Recognition of sport players' numbers using fast color segmentation

Cédric Verleysen* and Christophe De Vleeschouwer

ICTEAM institute, Université catholique de Louvain, Louvain-la-Neuve, Belgium

ABSTRACT

This paper builds on a prior work for player detection, and proposes an efficient and effective method to distinguish among players based on the numbers printed on their jerseys. To extract the numbers, the dominant colors of the jersey are learnt during an initial phase and used to speed up the segmentation of the candidate digit regions. An additional set of criteria, considering the relative position and size (compared to the player bounding box) and the density (compared to the digit rectangular support) of the digit, are used to filter out the regions that obviously do not correspond to a digit. Once the plausible digit regions have been extracted, their recognition is based on feature-based classification. A number of original features are proposed to increase the robustness against digit appearance changes, resulting from the font thickness variability and from the deformations of the jersey during the game. Finally, the efficiency and the effectiveness of the proposed method are demonstrated on a real-life basketball dataset. It shows that the proposed segmentation runs about ten times faster than the mean-shift algorithm, but also outlines that the proposed additional features significantly increase the digit recognition accuracy. Despite significant deformations, 40% of the samples, that can be visually recognized as digits, are well classified as numbers. Out of these classified samples, more than 80% of them are correctly recognized. Besides, more than 95% of the samples, that are not numbers, are correctly identified as non-numbers.

Keywords: Color segmentation, k-means, digit recognition, feature-based classification, SVM.

1. INTRODUCTION

In recent years, digit and character recognition have become very popular fields in image processing. Among the principal applications, one could think about car plate recognition[1], automatic sorting of post letters[2], digital archiving of books[3], etc. Automatic character recognition (OCR) is also a powerful tool that gives to the computers the ability to autonomously recognize objects that have been labelled in a structured manner. Such interpretation is for example required for the autonomous and low-cost production of personalized summaries of sport events[4,5]. In such context, the recognition of the digits printed on the players' jerseys appears to complete conventionnal people detection and tracking algorithms to understand the scene. In particular, our paper builds on earlier works[6,7] to detect the players, and proposes an efficient approach to identify them based on the segmentation and recognition of their jerseys' number.

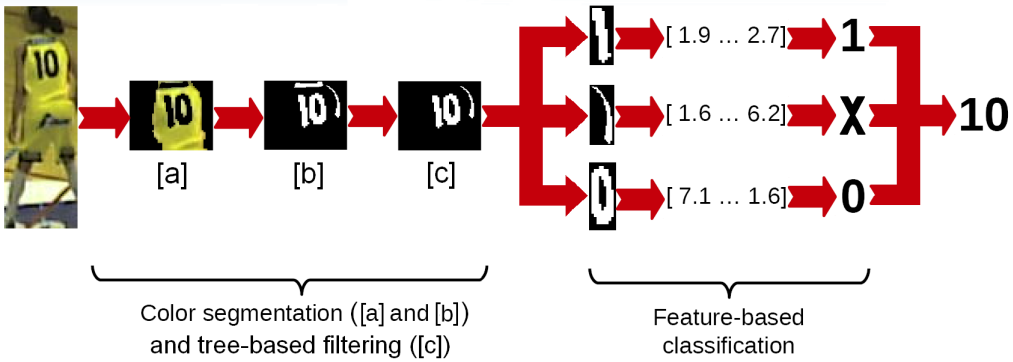

Figure 1. Digit recognition is divided into two stages: the digit extraction and its classification.

* cedric.verleysen@uclouvain.be

As illustrated in Fig.1, the proposed method can be divided into two main successive steps:

- Extraction of the regions that likely correspond to the digits printed on players' jersey;
- Automatic recognition of the candidate digit regions using a feature-based classifier.

The digits extraction step first relies on an autonomous learning of the dominant colors composing the players' jerseys in order to speed up a color segmentation algorithm. Once the silhouette of the player is segmented into a set of non-overlapping regions, a classification tree with relaxed contraints is applied to distinguish plausible digit regions out of the other segmented regions. More precisely, apriori information about the relative size of the digit (compared to the jersey bounding box size) and the pixel density of the digit (within its rectangular support) is exploited to discriminate between regions that are likely to correspond to a digit and the other ones. Additionally, the digit color can be estimated to be the dominant color appearing among those plausible digit regions, so that the knowledge of this color can be used to segment plausible digit candidates in subsequent frames. Once the plausible digit regions have been extracted, they are recognized based on a Support Vector Machine (SVM) classifier. Original features are introduced to make the classifier robust to changes in digit appearance, induced by both the jersey deformations during the game and the font thickness variability from one team to another.

This paper is organized as follows. Section 2 presents the general principle of the candidate digit extraction method. The rejection or acceptance of the plausible digit regions, and the definition of the related constraints, are explained in Section 3. Finally, the performances of the integrated approach are presented in Section 4. It first shows that the computation time of the proposed color segmentation is significantly smaller than the one of conventional segmentation methods working without color apriori, such as the mean-shift algorithm[8], while preserving the segmentation effectiveness. It then investigates how the proposed features improve both the performance and the robustness of the classifier to digit appearance changes.

2. DIGIT EXTRACTION

This section assumes that the bounding boxes of the players have been determined, e.g based on [6] or [7], and explains how to extract the digit printed on the jersey of the bounded player. Based on the foreground mask of the players, the proposed approach combines a learning of the jerseys' colors with a thresholding process to segment the jersey into regions of approximatively uniform color. It then uses some apriori knowledge about the digit to differentiate it from other parts of the jersey.

More precisely, since the jersey and the digit cover a significant fraction of the upper part of the player bounding box, those two uniform colors should appear as dominant modes of the upper bounding box color histogram. Hence, the dominant histogram modes provide useful apriori information to segment the player's jersey into its two main components. To decide which component actually corresponds to the digit, we can rely on the knowledge we have about the usual shape and position of the digit within the player bounding box. In practice, a set of shape and position criteria can be defined to select the regions that likely correspond to the digit region. The decision taken based on those criteria is error-prone. However, in average on a large number of samples, it is sufficiently reliable to define the digit color to be the dominant color observed within those selected regions. This color can then be exploited for direct digit segmentation by color thresholding.

The approach is generic in that it relies on the availability of complementary apriori information about the shape appearance of the region of interest. As an example, the method could as well be used to learn the color distribution of a ball by aggregating the histograms extracted from the moving foreground regions that are circular and have a size that is consistent with the ball size[9].

The remaining of this section first explains how dominant histogram modes are learnt and exploited for color-based segmentation. Then, the list of the shape and position criteria that are used to extract plausible digit regions out of the color-segmented regions is given.

2.1 Color segmentation and autonomous learning of the jerseys' dominant colors

The proposed approach computes the dominant colors of the player's jersey based on a recursive learning process. The learning set consists in a set of image samples corresponding to the upper part of the bounding box of the player. Each step of the learning process identifies one or two additional colors that appear frequently on the jersey. It works as follows. Let C_n denote the set of dominant colors identified among image samples, during the first n iterations of the recursive process. C_0 is defined to be empty. A pixel is said to be active at iteration n if its $\mathcal{L}^1$ distance to any color in C_{n-1} is larger than a threshold. The histogram associated to all active pixels of an image sample is then considered. In order to reduce both the memory requirements and computationnal cost of this detection, color components (RGB in our case) are processed separately. More precisely, if $h^n_{i,s}(j)$ is defined as the j^{th} bin of the histogram of the i^{th} color component of the s^{th} sample at the n^{th} step, we determine the first mode $f^n_{i,s}$ of the histogram $h^n_{i,s}$ as the value j that occurs most frequently in this histogram. Thus, the dominant color is denoted $\mathbf{f}^n_s$ and is given by $\mathbf{f}^n_s = [f^n_{1,s},\ f^n_{2,s},\ f^n_{3,s}]$. The dominant color $\mathbf{f}^n_s$ accumulated for the S image samples at iteration n are then stored in the 3D color space, and we use k-means clustering[10] to separate the space into 2 clusters. Two clusters are determined because there are two teams on the field with different jerseys. Hence, the 2-means clustering enables to learn the dominant color of each team. In the subsequent steps of the recursive learning process, the pixels associated with the dominant colors are ignored, and a 2-means clustering is applied on the residual histogram. The recursive learning process is repeated until the number of active pixels becomes insignificant or until a predefined number of dominant colors have been defined. In our case, because we assume that the color of the digit is one of the two dominant colors of the jersey, we adopt the second stopping criterion and only 4 colors (2 per team) have to be defined.

Once the main dominant colors have been learnt, color thresholding followed by 4-connectivity stack based flooding can be implemented to segment the jersey into regions of uniform color and connected pixels. Any (weak) classifier that is able to discriminate between digit and other regions (based on other features than color, e.g. shape, position, etc) might then be considered to partition the region into two sets (not digit and possibly digit), and to select the digit color as the dominant color that appears most often in the "possibly digit" category. The next section introduces the list of criteria that have been investigated to select plausible digit regions among the segmented ones.

In conclusion, this initial stage learns two sets of features for a given team, namely the color of its jersey (that corresponds to the team color) and the color of its associated digit. After the learning stage, digit color thresholding is used to rapidly segment plausible digit regions. More precisely, a pixel of the image is considered as being part of a number if both its $\mathcal{L}^1$ distance with the jersey color is higher than a first threshold and its $\mathcal{L}^1$ distance with the respective digit color is lower than a second threshold. However, as illustrated on Fig.2, it could happend that some regions (e.g. name of the player, advertisements, ...) have a similiar color than the number. Additional criterions than color should thus be considered to identify plausible digits among the regions that have the same color as the digits.

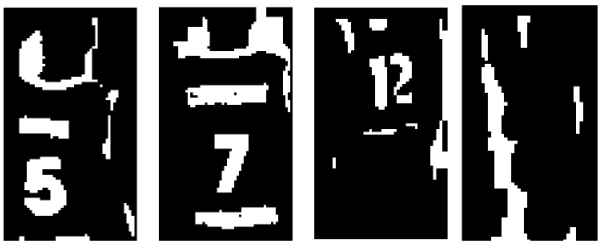

Figure 2. Other regions of similar color than the numbers are extracted by color segmentation and should be filtered out.

2.2 Digit extraction with a relaxed tree-based filtering

To differentiate plausible digit regions from the regions extracted based on color segmentation, a 4-connectivity stack-based flooding algorithm is applied on the resulting image to segment fastly the different extracted regions. As illustrated on Fig.3, each region goes independently through a tree that filters out non-numbers candidates based on weak apriori information about the numbers.

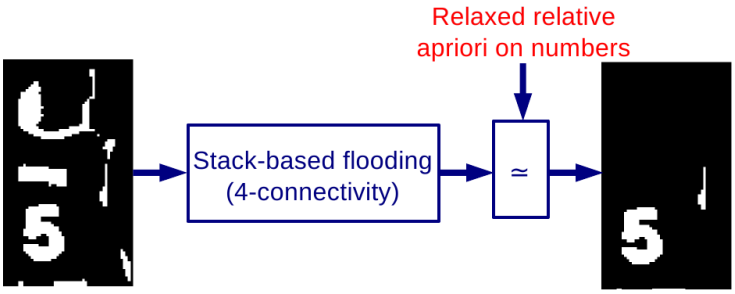

Figure 3. A tree-based filtering rejects the non-number regions using some weak apriori informations about numbers.

More precisely, both the candidate region and the isolated jersey are first described by their rectangular supports (defined by their extremity points respectively). After, these rectangular supports are compared to reject the candidate region if it does not satisfy one of the following criteria:

- The number width should fit inside $[15\%,\ 50\%]$ of the width of the jersey;
- The number height should fit inside $[15\%,\ 30\%]$ of the height of the jersey;
- The number density (amount of pixels representing the number divided by the area of the candidate region rectangular support) should fit inside the range $[35\%,\ 80\%]$.

3. DIGIT LABELIZATION

In order to recognize the plausible digit regions extracted in Section 2, a supervised machine learning technique is trained on manually labelized region samples to establish a (non)-linear parametric regression model that assigns a given test region to a specific digit or to a bin class. A multi-class Support Vector Machine (SVM) has been considered to build such a parametric model, as recommended and implemented by the LIBSVM library[11].
The training set mixes non-digit (to define a bin class) and digit samples. Because the jersey of a sport player is expected to rotate, to fold, to bend and to stretch, the training set has to integrate such kind of digit distortions to make the classification robust. Moreover, although each digit is generally extracted separately during the stack-based flooding (done before the relaxed tree-based filtering), the folds of the jerseys could imply an overlap of the multiple digits encountered in two digits numbers, which prevents the separation of the digits during the flooding. For these reasons, we generate a training database of distorted (rotation and shearing operations) digits going from 0 to 9, and complete it by overlapping and distorted numbers going from 10 to 99.

Both for the training and testing (on-the-fly recognition) phase, each candidate sample is described thanks to a 39-dimensional vector composed of 7 conventional and 32 original features. The conventional features are similar to the ones exploited in [7], namely:

- The ratio between the height and the width of the candidate sample;
- The number of holes in the candidate sample (using the fast computation of the Euler number of [12]);

After a resizing of the candidate to a 24×24 binary mask, the followed features are also extracted:

- The 2 centers of gravity (normalized spatial moments);
- Three values corresponding to second order moments m_{20}, m_{02} and m_{22}.

A set of additional original features have been introduced, both to increase the performance and the robustness of the recognition system. Those features are inspired from the horizontal and vertical projections of [7], but differ from previous works in two aspects. First, projections are computed independently on each of the quarters composing the digit. This better discriminates symetric patterns, like the ones composing a 2 and a 5. Second, two so-called *pseudo-skeletons* (one horizontal and the other vertical) of the digit are computed before projections, so as to be more robust to font thickness variations. As illustrated on Fig.4, the horizontal (vertical) *pseudo-skeleton* is defined as the superposition of the center of each horizontal (vertical) line segment composing the digit region. Totally, we accumulate the values of the six first bins of a quarter (vertical or horizontal) image projection into one feature and do the same for the six last bins. The feature vector is thus completed with 2 (the horizontal and vertical *pseudo-skeletons*) $\times$ 4 (the quarters of each *pseudo-skeleton*) $\times$ 2 (the horizontal and vertical projections) $\times$ 2 (two features per projection) $=$ 32 features.

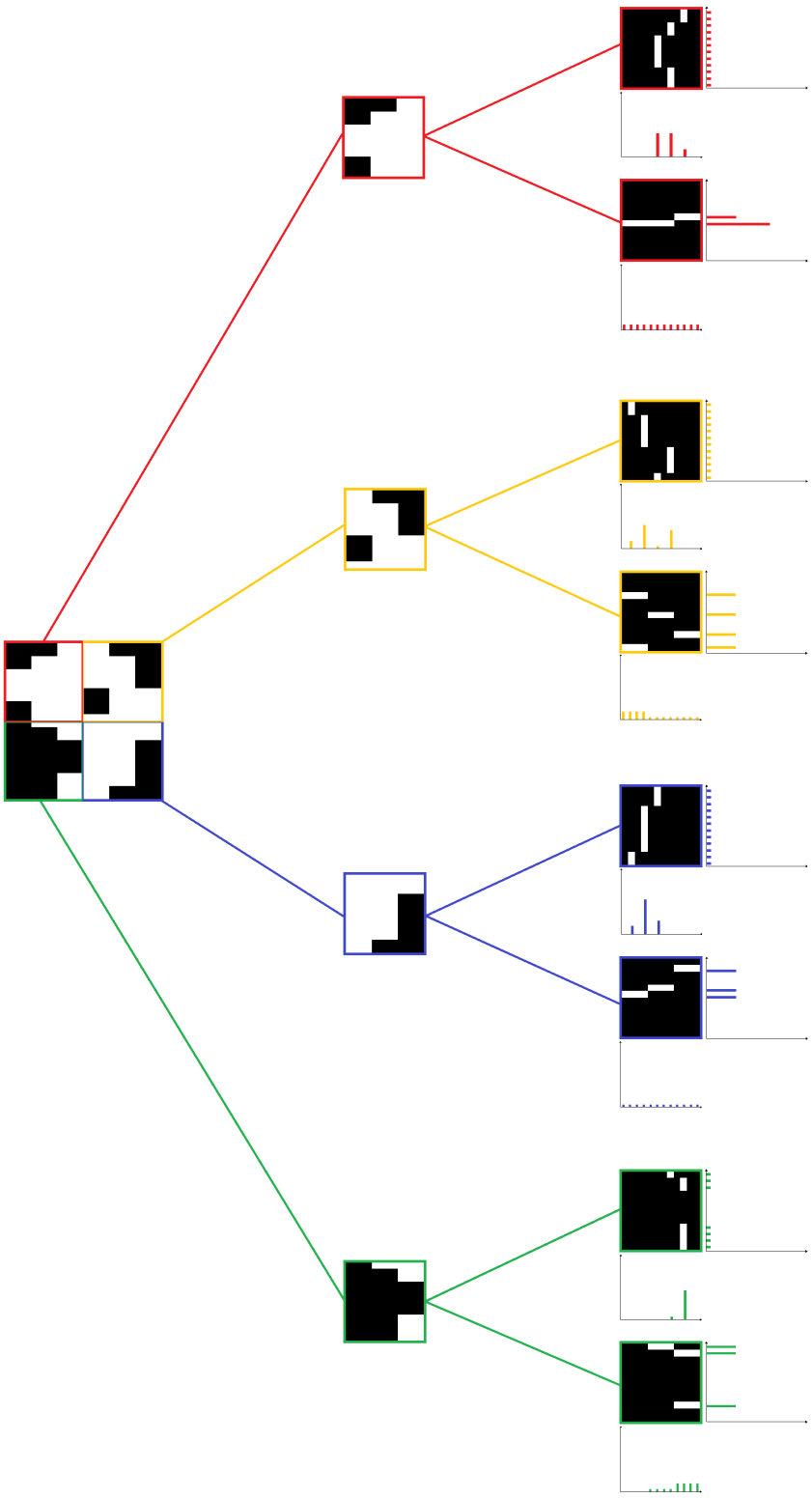

Figure 4. Each quarter of a candidate number is described thanks to two *pseudo-skeletons*. One represents the centers of the horizontal segments, the other the centers of vertical segments. For both kinds of *pseudo-skeletons*, the vertical and horizontal projections are computed. Finally, the accumulation of 6 successive bins represents a feature.

4. VALIDATION

In this section, both the efficiency and the effectiveness of the proposed method are evaluated. Two different test sets are considered to assess our approach. The first one consists of approximatively 100000 synthetic data generated by deforming reference number fonts. These deformations are obtained by applying a pseudo-random set of affine transforms followed by projections, as explained in Section 3. The second test set is composed of 3700 bounding boxes of players, extracted from a real basketball dataset. Specifically, we consider a scenario in which a basketball field is covered by a distributed set of cameras. The fusion of the foreground masks computed in each view is considered to detect the players on the ground field, as detailed in [6]. In the following, those two test sets are respectively named *synthetic dataset* and *real-life dataset*.

The remainder of this section is organized as follows. In Section 4.1, the segmentation of the candidate numbers is investigated using the *real-life dataset*. For the selection of plausible digit regions out of the segmented regions, the tree-based filtering is analyzed in Section 4.2. The performance of the SVM classifier are then investigated in Section 4.3, and a special attention is given to the choice of the training set and to the features selection. Finally, Section 4.4 summarizes the performances of the overall process and propose some paths to solve some of the weaknesses of our system.

4.1 Segmentation performances

Since the segmentation of the digit directly depends on the estimation of the jersey color, we first investigate the performance of correct team recognition. Then, we compare the proposed segmentation with the conventionnal mean-shift algorithm. Finally, it is shown that our approach is more than ten times faster than the mean-shift segmentation, while preserving the segmentation effectiveness.

Our experiments have revealed that, if the color of the jersey (and thus the team) is well recognized, the number represented on it is almost always perfectly segmented. The few inaccurate segmentations come from important deformations of the jersey, implying either some shadows (that change the color appearance) or an overlap of regions with similar colors. However, because these two extreme cases do not appear often, the absolute accurary of the proposed segmentation can roughly be estimated as the team recognition accuracy.

In Table 1, we present the team recognition performance on the *real-life dataset* in the single and multiple view scenarios. In the single view case, a team is recognized for each bounding box extracted from one of the views covering the game field. In contrast, the multiview case adopts a majority vote strategy to decide about the team of the player, based on the multiple individual decisions taken in each view.

Correct recognition of the teams	
Single view	92.2%
Majority vote on multiple views	97.8%

Table 1. A high team recognition performance is a necessary condition to get an accurate digit segmentation.

A deeper analysis of the erroneous team recognitions reveals that they are either induced by occlusion of players of different teams, or by the influence of the pixels of the bounding box that correspond to the players' skin. Specifically, because some bounding boxes represent the sideview of a player, it happens that more foreground pixels represent the arm of the player than the jersey of the player. In this case, the dominant color of the image is not anymore the one of his/her jersey, but the one of his/her skin. If the skin color of this player is similar to the jersey color of the opponent team, a wrong team decision is taken.

In order to compare the efficiency of the proposed segmentation approach with the mean-shift algorithm, the entire system (segmentation and digit recognition) has been implemented in C and optimized using the Integrated Performance Primitives library (IPP). For computationnal purpose and because of the approximatively homogenous colors and huge contrast that compose a jersey, the research of the dominant modes among the bins of a color component histogram is limited in this validation to the first two bins having the maximum peak values. Moreover, to segment the candidate digits, we have considered that the jersey is located in the upper 30% to 90% of the player bounding box height. Also, the $\mathcal{L}^1$ distance threshold used to compare the pixels

with the learnt color of the jersey is set to 150 and the one linked with the learnt color of the number is set to 120. Finally, both the computation time (in number of processed bounding boxes per second, refered as BB/s) presented in Table 2 have been evaluated on the same video and have been obtained with an Intel I7 processor of 3GHz and 8Gb of RAM.

	Average number of BB/s	Worst number of BB/s
Mean-shift	192	56
Proposed segmentation	2007	1103

Table 2. The entire process runs 10 times faster with our segmentation than with a conventionnal mean-shift algorithm.

This table proves that the proposed segmentation runs more than 10 times faster than a mean-shift algorithm, while Table 7 also shows that both the segmentation effectiveness are approximatively identical.

Hence, we can conclude that, as long as there is no occlusion between the players and that there are not too many skin pixels in the player bounding box, our method effectively extracts the digits, while segmenting more than 10 times faster than what the mean-shift algorithm does.

4.2 Tree-based filtering performances

Once the candidate digit regions have been extracted, the tree-based filtering enables to speed up the process by rejecting the regions that obviously do not correspond to digits. The method and associated parameters have been introduced in Section 2.2. Performances obtained on the *real-life dataset* are presented in Table 3.

Tree-based filtering performances		
(a)	Non number classified as number	45.4%
(b)	Number classified as number	91.9%

Table 3. The rejection of non-number regions speeds up the process, but impairs the performance due to the rejection of some of the true number regions.

These performances confirm the effectiveness of the tree-based filtering: almost all the numbers go through this tree but a significant amount of non-numbers are discarded. A deeper analysis of the number regions that are rejected by the tree shows that almost all the discarded numbers are connected (by deformation of the jersey) with other regions of the same color. It means that, even if the connected regions should be accepted by the tree-based filtering, it should anyway lead to a wrong recognition by the SVM. Hence, rejection is not dramatic in this particular case.

4.3 Digit recognition performances

In this section, the performances of the SVM are analyzed. This classifier is used to recognize the segmented regions of the bounding boxes that went through the tree-based filtering. More precisely, the classifier assigns the candidate regions either to a specific number or to a bin class. As represented in Fig.5, to investigate how "number" and "non-number" regions are processed by the SVM, we have chosen to evaluate the following rates:

(c) Non-numbers of (a) correctly classified as non-numbers by the SVM

(d) Numbers of (b) uncorrectly classified as non-numbers by the SVM

(e) Numbers of (b) correctly recognized by the SVM

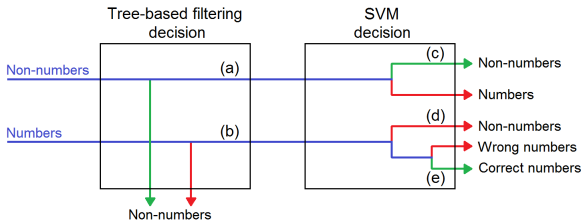

Figure 5. To compute the performances, segmented candidates are manually labelized into specific numbers or non-numbers and are passed through the system.

In practice, these performances highly depend on the training of the classifier. For this reason, the impact of the training strategy is first studied in the next section. After, it is shown that the proposed additional features strongly increase the correct classification of the candidate regions.

4.3.1 Impact of the training strategy

The objective of this section is to analyze the influence of a mismatch between the training set and the test set on the classification performances. Specifically, two kinds of mismatches are investigated. First, we analyze the case in which the distribution of the samples in the training set does not match the one of the validation set. Afterwards, the case in which the thickness of the reference numbers of the training set does not correspond to the font thickness observed in the test set is investigated. For both kinds of mismatches, the performances are obtained based on the features proposed by [7] (and presented in Section 3) and the SVM optimal parameters are determined with a 5-times cross-validation.

First, in order to determine the influence of a distribution mismatch, three different training sets are composed and evaluated on the same validation set. This validation set is composed of approximatively 1100 segmented candidate numbers coming from bounding boxes of players got from a real-life basketball game and labelized manually. As a first distribution of training set, one could think that all the numbers have the same probability to appear during a basketball game and that this probability is the same for the occurence of non-numbers. Then, the distribution of the training set is considered as uniform. The performances of such a training set are presented in the second column of Table 4. However, in a real-life basketball game, the International Basketball Federation (FIBA) has defined the legal jersey numbers as being the ones from 4 to 15. Moreover, because the recognition of a number composed of two spatially separated digits is splitted into two individual recognitions, the digits distribution can not be considered as uniform in our application. The third column of Table 4 presents the performances of this particularized distribution, taking into account that a non-number is ten times more likely to appear than a number. Finally, we found empirically that our method extracts in fact about 3 times more non-numbers candidates than number candidates. The "estimated" distribution of Table 4 take this ratio into account and is thus the one that more likely corresponds to a real-life basketball distribution.

		Uniform	FIBA	Estimated
(c)	Non-numbers of (a) correctly classified as non-numbers by the SVM	90.2%	91.5%	91.8%
(d)	Numbers of (b) uncorrectly classified as non-numbers by the SVM	90.4%	80.2%	75.9%
(e)	Numbers of (b) correctly recognized by the SVM	22.2%	28.9%	30.9%

Table 4. It can be observed that more the distribution of the training set matches with the distribution of the validation set (from the left column to the right one), better are the number recognition performances.

The performances presented in Table 4 show that the better is the match between the distribution of the training set and the distribution of the validation set, the higher is the number recognition accuracy. Because estimating the distribution of appearance of numbers and non-numbers in a real-life basketball match is beyond the scope of this paper, we use a training set based on the "estimated" distribution in the next sections.

To analyze the consequences of changes in font thickness between the training samples and the test samples, we train the classifier with a given number template and test its performances on both similar and different templates. More precisely, the classifier has been trained on a *synthetic dataset* and been tested on both the *synthetic* and *real-life datasets* presented in Section 4. Table 5 shows the influence of such a change on the performances of classification.

		Same font thickness	Different font thickness
(c)	Non-numbers correctly classified as non-numbers	98.5%	90.6%
(d)	Numbers uncorrectly classified as non-numbers	0.08%	77.3%
(e)	Numbers correctly recognized	98.4%	29.3%

Table 5. A variation of font thickness between the training and test samples leads to inaccurate digit recognition.

By comparing the results of the two test sets, the challenge of generalizing a classification model to real-life conditions is highlighted.

4.3.2 Impact of the features used by the SVM

This section studies the influence of adding the proposed features on the effectiveness of the number recognition. More precisely, for all segmented candidate numbers that have passed the tree-based filtering, the features presented in Section 3 are computed and are combined into the following different sets:

o. {The ratio, the number of holes, the two centers of gravity and the three second order moments};
i. {o. + the vertical and horizontal projections of the candidate number} (features proposed by [7]);
ii. {o. + the vertical and horizontal projections of the vertical and horizontal *pseudo-skeletons*};
iii. {o. + the vertical and horizontal projections of the four quarters of the vert. and horiz. *pseudo-skeletons*}.

For each set of features, Table 6 shows the performances of classification (represented by the symbol $\rightarrow$) and recognition (represented by the symbol $=$) of numbers (refered as N) and non-numbers (NN), both on the *synthetic* and *real-life datasets*.

	Impact of the proposed extra features					
	On *synthetic dataset*			On *real-life dataset*		
	(c) NN→NN	(d) N→NN	(e) N=N	(c) NN→NN	(d) N→NN	(e) N=N
i.	98.5%	0.08%	98.4%	90.6%	77.3%	29.3%
ii.	98%	0.14%	97%	88.5%	74.6%	39.1%
iii.	98.8%	0.04%	98.5%	89.8%	55.3%	80.6%

Table 6. The proposed additional features increase the robustness against digit appearance changes.

From Table 6, it can be observed that the additional features only influence the performances obtained on the *real-life dataset*. Because the classifier has been trained on *synthetic* samples, it means that the proposed additional features increase the robustness against digit appearance changes, resulting from the font thickness variability and from the natural deformations of the jersey during the game.

4.4 Performances of the entire system and weaknesses

This section first provides a quantitative comparison of our approach with the state of the art[7]. Then, we merge the performances obtained in the previous sections to evaluate the global performances of the entire system. Finally, the weaknesses of the proposed approach are presented, and some path to solve them are given.

In Table 7, we first show that there is almost no change of effectiveness when replacing the mean-shift algorithm in the approach proposed by [7] with the proposed faster segmentation. Then, by completing this fast segmentation with the proposed features, it shows that the effectiveness of the recognition is strongly increased.

	(c) NN→NN	(d) N→NN	(e) N=N
Mean-shift + i.	90.5%	79.4%	31.8%
Proposed segmentation + i.	90.6%	77.3%	29.3%
Proposed segmentation + iii.	89.8%	55.3%	80.6%

Table 7. The proposed segmentation speeds up the system without changing its effectiveness, while the proposed extra features strongly increase the number recognition performances.

Finally, to compute the performances of the entire system, it is important to realize that the correct recognition of the teams only influences the recognition of the numbers. Taking that into account, because approximatively 54.6% of the non-numbers are rejected by the tree-based filtering while 89.8% of the non-rejected ones are correctly classified as non-numbers by the SVM, the entire system correctly rejects more than 95% of the non-numbers candidates. For the number recognition, among the 97.8% correctly segmented digits, 91.9% go through the tree-based filtering and 44.7% of them are well recognized as numbers. So, totally, more than 40% of the numbers are classified as numbers, and 80.6% of them are correctly recognized.

These global performances show that the main weakness of the system is the erroneous classification (done by the SVM) of some numbers as non-numbers. In fact, it has been observed than almost 34% of the classification of numbers as non-numbers come from a strong difference of templates between the training numbers and the testing ones. More precisely, as illustrated on Fig.6, the template of the training numbers 4 is very different from

the one of the test dataset. The same observation can be done on number 9. This could be avoided by generating a training set using various distinct fonts of the same numbers. Finally, about 12% of the wrong recognition of the numbers come from a confusion between number 1 and number 7. However, there is no way to differentiate these two numbers without a global view of the jersey. In fact, all these errors are due to a rotation and folding of the jersey, making a number 1 perfectly similar to a number 7.

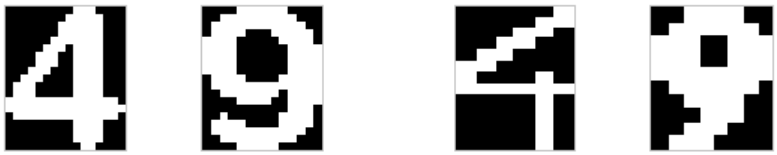

Figure 6. The first two number templates are the ones used to train the classifier to number 4 and 9. The two last ones are the templates of the same numbers from the *real-life dataset*. The important dissimilarity between those two sets leads to a high rate of erroneous classification of number 4 and 9 as non-numbers.

5. CONCLUSION

In this paper, a color segmentation method is coupled to a feature-based classification to automatically recognize a sport player based on the number printed on his/her jersey. A computationally efficient color-based segmentation is designed based on the pre-learning of the jersey and digit dominant color. In order to refine the extraction of the digit, the regions of the jerseys that have the same color than the digit are filtered out using a tree-based filtering with relaxed constraints. For the digit recognition, we proposed to describe each candidate digit by a mix of conventional and original features, including its number of holes, some central moments, the vertical and horizontal projections of its *pseudo-skeletons*, etc. The use of those features to train a SVM on a database composed of a distorted set of numbers has led to a good recognition rate of the digits despite their natural deformations in real-game situations.

ACKNOWLEDGMENTS

This work has been funded by the Belgian NSF (F.N.R.S) and the Walloon Region project WIST3 SPORTIC.

REFERENCES

[1] Chang, S. L., Chen, L. S., Chung, Y. C. and Chen, S. W, "Automatic license plate recognition," IEEE Transactions on Intelligent Transportation Systems 5(1), 42–53 (2004).
[2] Srihari, S. N., "High-performance reading machines," Proceedings of the IEEE 80(7), 1120–1132 (1992).
[3] Xu, Y. and Nagy, G., "Prototype extraction and adaptive OCR," IEEE Transactions on Pattern Analysis and Machine Intelligence 21(12), 1280–1296 (1999).
[4] European FP7 APIDIS project website on `http://www.apidis.org`.
[5] Ariki, Y., Kubota, S. and Kumano, M., "Automatic production system of soccer sports video by digital camera work based on situation recognition," Eighth IEEE International Symposium on Multimedia (ISM), 851–860 (2006).
[6] Alahi, A., Boursier, Y., Jacques, L. and Vandergheynst, P., "Sport players detection and tracking with a mixed network of planar and omnidirectional cameras," Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), 1–8 (2009).
[7] Delannay, D., Danhier, N. and De Vleeschouwer, C., "Detection and recognition of sports(wo)men from multiple views," Third ACM International Conference on Distributed Smart Cameras (ICDSC), 1–7 (2009).
[8] Fukunaga, K. and Hostetler, L., "The estimation of the gradient of a density function, with applications in pattern recognition," IEEE Transactions on Information Theory 21(1), 32–40 (1975).
[9] Parisot, P. and De Vleeschouwer, C., "Graph-based filtering of ballistic trajectory," IEEE International Conference on Multimedia and Expo (ICME), 1–4 (2011).
[10] MacQueen, J., "Some methods for classification and analysis of multivariate observations," Fifth Berkeley Symposium on Mathematic Statistics and Probability 1, 281–297 (1967).
[11] Chang, C. C. and Lin, C. J., "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology 2(3), 1–27 (2011).
[12] Dey, S., Bhattacharya, B. B., Kundu, M. K. and Acharya, T., "A fast algorithm for computing the Euler number of an image and its VLSI implementation," Thirteenth International Conference on VLSI Design, 330–335 (2000).