Neural Network Based Framework For Goal Event Detection In Soccer Videos

Kasun Wickramaratna, Min Chen, Shu-Ching Chen
School of Computing and Information Sciences, Florida International University,
Miami, FL 33199, USA
{kwick001, mchen005, chens}@cs.fiu.edu

Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami,
Coral Gables, FL 33124, USA
shyu@miami.edu

Abstract

In this paper, a neural network based framework for semantic event detection in soccer videos is proposed. The framework provides a robust solution for soccer goal event detection by combining the strength of multimodal analysis and the ability of neural network ensembles to reduce the generalization error. Due to the rareness of the goal events, the bootstrapped sampling method on the training set is utilized to enhance the recall of goal event detection. Then a group of component networks are trained using all the available training data. The precision of the detection is greatly improved via the following two steps. First, a pre-filtering step is employed on the test set to reduce the noisy and inconsistent data, and then an advanced weighting scheme is proposed to intelligently traverse and combine the component network predictions by taking into consideration the prediction performance of each network. A set of experiments are designed to compare the performance of different bootstrapped sampling schemes, to present the strength of the proposed weighting scheme in event detection, and to demonstrate the effectiveness of our framework for soccer goal event detection.

***Keywords**: video indexing, semantic event detection, content-based video retrieval, multi-modal analysis, neural networks, neural network ensemble, bootstrapping.*

1 Introduction

The continuous explosion of video data in the current era poses great needs in semantic indexing and querying of the video databases. As an essential issue, the detection of semantic events from various types of videos, especially sports video, has attracted great research attentions. However, it remains a challenge to infer the occurrences of events from various video features due to the so-called semantic gap. Moreover, the occurrence of interested events in many applications is fairly scarce, which poses additional difficulties in capturing such rare events from the huge amount of irrelevant data, especially with the presence of noisy or inconsistent data introduced in the video production process. To address these issues, many research studies have been carried out towards two directions for sports event detection, i.e., to explore the representative features and to capture the relations between the features and the events. Extensive researches have been devoted to explore the representative features for sports event detection, where the features are extracted at various levels (i.e., low-level, middle-level, object-level, etc.) and via different channels (i.e., visual, auditory, textual, etc.). Low-level features such as dominant color, motion vectors, and audio volume are acquired directly from the input videos by using simple feature extractions [2][26], which usually possess limited capabilities in presenting the semantic contents of the video events. In contrast, object-related features are attributes of the objects such as ball location and player shapes, which greatly facilitate the high-level domain analysis. However, their extraction is usually difficult and computationally costly for real-time implementation [10]. Middle-level features (e.g., grass areas/audience areas [5], audio keywords [29], etc.), on the other hand, offer a reasonable tradeoff between the computational requirements and the resulting semantics.

As for the problem of mapping the features to the semantic events, the hard-wired procedures or heuristic methods are adopted in most of the existing work [30], where the temporal templates [25] or heuristic rules [24] are created manually with the help of domain knowledge and/or data investigation. These approaches are normally efficient in the aspect of the computational cost, but they usually are not very robust and require heavy human involvement. Moreover, in many cases, it is difficult to intuitively define the relations between the features and the events. Alternatively, learning-based algorithms are adopted to detect the events via data mining and statistical analysis. For instance, Xie et al. [28] applied the Hidden Markov Model (HMM) to detect the *play* and *break* events from the soccer videos. However, the limitation of HMM for sports video event detection lies in the fact that it makes the so-called Markovian assumption about the data, i.e., the emission and the transition probabilities depend only on the current state, which does not fit well to the characteristics of the sports videos given various production styles and post-production effects. In addition, in [28], the framework was evaluated upon a very small testing data set, which failed to justify its generalization. Here, generalization means how well a trained decision function performs to classify the unseen data points, which is of great importance in evaluating a learning algorithm [22]. In [18], a three-layer feed-forward neural network was adopted to classify the shots into three semantic categories: "non-hitting," "in-field," and "out-field." The desirable feature of this work is that the feed-forward neural network is capable of carrying out data classification with more than two classes of objects in one single run. However, the generalization of the neural network is highly limited in detecting rare events. In order to make such kind of single neural network work in rare event detection, special techniques have to be employed in the training algorithm [1][9]. In [22], Support Vector Machine (SVM) was utilized for event detection in field-sports videos because of its promising generalization performance. However, SVMs were originally designed for binary classification and it remains an ongoing research issue regarding how to effectively extend it for multi-class classification [13].

To tackle these issues, a novel learning-based event detection framework is proposed in this paper, which incorporates both the strength of multimodal analysis and the ability of neural network ensembles to enhance the generalization capability. In addition, a bootstrapped sampling approach is adopted for rare event detection. Furthermore, a data reduction process and a robust weighting scheme are applied to further boost the classification performance.

The rest of paper is organized as follows. Section 2 gives an overview of the related work on the neural network techniques in classification. The proposed learning-based framework is discussed in Section 3. The experimental results are presented in Section 4, and Section 5 concludes the paper.

2 Related Work

2.1 Neural Networks in Classification

Neural networks have been used in many classification and event prediction problems due to their strength of identifying the relationship between predictor variables (inputs) and predicted variables (outputs) even when the relationship is far too complex to model with other mathematical approaches such as correlation. However, neural network based frameworks have rarely been applied in the domain of semantic event detection in video documents. In [18], the authors used a three-layer feed-forward neural network for semantic classification of baseball sport videos, where a back-propagation algorithm was used to train the network. In their work, the video shots are divided into three semantic categories as "Non-hitting," "In-field," and "Out-field" which are in fact three balanced classes, i.e., the number of instances in each class is comparably equal. For rare event detection, the back-propagation algorithm performs poorly in the sense that it converges very slowly for the classification problems in which most of the exemplars belong to one major class [1]. More specifically, the calculated initial net gradient vector is dominated by the major class so that the net error for the instances in the minor class (having a smaller number of instances) increases significantly [1]. Therefore, the approach proposed in [18] is not applicable for imbalanced event classification problems such as soccer goal event detection.

Several researches have been carried out to optimize the performance of neural networks in imbalanced classification problems. Among them, some studies focused on studying and modifying the training algorithms to achieve better performance. In [1], a modified training algorithm was proposed by treating both classes with equal importance, which eliminated the weaknesses in back-propagation algorithm and converged fast for two-class classification. However, such equality does not capture the characteristics of the event detection problems, where the focus is more leaned to identifying the event units rather than classifying the shots into two classes. Alternatively, [9] adopted the so-called "Stratifying coefficients" in the training algorithm, which introduces a higher weight to the minority class during the training process. More precisely, during the back-propagation training, a *weighted sum* instead of the summation of the derivatives was used, where the "weight" was determined by the ratio of the instances between the major and minor classes.

However, it is relatively complicated to modify the training algorithms. In contrast, a simpler approach was presented in [4], which generated multiple versions of a predictor and the final decision was reached by averaging the outcome of each predictor (in the case of numerical outcomes) or conducting a popularity vote when predicting a class. Those multiple predictors were formed using bootstrapped samples of the training data set, which is the idea we utilize in our work to alleviate the problem caused by the imbalanced training data set. That is, bootstrapped sampling is employed upon the training set to create a set of samples, where each sample contains a comparable amount of instances from both classes and is in turn utilized to train a neural network independently for the same task. This kind of a predictor collection is called an "ensemble" [23].

2.2 Neural Network Ensembles

In early nineties, Hansen and Salmon [11] discussed the applications of ensembles of similar multilayer neural networks and showed its strong capability in reducing the generalization error. Thereafter, this idea has been applied in different domains [9].

A Neural network ensemble is constructed in two steps: 1) to train a number of component neural networks and 2) to combine the component predictions. Each of these steps leaves ample opportunities and directions for the researchers to investigate.

Component Neural Networks

Many aspects need to be considered in this step, including the structure of the component networks (number of hidden layers and neurons, transfer function to be used in each layer, etc.), the training algorithm, the training terminating criteria, etc. As far as the neural network structure is concerned, a simple three-layer structure was used in most of the cases [16]. This is mainly because there is no established theoretical approach to decide the appropriate structure for a given problem. It also follows the Hornik's well-known statement that the multilayer standard feed-forward networks, even with as few as one hidden layer, possess the capability of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided that sufficiently many neurons are available [12].

Most of the other parameters are determined based on the experimental observations and experience. Training the neural network to avoid over-fitting the training set is a topic under discussion in many studies [3][17][21]. However, as suggested in [23], over-fitting can actually be useful in learning large neural network ensembles. In their work, it was shown that it is advantageous to use under-regularized component networks which overfit the training data for learning in large ensembles [23]. This constitutes the basic idea of our study in constructing the component networks.

Combining Predictions

In combining the predictions of component neural networks, the most prevailing methods are popularity voting [11], simple averaging [20], and weighted averaging [14][15][20]. For example, the weighted averaging methods in [15][20] assigned the weights to minimize the mean square error of the classification; whereas Jimenez [14] used dynamic weights determined by the confidence of the component network output, and the weights are dynamic in the sense that they are recalculated each time the ensemble output is evaluated.

In [31], a different approach was proposed where a subset instead of the entire set of component neural networks is used to construct the ensemble. In brief, an initial weight for each component network is assigned randomly and is then evolved using the genetic algorithm, where the goodness of the evolving population is evaluated using a validation data set that is bootstrap sampled from the training set. Finally, the component networks with weights greater than a preset threshold are selected to form the ensemble. However, it has three major drawbacks. First of all, it is extremely time consuming. Secondly, the aforementioned 'threshold' has to be defined with the aid of domain knowledge. Finally, since the goodness of the weights are evaluated using a random data set sampled from the training set, the constructed ensemble might be biased in the sense that the neural networks trained using these training instances could get higher weights.

All of the above mentioned approaches treated each of the predicted classes with equal importance. However, as discussed earlier, for event detection, normally one class (the event class) is of greater importance than the other classes. In addition, in popularity voting, simple averaging, and Jimenez's approach, they failed to take into consideration that the predictions of some neural networks may be more accurate than others. To address these issues, a novel weighting scheme is proposed for integrating the component network outputs (to be detailed in Section 3).

3 Proposed Methodology

In this study, the soccer application domain is used since soccer video analysis still remains a challenging task due to its loose game structure [27]. For soccer goal event detection, the event unit is defined in the shot level as shots are widely accepted as a self-contained unit. Therefore, the shot boundary detection algorithm proposed in our earlier work [6] consisting of the pixel-histogram comparison, segmentation map comparison, and object tracking subcomponents is first applied on the raw video data to obtain the shot units. Thereafter, a set of low-level and middle-level visual/audio features are extracted at the shot-level. More specifically, one middle-level feature, called the grass area ratio, is extracted on the basis of object segmentation and histogram analysis. In addition, the feature set also contains four low-level visual features and ten low-level audio features, where the visual features are obtained using pixel and histogram analysis and the audio features are exploited in both time-domain and frequency-domain. A complete list of all the features and their feature descriptions was presented in [7].

The problem of soccer goal event detection falls into the class of rare event detection. That is, the number of goal shots generally accounts for a small portion of the total shots in the soccer videos (e.g., less than 1% in our collected videos). As discussed earlier, in this work, bootstrapped sampling is applied to alleviate the learning problems caused by the *imbalanced* training data set. Then, a group of neural networks is trained by each of the samples and acts collectively as the predictor. This approach efficiently uses all the available data for training and the proposed weighting scheme effectively combines the outputs of all the component networks in the ensemble. The proposed framework is illustrated in Figure 1. As can be seen from this figure, it consists of conducting bootstrapped sampling, training the component neural networks, and combining the prediction.

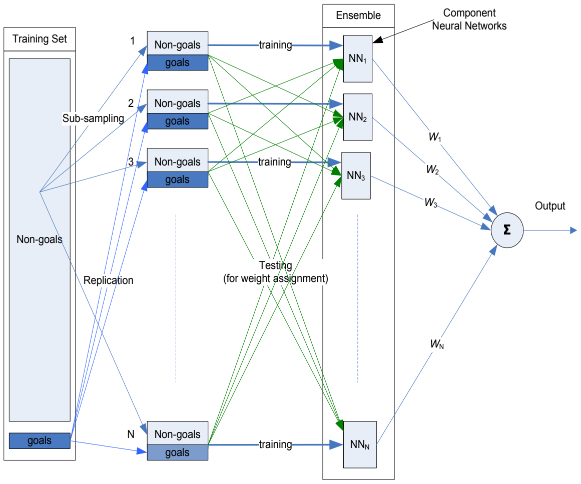

Figure 1. The proposed framework

• ***Bootstrapped Sampling***

The concern with bootstrapped sampling is that it improves the classification accuracy of the minor-class (i.e., goals) at the cost of decreasing that of the major-class (i.e., non-goals), which could lead to the miss-classification of non-goals and result in false-positives in the predicted goal class. In other words, it adversely affects the precision of goal event detection. Therefore, we have to come to a compromise between the recall and precision. For instance, if we change the original 1:99 ratio between goal and non-goal events to 50:50, the classification accuracy of the non-goal events is decreased. On the other hand, incrementing the goal ratio from 1% to 50% provides a big improvement in its classification accuracy. That is, the improvement in the recall value of the goal class will be larger compared to the deduction in precision. To experimentally prove this reasoning and to find a fair tradeoff point between the recall and precision values, three sub-sampling schemes with 1:1, 1:2, and 1:3 goal to non-goal example ratios are considered.

Formally, bootstrapped sampling can be defined as follows. Let n and m be the numbers of non-goal and goal instances in the training set, respectively. Let r be a constant that determines the non-goal to goal instance ratio in the bootstrapped sample, and N be the number of sub-samples randomly divided from the non-goal set, where N is determined by Eq. (1).

$$N = \lfloor n/rm \rfloor . \tag{1}$$

When n is not exactly divisible by N, any leftover non-goal examples are ignored in the training process. It is worth noting that if r is reasonably small, the instances being ignored from the non-goal set is negligible compared to the ones used in the training process. The goal instances are then bootstrapped into each of the non-goal samples. In other words, each of the resulting samples contains $(r + 1)m$ instances.

• ***Component Neural Networks***

Following the Hornik's [12] statement and based on the experimental evaluations on the training data set, the 3-layer feed-forward neural network structure is adopted for the component networks. The radial bias transfer function is used in the first two stages and the sigmoid transfer function is used in the output layer.

To fully utilize over-fitting for learning the large ensembles as suggested in [23], the component networks in our work are trained using a back-propagation algorithm to achieve maximum recall and precision for the training set, which, as discussed in [12], can reach 100% accuracy by properly adjusting the number of neurons in the neural network. In addition, it is observed that the classification performance varies with the initialization of the network biases and weights. Therefore, a two-step procedure is developed as follows to guarantee the 100% accuracy rate for each of the component networks with regard to the corresponding training samples. First, the experimental analysis is conducted to explore the proper combination of neurons in each of the layers, where the numbers of neurons in the input, hidden, and output layer are assigned to 10, 2, and 1, respectively for this study. Next, an iterative training scheme is designed with random initial network biases and weights such that each time the network is initialized at a different point of the weight space. The process continues until the maximum accuracy rate is achieved.

- ***Combining the Predictions***

In order to effectively traverse and combine the predictions of the component networks, a novel weighting scheme is proposed in this study. The basic idea is stated as follows.

The N bootstrapped data sets constructed earlier are used to train the N component neural networks independently, which in turn form the neural network ensemble. As mentioned earlier, each component network is trained to achieve the maximum accuracy upon the corresponding training subset, which might result in over-fitting and could exhibit a lower performance on a different subset. To test the performance of the component networks, each of them can be tested upon all the training subsets, where the one with a higher generalization ability gets a higher weight. This idea is realized in the following algorithm and the experimental results prove the effectiveness of the approach.

Let P_{ji} and R_{ji} be the precision and recall rates obtained by applying the j^{th} ($j = 1, 2, \ldots, N$) network on the training subset i ($i = 1, 2, \ldots, N$). The following procedure calculates the weights W_j for each of the component network such that those networks who consistently yield higher precision and recall values will get a higher weight. Define σ^p, σ^r as the deviation of precision and recall from 100%, respectively. Then σ^p can be calculated using Eq. (2).

$$\sigma_j^p = \sqrt{\frac{\sum_{i=1}^{N}(1 - P_{ji})^2}{N}}, \qquad (2)$$

where σ_j^p is the σ^p value of the j^{th} network. The weight W_j^p associated with the precision of the j^{th} network is defined as below.

$$W_j^p = \frac{1 - \sigma_j^p}{\sum_{j=1}^{N}(1 - \sigma_j^p)}. \qquad (3)$$

As can be observed from Eq. (3), if the deviation of the precision from 100% is lower compared to the others (i.e., the precision is consistently close to 100%), that network will get a higher weight. Similarly, σ_j^r and W_j^r values can be calculated using Eqs. (4) and (5). In addition, Eq. (6) defines the final weight for the j^{th} component network. It is worth noting that the weights are normalized so that all the weights sum to 1 for a certain ensemble.

$$\sigma_j^r = \sqrt{\frac{\sum_{i=1}^{N}(1 - R_{ji})^2}{N}}; \qquad (4)$$

$$W_j^r = \frac{1 - \sigma_j^r}{\sum_{j=1}^{N}(1 - \sigma_j^r)}; \qquad (5)$$

$$W_j = \frac{W_j^p W_j^r}{\sum_{j=1}^{N}(W_j^p W_j^r)}. \qquad (6)$$

Note that in our proposed framework, all the goal instances were used in the training process to construct each of the component networks. Therefore, the recall value reaches 100% in all the cases and it will show an equal effect on all the component networks. In other words, the weights are solely dependent on the precision value and are targeted to improve the overall precision value of the combined output. Nevertheless, the recall value is considered in the equations so that the idea can be generalized into other situations. This trained network ensemble is tested on the pre-filtered testing data set. Here, pre-filtering is employed to reduce the noisy and inconsistent data by using a limited set of domain rules, which further improves the precision of the detection. A detailed discussion regarding the pre-filtering process can be found in [5].

As mentioned earlier, each of the component networks in the ensemble has the log-sigmoid transfer function at the output layer. Therefore, the output produced by the network could lie anywhere between 0 and 1. Let $f_j(x)$ be the function computed by the j^{th} component network, the final results of the ensemble can be calculated as shown in Eq. (7). Note that since the log sigmoid function is symmetrical around coordinate (0, 0.5), intuitively the classification can be conducted by setting the threshold of decision as 0.5.

$$output = \sum_{j=1}^{N} W_j f_j(x). \qquad (7)$$

4 Empirical Study

In our experiments, 26 soccer videos were collected from different broadcasters, with a total time duration of 8 hours and 25 minutes. These videos are composed of 4,247 shots, out of which 37 are goals which account for 0.87% of the total shots.

4.1 Experiment Setup

The data set is randomly divided into a training set and a testing set, where the training set constitutes about 2/3 of the data set and the rest goes to the testing set. Seven such groups are formed randomly to employ 7-fold cross validation. As discussed in Section 3, the non-goal shots in the training set are randomly sub-sampled into N sets and the goal shots are bootstrapped into each set. Given m and n as the numbers of the goal and non-goal events, respectively, in the training set, the number N is determined by the required non-goal to goal ratio in the resulting sub-samples, i.e., the parameter r in Eq. (1).

Three experiments are designed to test the effectiveness of our proposed framework. In the first experiment, a single three-layer feed-forward neural network is trained using all the available data in the training set. In the second experiment, three sub-sampling schemes with different goal to non-goal ratios, namely 1:1, 1:2 and 1:3, are adopted and the performance of each approach is evaluated. In the third experiment, the component network predictions are combined using the proposed weighting scheme, whose performance is compared with the popularity voting method, Basic Ensemble method, and Dynamic Ensemble method. The outputs of the Basic Ensemble method [20] and Dynamic Ensemble method [14] are defined in Eq. (8) and Eq. (9), respectively.

$$f_{BEM} = \frac{1}{N}\sum_{j=1}^{N} f_j(x). \tag{8}$$

$$f_{DEM} = \sum_{j=1}^{N} w_j f_j(x), \quad \text{where} \quad w_j = \frac{c(f_j(x))}{\sum_{i=1}^{N} c(f_i(x))}. \tag{9}$$

Here, $c(y)$ is the certainty of a neural network output and is defined by:

$$c(y) = \begin{cases} y, & \text{if } y \geq 1/2; \\ 1-y, & \text{otherwise.} \end{cases} \tag{10}$$

Note that in this method, the weights are dynamic in the sense that they are recomputed each time the ensemble output is evaluated. In addition, in Tables 1 to 6, “RC”, “PR”, “Ident.” and “Mis-ident.” are used to denote “Recall”, “Precision”, “Identified” and “Mis-identified”, respectively.

4.2 Result Analysis

Experiment 1: Single Neural Network

In the first experiment, the performance of a single three-layer feed-forward neural network is evaluated. As shown in Table 1, for each of the cross-validation sets, the recall value is much lower than the precision value, which demonstrates the weakness of using neural networks in rare event detection where the influence of the rare event instances in the training process is overshadowed by the much higher number of the nonevent instances. However, as discussed earlier, the recall value is a more important measurement for event detection in the sense that it is desirable to detect as many events as possible even at the expense of adding a reasonable number of false positives. As can be observed in the following two experiments, bootstrapped sampling can greatly improve the recall of the detection.

Table 1. Cross validation results for the single neural network

Data set	Total	Ident.	Missed	Mis-ident.	RC (%)	PR (%)
1	12	9	3	0	75.00	100.00
2	12	10	2	1	83.33	90.91
3	16	12	4	1	75.00	92.31
4	10	8	2	0	80.00	100.00
5	10	8	2	1	80.00	88.89
6	13	8	5	4	61.54	66.67
7	17	14	3	1	82.35	93.33
Avg.	13	10	3	1	76.75	90.30

Experiment 2: Bootstrapped Sampling

In the second experiment, three different sampling schemes are adopted by changing the values of the parameter r from 1 to 3 in Eq. (1), and the results are presented in Tables 2, 3, and 4, respectively. As the purpose of this experiment is to demonstrate the effect of bootstrapped sampling, the popularity voting method, instead of our proposed weighting scheme, is applied to combine the predictions.

Table 2. Cross validation results for 1:1 goal to non-goal bootstrapped sampling

Data set	Total	Ident.	Missed	Mis-ident.	RC (%)	PR (%)
1	12	12	0	1	100.00	92.31
2	12	12	0	2	100.00	85.71
3	16	16	0	2	100.00	88.89
4	10	10	0	2	100.00	83.33
5	10	10	0	2	100.00	83.33
6	13	12	1	9	92.31	57.14
7	17	17	0	9	100.00	65.38
Avg.	13	13	0	4	98.90	79.44

Table 5 summarizes the average recall and precision values for the three sampling schemes. The results show that the decrement of the goal to non-goal ratio in the samples results in a decrease in the recall value and an increase in the

Table 3. Cross validation results for 1:2 goal to non-goal bootstrapped sampling

Data set	Total	Ident.	Missed	Mis-ident.	RC (%)	PR (%)
1	12	12	0	1	100.00	92.31
2	12	12	0	1	100.00	92.31
3	16	16	0	1	100.00	94.12
4	10	10	0	1	100.00	90.91
5	10	10	0	1	100.00	90.91
6	13	11	2	9	84.62	55.00
7	17	17	0	7	100.00	70.83
Avg.	13	13	0	3	97.80	83.77

Table 4. Cross validation results for 1:3 goal to non-goal bootstrapped sampling

Data set	Total	Ident.	Missed	Mis-ident.	RC (%)	PR (%)
1	12	11	1	1	91.67	91.67
2	12	11	1	1	91.67	91.67
3	16	16	0	1	100.00	94.12
4	10	10	0	1	100.00	90.91
5	10	10	0	1	100.00	90.91
6	13	11	2	8	84.62	57.89
7	17	17	0	6	100.00	73.91
Avg.	13	12	1	3	95.42	84.44

precision value. This observation follows our discussion in Section 3 about bootstrapped sampling. In addition, the network ensemble with bootstrapped sampling outperforms the single neural network in all the cases in the sense that the recall value improves dramatically with a small reduction in the precision value.

Experiment 3: Combining Predictions

In the third experiment, the performance of the proposed weighting scheme is compared with the popularity voting method (denoted by "Simple Maj."), Basic Ensemble method (denoted by "BEM"), and Dynamic Ensemble method (denoted by "DEM") under the three different bootstrapped sampling schemes in combining the component network predictions. Table 6 and Figure 2 present the performance comparison of the four methods when they are applied on the same network ensembles, where the average precision and recall values for 7-fold cross validation are used. As can be seen from the results, the proposed weighting scheme improves the precision of goal event detection without causing much reduction in the recall value. More precisely, the recall value only decreases in the case of 1:3 bootstrapped sampling, whereas it stays stable or even increases in the other two cases. This demonstrates the effectiveness of our proposed weighting scheme.

Table 5. Comparison of results between three sub-sampling schemas

Sampling	RC	PR
1-to-1	98.90%	79.44%
1-to-2	97.80%	83.77%
1-to-3	95.42%	84.44%

Table 6. Comparison of results between prediction combining techniques

Technique		Simple Maj.	BEM	DEM	Proposed Method
1:1 bootstrap	RC (%)	98.90	98.90	98.90	98.90
	PR (%)	79.44	79.82	79.44	82.73
1:2 bootstrap	RC (%)	97.80	97.80	97.80	98.90
	PR (%)	83.77	83.77	83.77	84.92
1:3 bootstrap	RC (%)	95.42	96.52	96.52	95.42
	PR (%)	84.44	84.74	84.74	85.41

5 Conclusions and Future Work

In this paper, an advanced framework for goal event detection in the soccer videos is proposed, which uses multimodal processing together with the classification power of neural network ensembles. The bootstrapped sampling scheme is adopted in our framework to address the challenges caused by the rareness of the events. In addition, the proposed weighting scheme presents strong capabilities in traversing and combining the predictions of all the component networks. The experimental results from diverse video sources fully demonstrate the viability and effectiveness of the proposed for semantic event detection. Currently, the framework is applied for the detection of a single event. Meanwhile, we are working on extending the framework for multiple event detection at different domains.

References

[1] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, November 1993.

[2] J. Assfalg, M. Bertini, C. Colombo, and A. D. Bimbo. Semantic annotation of sports videos. *IEEE Multimedia*, 9(2):52–60, April-June 2002.

[3] P. L. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, March 1998.

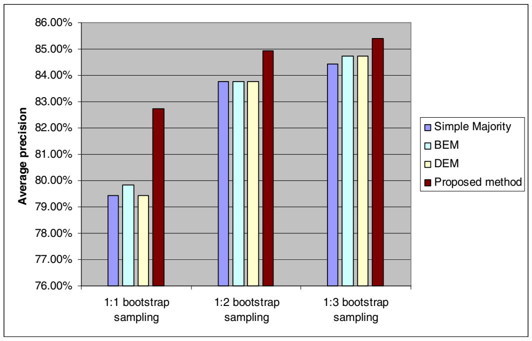

Figure 2. Precision of goal detection of four different prediction combining methods.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] S.-C. Chen, M.-L. Shyu, M. Chen, and C. Zhang. A decision tree-based multimodal data mining framework for soccer goal detection. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 265–268, Taipei, Taiwan, June 2004.

[6] S.-C. Chen, M.-L. Shyu, and C. Zhang. Innovative shot boundary detection for video indexing. In S. Deb, editor, *Video Data Management and Information Retrieval*, pages 217–236. Idea Group Publishing, Hershey, 2005.

[7] S.-C. Chen, M.-L. Shyu, C. Zhang, and M. Chen. A multimodal data mining framework for soccer goal detection based on decision tree logic, accepted for publication. *International Journal of Computer Applications in Technology, Special Issue on Data Mining Applications*, 2005.

[8] K. J. Cherkauer. Human expert level performance on a scientific image analysis task by a system using combined artificial neural networks. In *Proceedings of AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, pages 15–21, Portland, OR, 1996. AAAI Press.

[9] W. Choe, O. K. Ersoy, and M. Bina. Neural network schemes for detecting rare events in human genomic dna. *Bioinformatics*, 16(12):1062–1072, December 2000.

[10] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, July 2003.

[11] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, October 1990.

[12] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[13] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.

[14] D. Jimenez. Dynamically weighted ensemble neural networks for classification. *Proceedings of IJCNN-98*, 1:753–756, May 1998.

[15] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, 7:231–238, 1995.

[16] R. Kurino, M. Sugisaka, and K. Shibata. Growing neural network for acquisition of 2-layer structure. *Proceedings of the International Joint Conference on Neural Networks*, 4:2512–2517, July 2003.

[17] S. Lawrence, L. C. Giles, and A. C. Tsoi. Lessons in neural network training: Over-fitting may be harder than expected. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 540–45, Menlo Park, California, 1997. AAAI Press.

[18] W.-N. Lie, T.-C. Lin, and S.-H. Hsia. Motion-based event detection and semantic classification for baseball sport videos. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 1567–1570, Taipei, Taiwan, June 2004.

[19] R. Maclin and J. W. Shavlik. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *Proceedings of IJCAI-95*, pages 524–530, Montreal, Canada, August 1995.

[20] M. P. Perrone and L. N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In R. J. Mammone, editor, *Neural Networks for Speech and Vision*, pages 127–142. Chapman-Hall, London, UK, 1993.

[21] P. L. Rosin and F. Fierens. Improving neural network generalization. In *Proceedings of International Geoscience and Remote Sensing Symposium - IGARSS '95*, volume 2, pages 1255–1257, Firenze, Italy, July 1995.

[22] D. Sadlier and N. E. O'Connor. Event detection in field-sports video using audio-visual features and a support vector machine. *IEEE Transactions on Circuits and Systems for Video Technology (accepted for publication)*, 2005.

[23] P. Sollich and A. Krogh. Learning with ensembles: How over-fitting can be useful. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 190–196. MIT Press, Cambridge, MA, 1996.

[24] H. Sun, J.-H. Lim, Q. Tian, and M. S. Kankanhalli. Semantic labeling of soccer video. In *Proceedings of IEEE Pacific-Rim Conference on Multimedia (ICICS-PCM)*, volume 3, pages 1787–1791, Singapore, December 2003.

[25] V. Tovinkere and R. J. Qian. Detecting semantic events in soccer games: Towards a complete solution. In *IEEE International Conference on Multimedia and Expo*, pages 833–836, Tokyo, Japan, August 2001.

[26] K. W. Wan, X. Yan, and C. Xu. Automatic mobile sports highlights. In *IEEE International Conference on Multimedia and Expo*, Amsterdam, Netherlands, July 2005.

[27] J. Wang, C. Xu, E. Chng, K. Wah, and Q. Tian. Automatic replay generation for soccer video broadcasting. In *Proceedings of ACM Multimedia*, pages 32–39, New York, USA, 2004. ACM Press.

[28] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden markov models. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 29–32, Baltimore, MD, July 2003.

[29] M. Xu, N. C. Maddage, C. Xu, M. Kankanhalli, and Q. Tian. Creating audio keywords for event detection in soccer video. In *IEEE International Conference on Multimedia and Expo*, pages 281–284, Baltimore, MD, July 2003.

[30] X. Yu and D. Farin. Current and emerging topics in sports video processing. In *IEEE International Conference on Multimedia and Expo*, Amsterdam, Netherlands, July 2005.

[31] Z.-H. Zhou, J. Wu, and W. Tang. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137(1-2):239–263, 2002.