Erciyes Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 38, Sayı 3, 2022

Erciyes University
Journal of Institue Of Science and Technology
Volume 38, Issue 3, 2022

Spatio-temporal Object Features for Wildfire Detection in Dark Videos

Ahmet Kerim Agirman *[1], **Kasim Tasdemir** [2],

*[1] Abdullah Gül Üniversitesi Elektrik ve Bilgisayar Mühendisliği, KAYSERİ
[2] Abdullah Gül Üniversitesi Elektrik ve Bilgisayar Mühendisliği, KAYSERİ

(Alınış / Received: 05.07.2022, Kabul / Accepted: 26.08.2022, Online Yayınlanma / Published Online: 30.12.2022)

Keywords
Video based fire detection,
SVM,
majority voting,
Random Forest,
AdaBoostM1,
IBk

Abstract: In this paper, a wildfire detection algorithm from dark videos is proposed. Unlike the daytime wildfires, in the dark videos, neither the fire nor its surrounding has visually clearly perceptible texture. Its unique visual characteristics make it challenging to extract descriptive object features. This paper addresses the challenging problem by tracking the glowing objects in the darkness and extracting features based on the spatio-temporal behavior of them. It is experimentally shown that the proposed features are descriptive enough to classify wildfires with over 90% accuracy even there exists deceptive light sources such as city lights, flashlights, car headlights and reflections in the scene. Moreover, we investigate several conventional machine learning algorithms such as ensemble and kernel-based methods on the same spatio-temporal feature set. Comprehensive empirical test results demonstrate that the most accurate detection is obtained when the spatio-temporal feature set is classified using Random Forest.

Karanlık Videolarda Orman Yangını Tespitine Yönelik Uzaysal- Zamansal Nesne Öznitelikleri

Anahtar Kelimeler
Video tabanlı yangın tespiti,
SVM,
Çoğunluk Oylaması,
Rastgele Orman,
Adaboost,
IBk

Öz: Bu makalede, karanlık videolardan orman yangınının tespitine yönelik bir yöntem önerilmiştir. Gündüz orman yangınlarından farklı olarak, karanlık videolarda yangının kendisi de çevresi de görsel olarak açıkça algılanabilir bir desene sahip değildir. Karanlık videolardaki yangının bunun gibi kendine özgü görsel özellikleri, tanımlayıcı nesne öznitelikleri çıkarmayı zorlaştırmaktadır. Bu makale, karanlıkta parlayan nesneleri takip edip, uzaysal-zamansal davranışlarına dayalı öznitelikleri çıkararak bu zorlayıcı duruma çözüm üretmektedir. Önerilen özniteliklerin, videoda şehir ışıkları, el fenerleri, araba farları ve olay yerindeki yansımalar gibi aldatıcı ışık kaynakları olsa bile, orman yangınlarını %90'ın üzerinde doğrulukla sınıflandırmak için yeterince temsil edici olduğu deneysel olarak gösterilmiştir. Ayrıca, aynı uzaysal-zamansal öznitelik kümesinde topluluk ve çekirdek tabanlı sınıflandırma yöntemleri gibi çeşitli geleneksel makine öğrenmesi algoritmaları da karşılaştırma amacıyla denenmiştir. Kapsamlı deneysel test sonuçları, en yüksek tespit doğruluğunun, önerilen uzaysal-zamansal öznitelik kümesinin Rastgele Orman sınıflandırma yöntemi elde edildiğini göstermektedir.

*Corresponding Author, email: ahmet.k.agirman@gmail.com

1. Introduction

From 1988 to date, 72.360 forest fire incidents are reported [1]. As a result of these disasters, in Turkey, around 300.000 hectares of forest are lost. While 5.8 hectares per fire forest area is destroyed before the last five years period, this rate dropped to 2.4 hectares per fire in the last five years. Clearly, besides enhancements on firefighting techniques and development of related technology, this progress is due to efficient surveillance techniques and timely reports. Since December 2016, 23% of reported forest fires and 47 % of city fires in Istanbul occurred at no

daylight conditions [1-3]. Beginning from 1900s, watch towers have been an important part of fire detection across the world. However, due to human factors, fire announcement process didn't work properly all the time which increased forest loses especially at rural areas. For this reason, computer vision based automatic fire detection methods have been an important part of the forestry departments due to the fact that they don't require any human deployment to the risky areas and they support visual information via remote monitoring systems.

In order to overcome fire detection problem from video surveillance systems, various techniques have been proposed [8-20]. Front runner techniques include detection of fire, smoke or both depending on spatial and temporal features of objects and colors in a video and using different spectral or physical range cameras [16]. After processing video sequences, those techniques decide if a pixel, frame, sequence, or the whole video contain a fire.

However, very limited number of these works considered the fire detection in dark videos. Recent works shows evidence of semi or full daytime fires. Tasdemir's work at [22] proposes a method on distant night fire detection. Since the fire event is assumed to be at far distance, fire is considered as a slow-moving object. Even though this approach is fine for distant fires, it may not be correct for short or mid-range ones. In this paper, we propose a fire detection method that is able to detect wildfires from dark videos while overcoming false alarm sources, such as city lights, car headlights, streetlights, etc. This work extends the approach introduced in [23] with comprehensive comparisons with other machine learning methods and improves it by making structural additions.

Contribution of this paper can be counted as three folds:
1. A spatio-temporal feature extraction method including object tracking in dark video is proposed,
2. Comprehensive comparison of ensemble and kernel-based classification methods on wildfire detection in dark videos are demonstrated,
3. A final wildfire detection method which is robust against common source of false alarm sources in dark videos such as city lights or car headlights is proposed.

The organization of the paper is as follows: the proposed fire detection method is explained in Section II. It follows with the experimental setup section, Section III, then Section IV demonstrates the test and comparison results. The findings are discussed and concluded in Section V.

2. Material and Method

2.1. Extraction of Foreground Objects in Dark Videos

One challenging part of working on light emitting objects on the dark videos is they have limited visual features to be tracked or make any in depth visual analysis. For that reason, instead of visual ques of the object, we target to investigate its temporal behavior. However, we need to track an object throughout the video despite of the challenge. Light-emitting objects appear, disappear, flicker, move and even intersect with others or unmerge from the others in the video. All these cases are handled by the proposed object extraction and tracking algorithm.

Contrary to daytime counterparts, night-time videos contain very limited color information. They are very akin to digital binary images. Thus, without any color processing, each frame is first converted to a black & white image with a threshold of τ_0 by using Otsu's method [24]. As a result, the dark pixels are represented by 0 and bright ones by 1. Binary blobs in each frame are detected with 8 connectivity adjacency rule. This eliminated disconnected or isolated foreground pixels. Moreover, blobs having fewer pixels than τ_0 are discarded to reduce number of blobs considered as noise. The reason 8 connectivity is used instead of 4 is the nature of a fire which has a very fragmented structure, thus, when 4 connectivity is used there will be many small blobs belonging to same fire flame which makes analysis difficult.

Let $b_{n,k}$ be *k*'th fire candidate blob of *n*'th video frame and o_m be *m*'th object in the whole video. While in one frame o_m can be represented by *k*'th blob, $b_{n,k}$, in the succeeding frame it can be represented by *k+1*'th blob, $b_{n+1,k+1}$. Then a tagging procedure should be implemented for each blob in each frame to uniquely index each object across the video with an ID. A tag will have a lifetime; a tag is born, lives for a while, and then dies as the object disappears from the video. Basically, light blobs not only appear and disappear from the video, but they also move, intersect or unmerge. For that reason, we need to have an algorithm to track these light emitting objects. If the subsequent frames have intersecting blobs, then it is considered that they are the same objects and so tagged with the same ID. In other words, if tagging function, $b_{n,k} \rightarrow o_m$, is known, tagging procedure is performed as follows:

$$b_{n+1,i} \rightarrow \begin{cases} o_m & , b_{n,k} \cap b_{n+1,k} \neq \emptyset \\ o_{m+1} & , otherwise \end{cases} \quad (1)$$

The equation indicates that if two object in consecutive frames are spatially intersecting, they are the same objects and they need to have the same ID. Initially, blobs in the first frame also tagged with their arbitrarily initialized blob numbers. However, this approach has some difficulties. For example, if both $b_{n,k}$ and $b_{n,k+1}$ intersect with $b_{n+1,i}$, are those all the same objects? Another difficulty is if both $b_{n+1,i}$ and $b_{n+1,i+1}$ intersect with $b_{n,k}$, which objects are to be as separate? In Figure 1, both difficulties given above are represented. Assume the first frame n=1 contains seven blobs drawn in black circles and the second frame n=2 contains eight blobs drawn in red circles. Blobs in the first frame take their blob numbers as ID tags, i.e., $b_{1,1}$ gets tag 1, $b_{1,2}$ gets tag 2, etc. Now consider $b_{1,1}$ and $b_{2,4}$ intersect most, then $b_{2,4}$ gets the tag ID: 1. Next, $b_{1,2}$ intersects with $b_{2,2}$ most, thus $b_{2,1}$ dies, $b_{2,2}$ pairs with $b_{1,2}$ and gets the tag ID 2. Third, $b_{1,3}$ intersects with $b_{2,2}$ most, and thus both $b_{2,3}$ and $b_{2,5}$ die, $b_{2,2}$ pairs with $b_{1,3}$ and gets the tag 3. Fourth, $b_{1,4}$ intersects only with $b_{2,6}$ and $b_{2,6}$ gets the tag 4. Fifth, $b_{1,5}$ intersects only with $b_{2,6}$ and $b_{2,6}$ this time gets the tag 5. In similar way $b_{2,7}$ gets the tag 7, $b_{1,6}$ intersects with no one and dies. $b_{2,8}$ intersects with no one and is born by getting a new tag 8. This operation made the second difficulty apparent: $b_{2,2}$ and $b_{2,6}$ have two distinct tags transferred to them. This conflict is resolved in a similar way: $b_{2,2}$ intersects with $b_{1,3}$ more than $b_{1,2}$ does. Thus, it gets the tag 3 and $b_{1,2}$ dies; $b_{2,6}$ intersects with $b_{1,4}$ most, gets the tag 4, and $b_{1,5}$ dies. In summary, tags 2, 5, and 6 dies, tags 1, 3, and 4 survives, tag 8 is newly born, however $b_{2,3}$ and $b_{2,5}$ are stillbirths.

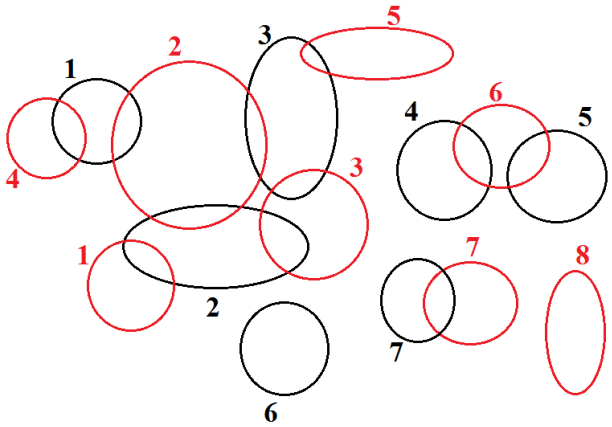

Figure 1. The figure shows possible scenarios that might come up during glowing object tracking in a dark video. Black and red circles indicate the foreground object location in the nth and the next frame, i.e. (n+1)'th frame. Since the flame has limited visual ques, their spatio-temporal locations are used to track the objects throughout the video.

2.2 Spatio-temporal Feature Extraction

In order to capture the temporal behavior of the flickering flame, the features are extracted from a number of video sequences. Size of the temporal window is a tradeoff between detection time of fire alarm and its accuracy. In order to extract features from same number of frames, a tag that is not apparent along a full window is discarded from computations.

Features of a full-window tag are extracted from change in various motion variables of the tagged object. Thirty features are derived from these 6 variables which can be listed as: pixel area of the object in frame, 2D position of the mass center of the object, height, width and area of smallest bounding box (BB) of the object.

By using these variables, we can realize distinctive characteristics of a night fire that are flickering and motion behavior. The variables are followed along a window and various features are extracted as explained presently.

While an object's variance of pixel area is large along a window, it is small for moving vehicles and fixed sources such as street, city, or house lights since area of such non-fire light sources does not change suddenly along a video. However, due to flickering motion of a fire, the area will change rapidly. Similarly, variance of height and width of BB will usually be large for fire objects and small for others. It is for this reason, mean and variance of height and width of a BB as well as their first and second order derivatives will be distinctive between fire and non-fire objects.

Let ψ_n be value of a variable at n'th frame in a window with N number of frames. For many of these variables, mean and variance is computed as follows, respectively:

$$\mu_0 = \frac{1}{N}\sum_{n=1}^{N} \psi_n \tag{2}$$

$$\sigma_0^2 = \frac{1}{N-1}\sum_{n=1}^{N} | \psi_n - \mu_0 |^2 \tag{3}$$

Mean and variance of first and second order derivative of some variables are computed as in Eq. (4-7), respectively.

$$\mu_1 = \frac{1}{N-1}\sum_{n=2}^{N} (\psi_n - \psi_{n-1}) \tag{4}$$

$$\sigma_1^2 = \frac{1}{N-2}\sum_{n=2}^{N} | (\psi_n - \psi_{n-1}) - \mu_1 |^2 \tag{5}$$

$$\mu_2 = \frac{1}{N-2}\sum_{n=3}^{N} (\psi_n + \psi_{n-2}) \tag{6}$$

$$\sigma_2^2 = \frac{1}{N-3}\sum_{n=3}^{N} | (\psi_n + \psi_{n-2}) - \mu_2 |^2 \tag{7}$$

For a fire object, variance of center of mass (CoM) is higher in vertical axis than in lateral axis. For a car moving horizontally, variance of CoM of headlights in vertical axis is very small. In the same manner, variance of CoM of fixed light sources in both axis is very small. These are the reasons we used variance of CoM as a feature.

Here, it is important to note that horizontal and vertical location of CoM is not considered as features since a fire can take place anywhere in the video. Otherwise, the system can be trained for a specific location that fire is expected to start. That's why position free features (i.e., mean and variance of first and second order derivatives) are used. In Table 1, variables and features are summarized.

If a feature belongs to a greater interval than other features, impact of small-bounded ones may be reduced. Normalization is the solution to avoid such a problem. Min-max normalization has the ability to preserve relation between elements of a feature vector, thus it is chosen. Let $\delta_{i,j}$ be value of *j*'th feature at sample *i*. Then, min-max normalization is defined as

$$\overline{\delta}_{i,j} = \frac{\delta_{i,j} - min\delta_{i,j}}{max\delta_{i,j} - min\delta_{i,j}} \tag{8}$$

Table 1. Extraction of Features from Variables.

	Feature		**Feature 1st Der**		**Feature 2nd Der**	
	Mean	*Var*	*Mean*	*Var*	*Mean*	*Var*
Pixel Area	x	x	x	x		
CoM x axis			x	x	x	x
CoM y axis			x	x	x	x
BB width	x	x	x	x	x	x
BB height	x	x	x	x	x	x
BB area	x	x	x	x	x	x

In real-time applications, video stream may be continuous. Therefore, after adding a new window, normalization should be implemented throughout up-to-date data.

2.3 Training

In this work, as a base classifier, Support Vector Machines (SVM) is used. Besides SVM, majority voting, Random Forests, AdaBoostM1, and IBk classifiers used and their performance compared to SVM. First a classifier model is

constructed and then the model predicts class of any test instance it is supplied. Here, we used LIBSVM library with radial based function (RBF) kernel since our data set has a nonlinear classification characteristic.

In order to get most accurate classification, best $c \in I^+$ cost and $\gamma \in I^+$ impact range parameters should be found. If $c_2 > c_1 > 0$ and $\gamma_2 > \gamma_1 > 0$ are predetermined intervals, then optimization requires a $[c_1, c_2] \times [\gamma_1, \gamma_2]$ size grid search for the best (c^*, γ^*) pair [25]. In our tests, an accelerating intervention to optimization saved time and gave a better (c^*, γ^*) pair compared to pairs obtained when not intervened. The intervention is simple: after at least ten trials, if the last five trials produce a mean absolute deviation of accuracy no greater than *1.5*, halt the search and use current pair as (c^*, γ^*).

Classes of instances in training set is determined by a professional for fire objects as *1* and for non-fire objects *-1*. With (c^*, γ^*) pair, a model is constructed in SVM and class of all instances from a distinct test set is predicted from the set $\{-1,1\}$. Accuracy and elements of confusion matrix (i.e., true positive rate, false negative rate, true negative rate, and false positive rate) are used as performance measures.

While SVM gives good results of predictions, majority voting (MV) improves these results significantly. In a test set, MV is implemented between distribution of fire or non-fire classification of an object tag. Then, class of the object is redetermined according to result of MV.

3. Setup of Experiments

Experiments implemented on a video dataset curated by the authors. Sketch of place that fire videos are recorded given in Figure 2. The maximum distance a camera can see in the night is *1km* and distance between fires and cameras changes from *30m* to *100m*. Besides fires, in 360^o sight of the cameras there are a series of streetlights, both bright and dark roads, city lights, tower lights flashing on and off, moving vehicle headlights in low- or high-density traffic, short distance house and streetlights.

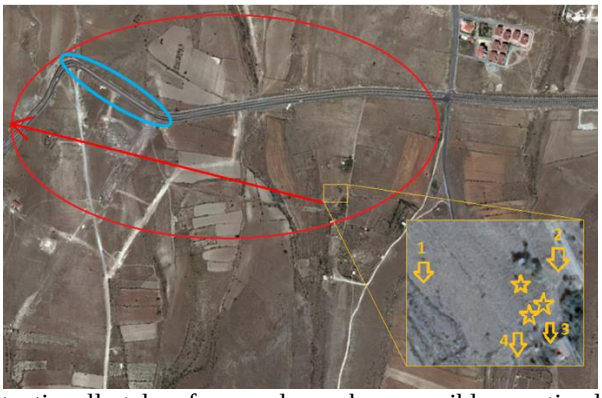

Figure 2. Dataset videos are intentionally taken from a place where possible negative light sources appear in the scene such as city lights or car lights. Location of test fires (stars), cameras (arrows) and sight of the scene is shown in red circle. The maximum distance of the sight from cameras is around 1 km. The road section (blue circle) generated challenging fire-like vehicle headlights during the fire recordings.

Four different cameras are used recording fires usually in 640x480 resolution: Casio Exilim EX-Z350, Nikon D3200, Samsung S850, and Samsung WB100. In total 15 night fires are recorded. In Table 2, some characteristics of the videos are tabulated. A montage of the videos are given in the Figure 3.

The global image threshold is experimentally determined to be $\tau_0 = 0.5$, thus objects with low luminance and noise can be eliminated. Furthermore, objects having pixels fewer than $\tau_1 = 16$ also discarded even fire objects since an event size lower then 16 pixels is not considered significant. Remaining fire candidates are labelled as 1, and other objects are labelled as -1. An object is decided to be fire if only the object represents a flame object. Some very close reflections of a flame object, mostly on the ground, also behave exactly as the flame object; however, these objects are considered as not-fire.

Analysis implemented in MATLAB® environment for window sizes of 5, 10, 20, 50, 100, and 200 pixels. SVM parameter optimization intervals are experimentally determined to be $c_1 = 5, c_2 = 9, \gamma_1 = 4, \gamma_2 = 8$.

Table 2. Properties of the samples in the video dataset.

Video	Duration	# of frames	# of negative samples	# of positive samples
1	9:41	17439	99	178
2	16:09	29091	265	148
3	21:23	38518	27	307
4	29:00	52231	17	671
5	04:00	6009	505	175
6	11:04	16608	2044	471
7	12:46	19158	530	405
8	20:00	30003	6	1343
9	12:05	21746	81	1508
10	14:14	25608	89	1608
11	18:10	32688	1208	927
12	20:00	35977	266	1876
13	08:34	15435	2681	1113
14	13:17	23913	7758	1839
15	03:25	6145	222	624

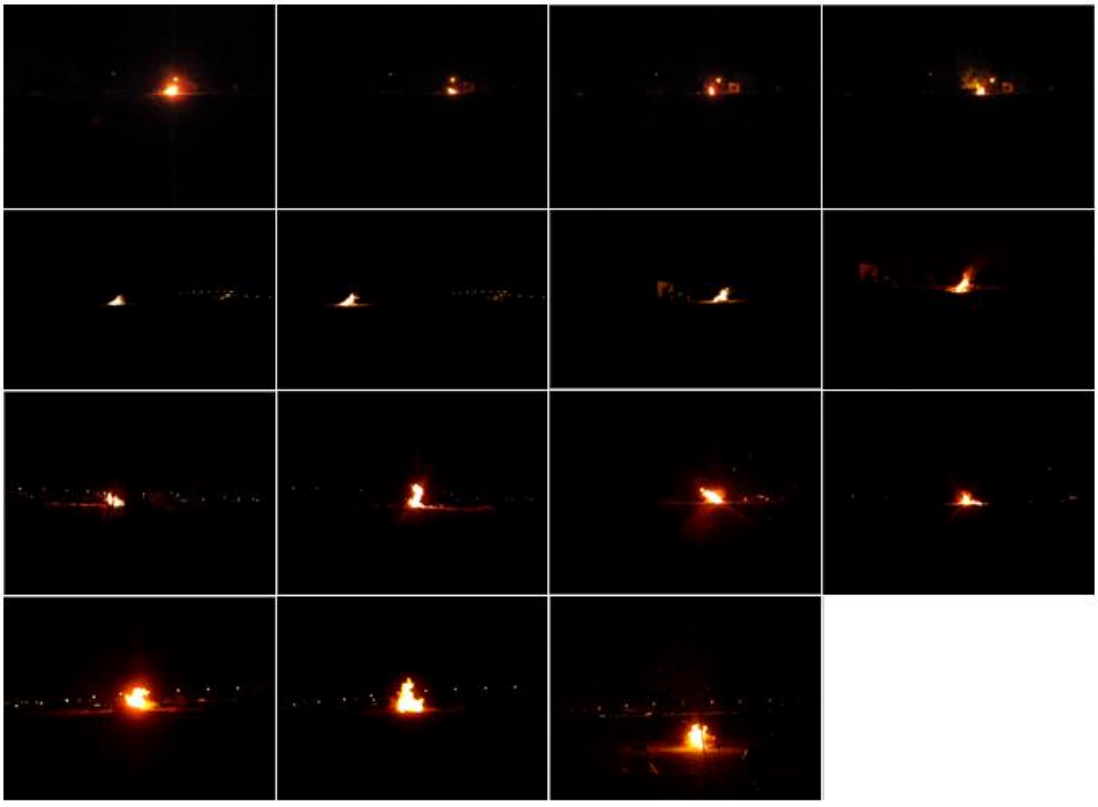

Figure 3. Representative images of the dataset videos indicated in Section 3. The videos include both fire and not-fire frames.

When a video is chosen for test, remaining ones are used for training (leave-one-out). For a total number of 90 experiments, average training and test set sizes are 7437 and 5058 instances, respectively. It should be noted that number of instances in a training set is limited to a maximum 10,000 while no restriction applied to test sets. Average distribution of fire and not-fire instances over 6 windows are tabulated in Table 3 and number of instances per window size is given in Table 4.

Table 3. Distribution of Not-Fire classes among the videos.

Video	Average #of negative objects (%)	Video	Average #of negative objects (%)
1	56.31	9	0.09
2	66.86	10	0.40
3	63.89	11	10.35
4	62.20	12	01.36
5	03.34	13	92.54
6	08.89	14	91.23
7	08.10	15	18.32
8	0.08	-	-

Table 4. Number of instances among windows.

Window size	# of instances
5	280.381
10	102.588
20	44.584
50	16.243
100	7.758
200	3.723

Some fires were able to reach up to *3m* height under low wind conditions. In videos 1, 2, 3, and 4, a very deceptive streetlight is apparent. From video 9 to 15, very deceptive city lights combined with semi-intense traffic are apparent. Besides these not-fire objects, a torch is also used to create false objects (Figure 4, video 3).

4. Experimental Results

In this section, performance of our method evaluated and experimental results are analyzed. The measures we use for evaluation are accuracy, true positive rate which implies state of "true alarm", and false negative rate which implies state of "false alarm". Other measures can be false negative rate or "missed alarm" and true negative rate or "true silent".

4.1 SVM Results

Selected performance measurements of the proposed method are shown in Table 5. When proposed features are used for a night fire, SVM is able to classify new instances correctly with an accuracy of usually over 90%. Implementing MV after SVM classification boosts accuracy rates usually over 95%. True Positive Rate (TPR) values are over 94% on average, however in some videos True Negative Rate (TNR) values are low due to reasons given below.

Table 5. Accuracy, TPR and TNR measurements on the videos. SVM and SVM+MV methods are compared.

Video	SVM (Accuracy)	SVM+MV (Accuracy)	SVM (TPR)	SVM (TNR)
1	0.90	**0.97**	0.91	0.89
2	0.88	**0.99**	0.78	0.92
3	0.96	**0.99**	0.92	0.98
4	0.98	**0.99**	0.97	0.99
5	**0.84**	**0.84**	0.84	0.69
6	0.92	**0.95**	0.95	0.45
7	**0.93**	**0.93**	0.99	0.30
8	0.98	**0.99**	0.97	0.00
9	0.95	**0.98**	0.95	0.22
10	0.96	**0.99**	0.97	0.15
11	0.92	**0.93**	0.99	0.19
12	0.97	**0.99**	0.97	0.52
13	0.98	**0.99**	0.99	0.97
14	0.97	**0.99**	0.99	0.97
15	**0.81**	**0.81**	0.99	0.02

In videos 5 and 6, the false alarm generating frame region is a sharp turn which is part of the road in the scene (Blue circle in Figure 2). This part of the road extends from front to back on the scene, which makes vehicles move not quite linearly. Since traffic is semi-intense or intense during the recording time, vehicles slowed down, and

overhead lights clustered to form fire-like moving objects. In Figure 4, on top, a sample frame is shown. Even though tag 11 is a not-fire object, it is detected as fire and boxed.

In videos 7, 8, 10, 12, and 15, reflections at not-fire regions on the ground up to half or one meter close to fire origin and intense luminance on objects very close to fire cause an error. In Figure 4, on middle, tag 30 is a fire objects that is predicted as fire, however though tag 172 is a reflection and not a fire, it is classified as fire. Notice that, since these types of errors appear when a fire is the case, they do not really cause a false alarm.

In videos 9 and 11, a moving torch initially turns to the camera and then turns quickly back which makes the illuminated area first grow and then suddenly shrink, eventually causes an error. In Figure 4, on bottom, both tags 1873 and 1975 predicted as fire while tag 1975 is a not-fire object.

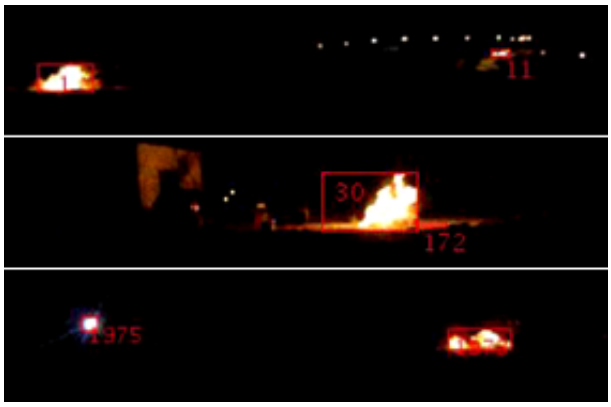

Figure 4. Examples of false detections are shown. TOP: The fire on the left (Object tag: #1) is correctly detected. However, flashlights of the cars passing by (Object tag: #11) are falsely labeled as fire object. MIDDLE: Fire (Object tag: #30) is correctly detected while its reflection (Object tag: #172) is also falsely detected, BOTTOM: (Object tag: #1975) is a correctly detected fire. (Object tag: #1873) is a flashlight and it is a false detection.

Window size also has an effect on accuracy. When window size increases, more evidence per window is collected for decision process which allows better predictions. For pre-processing, which includes tagging procedure, more computation is required. However, for SVM runs, less computation is the case. When window size decreases less evidence per window is collected, less pre-processing computation and more SVM computation is required. Table 5 shows performance measures for two windows: *N=5* and *N=200*. When *N=5*, average accuracy is 89.47% and when *N=200*, accuracy also increases to an average of 96.63%. An increase in window size also decreases false alarm rates. For example, in Figure 4 and video 1, the street, the house, and a torch light are predicted as fire objects, When window size is 200, at the beginning of the video house lights very short time, the torch never and due to move of the camera at the end of the video the streetlight very short time are predicted as fire. Even though this encourages us to use longer windows (preferably with high fps cameras) due to heavy work of pre-process, alarm response time will eventually decrease. In Table 6, NaN corresponds to existence of no not-fire objects in the video. In videos 11 and 15, TNR value is 0 due to misclassification of intense luminance of a vehicle standing very close to fire (Figure 4).

Table 6. Investigation of effect of window size on the classification accuracy. Window sizes N=5 and N=200 are used in the tests.

Video	**SVM (Accuracy)**		**SVM (*TPR*)**		**SVM (*TNR*)**	
	N=5	N=200	N=5	N=200	N=5	N=200
1	88.28	**91.09**	0.96	0.87	0.81	0.93
2	82.98	**91.32**	0.77	0.80	0.85	0.96
3	92.25	**98.00**	0.88	0.94	0.94	1.00
4	97.43	**98.78**	0.95	0.97	0.98	0.99
5	79.03	**91.89**	0.79	0.91	0.78	NaN
6	86.61	**100.0**	0.93	1.00	0.61	-
7	88.03	**100.0**	0.99	1.00	0.12	1.00
8	95.87	**99.20**	0.95	0.99	0.00	NaN
9	89.59	**100.0**	0.89	1.00	0.43	NaN
10	94.90	**98.04**	0.95	0.98	0.17	NaN
11	87.68	**98.47**	0.99	1.00	0.36	0.00
12	95.10	**98.78**	0.96	0.98	0.51	NaN

13	95.66	**99.34**	0.94	0.98	0.95	0.99
14	94,61	**99.54**	0.98	1.00	0.94	0.99
15	73.95	**85.00**	0.96	1.00	0.01	0.00

Apart from errors explained above, the proposed method successfully does not classify street or city lights, headlights of vehicles and many other not-fire objects.

4. 2. Comparisons with the other classification methods

SVM is a standard tool for image classification problems. However, there exist some other tools performs equally, some even better. In this section, we implement Random Forests (RF), AdaBoostM1 (AB), IBk and J48 machine learning tools on our data. Performance measures are the same as we used for SVM at previous section. The platform used for implementation is Weka data mining software by The University of Waikato. Default parameter set up is used for the tests. Contrary to SVM experiments, training data is not limited and full training set is used for building a model (see Table 4). In addition to SVM, 360 more tests are implemented, one test per video per window size, and per machine learning tool.

Accuracy results are given in Table 7, In the table, except videos 11 and 15, RF showed the best performance among other tools. Second best performance belongs to SVM. AB showed best performance for video 15 and IBk for video 11. Videos 3, 4, 6, 7, 8, 10, 11, 12,13, and 14 show a robust performance under any machine learning tool while videos 1, 2, 5, 9, 15 shows unstable performance. Performance of these methods is summarized in Figure 5. In Table 8, TNR values are tabulated. On average, IBk gives lowest average false alarm rate of 32.01% and SVM gives the highest average rate of 44.75%. Most robust videos in terms of TNR value are videos 3, 4, 13, and 14. After all, all these analysis shows us in terms of fire catch RF performs best, however in terms of false alarm avoidance IBk performs best.

Table 7. Comparison of the accuracies of SVM, Random Forest (RF) AdaBoostM1 (AB), IBk

Video	SVM	RF	AB	IBk
1	90.0	**94.4**	83.9	91.5
2	88.0	**94.0**	83.1	89.6
3	96.0	**98.6**	92.8	94.0
4	98.0	**98.4**	95.0	95.6
5	84.0	**88.3**	72.8	78.4
6	92.0	**92.6**	86.3	87.6
7	93.0	**95.1**	92.8	93.8
8	98.0	**98.8**	95.5	94.4
9	95.0	**97.2**	85.4	86.5
10	96.0	**98.9**	96.8	94.0
11	92.0	93.8	93.6	**94.0**
12	97.0	**98.6**	95.6	93.1
13	98.0	**99.3**	98.1	97.8
14	97.0	**98.3**	97.3	97.2
15	81.0	84.6	**88.0**	79.3

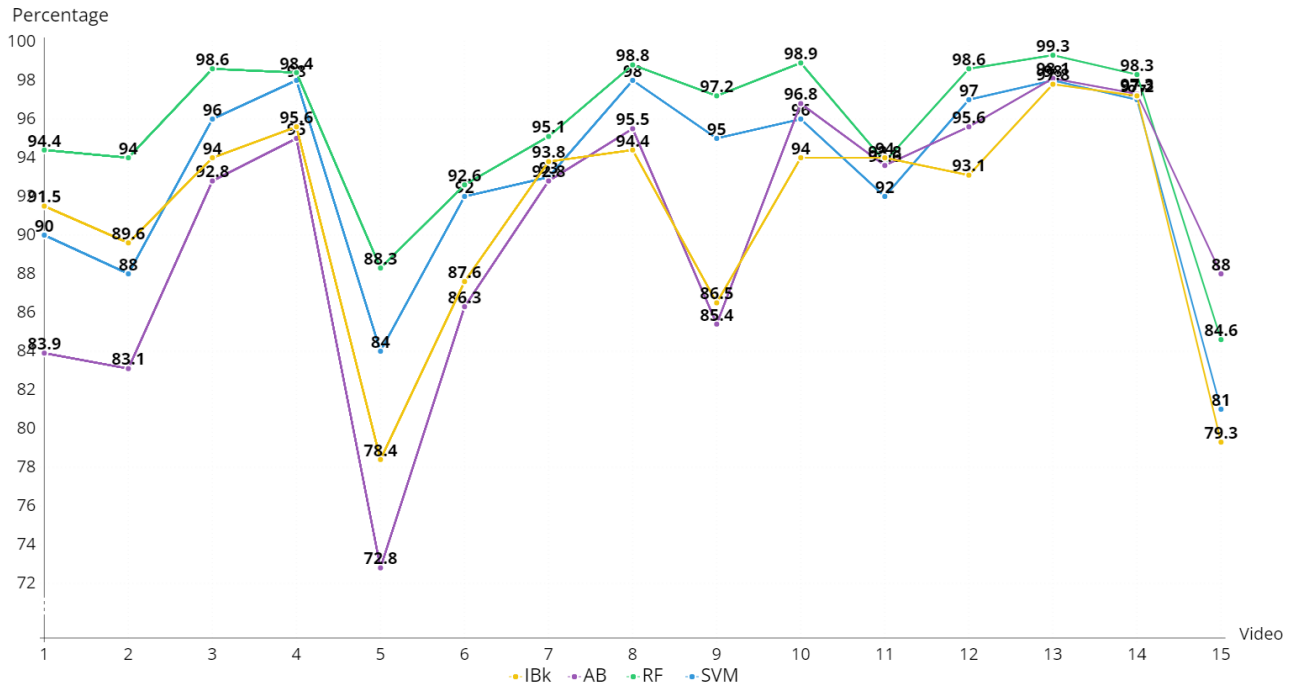

Figure 5. Comparison of the accuracies of SVM, Random Forest (RF) AdaBoostM1 (AB), and IBk given in the Table 7.

Table 8. Comparison of TNR of SVM, Random Forest (RF) AdaBoostM1 (AB), IBk

Video	SVM	RF	AB	IBk
1	10.40	08.38	17.20	09.00
2	07.70	05.55	21.26	03.90
3	01.90	00.76	07.58	01.50
4	01.10	01.12	04.29	01.30
5	30.80	13.31	15.77	18.86
6	54.50	53.91	51.19	58.10
7	69.90	68.90	84.50	78.00
8	100.0	83.30	100.0	00.00
9	77.20	54.70	80.00	29.70
10	85.30	89.20	93.90	85.52
11	80.20	57.12	60.80	48.60
12	47.70	04.84	56.60	44.20
13	02.00	00.72	01.70	01.90
14	02.80	01.86	02.83	02.90
15	99.80	68.90	37.72	96.70

5. Discussion and Conclusion

In this paper, a video-based wildfire detection method for under-illuminated environments is proposed. The experimental results show that temporal behavior of the flickering flame in a dark video has a distinct characteristic, and it is well suited for flame and fire detection in low light conditions. This temporal behavior of the fire allows us to extract descriptive spatio-temporal features from a fire video even the visual texture of the objects in the dark video are not visible. The proposed object features are taking advantage of temporal flickering motion of a night fire. The classification method can distinguish the deceptive false alarm sources such as city and streetlights, vehicle headlights and flickering reflections.

It is experimentally verified that the fire detection accuracy of the proposed method is over 90% on the average. The method is tested with various hyper-parameters such as temporal window size. It is shown that when the temporal window size is increased to include 200 consecutive frames, over 95% accuracy on average was obtained.

The proposed object features are tested with various classification methods such as SVM, Random Forests, AdaBoostM1, and IBk. The comprehensive comparison shows that Random Forests classification attains the

highest accuracy on the extracted features. It is also shown that the detection accuracy of IBk is comparable to the most accurate model. Moreover, among all tested machine learning algorithms, IBk gives the smallest false alarm rate, 32.01%, while SVM gives the highest. Therefore, when the reduction of the false alarm rate is more critical, IBk can be employed. The method is fundamentally applied to fires that are not far away. The method is based on temporal features of the fire object, which are derived from fires videos that are captured low to mid-range distances. It should be expected that flickering and other temporal behavior of a far distance fire should have differing features then that of a low to mid-range fire. Therefore, future research efforts can extend the method for detection of fire at far distance.

References

[1] DGF, "2021 Forestry Statistics", *Turkish Directorate General of Forestry*, Ankara, 2021.

[2] DGF, (2017, Jan 29), *State of Forest Fires* [Online]. Available: https://www.ogm.gov.tr/Lists/GuncelOrmanYanginlari/.

[3] Istanbul Fire Department, "2011-2016 Statistics," *Istanbul Municipality*, Istanbul, 2011.

[4] B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "Flame detection in video using hidden Markov models", *IEEE Int. Conf. Image Process*., vol. 2, no. October-2005, p. II-1230-3, 2005.

[5] Y.Dedeoglu, B.U.Toreyin, U.Gudukbay, and A.E.Cetin, "Real-timefire and flame detection in video", *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process.* - Proc., vol. II, pp. 669672, 2005.

[6] B.U.Toreyin, Y.Dedeoglu, and A.E. Cetin, "Contour based smoke detection in video using wavelets", *Eur. Signal Process. Conf*., no. EUSIPCO, pp. 610, 2006.

[7] Y. Dedeoglu, B.U. Toreyin, U. Gudukbay, and A.E. Cetin, "Real-timefire and flame detection in video", *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process.* - Proc., vol. II, pp. 669672, 2005.

[8] B. U. Toreyin and A. E. Cetin, "Online Detection of Fire in Video", *2007 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 15, 2007.

[9] B.U. Toreyin and A.E. Cetin, "Computer vision based forest fire detection", *2008 IEEE 16th Signal Process. Commun. Appl. Conf*., p. 6800, 2008.

[10] K. Dimitropoulos, K. Köse, N. Grammalidis, and A. E. Çetin, 'Fire detection and 3D fire propagation estimation for the protection of cultural heritage areas', *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. 8, pp. 620–625, 2010.

[11] O. Gunay, A. E. Cetin, and B. U. Töreyin, "Online adaptive decision fusion framework based on projections onto convex sets with application to wildfire detection in video," *Optical Engineering*, vol. 50, no. 7, p. 077202, 2011.

[12] Y. H. Habiboglu, O. Gunay, and A. E. Cetin, "Real-time wildfire detection using correlation descriptors," *in 2011 19th European Signal Processing Conference*, 2011, pp. 894–898.

[13] O. Gunay, B. U. Toreyin, K. Kose, and A. E. Cetin, "Entropy-functional-based online adaptive decision fusion framework with application to wildfire detection in video," *IEEE Transactions on Image Processing,* vol. 21, no. 5, pp. 2853–2865, 2012.

[14] Y. H. Habiboğlu, O. Günay, and A. E. Çetin, "Covariance matrix-based fire and flame detection method in video," *Machine Vision and Applications*, vol. 23, no. 6, pp. 1103–1113, 2012.

[15] F. Erden et al., "Wavelet based flickering flame detector using differential PIR sensors," *Fire Safety Journal,* vol. 53, pp. 13–18, 2012.

[16] S. G. Kong, D. Jin, S. Li, and H. Kim, "Fast fire flame detection in surveillance video using logistic regression and temporal smoothing," *Fire Safety Journal*, vol. 79, pp. 37–43, 2016.

[17] S. Verstockt et al., "A multi-modal video analysis approach for car park fire detection," *Fire safety journal*, vol.

57, pp. 44–57, 2013.

[18] K. Dimitropoulos et al., "Flame detection for video-based early fire warning for the protection of cultural heritage," *Euro-Mediterranean Conference*, 2012, pp. 378–387.

[19] T. Toulouse, L. Rossi, T. Celik, and M. Akhloufi, "Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods," *Signal, Image and Video Processing,* vol. 10, no. 4, pp. 647–654, 2016.

[20] K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection," *IEEE transactions on circuits and systems for video technology*, vol. 25, no. 2, pp. 339–351, 2014.

[21] A. E. Çetin et al., "Video fire detection–review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, 2013.

[22] K. Tasdemir, O. Gunay, B. U. Toreyin, and A. E. Cetin, "Video based fire detection at night," in *2009 IEEE 17th Signal Processing and Communications Applications Conference*, 2009, pp. 720–723.

[23] A. K. Agirman and K. Taşdemir, "Short to mid-range night fire detection," *in 2017 25th Signal Processing and Communications Applications Conference (SIU*), 2017, pp. 1–4.

[24] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST),* vol. 2, no. 3, pp. 1–27, 2011.

[26] Y. Wan, Y. Chen, and K. Li, "Identification and spatiotemporal distribution analysis of global biomass burning based on Suomi-NPP VIIRS Nightfire data," *Journal of Cleaner Production*, vol. 359, p. 131959, 2022.

[27] A. Bouguettaya, H. Zarzour, A. M. Taberkit, and A. Kechida, "A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms," *Signal Processing*, vol. 190, p. 108309, 2022.