applied sciences

MDPI

Article

Shot Classification of Field Sports Videos Using AlexNet Convolutional Neural Network

Rabia A. Minhas [1], Ali Javed [1], Aun Irtaza [2], Muhammad Tariq Mahmood [3] and Young Bok Joo [3,*]

[1] Department of Software Engineering, University of Engineering and Technology, Taxila 47050, Pakistan; afzalminhasrabia@gmail.com (R.A.M); ali.javed@uettaxila.edu.pk (A.J.)

[2] Department of Computer Science, University of Engineering and Technology, Taxila 47050, Pakistan; aun.irtaza@uettaxila.edu.pk

[3] School of Computer Science and Engineering, Korea University of Technology and Education, 1600 Chungjeolno, Byeogchunmyun, Cheonan 31253, South Korea; tariq@koreatech.ac.kr

* Correspondence: ybjoo@koreatech.ac.kr; Tel.: +82-041-560-1487

Received: 25 November 2018; Accepted: 28 January 2019; Published: 30 January 2019

Abstract: Broadcasters produce enormous numbers of sport videos in cyberspace due to massive viewership and commercial benefits. Manual processing of such content for selecting the important game segments is a laborious activity; therefore, automatic video content analysis techniques are required to effectively handle the huge sports video repositories. The sports video content analysis techniques consider the shot classification as a fundamental step to enhance the probability of achieving better accuracy for various important tasks, i.e., video summarization, key-events selection, and to suppress the misclassification rates. Therefore, in this research work, we propose an effective shot classification method based on AlexNet Convolutional Neural Networks (AlexNet CNN) for field sports videos. The proposed method has an eight-layered network that consists of five convolutional layers and three fully connected layers to classify the shots into long, medium, close-up, and out-of-the-field shots. Through the response normalization and the dropout layers on the feature maps we boosted the overall training and validation performance evaluated over a diverse dataset of cricket and soccer videos. In comparison to Support Vector Machine (SVM), Extreme Learning Machine (ELM), K-Nearest Neighbors (KNN), and standard Convolution Neural Network (CNN), our model achieves the maximum accuracy of 94.07%. Performance comparison against baseline state-of-the-art shot classification approaches are also conducted to prove the superiority of the proposed approach.

Keywords: AlexNet CNN; convolutional neural networks; deep learning; rectified linear unit layer; shot classification

1. Introduction

The last few decades have witnessed the production and transmission of a massive amount of multimedia content in the World Wide Web. Sports videos contribute a major portion of the available multimedia content in cyberspace. Sports video content analysis has been explored heavily due to the potential commercial benefits and massive viewership around the world. The manual annotation of such an enormous video content is a challenging task. Therefore, automated methods for sports video content analysis and management are required. Humans are inclined towards meaningful concepts, while viewing and exploring interactive databases; there is a growing need for indexing and semantic video content analysis. The major goal towards video content analysis and management is to provide semantic information relevant to the user. For this purpose, researchers have proposed various automated and semi-automated techniques to analyze the sports videos for shot classification, key-events detection, and video summarization.

Shot Classification approaches are commonly applied initially for sports video summarization and content retrieval applications to provide the semantic information. Shot Classification techniques are employed to categorize the sports videos into different views, i.e., long, medium, close-up, etc. The classification makes it convenient to further process the sports videos for various applications mentioned above. The effectiveness of these applications largely depends on accurate shot classification of the sports videos. However, shot classification is very challenging in the presence of camera variations, scene change, action speeds, illumination conditions (i.e., daylight, artificial light, and shadow), etc. Therefore, there exists a need to develop an effective shot classification method that is robust to the above-mentioned limitations. The main idea is to develop a shot classification scheme that can accurately classify the frames of field sports videos in the presence of the aforementioned limitations. Soccer and Cricket field sports are selected to evaluate the effectiveness of our framework due to the massive viewership of these sports across the globe.

Sports videos contain different types of shots that can be categorized into following categories: (i) Long shot, (ii) medium shot, (iii) close-up shot, and (iv) crowd/out-of-the-field shots. Long shots in sports videos are considered zoomed-out shots that covers a large area or a landscape of the field, the audience and the players are presented on a small scale. Medium shots are zoomed-in shots that mostly cover a full body of the player from top head of the body to the bottom of the feet or more than one player within the playground field. Close-up shots are the most zoomed-in shots that contain the players upper body footage in the scene. Crowd shots/out-of-the-field shots consist of viewers/audience footage in the sports videos.

Most of the existing methods classify the in-field segments of sports videos, i.e., long, medium, and close-up shots as these shots are more useful to generate the highlights. After viewing massive number of sports videos, it has been observed that crowd/out-of-the-field shots also contain few exciting events that can be included in the summarized video. Therefore, we proposed an automated shot classification method of sports videos into long, medium, close-up and crowd/out-of-the-field shots. Shot classification methods have applications in various domains, i.e., sports [1], medical [2], news [3], entertainment [4], and documentaries [5].

Existing approaches of shot classification can be categorized into (i) learning-based, and (ii) non-learning-based. Learning-based approaches employ various classifiers (i.e., Support Vector Machines (SVM), Kth Nearest Neighbor (KNN), decision trees, neural networks, etc.) to perform shot classification. Whereas, non-learning-based approaches use various statistical features, i.e., dominant grass color, histogram difference comparison, etc., to design different shot classification methods.

Existing methods [6–10] have used non-learning-based approaches for shot classification of sports videos. Divya et al. [6] proposed an effective shot classification method for basketball videos. Edge pixel ratio was used to classify the shots into two categories only: (i) Close-up view, and (ii) long view. Logo transition detection and logo sample selection were used to identify the replays. Rahul et al. [7] proposed a method to assign the semantic information to cricket videos. The input video was segmented into shots using the scene information extracted from the text commentary. Video shots were classified into different categories, i.e., batsman action of pull shot, etc. Rodrigo et al. [8] proposed a shot classification method for soccer videos using the Bhattacharyya distance [9]. Murat et al. [10] proposed a soccer video summarization method using cinematic and objects-based features. The dominant field color was used to detect the soccer field. Grass pixel ratio and color histogram comparison were used to detect the shot boundaries. Finally, statistical analysis was used to classify the shots into long, medium, and close-up shots.

In literature, approaches [11–22] also use learning-based techniques for shot classification. Learning-based approaches provide better accuracy as compared to non-learning-based approaches, but at the expense of increased computational cost. Rahul et al. [11] presented an automated approach to identify the types of views and detection of the significant events such as goal, foul, etc, using bag of words and SVM. Sigari et al. [12] used rule-based heuristics and SVM to classify the shots into far view/long shot, medium view, close-up view, and out-field view. Ling-Yu et al. [13] proposed a framework for semantic shot classification in sports videos. Low-level features were derived using color, texture, and DC images in compressed domain. These low-level features were converted to mid-level features such as camera motion, action regions and field shape properties, etc. Mid-level representation was necessary to use the classifiers that do not map on low-level features. Finally, shots were classified using different supervised learning algorithms like decision trees, neural networks, support vectors machine (SVM), etc. Camera shots were classified into mid shots, long shots, and close-up shots. Matthew et al. [14] proposed and evaluated deep neural network architectures that were able to combine image information across a video over longer time periods. They [14] have proposed two methods for handling full length videos. Results were evaluated over previously published datasets like Sports 1 million dataset, UCF-101 Dataset. The first method explored convolutional neural networks that was used to examine the design choices needed while adapting to this framework. The second method used an ordered sequence of frames that were employed to recurrent neural network based on Long Short-Term Memory (LSTM) cells connected to the output of underlying CNN. Karmaker et al. [15] proposed a cricket shot classification approach using motion vector detection through the optical flow method. 3D MACH filter was used for action recognition that was trained over six cricket videos to classify the different shots. Kapela et al. [16] proposed a learning-based method that used feed-forward neural networks, ELMAN neural network, and decision trees to classify the different events in field sports videos. Konstantinos et al. [17] presented a shot classification model using linear discriminant analysis (LDA) method to categorize the shots into full long view and player medium view. Atika et al. [18] proposed a shot classification approach using statistical features to classify the crowd shots. Nisarg et al. [19] proposed a multi labelled video dataset that contained over eight million videos, 500k hours of video, annotated with 4800 visual entities. The videos were labelled using Youtube vide annotation system. Each video was decoded using Deep CNN pre-trained on ImageNet to extract hidden representation immediately. Training was performed on different classification models on the dataset. The videos were classified onto different categories like vehicles, sports, concert, and animated, etc. Jungheon et al. [20] proposed a video event classification algorithm using audio-visual features. Convolutional neural networks were applied on the frames to extract the features followed by performing the classification. In addition, Mel Frequency Cepstral Coefficients (MFCCs) were also used to train the CNN for shot classification. Loong et al. [21] proposed a semantic shot classification algorithm for cinematography. Markov random field model based on motion segmentation was used to classify the film video into three types of shots, i.e., long, medium, and close-up. Ashok et al. [22], proposed a hybrid classifier-based approach for activity detection of cricket videos. Low-level features (i.e., grass pixel ratio) and mid-level features (i.e., camera view, distance, etc.) were used to train a hybrid classifier comprising of Naïve bias, KNN and multi-class SVM for shot classification into field-view and non-field views.

As we already discussed that effective shot classification improves the accuracy of video content analysis applications. However, shot classification is very challenging in the presence of camera variations, scene change, action speeds, illumination conditions (i.e., daylight, artificial light, shadow), etc. To address the challenges associated with shot classification, we proposed an effective shot classification framework for field sports videos. The major contributions of the proposed research work are as follows:

- AlexNet convolution neural network is designed to effectively classify the video shots into different views (i.e., long, medium, close-up, crowd/out-of-the-field) which is promising and novel in terms of its application to shot classification.
- The proposed framework is robust to camera variations, scene change, action speeds, and illumination conditions and can be reliably used for shot classification in the presence of these limitations.

Moreover, existing shot classification approaches focus on generating more classes for in-field segments (long, medium and close-up shots) because in-field segments are commonly used to generate the sports highlights. However, it has been observed after watching many sports videos that the crowd shots also contain few exciting segments that can be included in the summarized video. Therefore, the proposed research work focuses on classifying the shots of field sports videos into long, medium, close-up, and crowd/out-of-the-field shots. We categorize the crowd shots into separate class so that these shots can also be analyzed to further identify the interesting segments for video summarization. These crowd shots can also be used for different applications, i.e., activity detection, etc. The details of the proposed deep learning framework are provided in the next section.

2. Proposed Method

This section provides a comprehensive discussion on the proposed shot classification method. The proposed deep learning scheme applies the AlexNet Convolutional Neural Network (CNN) architecture to classify the shots into long, medium, close-up, and crowd/out-of-the-field shots. The input layer of the CNN is in three dimensions: The width, height, and depth of the pixel. The width and height represent horizontal and vertical pixels, whereas depth represents the RGB color channel. We have transformed the input sports video dataset in frames to reduce computational complexity involved in training the network. The process flow of the proposed framework is presented in Figure 1.

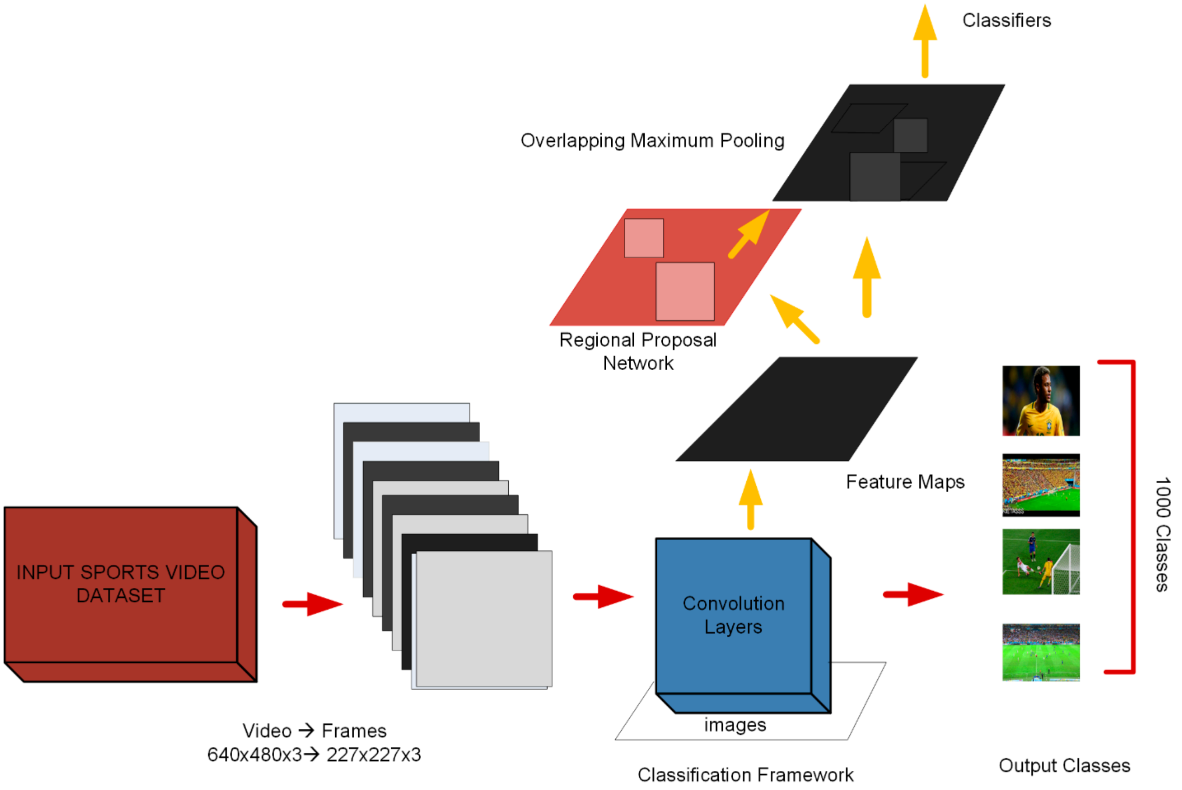

Figure 1. Proposed framework for shot classification.

2.1. AlexNet Architecture

In the proposed work, we employed the Alexnet CNN deep learning architecture [23] for shot classification in field sports videos. The network is deeper than standard CNN with five convolution layers followed by three maximum pooling layers. Dropout of 0.5% is applied on the fully connected layers to avoid over fitting of the data. The architecture consists of the following components:

- 1 Convolution with 11 × 11 kernel size (1CONV)
- Rectified Linear Unit Layer Activation (RELU)

- Response Normalization Layer
- 1 Maximum Pooling (4 × 4 kernel)
- 2 Convolution with 5 × 5 kernel size (2CONV)
- Rectified Linear Unit Layer (RELU)
- Response Normalization Layer
- 2 Maximum Pooling (3 × 3)
- 3 Convolution with 3 × 3 kernel size (3CONV)
- Rectified Linear Unit Layer Activation (RELU)
- 4 Convolution with 3 × 3 kernel size
- Rectified Linear Unit Layer Activation (RELU)
- 3 Maximum Pooling (3 × 3)
- Fully Connected Layer (4096 nodes)
- Rectified Linear Unit Layer Activation (RELU)
- Fully Connected Layer (4096 nodes)
- Rectified Linear Unit Layer (RELU)
- Soft-max out

The proposed AlexNet CNN architecture is presented in Figure 2. In the proposed research work, image input layer is defined as a pre-processing layer where the input frames are down-sampled from 640 × 480 to 227 × 227 in terms of spatial resolution to reduce the computational cost of our deep learning framework. The proposed system uses five convolutional (CONV) layers followed by three pooling layers (POOL) and Rectified Linear Unit (RELU). For the first convolutional layer, 96 kernels of relatively large size 11 × 11 × 3 are used. For the second convolutional layer, 256 kernels of size 5 × 5 are used. For the third, fourth, and fifth layers, 384 kernels of size 3 × 3 are used. Each convolutional layer generates a feature map. The feature maps of first, second and fifth convolutional layer are used in combination with pooling layers of 3 × 3 and stride of 2 × 2. The framework is comprised of eight layered architecture with 4096 nodes. This generates the trainable feature maps, i.e., feature extraction phenomena are performed in these layers. These feature maps are subjected to fully connected layers (FC) and then Soft-max activation is performed to determine the classification probabilities used by the final output classification layer. These classification probabilities in the Soft-max layer can make categories up to 1000 different classes, but in our dataset, we have four classes.

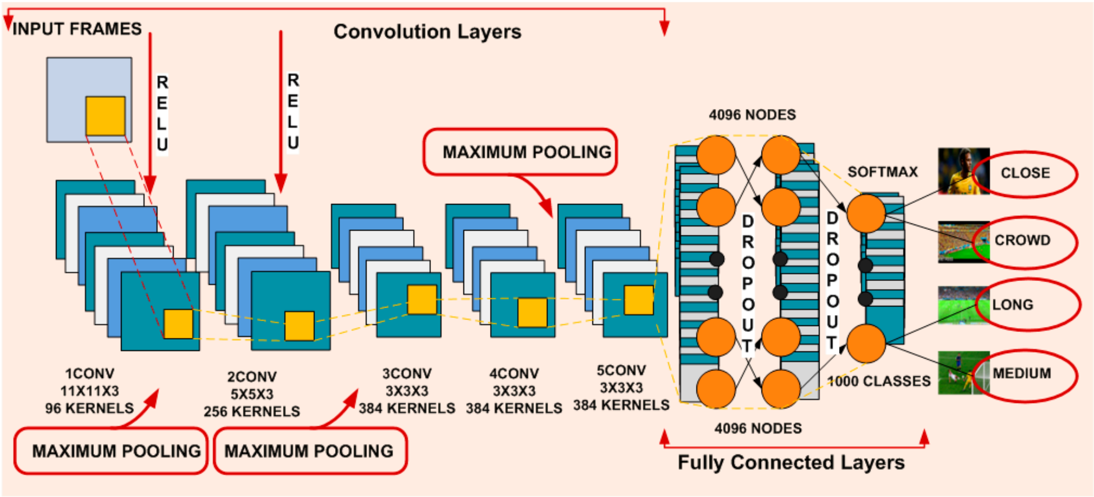

Figure 2. AlexNet convolution neural network architecture of the proposed framework.

2.1.1. Convolution Network Layer

This is the most significant layer in deep learning phenomena of neural networks that generates the feature maps which are subjected to classification layers. It consists of a kernel that slides over the input frame, which generates the output known as feature map. At every location on the input, we performed matrix multiplication followed by integrating the result. The output feature map is defined as:

$$N_x^r = \frac{N_x^{r-1} - L_x^r}{S_x^r} + 1;\; N_y^r = \frac{N_y^{r-1} - L_y^r}{S_y^r} + 1 \tag{1}$$

where (N_x, N_y) is the width and height of the output feature map of the last layer and (L_x, L_y) is the kernel size, ($S_{x,}$, S_y) that defines the number of pixels skipped by the kernel in horizontal and vertical directions and index r indicates the layer i.e., r = 1. Convolution is applied on the input feature map and a kernel to get the output feature map that is defined as:

$$X_1(m,n) = (J * R)(m,n) \tag{2}$$

where X_1 (*m*, *n*) is a two-dimensional output feature map obtained by convolving the two-dimensional kernel *R* of size (L_x, L_y) and input feature map *J*. The sign * is used to represent the convolution between *J* and *R*. The convolution operation is expressed as;

$$X_1(m,n) = \sum_{p=-\frac{L_x}{2}}^{p=+\frac{L_x}{2}} \sum_{q=-\frac{L_y}{2}}^{q=+\frac{L_y}{2}} J(m-p, n-q)R(p,q) \tag{3}$$

In the proposed framework, we used five CONV layers with RELU layer and response normalization layer to extract the maximum feature maps form the input frames to train the dataset with maximum accuracy.

2.1.2. Rectified Linear Unit Layer

In the next stage, we applied the RELU activation function to all the trainable layers to strengthen our network by making it non-linear. This layer accounts the non-linearities in an adequate manner. It is applied over the output feature map which is generated from the convolutional layer. The use of tanh(.) and RELU activation function saturates the non-linear gradient descent in terms of training time. tanh(.) is expressed as:

$$X_2(m,n) = \tanh(X_1(m,n)) = \frac{sinh(X_1(m,n))}{cosh(X_1(m,n))} = 1 + \frac{1 - e^{-2*X_1(m,n)}}{1 + e^{-2*X_1(m,n)}} \tag{4}$$

where $X_2(m,\ n)$ is a two-dimensional output feature map after applying tanh(.) on the input feature map $X_1(m,n)$, which is achieved after passing through the convolutional layer. The values in the final feature map are obtained after applying the RELU function as follows:

$$X(m,n) = \begin{cases} 0, & if\ X_2(m, n < 0) \\ X_2(m,n), & if\ X_2(m, n \geq 0) \end{cases} \tag{5}$$

where *X(m, n)* is obtained by transforming the negative values into zero and returns the same value back on receiving any positive value. We included the RELU layer in our proposed framework since deep convolutional neural networks train at a much faster pace when intact with the RELU layer.

2.1.3. Maximum Pooling Layer

A pooling layer is included in the proposed architecture after first and second convolution layer and then after the fifth convolution layer to decrease the spatial size of each frame to reduce the computational cost of the proposed deep learning framework. The pooling operation usually averages or simply pick the maximum value for each slice of the image. In the proposed work, we apply pooling by using the maximum value against each slice as we obtained better results on this setting. The application of the maximum pooling layer on the activation output for down-sampling the images is demonstrated in Figure 3.

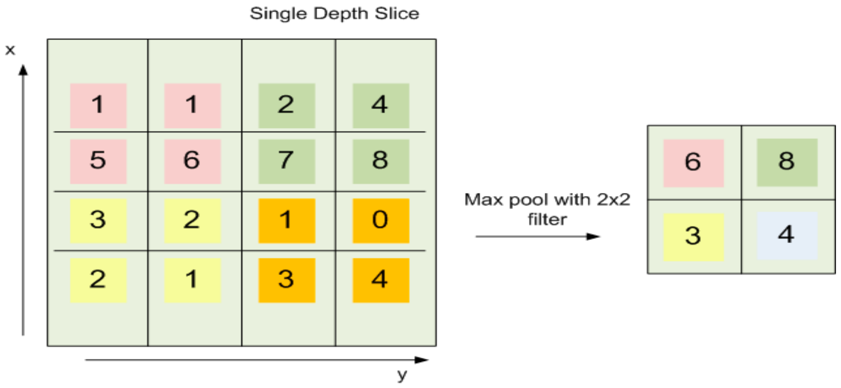

Figure 3. Maximum pooling layer.

2.1.4. Response Normalization Layer and the Softmax Activation

Response Normalization is performed after first two sessions to reduce the test error rate of the proposed network. This layer normalizes the input layers within networks along with the input of entire network. Normalization is performed as follows:

$$\mathrm{N}_{\mathrm{e,q}}^{\mathrm{x}} = \frac{\mathrm{b}_{\mathrm{e,f}}^{\mathrm{x}}}{\left(\mathrm{z} + \alpha \sum_{\mathrm{j=max(0,x-c/2)}}^{\mathrm{min(T-1,x+c/2)}} \left(\mathrm{b}_{\mathrm{e,f}}^{\mathrm{x}}\right)^2\right)^{\gamma}} \tag{6}$$

where $N_{e,f}^{x}$ represents the normalization of activity $b_{e,f}^{x}$ of neurons which is computed at position (e,f) with the use of kernel k. T is the total range of kernels within the layers. z, c, α, and γ are the constants hyperparameters and their values are adjusted by applying a validation set respectively.

Soft-max is a classifier on top of the extracted features. After performing five series of the convolutional network layer, the output is fed to the Soft-max layer for multi class classification that helps to determine the classification probabilities. These probabilities are then used by the final classification layer to classify the frames into long, medium, close-up, and crowd/out-field views.

2.1.5. Dropout Layer

The dropout layer is applied in the first two fully connected layers when the number of iterations doubles in our network to avoid overfitting of the data by increasing number of iterations by a factor of two, making the neurons dense. It performs the model averaging with neural networks and is a very efficient technique to regularize training data. Maximum pooling layers, kernel sizes of convolutional layer, and their skipping factors are processed such that the output feature maps are down sampled to one pixel per map. Fully connected layer also connects the output of the top most layers to 1D feature vector. The upper layer is always completely connected with the output unit for class label, such that extracting high level features form the training data Figure 4 depicts the regularization technique on fully connected layers before and after applying dropout.

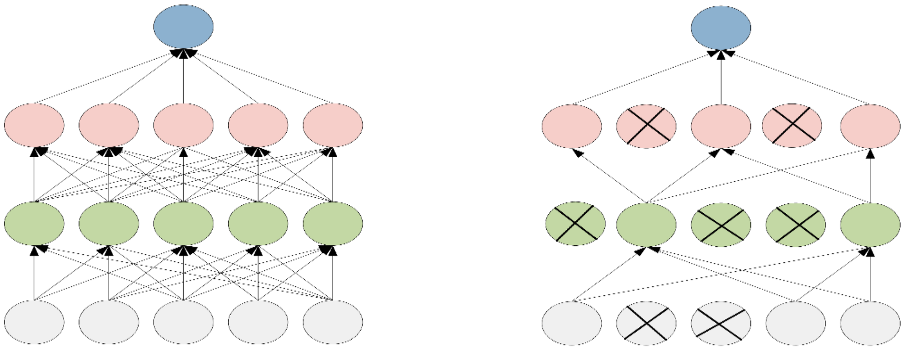

Figure 4. Fully connected layers (FC) before and after applying dropout.

3. Results and Discussion

This section presents the experiments designed to evaluate the performance of the proposed framework. The results of these experiments are also reported along with the discussion. Objective evaluation metrics (i.e., precision, recall, accuracy, error rate, F1-score) are employed for performance evaluation. The details of the dataset are also provided in this section.

3.1. Dataset

For performance evaluation we selected a diverse dataset comprising of soccer and cricket videos from YouTube as done by the comparative methods, i.e., [24–26]. The dataset includes 10 videos of 13 h from six major broadcasters, i.e., ESPN, Star Sports, Ten Sports, Sky Sports, Fox Sports, and Euro Sports. In addition, we included the sports videos of different genre and tournaments in our dataset. Cricket videos consist of 2014 One Day International (ODI) tournament between South Africa and New Zealand, 2006 ODI tournament between Australia and South Africa, 2014 test series and ODI series between Australia and Pakistan, 2014 ODI series between South Africa and New Zealand, 2014 Twenty20 cricket world-cup tournament, and 2015 ODI cricket world-cup tournament. For soccer videos, we considered the 2014 FIFA world-cup and 2016 Euro-cup videos. Soccer videos consist of Brazil vs. Germany semifinal match and Argentina vs. Germany final match of 2014 FIFA world-cup, Portugal vs. France final match and France vs. Germany semifinal match of 2016 Euro-cup.

Each video has a frame resolution of 640 × 480, frame rate of 30 fps and recorded in MPEG-1 format. The videos represent different illumination conditions (i.e., daylight, artificial lights). The dataset videos are comprised of different shot types, i.e., long, medium, close-up, and crowd shots, as shown in Figure 5. We used 70% frames of our dataset for training purpose and rest of the 30% for validation purpose. Our dataset videos can also be accessed at Reference [27] for research purposes.

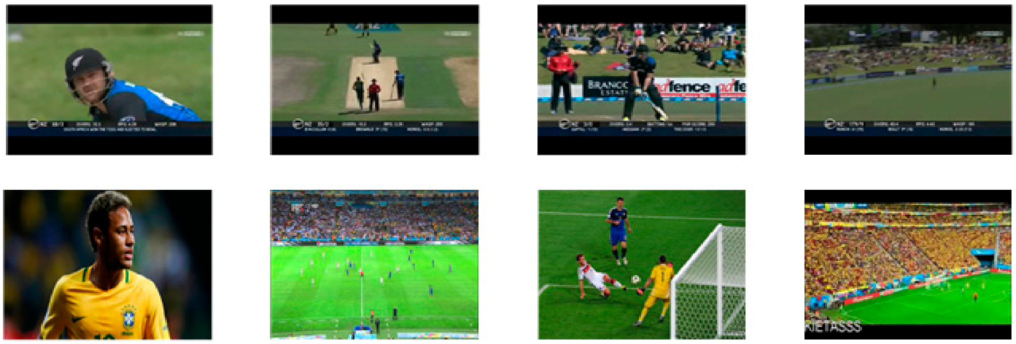

Figure 5. Snapshots for cricket (row-1) and soccer (row-2) videos.

3.2. Experimental Setup

We have trained our dataset using Alexnet CNN to classify four different classes presented in our dataset. Transfer learning of a network is presented in Figure 6.

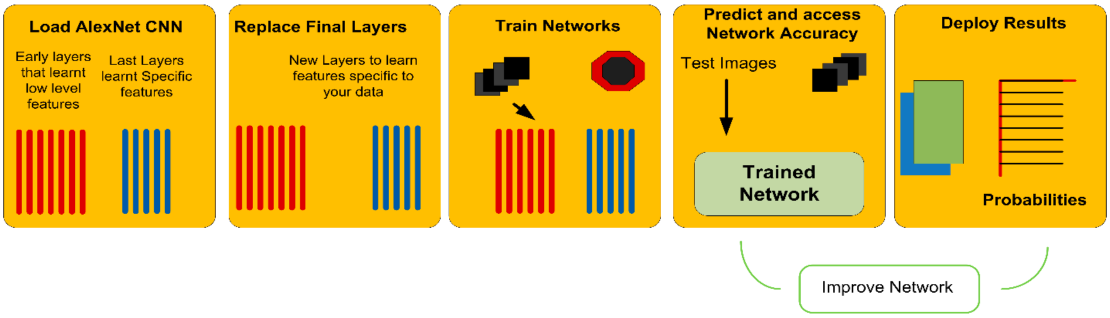
Load AlexNet CNN
Early layers that learnt low level features
Last Layers learnt Specific features
Replace Final Layers
New Layers to learn features specific to your data
Train Networks
Predict and access Network Accuracy
Test Images
Trained Network
Deploy Results
Probabilities
Improve Network

Figure 6. Transfer learning of AlexNet CNN network.

Training

The network takes four epochs in four to five days to train on two GTX 580 Graphic Processing Units (GPU). An epoch is the number of times training vectors are used once to update the weights. In our system, each epoch has 500 iterations for our dataset. A stochastic approximation of gradient descent is used to perform training iterations on the dataset. The stochastic gradient descent (SGD) is applied with a learning rate of 0.0001, momentum of 0.9 and weight decay of 0.0005, respectively. SGD is represented in Equations (7) and (8).

$$s_{l+1} := 0.9.\, s_l - 0.0005.\, \epsilon.\, x_l - \epsilon.\langle \frac{\partial L}{\partial x}\, |_{xl} \rangle_{B_l} \tag{7}$$

$$x_{l+1} = x_l + s_{l+1} \tag{8}$$

where l is the iteration index, s is the momentum variable, and ϵ is the learning rate. $\langle \frac{\partial L}{\partial x}\, |_{xl} \rangle_{B_l}$ is constant over the lth iteration of batch B_l of x evaluated at x_l. All the layers in our network have equal learning rate that can be adjusted during the training. Experiments have proved that, by increasing the learning features, validation set achieves better accuracy. We divided our dataset on videos level, that means we performed training on a dataset of soccer and cricket videos and tested the unique video dataset of soccer and cricket videos on the proposed network. Snapshots of the training progress are presented in Figure 7.

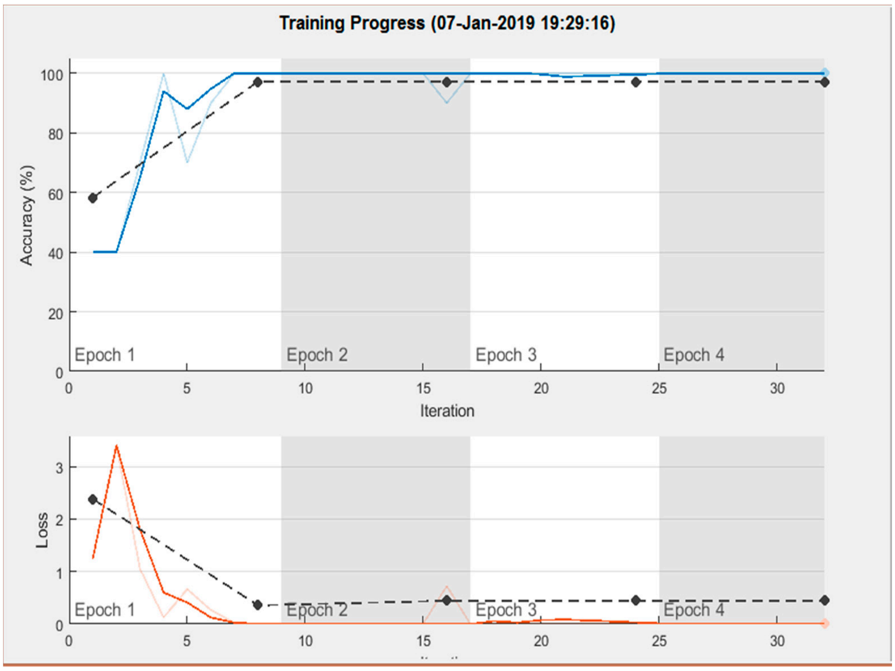

Figure 7. Training progress report.

3.3. Evaluation Parameters

We employed objective metrics to evaluate the performance of the proposed framework. For this purpose, we used precision (P), recall (R), accuracy (A), error rate (E), and F1-score to measure the performance. These metrics are computed in terms of correct/incorrect classification of shot types for each category. Finally, the results of all types of shots are averaged to obtain the final values.

For shot classification of sports videos, accuracy represents the ratio of correctly classified shots (True Positives and True Negatives) out of the total number of shots. Accuracy is computed as follows:

$$A = \frac{TP + TN}{P + N} \tag{9}$$

where true positives (TP) represent the correctly classified shots of a positive (P) class (i.e., long shot if we are measuring the accuracy of long shot). And, true negatives (TN) represent the correctly classified shots of negative (N) class (i.e., medium, close-up and crowd shots in case of measuring long shot accuracy).

Error rate refers to the ratio of misclassified shots (False Positives (FP) and False Negatives (FN)) to the total examined shots. Error rate is computed as:

$$E = \frac{FP + FN}{P + N} \tag{10}$$

where FP represent the N class shots misclassified as positive class shots. Additionally, FNrepresent the P class shot misclassified as the negative class shot.

Precision represents the ratio of correctly labelled shots over the total number of shots and computed as follows:

$$P = \frac{TP}{TP + FP} \tag{11}$$

Recall is the fraction of true detection of the shots over a total number of shots in the video and computed as:

$$R = \frac{TP}{TP + FN} \tag{12}$$

F1-score represents the harmonic mean of precision and recall. F1-score is useful metric for performance comparison in cases where some method have better precision but lower recall rate than the other method. In this scenario, precision and recall rates independently are unable to provide true comparison. Therefore, F1-score can reliably be used in such cases for performance comparison. F1-score is computed as:

$$F1 = \frac{P * R}{P + R} \tag{13}$$

3.4. Performance Evaluation

In this experiment, we computed the precision, recall, accuracy, error rate, and F-1 score against each shot category of 10 different sports videos (soccer and cricket). The results obtained for each class of shot are presented in Table 1. The proposed method achieves an average precision of 94.8%, recall of 96.24 %, F1-score of 96.49%, accuracy of 94.07% and error rate of 5.93% on these videos. These results indicate the effectiveness of our proposed AlexNet CNN framework for shot classification of sports videos.

Table 1. Performance evaluation for the proposed system.

Videos	Accuracy (%)	Error Rate (%)	Precision (%)	Recall (%)	F1-Score (%)
Video 1	93.99	6.01	94.5	95.46	96.39
Video 2	94.67	5.33	93.6	97.73	94.38
Video 3	92.39	7.61	96.7	96.7	98.7
Video 4	95.67	4.33	92.34	94.32	95.67
Video 5	96.78	3.22	94.35	97.8	96.78
Video 6	90.56	9.44	95.9	96.54	97.98
Video 7	94.23	5.77	95.46	95.46	96.97
Video 8	95.49	4.51	96.4	96.4	96.48
Video 9	95.64	4.36	94.32	97.56	95.34
Video 10	91.28	8.72	94.5	94.5	96.21
Average	94.07	5.93	94.807	96.24	96.49

3.5. Performance Evaluation Using Different Classifiers

In this experiment, we compared the performance of the proposed method against different classifiers. For this purpose, we designed different experiments to test the accuracy of shot classification on standard convolutional neural networks (CNNs), SVM, KNN, Centroid displacement-based K-Nearest Neighbors (CDNN) and ELM classifiers. We also employed different feature descriptors and classifiers for shot classification and compared the classification accuracy with our framework. More specifically, we used Local Binary Patterns (LBPs) and Local Tetra Patterns (LTRPs) descriptors for feature extraction and trained them on SVM and Extreme Learning Machine (ELM) classifiers. For SVM classifier we employed different kernel functions like linear, quadratic, multi-layer perception (MLP), and radial basis functions (RBF) to analyze the classification performance. The analysis and evaluation of these experiments are presented in detail in this section.

3.5.1. Classification Using Convolutional Neural Network

In our experiments, we used the CNN architecture having three convolutional layers followed by a batch normalization layer, a RELU layer and one fully connected layer (FC). The input video frames are transformed into grayscale and down-sampled from 640 × 480 to 64 × 64 resolution to reduce the computational cost. Each convolutional layer has a filter size of 3 × 3, and max-pooling was performed on every 2 × 2-pixel block. The output is fed to Soft-max layer for classification that helps to determine the classification probabilities used by the final classification layer. We used the learning rate of 0.01 in our experiments as we received the best results on this parameter setting. The details of each layer with the feature maps are as follows:

- 1st Convolutional layer (Conv1) with a kernel size of 3 × 3 and 64 feature maps.
- 2nd Convolutional layer (Conv2), kernel size of 3 × 3 and 96 feature maps.
- 3rd Convolutional layer (Conv3), kernel size of 3 × 3 and 128 feature maps.
- Fully connected layer, hidden units (4000).
- FC hidden units equal to several classes i.e., 4.
- Soft-max layer provides the classification probabilities.
- Final classification layer.

The results achieved by the standard CNN architecture for shot classification are presented in Table 2 that shows the classification accuracy, error rate, precision, recall, and F1-score.

Table 2. Performance evaluation using CNN.

Videos	Accuracy	Error Rate	Precision	Recall	F1-Score
Video 1	91.89	8.11	92.79	97.45	92.97
Video 2	91.27	8.73	91.78	95.67	93.34
Video 3	91.67	8.33	90.23	96.89	91.78
Video 4	92.01	7.99	92.01	98.42	91.37
Video 5	91.02	8.98	91.02	97.1	91.26
Video 6	92.35	7.65	92.31	94.32	90.65
Video 7	89.21	10.79	89.21	93.29	92.87
Video 8	90	10	90	95.79	92.8
Video 9	89	11	89	92.36	92.35
Video 10	90.13	9.87	90.13	97.89	93.45
Average	90.855	9.145	90.848	95.918	92.284

3.5.2. Classification Using Support Vector Machine (SVM)

For classification using SVM, we first extract features from the dataset and then performed training on the extracted features. We used Local Binary Patterns (LBPs) and Local Tetra patterns (LTrPs) for feature extraction [28], which is discussed in detail in this section.

We computed the LBPs by comparing the grayscale values of the center pixel of the given image with its neighbor as follows:

$$LBP_{Q,R} = \sum_{i=1}^{Q}\left(2^{(i-1)} \times \mathfrak{f}_1(S_i - S_c)\right) \tag{14}$$

$$\mathfrak{f}_1(x) = \begin{cases} 1, & x \geq 0 \\ 0, & else \end{cases} \tag{15}$$

where $LBP_{Q,R}$ represents the LBP value at the center pixel S_c. S_c and S_i represents the grayscale value of the center pixel and the neighboring pixels, respectively. Q is the number of neighbors and R is the radius of the neighborhood.

For LTrPs computation, we calculated the first order derivative in the vertical and horizontal directions and encoded the relationships between the referenced pixel and its neighbors. For image K, the first order derivative along zero and 90 degrees are calculated as:

$$K_0^1\ (S_c) = \ K(S_g) - K(S_C) \tag{16}$$

$$K_{90}^1\ (S_c) = \ K(S_h) - K(S_C) \tag{17}$$

where S_c denotes the center pixel in K, S_g and S_h represents the horizontal and vertical neighbors of S. The direction of center pixel S_c is calculated as::

$$K_{DIR}^1(S_c) = \begin{cases} 1, & K_0^1(S_c) \geq 0\ and\ K_{90}^1(S_c) \geq 0 \\ 2, & K_0^1(S_c) < 0\ and\ K_{90}^1(S_c) \geq 0 \\ 3, & K_0^1(S_c) < 0\ and\ K_{90}^1(S_c) < 0 \\ 4, & K_0^1(S_c) \geq 0\ and\ K_{90}^1(S_c) < 0 \end{cases} \tag{18}$$

From Equation (18), four different values, i.e., 1, 2, 3, and 4 is calculated and these values are named as the direction values. The second order derivative is calculated which converts the values into three binary patterns called local ternary patterns and direction is defined using the Equation (18). Local tetra patterns are generated by calculating Euclidean distance with respect to reference direction pixels. After creating the local patterns (LBPs, LTrPs), we represented each image through the histogram as:

$$H = \frac{1}{M \times N} \sum_{k=1}^{M} \sum_{l=1}^{N} f(LP(k,l), w) \tag{19}$$

$$f(x,y) = \left\{ \begin{array}{ll} 1, & if\ x = y \\ 0, & Otherwise \end{array} \right\} \tag{20}$$

where *LP* represents the local patterns (LBPs, LTrPs) and $M \times N$ is the size of the image.

We applied a multi-class SVM classifier using different kernels for performance evaluation. We computed the SVM classifier through minimizing the following expression:

$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \varepsilon_i \tag{21}$$

Subject to the constraints:

$$y_i \left(w^T \varnothing(x_i) + b \right) \geq 1 - \varepsilon_i\ and\ \varepsilon_i \geq 0,\ i = 1,\ \ldots,\ N$$

where C is the capacity constant, w is the vector of coefficients, b is a constant, and ε_i depicts parameters for handling non-separable data. i is the index that labels the N training cases. x_i depicts the independent variable. $\varnothing$ is the kernel that is used to transform data from input to the feature space.

The hyper-parameter for SVM classifier is the margin (C = 1) between two different classes. The largest the margin, the better will be the classifier results. Margin is the maximal width of the hyper-plane that has no interior data points. It has been observed that the larger the C, the more the error is penalized. For SVM classification, we obtained the average accuracy of 73.05% using LBP descriptor. The accuracy against each shot category is provided in Table 3. Similarly, we obtained an average accuracy of 74.46% for LTrP descriptor with SVM classifier. The accuracy for each shot category is presented in Table 4.

Table 3. Classification accuracy using SVM (linear kernel) with LBP features.

Videos	Crowd Shot	Long Shots	Medium Shots	Close-Up Shots
Video 1	77.89	72.12	70.33	72.34
Video 2	76.51	72.56	70.23	73.45
Video 3	78.34	73.57	70.75	73.41
Video 4	76.88	72.53	70.65	72.35
Video 5	72.34	71.91	70.55	72.85
Video 6	77.82	72.85	70.5	71.23
Video 7	76.43	73.44	70.46	72.46
Video 8	76.82	72.56	70.23	73.83
Video 9	76.23	71.23	70.43	72.76
Video 10	77.21	70.34	70.34	73.62
Average	76.64%	72.31%	70.45%	72.8%

Table 4. Classification accuracy using SVM (linear kernel) with LTrP features.

Videos	Crowd Shot	Long Shots	Medium Shots	Close-Up Shots
Video 1	77.49	74.32	72.32	73.21
Video 2	78.91	73.45	73.45	74.67
Video 3	77.65	75.62	72.78	73.91
Video 4	77.89	73.32	73.32	73.32
Video 5	76.57	75.4	72.11	72.11
Video 6	76.66	74.31	71.25	73.45
Video 7	78.64	74.81	70.19	74.56
Video 8	79.49	75.37	72.35	73.44
Video 9	76.34	73.67	71.23	72.16
Video 10	76.34	76.34	72.31	73.71
Average	77.59%	74.67%	72.12%	73.45%

The experiments reveal that the combination of LTrP with SVM provides better accuracy as compared to LBP with the SVM. Experiments signify that crowd shots were categorized effectively in the remaining shots, which is attributed to the fact that the crowd shots contain less dominant grass field color as compared to other categories.

We also used different kernels of SVM like quadratic, Multi-layer Perception (MLP) and radial basis function (RBF) during experimentation. MLP is the most popular kernel of SVM, it is the class of feed forward neural networks and is used when response variable is categorical. RBF kernel is used when there is no prior knowledge about the data as it induces Gaussian distributions. Each point in the RBF kernel becomes a probability density function of normal distribution. It is a sort of rough distribution of data. Whereas quadratic kernel is used to induce a polynomial combinations of the features, working with bended decision boundaries. Quadratic and RBF kernel are expressed in Equations (23) and (24), respectively.

$$K(p,q) = \left(p^t q + e\right)^d \tag{22}$$

where p and q represent input space vectors that are generated from training or validation sets. Note that e $\geq$ 0 is a free parameter that influences higher order versus lower order terms in the polynomial.

$$r(s) = \sum_{i=1}^{N} v_i \varnothing(||s - s_i||) \tag{23}$$

where r(s) is the approximating function, which is expressed as a sum of N radial basis functions, s_j is the center value weighted by $v_{j.}$ v_j is the estimated weight. The results obtained on LBP features with SVM using different kernel functions (quadratic, radial basis function (RBF) and MLP kernel) for shot classification are presented in Table 5.

Table 5. Classification accuracy using SVM (Quadratic, MLP, and RBF kernels) with LBP.

SVM Kernels	Crowd Shots	Close-Up Shots	Long Shots	Medium Shots
RBF	72.13%	68.73%	65.34%	67.65%
Quadratic	77.56%	76.24%	73.21%	72.34%
MLP	58.67%	60.98%	59.53%	61.23%

Similarly, the results of LTrP features with SVM using different kernel functions for shot classification are presented in Table 6.

Table 6. Classification accuracy using SVM (Quadratic, MLP and RBF kernels) with LTrP.

SVM Kernels	Crowd Shots	Close-Up Shots	Long Shots	Medium Shots
RBF	72.27%	71.97%	70.27%	73.37%
Quadratic	72.34%	78.34%	73.67%	73.45%
MLP	59.67%	59.23%	60.78%	62.23%

It has been observed from Tables 3–6 that linear SVM provides better performance as compared to quadratic, MLP, and RBF kernels. In addition, there is a slight difference in the performance accuracy between RBF and quadratic kernels, however, we received a very low accuracy rate for MLP kernel.

3.5.3. Classification Using ELM

We also designed an experiment to measure shot classification performance using the ELM classifier. For this purpose, we extracted the LBP and LTrP features in the similar manner as discussed in Section 3.1. For ELM classification, we obtained an average accuracy of 73.45%, precision of 72.67%, recall of 76.39%, and error rate of 26.55% using LBP descriptor. Similarly, an average accuracy of

75.56%, precision of 76.43%, recall of 77.89%, and error rate of 24.44% and 75.50% using LTrP descriptor was achieved. The results of LBP and LTrP descriptors with the ELM classifier are provided in Tables 7 and 8, respectively.

Table 7. Performance evaluation using extreme learning machine classifier with local binary pattern features.

Videos	Accuracy	Precision	Recall	Error
Video 1	73.21	73.21	76.32	26.79
Video 2	74.67	72.13	76.34	25.33
Video 3	73.91	73.91	76.58	26.09
Video 4	73.32	73.86	75.49	26.68
Video 5	72.11	72.69	76.45	27.89
Video 6	73.45	73.45	76.77	26.55
Video 7	74.56	71.73	77.49	25.44
Video 8	73.44	73.44	76.45	26.56
Video 9	72.16	72.16	75.86	27.84
Video 10	73.71	70.12	76.21	26.29
Average	**73.45%**	**72.67%**	**76.39%**	**26.55%**

Table 8. Performance evaluation using extreme learning machine classifier with local tetra pattern features.

Videos	Accuracy	Precision	Recall	Error
Video 1	76.54	77.43	79.45	23.46
Video 2	77.23	77.23	77.49	22.77
Video 3	75.21	75.21	77.48	24.79
Video 4	75.95	75.95	77.48	24.05
Video 5	74.28	77.84	78.81	25.72
Video 6	75.11	75.17	77.34	24.89
Video 7	74.56	76.39	78.39	25.44
Video 8	76.57	76.57	78.91	23.43
Video 9	75.66	77.65	77.65	24.34
Video 10	74.52	74.85	75.97	25.48
Average	75.56%	76.43%	77.89%	24.44%

From the results presented in Tables 7 and 8, it can be clearly observed that the combination of LTrP with ELM provides better performance as compared to LBP with the ELM. This illustrates that LTrP descriptors are more effective in comparison of LBP for shot classification because it includes magnitude and direction of the neighboring pixels, whereas in case of LBP only magnitude of the vertical and horizontal neighbor is concerned. It is to be noted that crowd shots classification results are far better than the remaining types of shot classification. One significant reason is the absence of playfield in crowd shots, whereas playfield exists for all the in-field shots.

3.5.4. Classification Using KNN

We also implemented and tested the shot classification performance on our cricket and soccer video dataset using K-Nearest Neighbor (K-NN) classifier. In KNN classification, an object is classified according to majority vote of its neighbors, with the object assigned to the most common class among its k nearest neighbors. We performed this experiment on different values of k and obtained the best results with k = 5, therefore, the value for K in KNN is set to five in our experiments. Nearest neighbors are computed by calculating the Euclidean distance formula. The results obtained using the KNN are provided in Table 9.

Table 9. Performance evaluation using k-nearest neighbor.

Approach	Accuracy	Precision	Recall	Error Rate
Video 1	90.45	90	92.31	9.55
Video 2	91.27	90.27	91.54	8.73
Video 3	91.67	90.12	90.12	8.33
Video 4	92.01	91.67	91.67	7.99
Video 5	91.02	91.02	91.02	8.98
Video 6	92.35	91.48	91.48	7.65
Video 7	89	89	90.89	11
Video 8	90.32	90	92.34	9.68
Video 9	89.24	89.24	90	10.76
Video 10	90.13	91	91.12	9.87
Average	90.75%	90.37%	91.25%	9.25%

3.5.5. Classification Using Centroid Displacement-Based K-Nearest Neighbors (CDNN)

We also implemented and tested the shot classification performance on our cricket and soccer video dataset using centroid displacement-based K-Nearest Neighbors (CDNN) [29] classifier. In CDNN classification, along with the distance parameter, an integral learning component that learns the weight of the view is added which helps in classifying new shots in the test dataset. The value of k for CDNN is set to five for our experiments as we obtained the optimal results on this value after checking the classifier performance on different values of k. The results obtained using the CDNN are far better than SVM and ELM classifiers, but lesser than our proposed method. The results are presented in Table 10.

Table 10. Performance evaluation using CDNN.

Videos	Accuracy	Precision	Recall	Error Rate
Video 1	93.45	93.23	94.67	6.55
Video 2	91.54	92.12	93.32	8.46
Video 3	93.67	93.45	93.78	6.33
Video 4	92.31	91.29	94.56	7.69
Video 5	91.02	92.49	92.36	8.98
Video 6	92.34	93.56	94.67	7.66
Video 7	93.47	91.23	94.78	6.53
Video 8	92.89	92.78	93.19	7.11
Video 9	92.91	92.89	93.67	7.09
Video 10	92.13	93.14	93.81	7.87
Average	92.5%	92.62%	93.9%	7.5%

Performance comparison of the proposed method with SVM, KNN, CDNN, ELM, and standard CNN classifiers are shown in Figure 8. The proposed method achieves an average accuracy of 94.07% in comparison with CDNN, CNN, KNN, ELM and SVM that provides 92.5%, 91.67%, 91.75%, 74.50% and 69.45%, respectively. From the results in Figure 8, it can be clearly observed that the proposed method performs far superior, as compared to SVM, ELM, and marginally better than KNN, CDNN, and standard CNN for shot classification. Therefore, we can argue that the proposed method is very effective in terms of shot classification of sports videos.

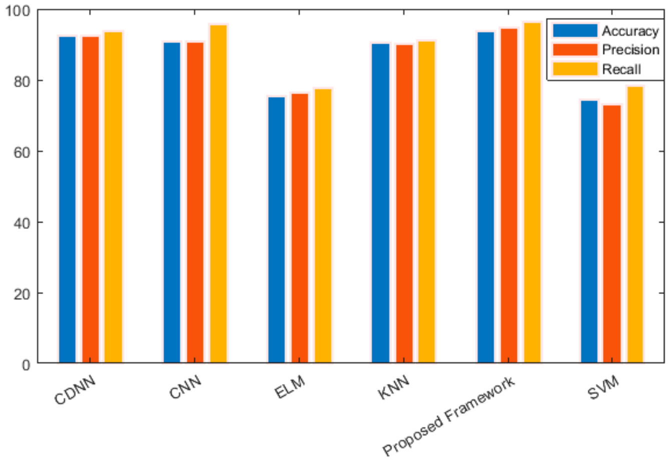

Figure 8. Performance comparison with different classifiers.

3.6. Performance Comparison with Existing Methods

In the last experiment, we compared the performance of our method against the existing shot classification methods [24–26] and [30–33] for sports videos.

Tavassolipour et al. [24] proposed a Bayesian network-based model for event detection and video summarization in soccer videos. Markov model and hue histogram differences were used to detect the shot boundaries. These shots were classified as long view, medium view, and close-up view depending upon the size of the players and dominant color. Bayesian networks were applied to classify the events, i.e., goal, foul, etc. Khaligh et al. [25] proposed a shot classification method for soccer videos. First, the in-field video frames were separated from the out-field frames. In the next stage, three features (i.e., number of connected components, shirt color in vertical and horizontal strips) were extracted from the in-field frames and fed to SVM to classify the long, medium, and close-up shots. Kapela et al. [26] used radial basis decomposition function (RBF) and Gabor wavelets to propose a method for scene classification of sports videos. The input image was transformed into HSV color space followed by applying the Fourier transform on the image. Gabor filters were applied and trained the SVM to classify the scene into different shots. Fani et al. [30] proposed a deep learning fused features-based framework to classify the shot types using the camera zoom and out-field information. The soft-max and fussing Bayesian layers were used to classify the shots into long, medium, close-up, and out-field shots. Chun et al. [31] proposed a system for automatic segmentation of basketball videos based on GOP (group of pictures). Long view, medium view and full court view were classified using the dominant color feature and length of the video clips. Kolekar et al. [32] proposed a system that generated highlights from the soccer videos. Bayesian network was employed to classify the video into replay, player, referee, spectator and playing gathering shots based on the audio features. Exciting segments from the soccer videos were detected that are assigned semantic concept labels like goals, save, yellow-cards, and kicks in sequence. Classification accuracy for the exciting segments was observed to be 86%. Raventos et al. [33] proposed a video summarization method for soccer based on audio-visual features. Shot segmentation was performed initially to select the frames for video summarization based on the relevance. Rule-based thresholding was applied on the grass field color pixels to detect the long shots in soccer videos. The average accuracy of the proposed and existing shot classification approaches is provided in Table 11.

From the results shown in Table 11, we can clearly observe that the proposed method was evaluated over a diverse dataset of sports videos; and achieved the highest precision and recall values, as compared to the existing state-of-the-art shot classification methods. Although the videos used in our method and comparative methods are different but the experimental setup of our method and comparative methods is similar in terms of video source and content selection. We selected the videos from YouTube as done by the comparative methods and the selected videos represent different broadcasters, different genres, and different tournaments. The videos also represent different illumination conditions (i.e., all-day videos, day, and night videos, and night videos recorded in electric lights), and various camera shot types (i.e., long, medium, close-up and crowd/out-field shots) as selected by the comparative approaches. By testing our method on a diverse set of sports videos captured under the challenging conditions for shot-classification we ensured the fair comparison against the state-of-the-art methods. Hence, based on our results, we can say that the proposed method is a reliable approach for the shot classification of the sports videos.

Table 11. Performance comparison with existing shot classification methods.

Shot Classification Methods	Technique	Dataset	Shots Type	Recall	Precision
Tavassolipour et al. [24]	SVM, KNN Classifiers	Soccer Videos	Long, medium, close, out-field	89.93	90.3
Khaligh et al. [25]	SVM	Soccer	Long, medium, close, out-field	93.85	91.07
Kapela et al. [26]	SVM Classifier	Field Sports	Close-up, Long	84.2	82.5
Fani et al. [30]	PFF-Net	Soccer Videos	Close, medium, long, out-of-the-field	91.3	90.6
Chun et al. [31]	Deep net (CC-Net)	Basketball videos	Close, medium close, medium, medium long, long, extreme long	NA	90
Kolekar et al. [32]	Bayesian Based Network (BB-Net)	Soccer videos	Long, Close-up	84	86
Raventos et al. [33]	Rule-based Thresholding	Soccer Videos	Long	80	86
Proposed System	AlexNet CNN	Cricket & Soccer Videos	Close, medium, long, crowd/out-field	96.24	94.07

3.7. Discussion

Different classification frameworks were presented using supervised and un-supervised learning-based approaches in the past. Experiments prove that convolution neural networks are effective for shot classification. We evaluated the performance of different convolution networks, and it has been observed that the proposed AlexNet convolution network performed better in classifying different shots of the sports videos. The use of response normalization rectified linear unit layer and the drop out layer on the training data makes the training much faster. In fact, once validation loss is observed to be zero, the network stops training and is ready for classification. In comparison with different classifiers like KNN++, KNN, SVM, ELM, and standard CNN, we found that the proposed system can train and validate the data by itself. It has also been brought into consideration that enhanced KNN and KNN classifiers perform significantly better than SVM and ELM classifier. The major reason is due to the integral weight factor and the distance parameters of these classifiers.

Moreover, we also observed during experimentation of the proposed method that the computers embedded with high performance Graphics Processing Unit (GPU) can further increase the speed and accuracy of the proposed framework. For fast training of dataset, AlexNet uses a Graphical Processing Unit (GPU) if its integrated on a system. It requires a parallel computing toolbox with CUDA enabled GPU, otherwise it uses Central Processing Unit (CPU) of a system. Our system is not integrated with GPU, therefore the proposed framework used CPU for training of sports videos.

It has also been observed that the proposed network stops training once it is confirmed that no validation loss is taking place, i.e., the dataset has been trained to a maximum limit. This is the advantage of our proposed network over standard CNN. Moreover, if Weight Learn Rate Factor (WLRF) and Bias Learn Rate Factor (BLRF) values of fully connected layers are increased, the leaning rate of training rises significantly. In addition, we observed during the experimentation that decreases the size of dataset to 25% increases the learning rate of the data.

4. Conclusions and Future Work

We proposed an AlexNet convolutional neural network-based model for shot classification of field sports videos. Our framework is robust to camera variations, scene change, action speeds, and illumination conditions (i.e., daylight, artificial light, shadow). The proposed framework successfully classifies the input sports video into long, medium, close-up, and crowd/out-field shots in the presence of these limitations. Experimental results signify the effectiveness of our framework in terms of shot classification of field sports videos. We compared the performance of the proposed method with existing state-of-the-art methods. In addition, we specifically designed an experiment to extract LBP and LTrP features and trained them on SVM and ELM separately for shot classification. We also evaluated the performance of shot classification using the KNN and standard CNN. Afterwards, we compared the results obtained on SVM, K-NNELM and standard CNN with the proposed method. The comparison clearly shows that the proposed framework provides better classification performance, as compared to SVM ELM, K-NN, and standard CNN classifiers. It is to be noted that the use of CUDA Graphics Processing Unit namely GTX 580 can further increase the processing speed of the proposed method. Optimization level is achieved by just changing CUDA 480 to CUDA 580 GB graphics card.

Currently we are investigating the performance of the proposed method on a more diverse and larger dataset. We would preferably be working on small data that is able to define some training and validation percentage of data required to make a trade-off between efficient and effective classification phenomena. Moreover, it would be interesting to investigate the performance of various classifiers for the proposed features and the undertaken problem. Particularly, in future work, the performance of the resultant classifiers from the combination of weak classifiers such as random forest will be analyzed. In addition, these excellent results of shot classification can be further used to increase the accuracy of video summarization systems.

Author Contributions: Data curation, J.B.J.; Funding acquisition, M.T.M.; Investigation, A.I.; Methodology, A.I.; Project administration, J.B.J.; Resources, M.T.M.; Software, R.A.M. and A.J.; Validation, A.J.; Writing—original draft, R.A.M.; Writing-review & editing, A.J., A.I., M.T.M. and J.B.J.

Funding: This research work was supported by Korea University of Technology and Education (KOREATECH), Education and Research Promotion Program (2017).

Conflicts of Interest: The authors have no conflict of interest.

References

1. Choroś, K. Automatic playing field detection and dominant color extraction in sports video shots of different view types. In *Multimedia and Network Information Systems*; Springer: Cham, Switzerland, 2017; pp. 39–48.
2. Petscharnig, S.; Schöffmann, K. Learning laparoscopic video shot classification for gynecological surgery. In *Multimedia Tools and Applications*; Springer: New York, NY, USA, 2018; Volume 77, pp. 8061–8079.
3. Choroś, K. Application of the temporal aggregation and pre-categorization of news video shots to reduce the time of content analysis. *J. Intell. Fuzzy Syst.* **2017**, *32*, 1615–1626. [CrossRef]
4. Wei, W.-L.; Lin, J.-C.; Liu, T.-L.; Yang, Y.-H.; Wang, H.-M.; Tyan, H.-R.; Liao, H.-Y.M. Deep-net fusion to classify shots in concert videos. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 1383–1387.
5. Hmayda, M.; Ejbali, R.; Zaied, M. Automatic topics segmentation for TV news video. In Proceedings of the Ninth International Conference on Machine Vision (ICMV 2016), Nice, France, 18–20 November 2016; International Society for Optics and Photonics: Bellingham, DC, USA, 2017; Volume 10341, p. 1034114.
6. Chauhan, D.; Patel, N.M.; Joshi, M. Automatic summarization of basketball sport video. In Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 14–16 October 2016; pp. 670–673.
7. Sharma, R.A.; Gandhi, V.; Chari, V.; Jawahar, C.V. Automatic analysis of broadcast football videos using contextual priors. *Signal Image Video Process.* **2017**, *11*, 171–178. [CrossRef]
8. Chacon-Quesada, R.; Siles-Canales, F. Evaluation of Different Histogram Distances for Temporal Segmentation in Digital Videos of Football Matches from TV Broadcast. In Proceedings of the 2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI), Funchal, Portugal, 10–12 July 2017; pp. 1–7.
9. Chattopadhyay, A.; Chattopadhyay, A.K.; B-Rao, C. Bhattacharyya's distance measure as a precursor of genetic distance measures. *J. Biosci.* **2004**, *29*, 135. [CrossRef] [PubMed]

10. Ekin, A.; Tekalp, A.M.; Mehrotra, R. Automatic soccer video analysis and summarization. *IEEE Trans. Image Process.* **2003**, *12*, 796–807. [CrossRef] [PubMed]
11. Sharma, R.A.; Sankar, K.P.; Jawahar, C.V. Fine-grain annotation of cricket videos. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 421–425.
12. Sigari, M.-H.; Soltanian-Zadeh, H.; Kiani, V.; Pourreza, A.-R. Counterattack detection in broadcast soccer videos using camera motion estimation. In Proceedings of the 2015 International Symposium on Artificial Intelligence and Signal Processing (AISP), Mashhad, Iran, 3–5 March 2015; pp. 101–106.
13. Duan, L.-Y.; Xu, M.; Tian, Q.; Xu, C.-S.; Jin, J.S. A unified framework for semantic shot classification in sports video. *IEEE Trans. Multimed.* **2005**, *7*, 1066–1083. [CrossRef]
14. Joe, Y.-H.N.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; Toderici, G. Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4694–4702.
15. Karmaker, D.; Chowdhury, A.Z.M.E.; Miah, M.S.U.; Imran, M.A.; Rahman, M.H. Cricket shot classification using motion vector. In Proceedings of the 2015 Second International Conference on Computing Technology and Information Management (ICCTIM), Johor, Malaysia, 21–23 April 2015; pp. 125–129.
16. Kapela, R.; Świetlicka, A.; Rybarczyk, A.; Kolanowski, K. Real-time event classification in field sport videos. *Signal Process. Image Commun.* **2015**, *35*, 35–45. [CrossRef]
17. Papachristou, K.; Tefas, A.; Nikolaidis, N.; Pitas, I. Stereoscopic video shot classification based on Weighted Linear Discriminant Analysis. In Proceedings of the 2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Reims, France, 21–24 September 2014; pp. 1–6.
18. Burney, A.; Syed, T.Q. Crowd video classification using convolutional neural networks. In Proceedings of the 2016 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 19–21 December 2016; pp. 247–251.
19. Abu-El-Haija, S.; Kothari, N.; Lee, J.; Natsev, P.; Toderici, G.; Varadarajan, B.; Vijayanarasimhan, S. Youtube-8m: A large-scale video classification benchmark. *arXiv* **2016**, arXiv:1609.08675.
20. Lee, J.; Koh, Y.; Yang, J. A deep learning-based video classification system using multimodality correlation approach. In Proceedings of the 2017 17th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, 18–21 October 2017; pp. 2021–2025.
21. Wang, H.L.; Cheong, L.-F. Taxonomy of directing semantics for film shot classification. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 1529–1542. [CrossRef]
22. Kumar, A.; Garg, J.; Mukerjee, A. Cricket activity detection. In Proceedings of the 2014 First International Image Processing, Applications and Systems Conference (IPAS), Sfax, Tunisia, 5–7 November 2014; pp. 1–6.
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Neural Information Processing System Foundations Inc.: San Diego, CA, USA, 2012; pp. 1097–1105.
24. Tavassolipour, M.; Karimian, M.; Kasaei, S. Event detection and summarization in soccer videos using Bayesian network and copula. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 291–304. [CrossRef]
25. Bagheri-Khaligh, A.; Raziperchikolaei, R.; Moghaddam, M.E. A new method for shot classification in soccer sports video based on SVM classifier. In Proceedings of the 2012 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), Santa Fe, NM, USA, 22–24 April 2012; pp. 109–112.
26. Kapela, R.; McGuinness, K.; O'Connor, N.E. Real-time field sports scene classification using colour and frequency space decompositions. *J. Real-Time Image Process.* **2017**, *13*, 725–737. [CrossRef] [PubMed]
27. Rabia and Ali Javed. Available online: https://datadryad.org/handle/10255/2/submit/58131f06892432862112314b2c7134460f387284.continue?processonly=true?processonly=true (accessed on July 2018).
28. Murala, S.; Maheshwari, R.P.; Balasubramanian, R. Local tetra patterns: A new feature descriptor for content-based image retrieval. *IEEE Trans. Image Process.* **2012**, *21*, 2874–2886. [CrossRef] [PubMed]
29. Nguyen, B.P.; Tay, W.L.; Chui, C.K. Robust Biometric Recognition from Palm Depth Images for Gloved Hands. *IEEE Trans. Human Mach. Syst.* **2015**, *45*, 799–804. [CrossRef]
30. Fani, M.; Yazdi, M.; Clausi, D.A.; Wong, A. Soccer Video Structure Analysis by Parallel Feature Fusion Network and Hidden-to-Observable Transferring Markov Model. *IEEE Access* **2017**, *5*, 27322–27336. [CrossRef]

31. Tien, M.-C.; Chen, H.-T.; Chen, Y.-W.; Hsiao, M.-H.; Lee, S.-Y. Shot classification of basketball videos and its application in shooting position extraction. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007), Honolulu, HI, USA, 15–20 April 2007; Volume 1, pp. 1085–1088.
32. Kolekar, M.H.; Sengupta, S. Bayesian network-based customized highlight generation for broadcast soccer videos. *IEEE Trans. Broadcast.* **2015**, *61*, 195–209. [CrossRef]
33. Raventos, A.; Quijada, R.; Torres, L.; Tarrés, F. Automatic summarization of soccer highlights using audio-visual descriptors. *SpringerPlus* **2015**, *4*, 301. [CrossRef] [PubMed]