# Application of AdaBoost Algorithm in Basketball Player Detection

**Branko Markoski*, Zdravko Ivanković*, Ladislav Ratgeber**, Predrag Pecev*, Dragana Glušac***

[*]University of Novi Sad, Technical faculty “Mihajlo Pupin”, Djure Djakovica bb, 23000 Zrenjanin, Republic of Serbia, markoni@uns.ac.rs, zdravko@tfzr.uns.ac.rs, pecev@tfzr.uns.ac.rs, gdragana@tfzr.uns.ac.rs

[**]PTE-ETK University of Health Sciences Pécs - Doctor School, Havihegyi út 4, 7627 Pécs, Hungary, ratgeber@ratgeber.hu

*Abstract: Video materials contain huge amount of information. Their storage in databases and analysis by various algorithms is a constantly developing area. This paper presents the process of basketball game analysis by AdaBoost algorithm. This algorithm is mainly used for face and body parts recognition, and was not tested on player detection in basketball. It consists of a linear combination of weak classifiers. In this paper, we used stumps, i.e. decision trees with only one level as such classifiers. The aim of this research is to assess the accuracy of this algorithm when applied in player detection during basketball games. We examined the capabilities of AdaBoost algorithm on a video footage obtained from the single moving camera, without any previous processing. First training was performed using images of a basketball player’s entire body (head, legs, arms and torso), while the second training was performed using images of a head and torso. By applying the algorithm to the given set of images that include head and torso, the algorithm obtained an accuracy of 70.5%. Training on the set of entire body images was not successful due to the large amount of background that goes into the training, and which represents noise in training process. This research concluded that AdaBoost could not be applied to object detection in sports events. We also concluded that this algorithm gives much better results when applied on simpler objects (like face recognition) and that its application could be in detection of players’ body parts or as a first step in object detection in order to eliminate as much area as possible. Its application in detecting players' upper body or entire players gives large number of false positive, which makes algorithm inapplicable in real situations.*

*Keywords: AdaBoost; Object detection; Basketball*

## 1 Introduction

With the advance of information technology, the amount of created, transmitted and stored multimedia content constantly increases. As a result, the multimedia content is widely used in many applications. Therefore, there arises the need for its organization and analysis, both from commercial and academic aspects. Computer vision represents a technology that can be applied in order to achieve effective search and analysis of video content.

Computer vision represents a process that consists of several phases [1]. First phase is initialization, which is a process of removing background and extracting objects of interest by creating their models using markers, images or predefined shapes. Next phase is tracking, which is a process of object recognition in successive frames. This phase lasts until the object leaves observed area, or until the tracking is terminated. Third phase is pose estimation, which in process of human recognition and represents analysis of the arms, legs, torso and head, according to which the object is classified into one of the previously defined poses. The final phase is recognition. Recognition can be achieved by recognizing person's face or some other characteristic feature. Computer vision, therefore, consists of four phases (initialization, tracking, pose estimation and recognition), but the subject of this study is the first phase, e.g. initialization, which represents the model creation.

The model is created by players' images, which are obtained using specially developed software. This software stores one frame from video material in every 0.5 seconds on provided location in computer. From those frames, we cut rectangles that contain basketball players. These rectangles will be used in training process. Training is done with AdaBoost algorithm, which represents an often-used algorithm in the shape recognition. It is primarily used for face and body parts recognition. The aim of this study is to assess its capabilities in order to detect players in basketball games. Images of basketball players are objects with high degree of diversity, depending on whether the player has the ball, plays defense, shoots on the basket, jumps for the ball etc. Therefore, the training set has many variations, and the goal of this research is to assess whether AdaBoost can be successfully applied in such training process. Its main feature is execution speed, which is especially important when analyzing basketball games. Coaches often want to have complete analysis of the game as soon as possible, in order to make some changes in their team play. For that reason, the use of this algorithm in the process of basketball game analysis would be useful. AdaBoost requires a large training set. Therefore, six thousand positive examples (images of basketball players) and six thousand negative examples (images that do not contain any basketball player) were used. We have combined positive and negative images in testing process (images of basketball players are "glued" over negative examples). Images combined in this way are used in order to assess the performance of AdaBoost algorithm.

The second chapter in this paper provides an overview of application of data mining in sports, as well as research related to computer vision. Basketball player recognition belongs to the broader group of analysis in computer vision that is called human motion capture. The third chapter explains AdaBoost algorithm and its variations in order to achieve better performance. The fourth chapter contains the procedure for training AdaBoost algorithm, which consists of creating training set, marking positive and negative examples, training and testing. The fifth section contains conclusion remarks and further stages of computer vision that can be applied in order to create solution that can be applicable in practice.

# 2 Data Mining in Sport

Data mining in sport is experiencing rapid growth in recent years and is gradually attracting the attention of largest sports associations. Baseball team Boston Red Sox and football club AC Milan were among the first organizations that started to apply the benefits of data mining in the sports. Special merits for the introduction of data mining in the sport belong to Dean Oliver, who introduced this methodology in basketball [2], and Bill James, who did the same in baseball [3].

Before the implementation of data mining, sports organizations have relied almost exclusively on human factor. They believed that experts in a given field (coaches, managers, scouts) could successfully convert collected data into practical knowledge. As the amount of collected information increased, these organizations started looking for methods that are more practical. Appropriate usage of large amounts of data available to sports organizations led from engaging additional statisticians to adopting techniques of data mining. Application of data mining can lead to better overall team performance, by analyzing the behavior of players in certain situations, determining their individual impact, revealing the opponent's tactics and pointing possible weaknesses in play.

According to Schumaker et al. [4] in the next few years, the application of data mining in sports will face several challenges and obstacles. The biggest obstacle will be to overcome opposition to new technologies that is present in some sports organizations.

While the use of statistics in decision-making is certainly an improvement over the use of instinct of coaches, managers and scouts, statistics alone can go in the wrong direction without knowledge of the problem domain. The first part of the problem is to determine the performance metrics. A large number of existing sport metrics can easily be used inappropriately. Ballard [5] has presented a typical example of inaccuracy in data collection in basketball. He gives an example of a jump in defense, which represents the number of times a player catches the ball in defense after opponent’s unsuccessful shot. In order to record the jump in defense, teammates have to block opponent players and keep them away from the basket, but only the player who catches the ball is awarded with the rebound. The second part of the problem is to find interesting patterns in data. These patterns may display movement and intentions of opponent players, reveal the beginning of injury during training or predict outcome of observed game. A practical method in finding those patterns could be application of neural networks [6] [7].

With the development of technology, sports events have become available in digital form as part of multimedia databases. Search of video and multimedia content is becoming more common in sports due to large number of available tools. Automated methods of detection are used for parsing video content, and translating it into a form that can be searched [4] [8].

Traditional sports statistics has quickly become insufficient in comparison to the advantages of multimedia technology [9]. In recent years the usage of videos for recording certain events for later analysis, has become a common place. For example, baseball players in American professional league visit a team multimedia room and study different ways in which the pitcher sends the ball, in order to prepare for new game or to play correctly during the match [10]. Another technology that allowed faster analysis and transfer of video materials and knowledge obtained by their analysis is the internet [11] [12]. Due to these technologies, videos are almost immediately available to players, coaches and scouts.

Multimedia analysis of sports events is presented in menu scientific papers. Among them, there are almost no papers about application of AdaBoost in basketball games. Lehuger et al. have applied AdaBoost in soccer [13], while Zahid et al have applied AdaBoost in baseball [14].

Our goal is to develop system that would automatically gather knowledge from footages of basketball games. In order to achieve this, the first step is to recognize players on the court. Next steps would involve court detection, ball and basket detection and analysis of ball trajectory.

# 3 Adaboost Algorithm

Boosting takes its origin from the theoretical framework for studying machine learning called "PAC" (Probably Approximately Correct) developed by Kearns and Valiant [15]. They were the first who questioned whether "weak" learning algorithm, which behaves slightly better than random guessing, could be a building block for general accurate "strong" learning algorithm.

AdaBoost, short for Adaptive Boosting, is a machine-learning algorithm first formulated by Freund and Schapire [16]. It is adaptive in the sense that classifiers that come in next for execution are adjusted according to those instances that were wrongly classified width the previous classifiers. It is sensitive to noisy data and information that does not belong to the required set. However, in some situations, this algorithm may be less susceptible to memory input set in comparison to many other algorithms. AdaBoost calls the weak classifiers repeatedly, performing a series of $t = 1,...,T$ classifiers. In each execution, "weight" calculated by incorrectly classified examples increases (or, alternatively, weights of each correctly classified examples decreases). New classifiers are constrained to focus on those examples that were incorrectly classified by previous classifiers. This is a meta-algorithm that can be used together width a number of other algorithms in order to improve performance. Pseudo-code for AdaBoost is given in Listing 1.

Listing 1. Pseudo code for AdaBoost algorithm.

---

Given: $(x_1, y_1),...,(x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1,+1\}$

Initialize : $D_1(t) = 1/m$

For $t = 1,...,T$ :

- Train weak learner using distribution $D_t$
- Get weak hypothesis $h_t : X \to \{-1,+1\}$ width error

  $\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$
- Choose:

  $$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$
- Update:

  $$D_{t+1}(t) = \frac{D_t(i)}{Z_t} \times \begin{Bmatrix} \varepsilon^{-\alpha_t}, if\ h_t(x_i) = y_i \\ \varepsilon^{\alpha_t}, if\ h_t(x_i) \neq y_i \end{Bmatrix} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

  Where $Z_t$ is a normalization factor (t is chosen so that $D_{t+1}$ will be a distribution). The output is the final hypothesis:

  $$H(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

---

The algorithm receives as input some training set $(x_1, y_1),...,(x_m, y_m)$ where each $x_i$ belongs to a particular domain or instance space X, and each label $y_i$ is in label space Y. In most cases, it is assumed that $Y = \{-1,+1\}$, except when looking at extending of AdaBoost with more classes. AdaBoost calls "weak" learning algorithm repeatedly in $t = 1,...,T$ execution steps. One of the basic ideas of the algorithm is to maintain a distribution or set of weights over the training set. Weighting distribution in training example *i* in step t is denoted by $D_t(i)$. At the beginning, all the weights are placed on the same value, but at each step, the weights of incorrectly classified examples are increased and the weak learning algorithm is forced to focus on more difficult examples in training set.

The task of the weak learning algorithm is to find a weak hypothesis $h_t : X \rightarrow \{-1,+1\}$, which corresponds to the distribution $D_t$. The accuracy of the weak hypothesis is measured by its error:

$$\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i) \neq y_i} D_t(i) \tag{1}$$

The previous expression shows that the error is measured in accordance to the distribution $D_t$ over which the weak learning algorithm was trained. In practice, the weak learning algorithm can be any algorithm that uses weights $D_t$ from the training set.

Looking at the example of recognizing players in basketball games, $x_i$ represents a player (standing motionless on the court, shooting to the basket, jumping for the ball, playing defense, attempting dribbling penetration, etc.), while labels $y_i$ show whether a given detection represents a basketball player or something else in the frame. Weak hypothesis is presumption that certain objects are players, and sub collections examined by that hypothesis are selected according to the distribution $D_t$.

When it comes to the hypotheses $h_t$, AdaBoost determines parameter $\alpha_t$. Intuitively, $\alpha_t$ measures importance assigned to the hypothesis $h_t$. Listing 1 shows that $\alpha_t \geq 0$ if $\varepsilon_t \leq 1/2$, and that $\alpha_t$ increases its value for error $\varepsilon_t$ becomes smaller.

Next step is to update distributions $D_t$ by using rules shown in Listing 1. The effect of this rule is to increase the weight of examples that are misclassified by the hypothesis $h_t$, and to reduce weights of well-classified examples. Thus, the weights are trying to concentrate on "harder" examples.

The final hypothesis $H$ is weighted majority of votes of $T$ weak hypotheses, where $\alpha_t$ represents weight assigned to hypothesis $h_t$.

In practice, AdaBoost has many advantages. It is fast, simple and easy to program. AdaBoost does not have any parameters that have to be adjusted separately (except for the number of steps $T$). On the other hand, actual performances of boosting in a particular problem are largely dependent on data and weak learning algorithm. In accordance with theory, boosting can give wrong results when there is not enough data for training, when weak hypothesis are very complex, or when weak hypothesis are too weak.

In 2001, Viola and Jones [17] presented their work that was a milestone in the implementation of AdaBoost algorithm. Their work had three important contributions in fast and accurate image analysis:

- Appliance of integral images
- Learning classification functions
- Cascade creation

## 3.1 Stump

In this paper, we used stump as a weak learning algorithm. Stump is a machine-learning model that consists of a decision tree width one level. In other words, it is a decision tree width one internal node (root node) that is directly related to end-nodes. Stump gives a prediction based on the value of a single input. It is sometimes called 1-rule. An example of stump is shown in Fig. 1.

Depending on the type of input characteristic, there are several variations. For nominal characteristics, a stump can be created that contains a leaf for each possible characteristic value, or stump width only two leafs where first leaf corresponds to one category of results and second leaf corresponds to all other categories. For binary characteristic, these two approaches are the same.

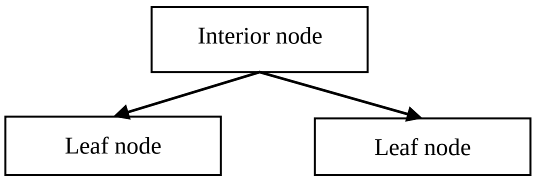



Figure 1
Functional structure of the system that analyzes the movements of the human body

# 4 Creating Model of Basketball Players

In order to create a model of basketball players, AdaBoost algorithm is applied over the training set, without any previous processing. During the process of training a new classifier, it is necessary to go through several stages:

1. Image acquisition
2. Example creation
3. Training
4. Testing

## 4.1 Image Acquisition

Different sources can be used in image acquisition process. Since the aim of this paper is player detection in basketball games, the training process uses videos broadcasted by television stations. These videos are stored in multimedia database. In order to apply AdaBoost algorithm we use additional software. The purpose of this software is to capture one frame from the video material in every 0.5 seconds and store it in predefined location on user's hard disk. We used

SampleCreator software, which is implemented in C# programming language and .NET 4.0 framework. Frames were captured and stored using DirectShow technology. A game in NBA league lasts 48 minutes (four quarters of 12 minutes). In addition to this playtime, there are number of interruptions (fouls, time outs, breaks between periods). Including those, we can assess empirical value of the average game duration around 100 minutes. By applying this method of storing frames from single game, we get 100 * 60 * 2 = 12.000 frames. This provides a sufficient amount of data to train AdaBoost algorithm. This algorithm works width black and white pictures so color of shirt itself is not a major problem and an already trained algorithm can be applied on large number of games.

By using DirectShow technology, we obtain images that will mainly serve as positive examples (images that contain objects of interest). However, during the game we have frames that do not contain any player. These frames can be used as negative examples. In addition, negative examples could be found in other sources. It can be any picture without basketball players. Fig. 2 shows positive and negative examples obtained from a basketball game.

Figure 2

Positive and negative example obtained from the basketball game

Besides positive and negative examples, some frames may not fall into any of the above categories. In most cases, those frames contain basketball players with the poor display quality. Their use in the training process could direct AdaBoost algorithm in the wrong direction, because the algorithm is sensitive to the noise data. Those frames should be excluded from the training process.

## 4.2 Example Creation

Examples are objects of interest, which are used in the training process. Depending on the look of required objects, different authors have used different sizes of training sets: 5000 examples in face detection [18], 6000 examples in pedestrian detection [19]. Training set creation is performed by cutting objects from the frames of basketball game. SampleCreator software is used for this purpose. It allows marking objects, while maintaining the ratio between width and height. Fig. 3 shows the process of marking the whole basketball players that will be used in training, as well as the process of marking player’s upper body.

Different ratios of height and width were used in marking process. In marking the whole players, ratio of 2.2 was empirically determined (e.g. width 100 px, height 220 px), while in marking the upper body, that ratio was 1.8 (e.g. width 100 px, height 180 px). The ratios were determined in order to optimally cover player on the observed frame. The ratio of player in one training set must be fixed, because all images in training process are scaled to same size. If ratio is not the same, some images are stretched while others are elongated.

Figures show that not all players are marked as positive examples. The reason is that some objects may adversely affect the training if they are not shown clearly, if they are entering or leaving the frame, or if other objects obscure them. If we compare images on Fig. 3, it is evident that we have not marked the same players. Some players are not marked because of the background that would enter in the training process, which would adversely affect the AdaBoost algorithm. Unmarked players will not interfere in the training process because we crop positive examples from the picture and "glue" them on negative examples at pre-defined locations.

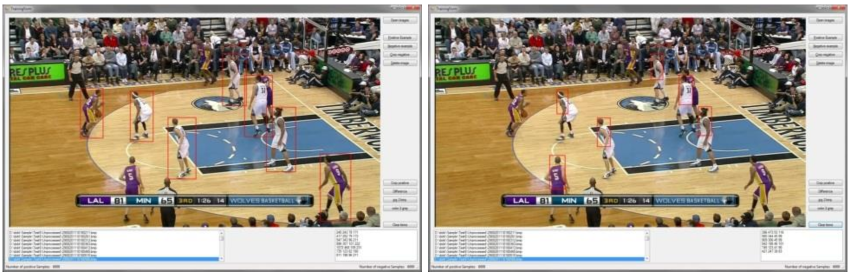


Figure 3

Marking of whole body and upper body of basketball players by SampleCreator software

## 4.3 Training Example Creation

In order to put labeled examples into a training set, they should be “cut” from the observed images. In addition, example size can be changed, or we can apply function that will distort the image in certain limits. The program is not able to create more training examples based on one picture. It is therefore necessary to have a sufficient number of examples, which is in various applications ranging from 5000 to 7000. The marked areas, which will be used in the training process, are shown in Figs. 4~5.

Figure 4
Images of whole basketball players' body that will be used in the training process

From the previous image, it can be seen that background takes a large area in the training set. Size of this area is different depending to whether a player is close to the paint (the area on the ground painted a darker color), close to the audience (the audience is much darker than the floor), or whether there are other players in its vicinity. In addition, during the game, basketball player can be in different positions depending on whether he is running, having the ball, playing defense or taking a shoot. Depending on that, their limbs can be in different position, which makes a training process much harder. Because of this, training was performed width another set of examples that included basketball player’s upper body. In this training, limbs were excluded, which reduced dimensionality of the search problem. It also reduces the amount of background that is essentially a noise. Those examples are shown in Fig. 5.

In our research we have used 6000 positive examples that contain players whole body and 6000 positive examples of players upper body in order to train algorithm.

Figure 5

Images of basketball player upper body that will be used in the training process

## 4.4 Testing Example Creation

Testing set can be created by putting one positive example over a negative example. In testing process, we used a new set of 1000 positive examples. During this process, a positive image can be resized or distorted. In this way, the application knows the exact position of the positive examples, based on which it can provide assessment of whether the algorithm recognized the positive example on the observed image. Examples of test images are given in Fig. 6.

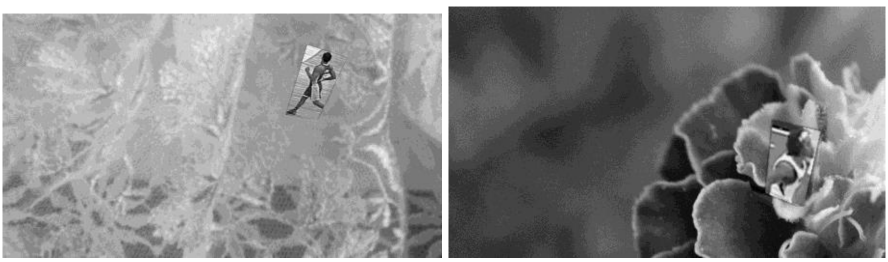


Figure 6

Testing examples width whole body and upper body of basketball player

## 4.5 Training

In training process, we apply AdaBoost algorithm over the previously marked examples. Regarding the size of examples in training set, Kuranov et al. [18] have shown that the best results are achieved when the dimensions of examples that contain faces are reduced to 20x20 pixels. In the training process with images that contain whole basketball players, the ratio between the width and height cannot be 1:1. Therefore images were reduced to 20x44 pixels in training with the entire basketball players (574,479 features), and 20x36 pixels for training with player's upper body (392,394 features).

The same authors have also suggested a 20-stages training. If as a training parameters we use degree of false positive of 0.5 and detection rate of 0.999, after the entire training we can expect degree of false positive of $0.5^{20} \approx 9.6e - 07$, and the detection rate of . $0.999^{20} \approx 0.98$

During AdaBoost training on the examples that have symmetry (human face), we can apply some type of optimization that significantly speeds up the processing. This is due to the fact that in these cases only one-half (left of right) of the Haar feature is used. However, although players are objects that have symmetry, when observed during the game and in all positions in which they could be found, the application of symmetry in the training would not lead to desired results. In order to achieve higher accuracy, we used the extended set of Haar features [20] (vertical features and features rotated by 45 degrees).

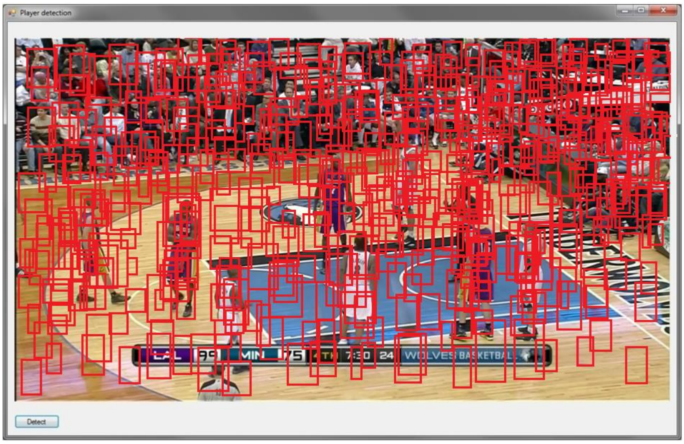
GAB (Gentle AdaBoost) classifier is used in the training that puts less emphasis on outliers [21]. The main reason was the work of Kuranov et al. [18], in which they have proven that GAB algorithm achieves highest results in object detection. This classifier is also the fastest one considering the time required for training.

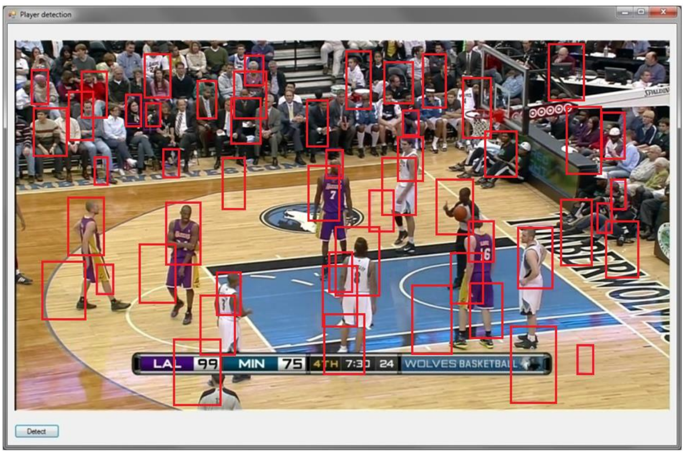
During the training process, we used six thousand positive examples and six thousand negative examples. Training can be completed in several sub-stages when the minimal desired degree of search is fulfilled, or when the degree of false positive is reached, because the additional phases will certainly reduce this level (0.99 after current phase * 0.99 for next phase = 0.981 after next phase). A valid algorithm is the one that rejects all incorrect examples.

During the training process over the set that included images of whole basketball players, the algorithm failed to reduce the level of false positive below 0.5. The algorithm continues training until the percentage of the original examples used in training falls to 0%, i.e. until all the examples are used. This would lead to interruption of the training without the wanted result. The reason is, primarily, a large degree of diversity that is encountered in the training set. This diversity is reflected in the position of players depending on whether they are standing, walking or running, and whether they have the ball, play an active defense or shooting on the basket. Another reason is a relatively large amount of background, which is located in the training examples, and which represents a noise.

Training on the set that contains players' upper body was successfully finished. This set does not include player arms and legs and is therefore much more balanced. In addition, the amount of background in these images is far lower, as well as the noise.

If the results of training are included in the application and executed on an arbitrary example taken from a basketball game, we get the result as shown in Fig. 7. In figure we can see three images. First image is taken after twelve of twenty total stages in training process. After this step, all players are detected, but also very large number of false positive is detected (algorithm detects almost everything on the image). The second image is taken after eighteen steps in training process. After this stage, algorithm also detects all players, but there are almost twenty times more false positive detections. From the third image it can be seen that the algorithm correctly detects approximately two thirds of the players. In addition, the algorithm still gives a large number of false positive in the analysis of the audience (identified as basketball players). There are seven times more false positive then true positive detections. Among the identified objects are also some unexpected results, such as parquet parts shown in one color. The reason for such poor result is large number of diversity in training set.



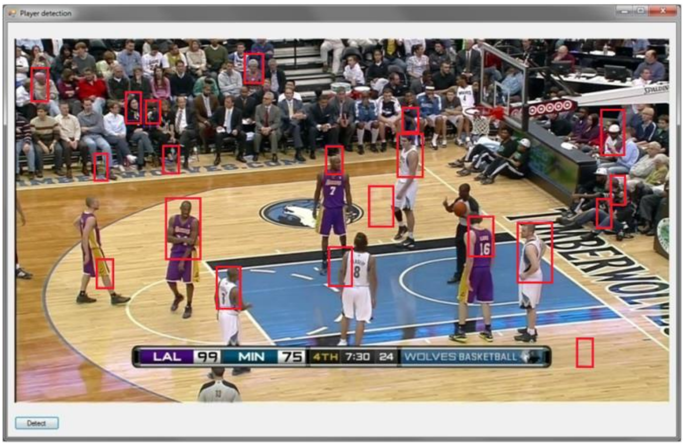



Figure 7
Result of algorithm application

## 4.6 Testing

In order to assess the performances of a trained classifier, a set of testing images was used. Those images have precise locations for each object of interest. The tool that is used to assess performance as input receives a collection of tagged images over which the classifier is applied. The performance of output is obtained by calculating number of found objects, number of objects that are not found and number of objects that are incorrectly classified as positive.

In this paper, we measured performances of training over the set that contains upper body of basketball players, because training over the set that contains whole players was not possible. With the obtained results, a ROC (Receiver Operating Characteristic) curve was created. It graphically represents sensitivity, i.e. ratio of true positive versus false positive, for a binary classification system.

The output of the testing tool shows the number of objects that are identified (Hits), the number of objects that are not identified and which represents false negative (Missed), and the number of objects that are classified as positive, but actually are not required objects (False). Looking at all images that were used in the training process, we get that the algorithm has successfully classified 705 objects from total number of 1000 used objects (70.5%). Detection is classified as positive if it overlaps at least 60% with real area. Algorithm did not recognize the remaining 295 objects (29.5%). In addition, the algorithm has recognized 7600 objects as requested objects, though they are not. This means that the algorithm, for each successfully recognized object, averagely has to recognize seven to eight points as positive. These results can be characterized as expected when compared with other researches that have used the AdaBoost algorithm. Baluja and Rowley have used the same algorithm, trying to recognize gender based on the images that contain people faces [22]. The training gave an accuracy of 80%. This is somewhat higher accuracy, but the images used in testing used a face only, which implicates minor amount of noise. When the training sets contain images with variable background, the accuracy of training process is decreasing. Yuan et al. showed differences in accuracy in recognizing faces when the background changes [23]. When the images contained different amounts of light, the resulting accuracy was 68.4%. In recognition of computer-generated characters with a constant background, AdaBoost achieves an accuracy of 90.5% [24].

When comparing our results to other papers that are also interested in player detection, our work is most similar to Lu et al. [25]. They recognize players from basketball games broadcasted via television stations by applying CRF (Conditional Random Fields algorithm) over DPM (Deformable Part Model). In their study they obtained an accuracy of 73% in player detection but much less number of false positive detection. Lehuger obtained an accuracy of 78.03% by applying AdaBoost algorithm, but algorithm was applied on soccer videos where the background is almost constant and player overlapping is much lower. Those results are shown in Table 1.

The complete output obtained after testing can be presented using ROC curves as given in Fig. 8. This curve shows recognition we can expect when we allow a certain degree of false positives. Figure shows that when we allow seven or more false positive, the level of hits is about 70%. By reducing allowed number of false positive, the percent of true positive is reduced as well. This decrease is approximately linear up to the value 2 for false positives, where the percentage of correct hits is just below 55%. By further reducing the level of false positive, the level of true positive decreases exponentially and if we do not allow false positive, i.e. when this value approaches zero, we can expect only about 25% of successful recognition.

Table 1

Detection results of applying AdaBoost algorithm on different training sets

| Researcher(s) | Training set | Result |
|---|---|---|
| Baluja and Rowley [22] | Face recognition | 80% |
| Yuan et al. [23] | Face recognition with changing background | 68.4% |
| Hoe et al. [24] | Recognition of computer-generated characters | 90.5% |
| Lehuger et al. [13] | Player recognition in soccer | 78.03% |
| Our work | Player recognition in basketball | 70.55 |

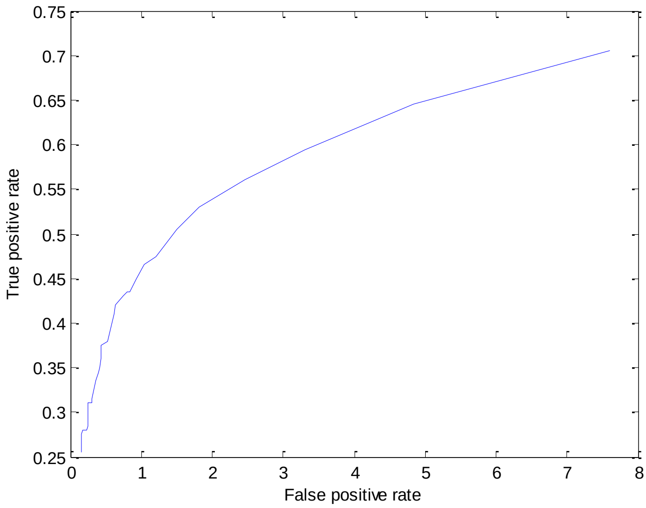



Figure 8
Performance presented by ROC curve

## Conclusion

Application of computer vision in analysis of sport events is quite a common practice, especially in recent years. Basketball, as one of the most popular sports, does not deviate from this trend. Unlike some other sports, it is played almost exclusively in halls, which simplifies the process of analysis due to constant lighting. Additional benefits in analysis of basketball games are the facts that all players in any team have jerseys of the same color, and that a large number of players are constantly in the camera view field.

Three approaches can be applied in analyzing basketball games: an analysis using markers that are placed on basketball players, an analysis using multiple synchronized cameras that cover the whole court, and image analysis using single camera. This paper reports the third type of analysis, i.e. analysis of games using video broadcasted by TV stations. This type of analysis brings the greatest amount of assumptions and inaccuracies, because players are often obscured by other players, or are outside the current view field of active camera. Another drawback is the fact that in this type of analysis camera and objects of interest are both moving.

In player detection process, we used Gentle AdaBoost algorithm, which is trained on two sets of examples. First set of images represents entire basketball players' body (head, torso, arms and legs), while another set of images represents basketball players' upper body (head and torso). AdaBoost failed to create a classifier based on images from the first training set. The reason is the large difference in training examples. The appearance of basketball players varies greatly depending on whether they walk, run, play defense, dribble the ball or shoot on the basket. In addition, all these variations occur when a basketball player is turned in some direction (towards the camera, towards a basket ...). In contrast, another training set, which included only head and torso of basketball players, contained a much smaller degree of variation. This resulted in successfully finished training process by AdaBoost algorithm, which produced a classifier that can be applied in basketball player detection. Nevertheless, the algorithm still has a number of areas in the pictures marked as basketball players (false positive).

This paper presented a degree of applicability of AdaBoost algorithm in the recognition of basketball players, without any prior processing. Obtained results are not applicable in real life situations because of low detection rate and very high rate of false positive detections. AdaBoost algorithm was applied for several reasons. The first is its speed, which is very important in players' recognition in order to get game knowledge as fast as possible. Another reason is very successful application of this algorithm in pedestrian detection on the CCTV footage. Our assumption was that the same algorithm could be successfully applied to players' recognition because that is also movement of people which is captured with the camera. However, the players move in different directions, they can be obscured by other players; a camera that captures them can pan and zoom in order enable viewers better perspective on the current action, and which is most important, their limbs (arms and legs) can be found in almost every position depending on whether they play offense, defense, jump for a ball etc.. When we take a look at pedestrian detection, we have pan, their hands are near the body and leg movements are the same. These are the reasons why AdaBoost did not give applicable results in the recognition of players in basketball games.

In order to improve obtained performances, we could apply background subtraction techniques that would leave only the objects that are likely to represent basketball players, on which the algorithm would then be applied. Further improvement would be achieved by mapping areas of interest, i.e. play field in observed application. This would remove everything that is not on the play field, which would make the search faster and more accurate. Another possible improvement would be training of AdaBoost algorithm for body parts (head, legs, arms, torso), which can then be combined in order to identify players.

AdaBoost primary characteristic is its speed. Some other algorithms achieve better results, if we observe number of identified objects, but their execution is much slower. Identifying players in basketball games is an activity that needs to be done in a short period, because coaches want to have analysis of both teams during the match, in order to make right decisions. Therefore, AdaBoost appears to be a proper solution. One possible solution to achieve better results and fewer false positive requires application of another algorithm that has better accuracy but worse execution time after classification by AdaBoost algorithm. This would not affect the overall performance, because most of the areas in the picture are already rejected by fast AdaBoost algorithm.

## Acknowledgment

Research was partially supported by the Ministry of Science and Technological Development of Republic of Serbia, through project no. 171039.


## References


[1] Moeslund, T., Hilton, A. and Kruger, V. "A Survey of Advances in Vision-based Human Motion Capture and Analysis", *Computer Vision and Image Understanding*, Vol. 104, No. 2, 2006, pp. 90-126

[2] Dean Oliver, Using Statistics to Make Forecasts, Ph.D., University of North Carolina, 1994

[3] Bill James, "The Bill James Historical Baseball Abstract", Villard 1985

[4] Schumaker, R., Soliman, O. and Chen, H. "Sports Data Mining", *Springer*, 2010

[5] Ballard, C. "Measure of success", *Sports Illustrated*, 2005

[6] Ivankovic, Z., Rackovic, M., Markoski, B., Radosav, D. and Ivkovic, M. "Appliance of Neural Networks in Basketball Scouting", *Acta Polytechnica Hungarica*, Vol. 7, No. 4, 2010, pp. 167-180

[7] Ratgeber, L., Markoski, B., Pecev, P., Lacmanovic, D. and Ivankovic, Z. "Comparative Review of Statistical Parameters for Man's and Women's Basketball Leagues in Serbia", Acta Polytechnica Hungarica, Vol. 10, No. 6, 2013, pp. 151-170

[8] Vasiljevic, P., Markoski, B., Ivankovic, Z., Ivkovic, M., Setrajcic, J. and Milosevic Z. "Basket Supervisor – Collecting Statistical Data in Basketball and Net Casting", *Technics Technologies Education Management*, Vol. 6, No. 1, 2011, pp. 169-178

[9] Lajos Izsó - Péter Tóth: Applying Web-mining Methods for Analysis of Student Behaviour in VLE Courses, Acta Polytechnica Hungarica, 5(4), 2008, pp. 79-92

[10] Lewis, M. "Moneyball: The Art of Winning an Unfair Game", *W. W. Norton & Company*, 2003

[11] Markoski, B., Ivankovic, Z., Pecev, P., Milosevic, Z. and Istrat, V. "Transfer of Basic Statistical Parameters in Basketball by Internet" *Techniques, Informatics and Education TIO 06*, 2011, pp. 792-797

[12] Vasiljevic, P., Ivankovic, Z., Milosevic, Z., Pecev, P. and Markoski, B. "Ajax Web Application for Basketball Statistics", *Information and Communication Technologies for Small and Medium Enterprises*, Arandjelovac, Serbia, 2011

[13] Lehuger, A., S. Duffner, and C. Garcia. "A Robust Method for Automatic Player Detection in Sport Videos." *Compression et Représentation des Signaux Audiovisuels (CORESA'07)*, Montpellier, France, November 2007

[14] Mahmood, Zahid, Tauseef Ali, and Shahid Khattak. "Automatic Player Detection and Recognition in Images using AdaBoost." Applied Sciences and Technology (IBCAST), 2012 9th International Bhurban Conference on IEEE, 2012

[15] Kearns, M. and Valiant, L.G. "Cryptographic Limitations on Learning Boolean Formulae and Finite Automata", *Journal of the Association for Computing Machinery*, Vol. 41, No. 1, 1994, pp. 67-95

[16] Freund, Y. and Schapire, R. E. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", *Journal of Computer and System Sciences*, Vol. 55, No. 1, 1997, pp. 119-139

[17] Viola, P. and Jones, M. "Rapid Object Detection using a Boosted Cascade of Simple Features", *Computer Vision and Pattern Recognition*, 2001, pp. 511-518

[18] Kuranov, A., Lienhart, R. and Pisarevsky, V. "An Empirical Analysis of Boosting Algorithms for Rapid Object with Extended Ser of Haar-like Features", *Intel Technical Report MRL-TR-July02-01*, 2002

[19] Viola, P., Jones, M. J. and Snow, D. "Detecting Pedestrians using Patterns of Motion and Appearance", *International Journal of Computer Vision*, Vol. 63, No. 2, 2005, pp. 153-161

[20] Lienhart, R., Kuranov, A. and Pisarevsky, V. "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection", *Pattern Recognition*, 2003, pp. 297-304

[21] Friedman J., Hastie T., Tibshirani R., "Additive Logistic Regression: A Statistical View of Boosting", The Anals of Statistics, Vol. 28, No. 2, 2000, pp. 337-407

[22] Baluja, S. and Rowley, H "Boosting Sex Identification Performance", *International Journal of Computer Vision*, Vol. 71, No. 1, 2007, pp. 111-119

[23] Yuan, Z., Lu, Z. and Pan, H. "LPP-AdaBoost-based Face Detection in Complex Backgrounds", *Fourth International Conference on Computer Sciences and Convergence Information Technology*, 2009, pp. 451-456

[24] Li, L., Hoe, K. E., Yu, X., Dong L. and Chu X. "Human Upper Body Pose Recognition Using Adaboost Template for Natural Human Robot Interaction", *Canadian Conference Computer and Robot Vision*, 2010, pp. 370-377

[25] W. L. Lu, J. A. Ting, K. P. Murphy, J. J. Little "Identifying Players in Broadcast Sports Videos using Conditional Random Fields". *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3249-3256