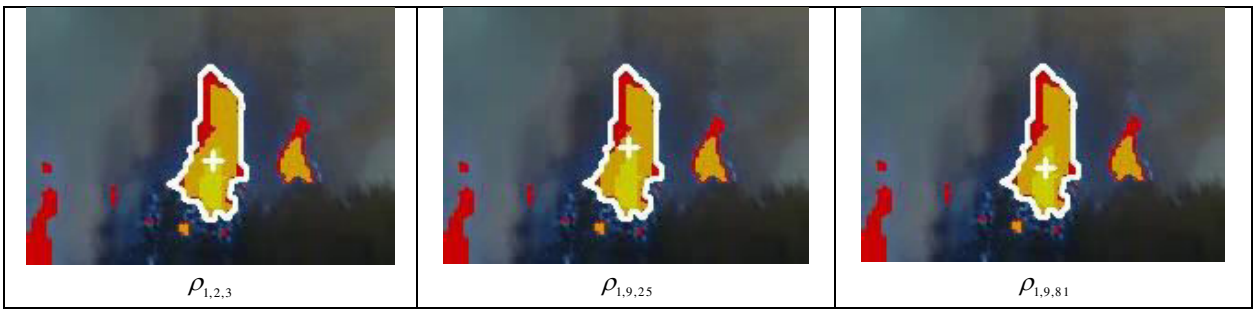


Figure-13. Visualization of the selection of the extinguishing loop point for the hybrid scheme (highlighting the common flame contour and finding the center of mass point).

A comparison of binary, multi-class, and hybrid schemes is presented in Table-6. The multi-class variant shows the best indicators of the center point hitting the flame circuit (q_2) and the average flame class at the center of the mass point (q_3). However, the point shift from frame to frame (q_4) is much higher than that of the rest of the circuits, which is unacceptable for flame guidance systems.

A hybrid scheme with low densities works similarly to a binary scheme, and a hybrid scheme with

high densities of orange and yellow flame works similarly to the values of a multi-class scheme, except for the value of the extinguishing point offset metric (q_4). For all hybrid schemes, it is significantly smaller, which allows us to eliminate the main drawback of the multi-class scheme and consider only binary and hybrid schemes in the future.

Table-6. Comparison of quenching point selection quality metrics.

Pattern	q_1	q_2	q_3	q_4
Multi-class	99.98%	**99.00%**	**2.65**	17.31px
Binary	99.98%	91.93%	1.64	7.98px
Hybrid $\rho_{1,2,3}$	99.98%	91.62%	1.70	**7.89px**
Hybrid $\rho_{1,9,25}$	99.98%	92.65%	1.81	8.49px
Hybrid $\rho_{1,9,81}$	99.98%	92.55%	1.86	9.16px

To stabilize the targeting point, the result of flame segmentation from frame to frame is averaged using a moving average. For binary and hybrid schemes, the averaging windows are set to $\tau = 0.1, 0.25, 0.5, 1, 2.5$, and 5 seconds (17). In the case of a binary scheme, $S = B$ of formulas (16) and (18). The resulting average EMA_B mask is binarized at the threshold $\beta = 0.9$ and the clipped signal (above the threshold) is used to find the contour and point of the center of mass. For the hybrid scheme, the search for the largest contour is also carried out using the binarized EMA_B mask. By extracting the maximum contour of the fla, the point of the center of mass in it is searched using the average density of the formula (21).

A comparison of the time-averaged τ of binary and hybrid schemes is shown in Table-7. There is a decrease in the value of the metric of the presence of the extinguishing point in the frame (q_1) with an increase in the averaging time τ. This is due to the lag in the detection of the extinguishing point due to the accumulation of a signal to exceed the threshold $\beta = 0.9$. To mitigate this effect, the following is a demonstration of how to use a state machine to switch states between averaging windows.

The presence of averaging contributes to an increase in the quality of the metric of the extinguishing point entering the flame circuit (q_2). For τ values between *0.1* and *1* second, q_2 has similar values. With averaging window values of *2.5* and *5* seconds, the q_2 metric is lower than in the range of *0.1* to *1* second, but much higher than in the scheme without averaging. Averaging helps to filter areas of flame (tongues) that vary greatly from frame to frame, without delaying the point of aiming at the source of the fire. This effect is demonstrated in Figure-14, where, in the first line without averaging, the targeting point moves much more strongly than in the schemes of the small and large averaging windows, in the second and third rows, respectively.

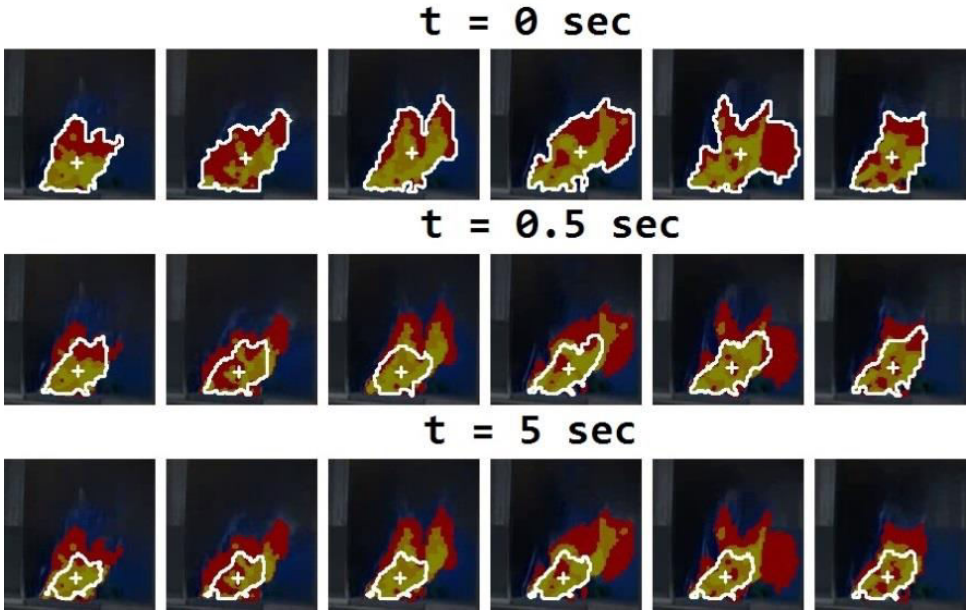


Figure-14. Visualization identification of the averaged flame contour with the designation of the center of mass.

Table-7. Comparison of flame moving average signal schemes.

Window size τ	**Method**	q_1	q_2	q_3	q_4
$\tau = 0s$ (without averaging)	Binary	99.98%	91.93%	1.64	7.98px
	Hybrid $\rho_{1,2,3}$	99.98%	91.62%	1.70	7.89px
	Hybrid $\rho_{1,9,25}$	99.98%	92.65%	1.81	8.49px
	Hybrid $\rho_{1,9,81}$	99.98%	92.55%	1.86	9.16px
$\tau = 0.1c$	Binary	96.40%	93.76%	1.69	5.66px
	$\rho_{1,2,3}$	96.40%	93.60%	1.74	5.58px
	$\rho_{1,9,25}$	96.40%	94.13%	1.83	5.83px
	$\rho_{1,9,81}$	96.40%	94.12%	1.88	6.07px
$\tau = 0.25c$	Binary	92.44%	94.60%	1.72	4.04px
	$\rho_{1,2,3}$	92.44%	94.46%	1.77	3.99px
	$\rho_{1,9,25}$	92.44%	94.13%	1.85	4.12px
	$\rho_{1,9,81}$	92.44%	94.12%	1.90	4.26px
$\tau = 0.5c$	Binary	91.08%	94.54%	1.75	3.16px
	$\rho_{1,2,3}$	91.08%	94.65%	1.79	3.13px
	$\rho_{1,9,25}$	91.08%	94.31%	1.87	3.20px
	$\rho_{1,9,81}$	91.08%	94.13%	1.90	3.29px
$\tau = 1c$	Binary	81.87%	94.16%	1.75	1.66px
	$\rho_{1,2,3}$	81.87%	94.24%	1.80	1.63px
	$\rho_{1,9,25}$	81.87%	93.61%	1.85	1.63px
	$\rho_{1,9,81}$	81.87%	93.74%	1.88	1.65px
$\tau = 2.5c$	Binary	69.82%	93.42%	1.81	0.86px
	$\rho_{1,2,3}$	69.82%	93.47%	1.84	0.84px
	$\rho_{1,9,25}$	69.82%	92.29%	1.87	0.83px
	$\rho_{1,9,81}$	69.82%	92.08%	1.89	0.82px
$\tau = 5c$	Binary	52.30%	92.93%	1.78	0.66px
	$\rho_{1,2,3}$	52.30%	92.45%	1.78	0.64px
	$\rho_{1,9,25}$	52.30%	91.01%	1.80	0.63px
	$\rho_{1,9,81}$	52.30%	90.48%	1.80	0.63px

For the metric of the average value of the flame class at the targeting point (q_3), it is important to note the following fact. It has been found that as the averaging time increases, the value of the q_3 metric for a binary scheme increases to the values of a hybrid scheme. This means that at the signal point of the largest time-averaging window, there is a steady source of flame in color, more often bright (yellow) than dark (red), and the brightness value correlates with the temperature of the fire. Thus, it is important to be able to find the source of fire energy using the methods of clarifying multi-class segmentation of fire by color, which is also demonstrated in Figure-14 for a 5-second averaging scheme (3rd frame line).

Considering the smallest mean displacement of the center of mass point (q_4), for the small averaging window τ, the best (smallest) value is obtained for the binary and hybrid scheme with a low density of orange and yellow flames. But with the growth of τ, hybrid schemes with large values of yellow and orange flame densities, such as and $\rho_{1,9,81}$, show better results. This means that over a long period, the source of the fire has predominantly yellow and orange flame zones. In the focus, the targeting point is more stable and has a smaller spread, as it is "attracted" to the zones of high-temperature yellow and orange flames due to the high density of the latter in the schemes $\rho_{1,9,25}$ and $\rho_{1,9,81}$.

Schemes with an averaging window have the effects of late detection of the flame contour (shown in Figure-15) and a delayed response to background changes as a result of camera rotation (shown in Figure-16). These effects are especially noticeable at large values of the

averaging window, but the frame-by-frame representation in Figures 15 and 16 can also be seen for small windows.

The solution to these problems is a multi-window scheme of flame averaging with state switching by the finite state machine method. There are two types of state switching: activation - switching to the next averaging window if a flame contour appears on it, and deactivation - switching to the previous averaging window if it detects the absence of flame on the current one. The mathematical formulation of this algorithm is represented by formulas (19, 20).

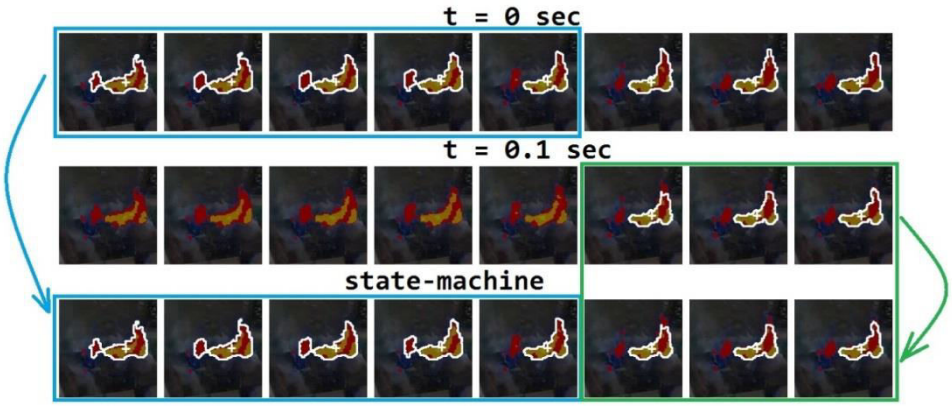


Figure-15. Visualization of the effect of the delayed reaction to the appearance of a flame at the beginning of the video for a scheme with averaging, compared to a circuit without it and a state machine.

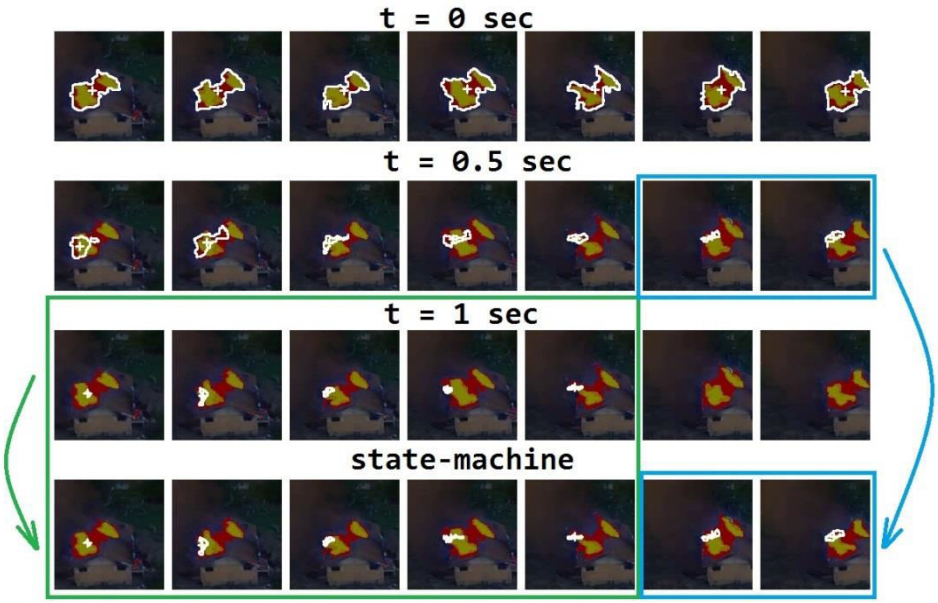


Figure-16. Visualization of the effect of the loss of the average flame zone when the the video camera is rotated for a non-averaging circuit, with averaging and a finite state machine.

The results of the finite state machine method are shown in the lower part of Figures 15 and 16. At the stage of the beginning of flame detection in Figure-15, the diagram immediately gives the result of the detected flame in the form of a contour and an extinguishing point without averaging. The *0.1-second* averaging scheme accumulates the signal above the cut-off fault only by frame 6, and the state machine circuit uses the unaveraged result of the scheme up to that frame (highlighted in blue) and then activates the averaging circuit (highlighted in green).

At the stage of changing the background and deactivating the state machine signal, shown in Figure-16, up to 6 frames, the result of the circuit with *1* second averaging is used, and then with *0.5* seconds as a result of the absence of a signal on the first frame.

Table-8 shows a comparison of state machine methods for binary and hybrid schemes of different densities. A hybrid scheme with density shows the best

metric values on average. A comparison of schemes without averaging, with a constant averaging window and a finite state machine for density switching is presented in Table-9. The value of the state machine mean offset (q_4) of *2.48px* is higher than the values with large averaging windows of *0.83* and *0.63px*, respectively. This increase is due to the use of small averaging time windows in the initial phase of the algorithm.

Table-8. Comparison of quality metrics for the selection of extinguishing points of the finite machine method of binary and hybrid flame detection schemes.

Method	q_1	q_2	q_3	q_4
Binary	98.83%	94.76%	1.72	2.46px
Hybrid $\rho_{1,2,3}$	98.83%	94.70%	1.74	2.43px
Hybrid $\rho_{1,9,25}$	*98.83%*	*93.74%*	*1.77*	*2.48px*
Hybrid $\rho_{1,9,81}$	98.83%	93.24%	1.79	2.54px

Table-9. Comparison of quality metrics for the selection of quenching points of previously considered averaging variations of the hybrid $p_{1,9,25}$ flame detection scheme.

Method	q_1	q_2	q_3	q_4
$\tau = 0$	99.98%	92.65%	1.81	8.49px
$\tau = 0.1$	96.40%	94.13%	1.83	5.83px
$\tau = 0.25$	92.44%	94.13%	1.85	4.12px
$\tau = 0.5$	91.08%	**94.31%**	**1.87**	3.20px
$\tau = 1$	81.87%	93.61%	1.85	1.63px
$\tau = 2.5$	69.82%	92.29%	**1.87**	0.83px
$\tau = 5$	52.30%	91.01%	1.80	**0.63px**
Finite machine	*98.83%*	*93.74%*	*1.77*	*2.48px*

A graph of the extinguishing point displacement metric relative to time for schemes without averaging, with a small averaging window (*0.5* seconds) and with a finite state machine is shown in Figure-17. The state machine diagram shows high displacement values on the left side of the graph, decreasing with increasing time, i.e. shifting the graph to the right. At a later stage, the offset becomes smaller than the constant averaging scheme. This is due to the *adaptability* of the state machine method, by switching the state to an average frame with a large window in which only the flame is displayed and the "tongues" are filtered.

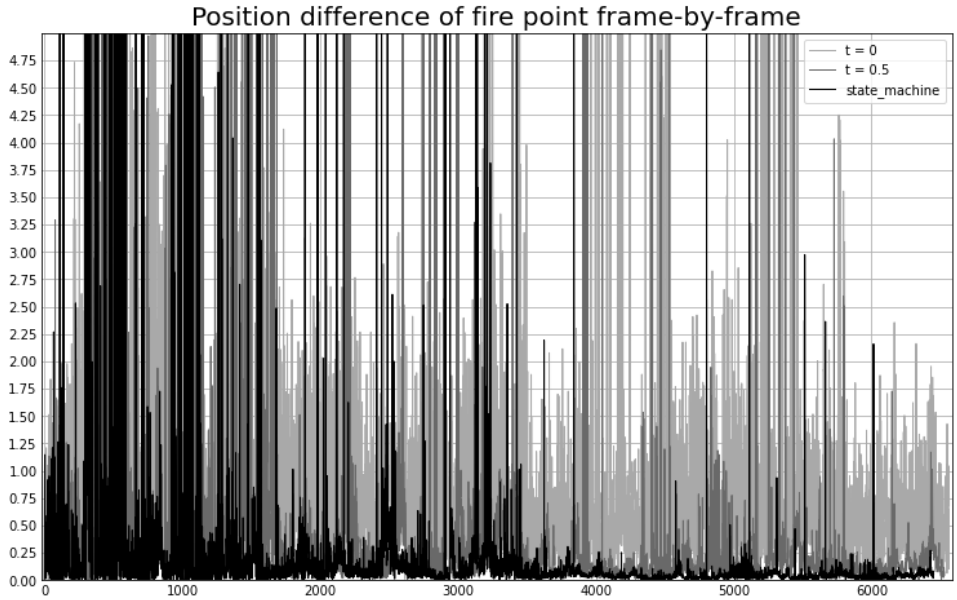


Figure 17. Graphs of the amount of extinguishing point offset between frames for the schemes: unaveraged (light gray), averaging (dark gray), and finite state machine of the set averaging schemes (black), showing the effect of decreasing the value of the offset function of the state machine method, with an increase in the number of video frames.

The finite state machine circuit, despite the high offset values, detects the flame signal without delay after detection, which is also shown in Figure-18 compared to the scheme with a large averaging window.

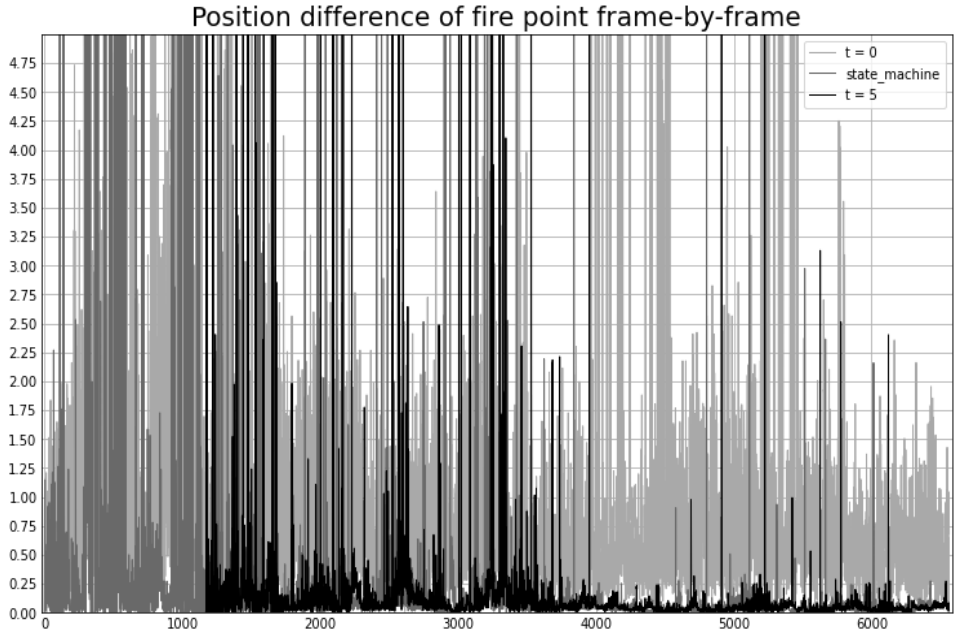


Figure-18. Graphs of the magnitude of the displacement of the extinguishing point between frames for the schemes: unaveraged (light gray), averaging (black), and finite state machine of the set averaging (dark gray), showing the elimination of the delayed response of the finite state machine method in comparison with a large averaging window.

In this way, the finite state machine method for switching averaging schemes is fast feedback from the observed appearance of the flame in the video, adaptable to minimize the movement of the water cannon, and resistant to changes in the scene in the form of moving or rotating the videographer.

4. CONCLUSIONS

The article provides a practical comparative analysis of flame segmentation models in video and algorithms for finding the optimal fire extinguishing point, using a compact mobile device as an autonomous computing node. An optimized version of the BM-UNet-32+ binary-multiclass architecture demonstrates high accuracy, with optimal computational costs for real-time mode, and is applicable as a model operating on a water-cannon device.

Based on this model, averaging schemes for segmentation results have been designed, within the framework of which a hybrid binary-multiclass finite state machine of averaging schemes based on odds shows satisfactory results in quality metrics and can be used in the software of a water cannon robot.

REFERENCES

[1] Bochkov V. S. and Kataeva L. Y. 2021. wUUNet: Advanced Fully Convolutional Neural Network for Multiclass Fire Segmentation. Symmetry 2021, 13, 98.

[2] Korobeinichev O. P., Paletsky A. A., Gonchikzhapov M. B., Shundrina I. K., Chen H. and Liu N. 2013. Combustion chemistry and decomposition kinetics of forest fuels. Procedia Engineering. 62, 182-193.

[3] Kataeva L. Y., Maslennikov D. A. and Loshchilova, N. A. 2016. On the laws of combustion wave suppression by free water in a homogeneous porous layer of organic combustible materials. Fluid dynamics. 51(3): 389-399.

[4] Kataeva L. Y., Ilicheva M. N. and Loshchilov A. A. 2022. Mathematical Modeling for Extinguishing Forest Fires Using Water Capsules with a Thermoactive Shell. Journal of Applied Mechanics and Technical Physics. 63(7): 1227-1242.

[5] Lu H., She Y., Tie J. and Xu S. 2022. Half-UNet: A simplified U-Net architecture for medical image segmentation. Frontiers in Neuroinformatics. 16, 911679.

[6] Wang Z., Wang Z., Zhang H. and Guo X. 2017. A novel fire detection approach based on CNN-SVM using tensorflow. In Intelligent Computing Methodologies: 13th International Conference, ICIC 2017, Liverpool, UK, August 7-10, 2017, Proceedings, Part III 13 (pp. 682-693). Springer International Publishing.

[7] Saponara S., Elhanashi A. and Gagliardi A. 2021. Real-time video fire/smoke detection based on CNN in antifire surveillance systems. Journal of Real-Time Image Processing. 18, 889-900.

[8] Chen X., Hopkins B., Wang H., O'Neill, L., Afghah F., Razi A. and Watts A. 2022. Wildland Fire Detection and Monitoring Using a Drone-Collected RGB/IR Image Dataset. IEEE Access. 10, 121301-121317.

[9] Dogan S., Barua P. D., Kutlu H., Baygin M., Fujita H., Tuncer T. and Acharya U. R. 2022. Automated accurate fire detection system using ensemble pretrained residual network. Expert Systems with Applications. 203, 117407.

[10] Xue Z., Lin H. and Wang F. 2022. A small target forest fire detection model based on YOLOv5 improvement. Forests. 13(8): 1332.

[11] Ronneberger O., Fischer P. and Brox T. 2015. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

[12] Chen L. C., Zhu Y., Papandreou G., Schroff F. and Adam H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 801-818).

[13] Harkat, H., Nascimento, J., and Bernardino, A. 2020, September. Fire segmentation using DeepLabv3+ architecture. In Image and signal processing for remote sensing XXVI. 11533: 134-145). SPIE.

[14] Zhou Z., Rahman Siddiquee M. M., Tajbakhsh N. and Liang J. 2018. Unet++: A nested u-net architecture for medical image segmentation. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4 (pp. 3-11). Springer International Publishing.

[15] Lin T. Y., Dollár P., Girshick R., He K., Hariharan B. and Belongie S. 2017. Feature pyramid networks for

object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117-2125.

[16] Wang Z., Peng T. and Lu Z. 2022. Comparative research on forest fire image segmentation algorithms based on fully convolutional neural networks. Forests. 13(7): 1133.

[17] Perrolas G., Niknejad M., Ribeiro R. and Bernardino A. 2022. Scalable fire and smoke segmentation from aerial images using convolutional neural networks and quad-tree search. Sensors. 22(5): 1701.

[18] Kingma D. P. and Ba J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv: 1412.6980.

[19] Loshchilov I. and Hutter F. 2016. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983.

[20] Jaccard P. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc. Vaudoise Sci. Nat. 37, 547-579.

[21] Yu F. and Koltun V. 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

[22] Suzuki S. 1985. Topological structural analysis of digitized binary images by border following. Computer vision, graphics, and image processing. 30(1): 32-46.