S. Krajči (ed.): ITAT 2018 Proceedings, pp. 100–107
CEUR Workshop Proceedings Vol. 2203, ISSN 1613-0073, © 2018 Oliver Keruľ-Kmec, Petr Pulc, and Martin Holeňa

Semisupervised Segmentation of UHD Video

Oliver Keruľ-Kmec[1], Petr Pulc[1,2], Martin Holeňa[2]

[1] Faculty of Information Technology, Czech Technical University, Thákurova 7, Prague, Czech Republic
[2] Institute of Computer Science, Czech Academy of Sciences, Pod vodárenskou věží 2, Prague, Czech Republic

Abstract: One of the key preprocessing tasks in information retrieveal from video is the segmentation of the scene, primarily its segmentation into foreground objects and the background. This is actually a classification task, but with the specific property that it is very time consuming and costly to obtain human-labelled training data for classifier training. That suggests to use semisupervised classifiers to this end. The presented work in progress reports the investigation of semisupervised classification methods based on cluster regularization and on fuzzy c-means in connection with the foreground / background segmentation task. To classify as many video frames as possible using only a single human-based frame, the semisupervised classification is combined with a frequently used keypoint detector based on a combination of a corner detection method with a visual descriptor method. The paper experimentally compares both methods, and for the first of them, also classifiers with different delays between the human-labelled video frame and classifier training.

1 Introduction

For the indexing of multimedial content, it is beneficial to have annotations of actors, objects or any other information that can occur in a video. A vital preprocessing task to prepare such annotations is the segmentation of the scene into foreground objects and the background.

Traditional methods, such as Gaussian mixture modeling, work on the pixel level and are time consuming on higher resolution video [1]. Another simple method models the background through image averaging, however it requires a static camera [6]. Our approach, on the other hand, is based on the level of detected interest points, and uses semi-supervised classification to assign those points as belonging either to the foreground objects or to the background.

In the next section, we introduce the key points detector we employed for the detection of points of interest. Section 3 recalls two methods of semi-supervised classification we used in our approach. The approach itself is outlined in Section 4. Finally, Section 5 presents the results of its experimental validation performed so far.

2 Scene Segmentation in the Context of Video Preprocessing

In each frame of the video, a keypoint detector is used to detect points of interest and compute their descriptors. In our research, a combination of a corner detection method FAST (Features from Accelerated Segment Test) with a visual descriptor method BRIEF (Binary Robust Independent Elementary Features) is used to this end, known as ORB (oriented FAST and rotated BRIEF) [7]. Points of interest detected in a frame are always attempted to match those detected in the next frame. Such matching points are searched in a two-step fashion:

(i) Only the points of interest in the spacial neighbourhood of the expected position are considered. That position is based on last known interest point position and its past motion (if available).

(ii) Among the points of interest resulting from (i), as well as among all detected in the current frame for which no information about their past motion is available, points in the previous frame are searched based on the Hamming distance between the descriptors of both points.

Whereas the dependence of matching success on the difference between positions of the points and on the movement of the first point has a straightforward geometric meaning, its dependence on the Hamming distance between their descriptors has a probabilistic character. In [7], this dependence was investigated and was found that if the Hamming distance between 256-bit binary descriptors of the points is greater than 64, then the probability of successful match is less than 5%.

If two points of interests in subsequent frames are considered matching, the point in the later frame is added to the history vector of the point in the previous frame. In this way, we get the motion description of each point of interest.

3 Semi-supervised Classification

Traditional supervised classification techniques use only labelled instances in the learning phase. In situations where the number of availabe labelled instances is insufficient, labelling is expensive and time consuming, semi-supervised classification can be employed, which uses both labelled and unlabelled instances for learning.

In the reported research, we used the following two methods for semisupervised classification.

3.1 Semisupervised Classification with Cluster Regularization

The principle of this method, in detail described in [8], consists in clustering all labelled and unlabelled instances and estimating, for the instance x_k, $k = 1,\dots,N$, its probability distribution q_k on the set of clusters. In addition, the following penalty function is proposed for the differences between the pairs (q_k, q_n) of probability distributions of the instances.

$$P(q_k,q_n) = \sin\left(\frac{\pi}{2}\left(r(q_k,q_n) * s(q_k,q_n)\right)^{\kappa}\right), \quad k,n = 1,\dots,N, k \neq n, \qquad (1)$$

where $r(q_k,q_n)$ denotes the Pearson correlation coefficient between q_k and q_n, κ is a parameter controlling the steepeness of the mapping from similarity to penalty, and $s(q_k,q_n)$ is a normalized similarity of the probability distributions q_k and q_n, defined

$$s(q_k,q_n) = 1 - \frac{\|q_k - q_n\| - d_{\min}}{d_{\max} - d_{\min}} \qquad (2)$$

using the notation

$$d_{\min} = \min Q,\ d_{\max} = \max Q, \quad \text{with } Q = \{\|q_k - q_n\| | k,n = 1,\dots,N, k \neq n\}. \qquad (3)$$

The results of clustering allow to assign pseudolabels to unlabelled instances. In particular, the pseudolabel assigned for the j-th among the M considered clusters to an unlabelled instance x_n in a cluster Ψ is

$$\hat{y}_{n,j} = \frac{\exp\left(\sum_{x_k \in \Psi \text{ is labelled}} y_{k,j}\right)}{\sum_{i=1}^{M} \exp\left(\sum_{x_k \in \Psi \text{ is labelled}} y_{k,i}\right)}, \qquad (4)$$

where $y_{k,i}, i = 1,\dots,M$ is a crisp or fuzzy label of the labelled instance x_k for the class i. For uniformity of notation, the symbol $\hat{y}_{k,j}, j = 1,\dots,M$ can also be used for $y_{k,j}$ if x_k is labelled.

The penalty function (1) can be used as a regularization modifier in some loss function $L : [0,1]^2 \to [0,+\infty)$ measuring the discrepancy between the classifier outputs $F(x_n) = ((F(x_n))_1, \dots, (F(x_n))_M)$ for an instance x_n, and the corresponding labels $(y_{n,1},\dots,y_{n,M})$ or pseudolabels $(\hat{y}_{n,1},\dots,\hat{y}_{n,M})$:

$$E = \frac{1}{N}\sum_{j=1}^{M}\left(\sum_{x_n \text{ labelled}} L((F(x_n))_j, y_{n,j}) + \sum_{x_n \text{ unlabelled}} \frac{\lambda \max(q_n)}{|\phi(x_n)|} \sum_{x_k \in \phi(x_n)} P(q_k,q_n) L((F(x_k))_j, \hat{y}_{k,j})\right), \qquad (5)$$

where $\lambda > 0$ is a given parameter determining the tradeoff between supervised loss and unsupervised regularization, and the set of instances $x_k \neq x_n$ with the highest value of $P(q_k,q_n)$ is denoted $\phi(x_n)$.

In [8], the following design decisions have been made for the loss function and the classifier in (5):

1. The employed loss function can be derived from $D_{\mathrm{KL}}\left((\hat{y}_{n,1}, \dots, \hat{y}_{n,M}) \| F(x_n)\right)$, the Kullback-Leibler divergence, from classifier outputs to labels or pseudolabels. If both the labels or pseudolabels and the classifier outputs form probability distributions on classes, then

$$D_{\mathrm{KL}}((\hat{y}_{n,1},\dots,\hat{y}_{n,M}) \| F(x_n)) = = \sum_{j=1}^{M} \hat{y}_{n,j} \ln\left(\frac{(F(x_n))_j}{\hat{y}_{n,j}}\right), n = 1,\dots,N. \qquad (6)$$

Therefore, the considered loss function is

$$L((F(x_k))_j, \hat{y}_{k,j}) = = \hat{y}_{n,j} \ln\left(\frac{(F(x_n))_j}{\hat{y}_{n,j}}\right), n = 1,\dots,N, j = 1,\dots,M. \qquad (7)$$

2. As a classifier, a multilayer perceptron with one hidden layer is used, such that the activation function g in its hidden layer is smooth and includes no bias, and its output layer performs the softmax normalization of the hidden layer. Hence,

$$(F(x))_j = \frac{\exp(g(w_{j\cdot}^{\top} x))}{\sum_{i=1}^{M} \exp(g(w_{i\cdot}^{\top} x))}. \qquad (8)$$

The weight vectors $w_{1\cdot}, \dots, w_{M\cdot}$ in (8) are learned through the minimization of the error function (5).

3.2 Semi-supervised Kernel-Based Fuzzy C-means

This method, in detail described in [9], originated from the fuzzy c-means clustering algorithm [2]. Similarly to the original fuzzy c-means, the method is parametrized by a parameter $m > 1$. What makes this method more general than the original fuzzy c-means, is its dependence on the choice of some kernel K, i.e., a symmetric function on pairs (x,y) of clustered vectors, which has positive semidefinite Gramm matrices (e.g., Gaussian or polynomial kernels). In fact, the fuzzy c-means algorithm corresponds to the choice $K(x,y) = x^{\top} y$.

First, the membership matrix U^l is constructed, for clustering n_l labelled instances $x_1^l,\dots,x_{n_l}^l$ into as many clusters as there are classes, i.e., M. For $j = 1,\dots,M, k = 1,\dots,n_k$,

$$U_{j,k}^{l} = \begin{cases} 1 & \text{if the instance } x_k^l \text{ is labelled with the class } j \\ 0 & \text{else.} \end{cases} \qquad (9)$$

From U^l, the initial cluster centers are constructed as

$$v_j^0 = \frac{\sum_{k=1}^{n_l} U_{j,k}^l x_k^l}{\sum_{k=1}^{n_l} U_{j,k}^l}, j = 1,\dots,M. \qquad (10)$$

If for some $t = 0,1,\dots$, the cluster centers $v_1^t,\dots,v_M^t$ are available, such as (10), then they are used together with the chosen kernel K to construct the membership matrix

$U^{u,t}$ for clustering n_u unlabelled instances $x_1^u, \dots, x_{n_u}^u$, as follows:

$$U_{j,k}^{u,t} = \frac{(1 - K(x_k^u, v_j))^{-\frac{1}{m-1}}}{\sum_{i=1}^{M} (1 - K(x_k^u, v_i))^{-\frac{1}{m-1}}}, \quad j = 1, \dots, M, \; k = 1, \dots, n_u. \tag{11}$$

Finally, the cluster centers are updated, for $t = 0, 1, ..$ by calculating

$$v_j^{t+1} = \\ = \frac{\sum_{k=1}^{n_l} (U_{j,k}^l)^m K(x_k^l, v_j^t) x_k^l + \sum_{k=1}^{n_l} (U_{j,k}^{u,t})^m K(x_k^u, v_j^t) x_k^u}{\sum_{k=1}^{n_l} (U_{j,k}^l)^m K(x_k^l, v_j^t) + \sum_{k=1}^{n_l} (U_{j,k}^{u,t})^m K(x_k^u, v_j^t)}. \tag{12}$$

The computations (11)–(12) are iterated until at least one of the following termination criteria is reached:

(i) $\|U^{u,t} - U^{u,t-1}\| < \varepsilon, t \geq 1$, for a given matrix norm $\|\cdot\|$ and a given $\varepsilon > 0$;

(ii) a given maximal number of iterations $t_{\max}$.

4 Proposed Approach

4.1 Overall Strategy

Our methodology for the segmentation of video frames into foreground objects and background relies on the assumption that the user typically assigns corresponding labels to points of interest only in the first frame, and even not necessarily to all detected points of interest.

No matter whether the considered method of semisupervised classification is semisupervised classification with cluster regularization or semi-supervised kernel-based fuzzy c-means, the methodology always proceeds in the following steps:

1. In the first frame, the user labels some of the points of interest detected by the ORB detector.

2. Using the considered method of semisupervised classification, the remaining detected points of interest are labelled.

3. Matching points detected in the next frame are assigned the same labels as the points to which they are matched.

4. Using the considered method of semisupervised classification, the remaining points of interest detected in the next frame are labelled.

5. Steps 3 and 4 are repeated till either the points of interest in all frames have been classified or the scene has been so much disrupted between two frames that no points of interest could be matched between them (in such a case, new labelling by the user is needed).

4.2 Implementation of Object Segmentation

The Cartesian coordinates $([p]_1, [p]_2)$ of a point p of interest are expressed with respect to top left corner of the frame, using as units the frame height and width. Due to that, $[p]_1$ and $[p]_2$ are normalized to $[0,1]$.

For a match between points of interest p_k and p_{k+1} in subsequent frames k and $k+1$, the following criteria have been used:

(i) The point p_{k+1} must lie within the radius r_k^p from the estimated new position of the point $\hat{p}_k$

$$\|p_{k+1} - \hat{p}_k\| < r_k^p. \tag{13}$$

Here, the estimated position $\hat{p}_k$ is calculated as

$$\hat{p}_k = \begin{cases} p_k + c_1(p_k - p_{k-1}) & \text{if } p_{k-1} \text{ is available,} \\ p_k & \text{else,} \end{cases} \tag{14}$$

where $c_1 > 0$, and the radius r_k^p is calculated as

$$r_k^p = (u_k^p W)^2, \tag{15}$$

where u_k^p quantifies the uncertainty pertaining to the point p_k in the k-th frame and W denotes the frame width (in the units in which point positions are expressed). The uncertainty u^p is set to $u_1^p = c_2 > 0$ in the first frame and is then evolved from frame to frame through linear scaling above a lower limit $c_3 > 0$:

$$u_{k+1}^p = \begin{cases} \max(c_3, c_4 u_k^p) & \text{if } p_k \text{ is matched,} \\ c_5 u_k^p & \text{if } p_k \text{ is not matched,} \end{cases} \tag{16}$$

where $0 < c_4 < 1, c_5 > 1$.

Moreover, if the evolution (16) leads to $u_{k+1}^p > c_6$ for some $c_6 > c_3$, then the point p is deactivated and not any more considered for matching.

(ii) Hamming distance between the 256-bit binary desciptors of the points is at most 64.

The choice of the real-valued constants in the criterion (i) has been based on the resolution of the video (4K), on the frame rate (25) and on the defaults in the ORB implementation based on [7]. They have been set to the following values: $c_1 = 0.6, c_2 = 0.02, c_3 = 0.009, c_4 = 0.9, c_5 = 1.1, c_6 = 0.03$.

In each frame, the described implementation was used to find 500 most interesting points. On a linux computer with a 3.3 GHz Intel Xeon E3-1230 processor, this took 95.32 ms.

4.3 Implementation of Semi-supervised Classifiers

As input features for both classification methods, the Cartesian coordinates $([p_k]_1, [p_k]_2)$ of the point in the k-th frame and and the polar coordinates $([p_k - p_{k-1}]_{||}, [p_{k+1} -$

$p_k]_\varphi$) of its movement with respect to the previous frame are used.

In the implementation of the semisupervised classification with cluster regularization method described in 3.1, we used k-means clustering for an initial clustering of all instances. Although this method allows choosing the number of clusters independently of the number of classes, we have set it to the same value for comparability with semi-supervised kernel-based fuzzy c-means, i.e., to the value 2 corresponding to the classes of foreground objects and background. Hence, we performed k-means clustering with $k = 2$. Since the k-means algorithm does not output a probability distribution on the set of clusters, we employed a simple procedure proposed in [8] to transform the original distances from an instance x_n to cluster centers $v_1, \dots, v_k$, to a probability distribution q_n, which assures that x_n more likely belongs to clusters to which centers it is closer:

$$(q_n)_i = \frac{1 - \left(\frac{\|x_n - v_i\|}{\sum_{j=1}^{k} \|x_n - v_i\|} \right)}{k-1}. \tag{17}$$

Consequently, for our case $k = 2$:

$$(q_n)_1 = \frac{\|x_n - v_2\|}{\|x_n - v_1\| + \|x_n - v_2\|}, \tag{18}$$

$$(q_n)_2 = \frac{\|x_n - v_1\|}{\|x_n - v_1\| + \|x_n - v_2\|}. \tag{19}$$

The remaining parameters pertaining to semisupervised classification with cluster regularization were set as proposed in [8]: $\lambda = 0.2, \kappa = 2, |\phi(x_n)| = 10$.

For the semi-supervised kernel-based fuzzy c-means algorithm described in 3.2, we used a Gaussian kernel function for updating the membership matrix $K(x,y) = exp(-\|x-y\|^2/\sigma^2)$, where the parameter σ is computed as proposed in [9]:

$$\sigma = \frac{1}{M} \sqrt{\frac{\sum_{n=1}^{N} \|x_n - v\|^2}{N}}, \tag{20}$$

where v is the center of all instances. The remaining parameters were set as follows: $m = 2, \varepsilon = 0.001, t_{max} = 50$.

5 Experimental Validation

5.1 Employed Data

For the validation of the proposed approach we prepared 12 short videos. In all videos, there is a yellow or blue balloon as a foreground object and a green background. On the background, there are a few small red sticky notes to help detecting some key points. The videos were recorded in a UHD resolution.

Here is a brief characterization of all employed videos:

- a handheld camera, both the foreground object and the background are sharp,
- a handheld camera, only the foreground object is sharp (2 videos),
- a static camera, only the background is sharp (2 videos),
- a static camera, only the background is sharp, the foreground object is close to the camera,
- a static camera, only the foreground object is sharp, a hand is interfering with the background (2 videos),
- a static camera, only the foreground object is sharp, it is moving towards the camera,
- a static camera, only the foreground object is sharp, it is moving away from the camera,
- static camera, only the foreground object is sharp, it passes the scene multiple times (2 videos).

For the testing, labels were available for all points of interest. Unfortunately, those labels were often unreliable.

5.2 Results and Their Analysis

On all the employed videos, we measured the quality of classification by means of accuracy, sensitivity, specificity and F-measure of both implemented classification methods.

For the fuzzy c-means method, the accuracy and specificity on the unlabelled data are illustrated for four particular videos in Figure 1.

For the cluster-regularization method, we compared the values of the considered four quality meaures obtained with five classifiers trained in each of the five first video frames with respect to the delay between classifier training and measuring its quality. The results of their comparison are for three particular delays, 1 frame, 5 frames and 10 frames, summarized in Table 1. In addition, for delays up to 50 frames, they are again illustrated for accuracy and sensitivity on the four videos used already in connection with the fuzzy c-means classifier, in Figures 2–5.

The figures (2)–(5) indicate that classifiers trained in a later frame tend to have higher accuracy and specificity, but in general, the differences between classifiers trained in different frames are small. This is confirmed by the Friedman test for delays 1, 5 and 10 frames between classifier training and measuring its quality and for all four considered quality measures. The hypothesis of equality of all five classifiers is rejected (p-value $< 5\%$) only for the delay 1 frame and the F-measure, and weakly rejected (p-value $< 10\%$) for the delay 1 frame and the sensitivity, as well as for the delay 5 frames and the F-measure. A posthoc test expectedly reveals that the equality of all five classifiers was rejected mainly due to differences between classifiers trained in the early and in later frames; in particular between those trained in the 1st and 4th frame (delay 1, both sensitivity and F-measure), classifiers trained

Table 1: Comparison of the values of the considered quality measures obtained with classifiers trained in each of the 5 first video frames for different delays between classifier training and testing, obtained on data from the 12 employed videos. The result in a cell of the table indicates on how many videos the considered measure of classifier quality (accuracy, sensitivity, specificity, F-measure) was higher for the row classifier : on how many videos it was higher for the column classifier. A result in italic, respectively bold italic, indicates that after the Friedman test at least weakly rejected (p-value < 10%) the hypothesis that the considered quality measure is equal for all classifiers (cf. Table 2), the post-hoc test according to [3, 4] weakly rejects, respectively rejects (p-value < 5%) the hypothesis that it is equal for the particular row and column classifiers. All simultanously tested hypotheses were corrected in accordance with Holm [5]

	Delay between the frame on which the classifier is trained and the frame when it is tested														
	1 frame					5 frames					10 frames				
# 1	Frame in which the compared classifier was trained # 2														
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
	Accuracy														
1		5:7	5:7	7:5	6:6		4:8	7:5	7:5	10:2		4:8	4:8	4:8	2:10
2	7:5		4:8	8:4	6:6	8:4		6:6	7:5	10:2	8:4		7:5	5:7	5:7
3	7:5	8:4		9:3	6:6	5:7	6:6		7:5	11:1	8:4	5:7		6:6	5:7
4	5:7	4:8	3:9		5:7	5:7	5:7	5:7		10:2	8:4	7:5	6:6		5:7
5	6:6	6:6	6:6	7:5		2:10	2:10	1:11	2:10		10:2	7:5	7:5	7:5	
	Sensitivity														
1		8:4	8.5:3.5	*9.5:2.5*	8.5:3.5		8:4	7:5	7:5	9:3		6.5:5.5	8:4	8.5:3.5	8.5:3.5
2	4:8		8:4	10:2	9:3	4:8		6:6	7.5:4.5	9.5:2.5	5.5:6.5		6:6	8:4	8:4
3	3.5:8.5	4:8		10.5:1.5	9.5:2.5	5:7	6:6		7.5:4.5	8.5:3.5	4:8	6:6		8.5:3.5	9.5:2.5
4	*2.5:9.5*	2:10	1.5:10.5		6.5:5.5	5:7	4.5:7.5	4.5:7.5		8:4	3.5:8.5	4:8	3.5:8.5		8.5:3.5
5	3.5:8.5	3:9	2.5:9.5	5.5:6.5		3:9	2.5:9.5	3.5:8.5	4:8		3.5:8.5	4:8	2.5:9.5	3.5:8.5	
	Specificity														
1		7.5:4.5	6.5:5.5	7:5	7:5		3.5:8.5	5:7	5:7	4:8		6.5:5.5	3.5:8.5	2:10	3.5:8.5
2	4.5:7.5		4.5:7.5	6:6	6.5:5.5	8.5:3.5		5:7	4.5:7.5	4:8	5.5:6.5		4:8	4:8	4:8
3	5.5:6.5	7.5:4.5		6:6	7:5	7:5	7:5		7:5	6.5:5.5	8.5:3.5	8:4		4.5:7.5	6:6
4	5:7	6:6	6:6		4.5:7.5	7:5	7.5:4.5	5:7		4:8	10:2	8:4	7.5:4.5		6.5:5.5
5	5:7	5.5:6.5	5:7	7.5:4.5		8:4	8:4	5.5:6.5	8:4		8.5:3.5	8:4	6:6	5.5:6.5	
	F-measure														
1		8:4	9:3	***10:2***	8:4		6:6	7:5	8:4	***11:1***		5.5:6.5	9:3	8.5:3.5	9.5:2.5
2	4:8		7:5	***12:0***	9:3	6:6		6.5:5.5	7:5	*10:2*	6.5:5.5		6.5:5.5	7.5:4.5	9.5:2.5
3	3:9	5:7		11:1	8:4	5:7	5.5:6.5		8:4	*11:1*	3:9	5.5:6.5		8:4	9:3
4	***2:10***	***0:12***	1:11		6:6	4:8	5:7	4:8		8.5:3.5	3.5:8.5	4.5:7.5	4:8		9:3
5	4:8	3:9	4:8	6:6		***1:11***	*2:10*	*1:11*	3.5:8.5		2.5:9.5	2.5:9.5	3:9	3:9	

Table 2: Results of the Friedman test of the hypothesis that for a given delay between classifier training and measuring its quality, a given quality measure is equal for the classifiers trained in each of the 5 first video frames, for the 12 combinations of delays and quality measures considered in Table 1. The combinations for which the tested hypotheseis was weakly rejected (p-value $< 10\%$) are in italic, the single combination for which it was rejected (p-value $< 5\%$) is in bold italic. All simultanously tested hypotheses were corrected in accordance with Holm [5]

Quality measure	Delay	p-Value
accuracy	1	1
accuracy	5	0.117
accuracy	10	1
sensitivity	1	*0.052*
sensitivity	5	0.428
sensitivity	10	0.238
specificity	1	1
specificity	5	1
specificity	10	0.25
F-measure	1	***0.043***
F-measure	5	*0.089*
F-measure	10	0.238

in the 1st and 4th frame (delay 1, F-measure) and classifiers trained in the 1-3 frame and in the 5th frame (delay 5, F-measure).

6 Conclusion

The presented research integrates two comparatively recent approaches, the keypoint detector ORB, which is a combination of a corner detection method with a visual descriptor method, and two semi-supervised classification methods. To our knowledge, this is the first time these approaches are used together for the task of scene segmentation into the foreground objects and the background.

On the other hand, this is a work in progress and the presented results are still rather preliminary, being obtained on 12 artificially created videos with a quite simple scene segmentation. Both approaches should be investigated in the context of more complex segmentations and more realistic scenes. To this end, however, especially the ORB detector needs to be more deeply elaborated with methods of semisupervised classification.

Acknowledgement

The research reported in this paper has been supported by the Czech Science Foundation (GAČR) grant 18-18080S.

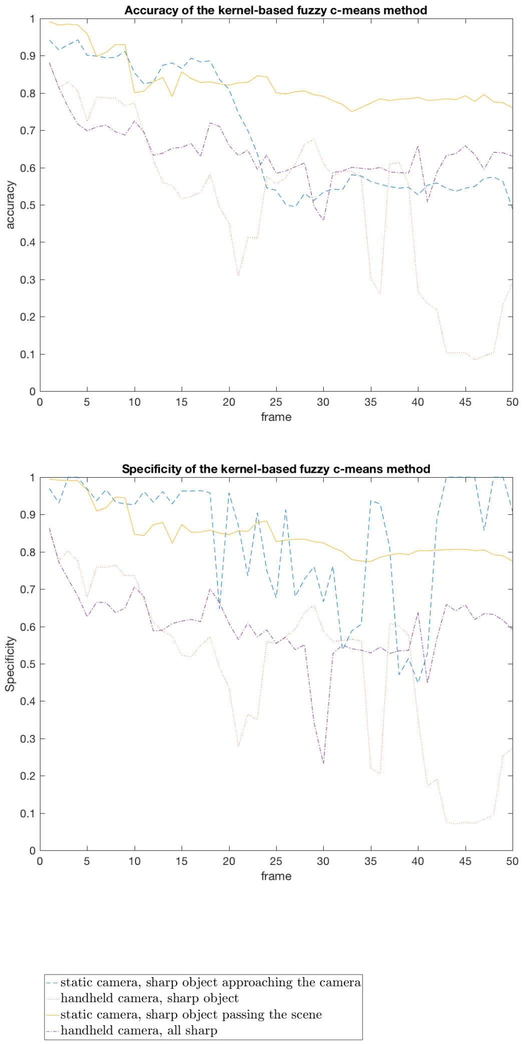

Figure 1: The evolution of accuracy (top) and specificity (bottom) of the c-means method on the unlabelled data for four particular videos

References

[1] M.S. Allili, N. Bouguila, and D. Ziou. Finite general Gaussian mixture modeling and application to image and video foreground segmentation. *Journal of Electronic Imaging*, 17:paper 013005, 2008.

[2] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[3] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[4] S. Garcia and F. Herrera. An extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

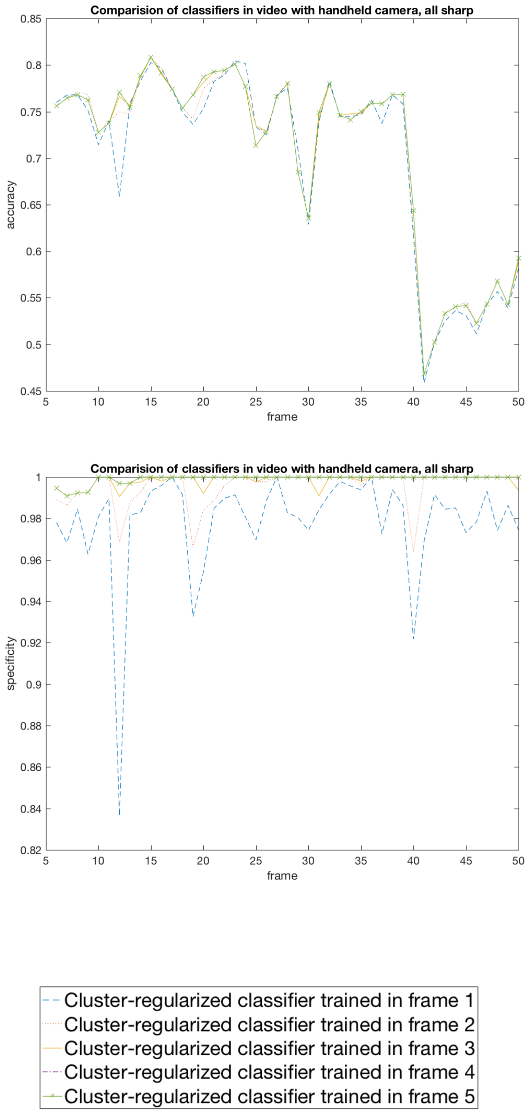

Figure 2: The evolution of accuracy (top) and specificity (bottom) of the classifiers trained in each of the 5 first video frames for a handheld-camera video with both the foreground object and the background sharp

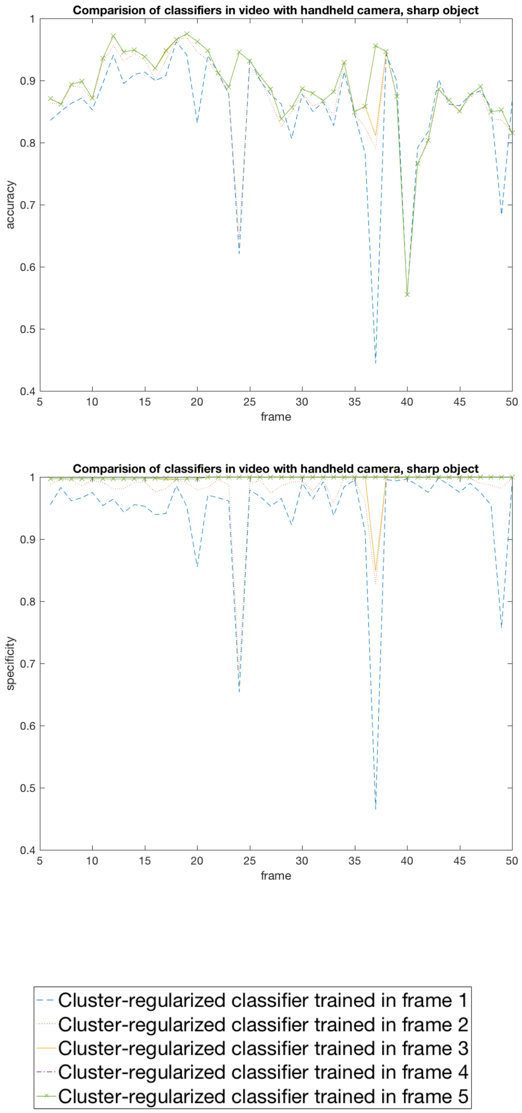

Figure 3: The evolution of accuracy (top) and specificity (bottom) of the classifiers trained in each of the 5 first video frames for a handheld-camera video with only the foreground object sharp

[5] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[6] L. Li, W. Huang, I.Y.H. Gu, and Q. Tan. Foreground object detection from videos containing complex background. In *11th ACM Conference on Multimedia*, pages 2–10, 2003.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision*, pages 2564–2571, 2011.

[8] R.G.F. Soares, H. Chen, and X. Yao. Semisupervised classification with cluster regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 23:1779–1792, 2012.

[9] D. Zhang, K. Tan, and S. Chen. Semi-supervised kernel-based fuzzy c-means. In *ICONIP'04*, pages 1229–1234. Springer, 2004.

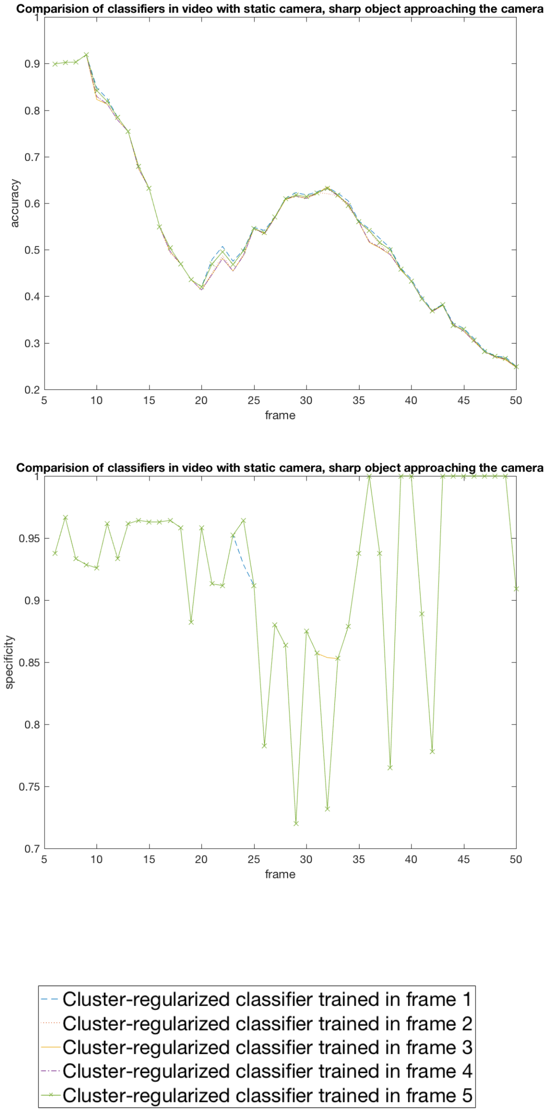

Figure 4: The evolution of accuracy (top) and specificity (bottom) of the classifiers trained in each of the 5 first video frames for a static-camera video, in which only the foreground object is sharp and is moving towards the camera

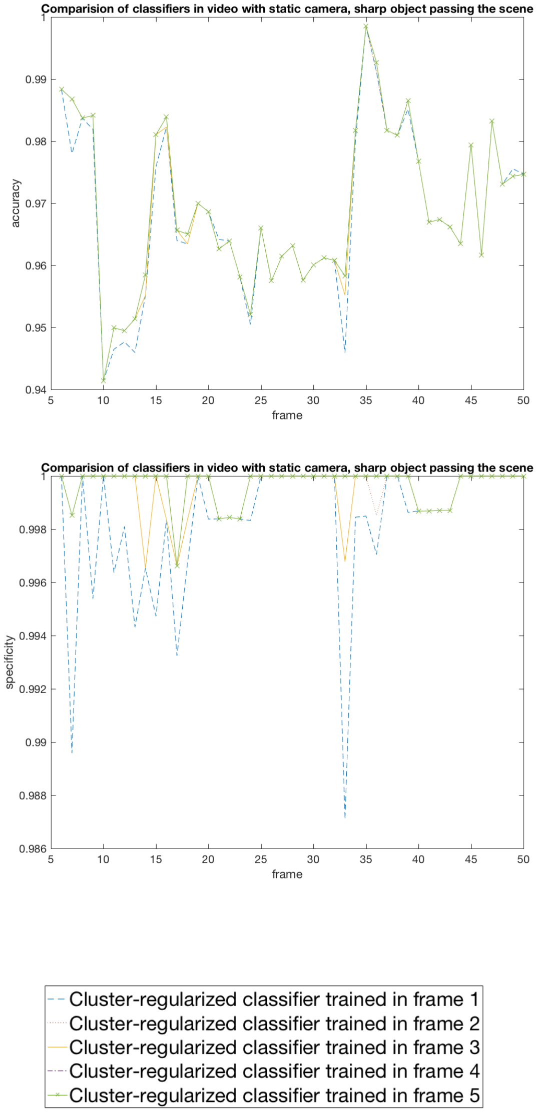

Figure 5: The evolution of accuracy (top) and specificity (bottom) of the classifiers trained in each of the 5 first video frames for a static-camera video, in which only the foreground object is sharp and passes the scene multiple time