Available online at www.sciencedirect.com

ScienceDirect

ICT Express 8 (2022) 65–71

www.elsevier.com/locate/icte

Categorization of actions in soccer videos using a combination of transfer learning and Gated Recurrent Unit

Anik Sen[a,b], Kaushik Deb[a,*]

[a] *Department of Computer Science & Engineering, Chittagong University of Engineering & Technology (CUET), Chattogram 4349, Bangladesh*
[b] *Department of Computer Science & Engineering, Premier University, Chattogram 4000, Bangladesh*

Received 27 July 2020; received in revised form 9 March 2021; accepted 10 March 2021
Available online 26 March 2021

Abstract

Extraction of knowledge from soccer videos has enormous applications like context-based advertisement, content-based video retrieval, match summarization, and highlight extraction. Overlapping soccer actions and uncontrolled video capturing conditions make it challenging to detect action accurately. For overcoming these problems, Convolutional Neural Network and Recurrent Neural Network are used jointly to classify different lengths of soccer actions. Initially, transfer learning from pre-trained VGG network extracts characteristic spatial features. Afterwards, Gated Recurrent Unit deals with temporal dependency and solves the vanishing gradient problem. Finally, softmax layer assigns decimal probabilities to each class. Experimental results demystify the significance of the proposed architecture.

Keywords: Convolutional Neural Network (CNN); Gated Recurrent Unit (GRU); RecurrenT Neural Network (RNN); Transfer learning

1. Introduction

Though abounding research has been conducted with the goal of video classification, this field still poses multifold novel challenges. In recent years, sports-based work is becoming popular among researchers. Of all types of sports, soccer is one of the most popular and viewed across the globe. However, in our knowledge, there are only a few works dedicated to soccer, which can have a wide range of applications in the field of computer vision. For example, our work can be combined with some Natural Language Processing (NLP) models to generate Artificial Intelligence (AI) based automatic, unbiased, and blind commentary system. Moreover, to eliminate the manual, tedious task like automatic soccer highlight generation and automated match summarizing, it is required to detect the vital soccer actions and categorize them accordingly.

In the upcoming sections, we expound on video classification by taking soccer-based sports actions. Moreover, according to [1], it can be seen that more than 84% of soccer fans watch past world cups or other soccer games on YouTube. Due to the overlapping scenes between different soccer actions, classifying the proper actions is a challenging task. Currently, there exist only a few works dedicated to soccer. The dataset is not yet to that scale among the current works, and many of the vital actions are ignored. However, automatically categorizing various soccer actions has a high commercial impact on the broadcasters. The above facts and the tremendous achievement of deep learning algorithms motivated us for this work.

To outline the central handouts of our work, we have considered 10 soccer actions corner, foul, free-kick, goal-kick, long-pass, penalty, short-pass, shot-on-goal, substitution, and throw-in and developed a new dataset termed as SoccerAct10. To incorporate both the spatial-features and temporal-features, we seize the benefit of CNN and RNN. In the ongoing trend, researchers have intensive use of deep neural networks as it pushes the boundaries of many hand-engineered processes.

Researches related to soccer can be further categorized. Video retrieval from soccer to generate the highlights is performed in [2]. Moving object detection is accomplished by motion intensity. For audio information extraction, short time energy and speech components are used. In [3], a framework for summarizing soccer videos is introduced which allows real-time event detection by cinematic features.

[*] Corresponding author.
E-mail address: debkaushik99@cuet.ac.bd (K. Deb).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.03.004

Event detection from tiny set of soccer videos is proposed in [4]. Mean-gradient features was compounded to shot boundary detection. Furthermore, low and mid level feature descriptors were fed into the MKL based SVM classifier. However, a study in [5] exemplifies that the softmax classifier can achieve better performance than the SVM classifier.

In [6], 4 different actions of soccer, particularly shot-on-goal, placed-kick, throw-in and goal-kick, are classified. 3-fold cross validation was performed with a tiny dataset containing only 20 videos as training and 5 as testing. The proposed SVM string kernel based model attains a mean accuracy of 73%. In [7], a method for detecting goals in soccer video is proposed which also combines audio clues. However, this work paves the way for more improvement. In [8], 3 group actions during soccer, namely left side attacking (LA), left side defending (LD), and stalemate (ST) are classified. This method commits comparatively low accuracy of 83.87% for the LD and ST. In [9], 4 soccer actions: goal-kick, placed-kick, shot-on-goal, and throw-in are classified. Using Bag of Words (BoW), dominant motion, and Long Short-Term Memory (LSTM), 92% accuracy is achieved. However, in a study comparing BoW and CNN performance, it is shown that CNN based method attains higher accuracy than BoW [10].

The contributions of this work are depicted below:

(i) Developed a custom dataset having a variable length of videos
(ii) Proposed an algorithm for sampling a fixed number of frames from a variable length of videos
(iii) Implemented deep learning algorithms, namely CNN, GRU to overcome the limitations of the conventional feature extraction approach. Moreover, to overcome the limitation of a popular linear classifier SVM, we utilized softmax classifier
(iv) Rigorous experiments are conducted to find the best architecture, and eventually designed a hybrid CNN–GRU architecture which can categorize 10 distinct soccer actions considering diverse capturing conditions (e.g., abrupt change of camera angle while taking a corner) and overlapping sub-patterns (e.g., players moving toward goal post can be similar for both free-kick and corner)

2. Proposed soccer actions categorization (SAC) architecture

Sequence data are generated from variable length of videos. Then, sequence data are fed into CNN (VGG), and extracted features are passed into GRU. The output from GRU is passed into softmax, and the class level is predicted. The steps to develop the architectures for SAC are depicted in Fig. 1.

2.1. Network architecture

From various lengths of videos, features are extracted using VGG. These extracted features are passed to a fully connected layer with ReLU activation. Then, flatten layer is stacked. Time distributed wrapper encloses the layers mentioned above

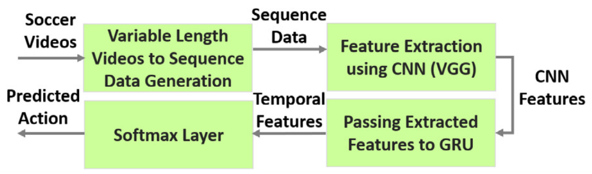

Soccer Videos
Variable Length Videos to Sequence Data Generation
Sequence Data
Feature Extraction using CNN (VGG)
CNN Features
Passing Extracted Features to GRU
Temporal Features
Softmax Layer
Predicted Action

Fig. 1. Workflow of the proposed architecture for SAC.

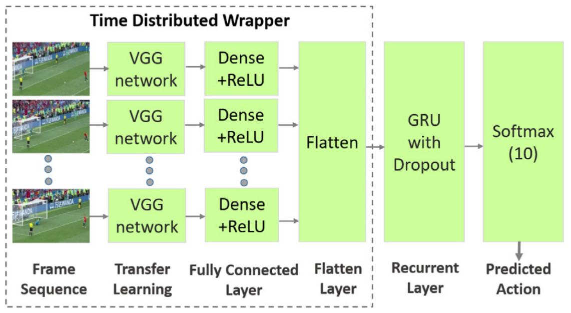


Fig. 2. VGG–GRU network architecture for SAC.

to apply the same layer to each frame. GRU layer is stacked afterwards. Dropout is added with GRU to prevent overfitting. Various configurations of hidden units for Dense and GRU layers are depicted in Table 1. Finally, softmax activation function as given in Eq. (1) is used, which normalizes the probabilities of different classes that sums up to 1.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} for j = 1, \ldots, k \tag{1}$$

Stochastic gradient descent (SGD) optimizer having 0.0001 learning rate and 0.9 momentum is used throughout all the experiments. These hyperparameters are chosen using grid search. ModelCheckpoint and EarlyStopping callbacks are used here. The former saves the best weight, and the latter stops training if there is no improvement in validation loss for 5 consecutive epochs. We used categorical crossentropy loss function as defined in Eq. (2).

$$L(\hat{y}_i, y) = -\sum_{i=1}^{K} y_i \log(\hat{y}_i) \tag{2}$$

Here, K is the total no of classes, y_i is the desired one hot encoded vector, and $\hat{y}_i$ is the model predicted output vector. During backpropagation, the gradient starts to backpropagate through the derivative of the loss function with respect to the output of the softmax layer.

The initial batch size was 5 and varied this value ranging from 1 to 8 while resuming the training. The combined VGG–GRU architecture is highlighted in Fig. 2.

2.2. Variable length videos to sequence generation

Incorporating an uneven number of frames is a challenging task while dealing with videos. The main action can be at any video part; it can be at the starting part, middle part, or end part. Therefore, when considering the frames, it is to be done so that the frames of the whole video come into consideration.

Table 1
Network configuration and accuracy of our proposed networks.

Model	Pre-trained model	Trainable layers (excluding fully-connected)	Hidden units in TimeDistributed Dense layer	Hidden units in GRU layer (dropout)	Total trainable parameters	Training accuracy (%)	F1-Score (%)
Model-1	VGG16	None	256	32 (0.4)	5,28,042	97.11	88.00
Model-2	VGG16	All	2048	84 (0.6)	2,40,45,118	99.97	93.00
Model-3	VGG16	Final 4	512	64 (0.4)	89,28,074	100	92.00
Model-4	VGG16	Final 8	512	64 (0.5)	1,48,27,850	99.98	**94.00**
Model-5	VGG19	None	1024	64 (0.4)	36,84,170	97.20	85.00
Model-6	VGG19	All	512	64 (0.4)	2,18,73,034	99.90	**94.00**
Model-7	VGG19	Final 4	2048	96 (0.5)	1,75,96,138	99.74	92.00
Model-8	VGG19	Final 8	2048	72 (0.5)	2,23,038,58	99.97	93.00
Model-9	Xception	Final 4	512	32 (0.25)	16,492,122	96.67	77.00
Model-10	InceptionV3	Final 9	512	32 (0.25)	8,553,866	98.97	84.00

In our work, we accepted this challenge and proposed an algorithm that generates a fixed sequence length from uneven number of video frames. The steps are outlined in Algorithm 1. The outer loop executes for all the files in a directory. Let us consider the total number of files to be n. Until the loop of line 4, the algorithm will execute the instructions in constant time. The inner loop considers each 2D RGB frame of a video file. Considering the highest number of frames to be m, the proposed algorithm's overall complexity is $\mathcal{O}(mn)$.

Algorithm 1: Variable length videos to sequence generation.

```
Input: Integer $seqlen$, where $seqlen$ < total frames of a video.
Output: List of data and labels
1  $data \leftarrow []$, $labels \leftarrow []$;
2  for each video file do
3      $frame\_count \leftarrow$ total frames in the file; $label \leftarrow$ Get Label of the video file;
       $x \leftarrow round(frame\_count \div seqlen)$; $count \leftarrow 0$, $i \leftarrow 0$, $frames \leftarrow []$;
       /* initialize count and i with 0. For each frame, count will be incremented. If a frame is selected, i will be incremented. */
4      for each frame in a file do
5          $frame \leftarrow$ read $frame$ values;
           /* below condition checks if current frame is to be selected or not. */
6          if $count \bmod x = 0$ and $i < seqlen$ then
7              Append $frame$ to $frames$; $i++$;
8          end
9          $count++$;    // increment for each frame
10     end
11     data.append(frames), labels.append(labels);
12 end
```

2.3. CNN and transfer learning

CNN is a version of multilayer perceptrons that is designed to automate the spatial feature extraction process from image. It has already proven its tremendous success in the field of computer vision. This feature extraction process is performed using some stack of layers, namely Convolution Layer, Pooling Layer, Flatten Layer, and Fully-connected Layer. The convolution operation is performed with some filters, thus producing the convolved feature maps. Pooling reduces the parameters by reducing the spatial size of the image. The initial filters generally expose the low-level features like colors, texture, and lines, while the downstream layers detect more complex patterns, which evidently describe an image. Weights are updated in a similar fashion as multilayer perceptron (MLP).

A deep neural network must have to be trained with a massive amount of data to attain high accuracy. However, for a domain-specific problem, it sometimes becomes challenging to manage this well-annotated vast dataset [11]. Transfer learning alludes to the process of picking up knowledge from one problem and applying to another problem. Recently, a lot of researchers are using transfer learning instead of designing the whole network from scratch. It also provides the offer of adding layers accordingly or re-train all the weights. By layer freezing technique, weight updating of specific layers can be stopped. Pre-trained models, already trained on massive datasets like ImageNet, provide an excellent starting point for another problem where the dataset size is not yet well enough. Several pre-trained CNN models are the most widely used. VGG, ResNet, Inception are worth mentioning. With some initial experiments, it is observed that VGG yields comparatively high accuracy than other pre-trained models, namely Xception and InceptionV3, as shown in Table 1. Thereupon, transfer learning from deep VGG models are used [12] for our specific task of SAC. VGG uses very small (3×3) convolution filters with stride of 1. Max-pooling is performed over a 2×2 pixel window, with stride 2. We used 2 versions of VGG, VGG16 with ≈138M parameters, and VGG19 with ≈143M parameters.

2.4. Gated recurrent unit (GRU)

For video classification, it requires temporal features along with spatial features. For temporal features, various RNN has been proposed. However, due to the gradient's long propagation through all the timesteps, the vanilla RNN suffers a lot from vanishing gradient and exploding gradient problem [13]. Hence, LSTM [14] was proposed later, which proves its effectiveness in maintaining the long temporal dependency. GRU [15] is used in our work because this network is even a simplified version of LSTM, which can achieve similar results as per LSTM, using fewer parameters. Update gate, Reset gate, and Current memory gate are used. Update gate determines the amount of previous knowledge that is to be passed along into the future. Reset gate determines the amount of prior knowledge that is to be forgotten. Using the current memory gate, it calculates the present state value by combining the previous hidden state with the current input. The equations for the update gate, reset gate, current memory state, and final memory are given in Eq. (3), Eq. (4), Eq. (5), and Eq. (6) respectively:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{3}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{4}$$

$$\tilde{h_t} = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b) \tag{5}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{6}$$

Here, W_z, W_r, and W are learnable weight matrices, h_{t-1} is the previous hidden state, x_t is the input vector, σ and tanh are the sigmoid and tanh activation function, $*$ represents the hadamard product, and b_z, b_r, and b are biases.

3. Experimental results and discussions

The experiment was conducted on the configuration of AMD Ryzen 7 2700X Eight-core 3.7 GHz Processor with Ubuntu 18.04 OS, 32 GB RAM, NVIDIA GEFORCE RTX 2060 SUPER of 8 GB GPU Memory. Keras using the TensorFlow backend was used. We extracted the dataset video clips from different events of World Cup, AFC Asian Cup, UEFA, Copa America, English Premier League, Italian Serie A, German Bundesliga, and Spanish La Liga. Our custom dataset, termed SoccerAct10, consists of a total of 3110 training videos, the lowest 250 videos of foul, and the highest 375 videos of goal-kick. Additional 500 videos from different games, apart from the training set, were generated for the test set to remove the biasness. Videos were ranging from a minimum of 1.0 s to a maximum of 7.8 s Augmentation techniques, namely scaling, rotation, and horizontal flip are applied to the training set. The average no. of frames for the whole dataset is 74.95. Meanwhile, the standard deviation is 26. In Figs. 3(a)–3(j) sequential frames of 10 actions are shown.

With 100 samples of each class, we conducted an experiment using the VGG16 network to choose the best sequence length. Furthermore, almost all the sequence lengths from 5 to 15 provide more than 90% validation accuracy. Moreover, for the sequence of length 10, the model works out its best with 96.57% validation accuracy. Hence, the sequence length of 10 is considered for further investigation. 150 × 150 pixels frame size is considered through all the experiments. We conducted various experiments to determine the best model. At first, for each model, all the dense layers of VGG16 and VGG19 are dropped. Then a variety of configurations were tested to determine the best one. Some of the best configurations are shown in Table 1. In *Model-1* and *Model-5*, layer freezing is applied to all the layers of VGG16 and VGG19, respectively. *Model-2* and *Model-6* is derived by making all the VGG16 and VGG19 layers trainable. *Model-3* and *Model-7* keep the final 4 VGG16 and VGG19 trainable. *Model-4* and *Model-8* were executed by freezing all the layers except the final 8 layers of VGG16 and VGG19, respectively. *Model-9* takes output from an intermediate layer (*block4_sepconv2_act*) of pre-trained Xception model and makes final 4 layers trainable. *Model-10* takes output from an intermediate layer (*average_pooling2d_4*) of pre-trained InceptionV3 model and makes final 9 layers trainable.

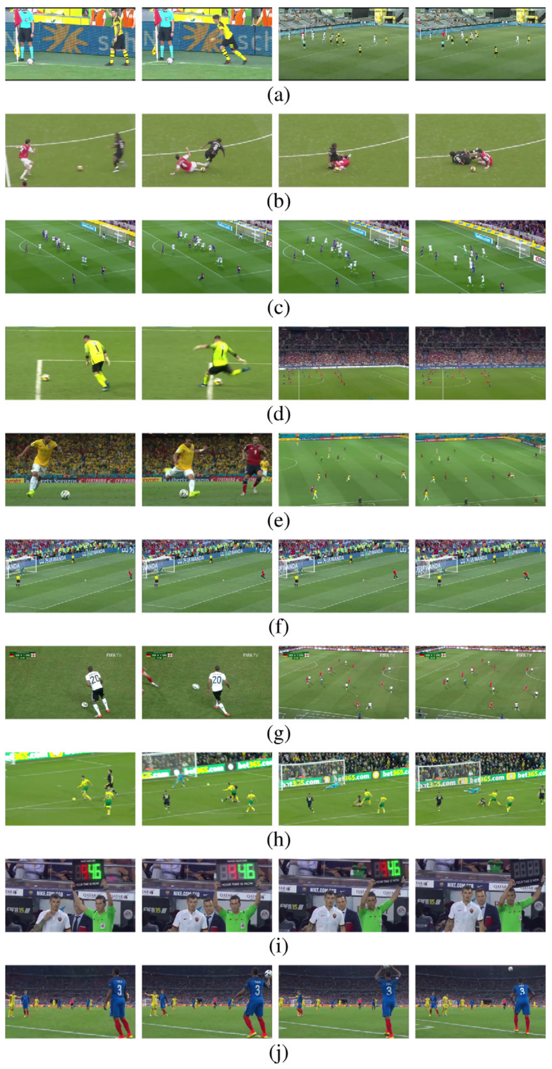


Fig. 3. Sequential frames of (a) corner, (b) foul, (c) free-kick, (d) goal-kick, (e) long-pass, (f) penalty, (g) short-pass, (h) shot-on-goal, (i) substitution, and (j) throw-in.

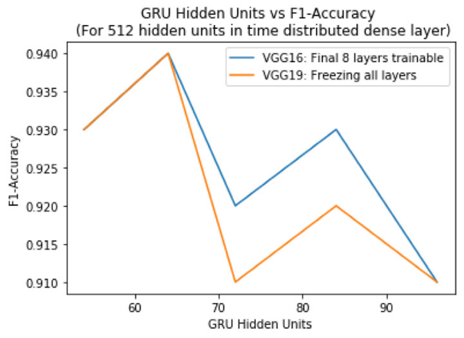
Analyzing the data, we see that transfer learning from Xception, and InceptionV3 models provide lower accuracy than VGG-based models. So, we were inspired to derive the best architectures for our SAC by fine-tuning the VGG-based models in more detail.

We used grid search to determine the best optimizer and learning rate. From SGD, Adam, and RMSprop optimizers, SGD with 0.0001 learning rate gave the best result. 0.9 momentum was used with the SGD optimizer. ReLU activation function is used in the TimeDistributed Dense layer as it produced better results than tanh and sigmoid. The F1-Score analysis is considered to evaluate model performance as it considers both false positive and false negative samples. *Model-4* and *Model-6* both provide an accuracy of 94% and among these two models, *Model-4* is the best as it yields comparatively less trainable parameters.

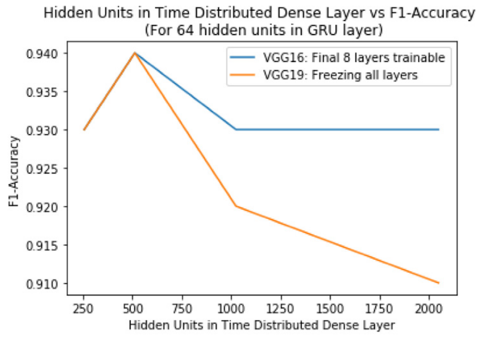
For further exploration, the VGG configuration for both the best models is kept fixed. Afterwards, two types of experiments were conducted: one by keeping the hidden unit of TimeDistributed dense layer as 512 and changing the GRU hidden units as shown in Fig. 4(a), and another by keeping the GRU hidden unit as 64 while changing the hidden units in TimeDistributed dense layer as depicted in Fig. 4(b). In either way, we can conclude that our *Model-4* and *Model-6* gives us the best architecture for SAC. The confusion matrices for the best 2 models, *Model-4* and *Model-6* are depicted in Figs. 5(a) and 5(b).

Our neural network is made up of CNN and RNN. Instead of building CNN from scratch, we leverage the pre-trained VGG model, which extracts the dominant spatial features from video frames. Being trained on a massive dataset like ImageNet, transfer learning models give us the ability to work with a comparatively smaller dataset. However, to detect the actions from videos, spatial features are not well enough. It also necessitates keeping track of the changes that occur throughout the video. Considering this fact, we adapted GRU, a simpler version of RNN, which tracks video frames' changes across the timesteps. Moreover, it solves the vanishing gradient problem of our architecture and has fewer trainable parameters compared to LSTM, another version of RNN.

The best model performance is shown in Fig. 6. From this figure, up to first 5 epochs, test accuracy is greater than training accuracy, which signifies that initially, the test set contains comparatively easier data to predict. Gradually, test accuracy is increasing with the training accuracy. After 25th epoch, the test accuracy is increasing very slowly, which indicates that the remaining test samples are hard to predict. EarlyStopping callback, monitoring no improvement of test loss for 5 consecutive epochs, stops the training at 50th epoch. We resume the process again with the best model saved to check if accuracy can be further increased. Despite having better accuracy, some actions are misclassified due to the actions being obscured by the surrounding players. Our best-tuned *Model-4* architecture, getting the best weights for SAC under complex scene conditions and overlapping sub-patterns, attains better accuracy of 94% compared with [9] (i.e. BoW, Dominant Motion, LSTM based approach) and [16] (i.e. CNN–GRU based architecture), as shown in Table 2. The architecture used in [16] is not so deep as VGG and giving a very low accuracy of only 32% for our dataset. So, we can conclude that, for SAC, our proposed VGG–GRU based deep architecture attains higher accuracy.



(a)



(b)

Fig. 4. F1-Score graph against (a) hidden units in GRU layer, (b) hidden units in TimeDistributed dense layer.

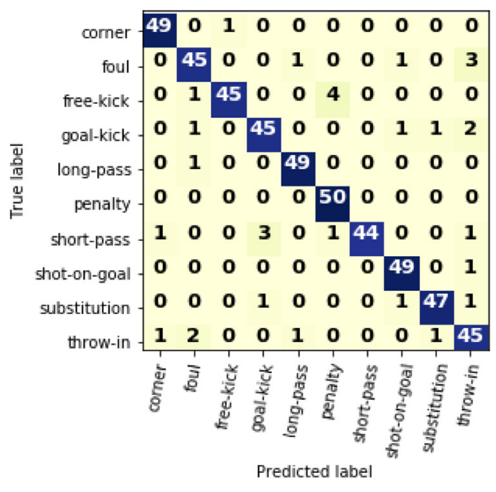
Table 2
Accuracy comparison with other works.

Reference	Dataset size	Accuracy
[9]	100	92%
[16]	300	32%
Our proposed architecture	3610	94%

The model proposed in [9] deals with only 4 soccer actions, while our work has considered 10 soccer actions. If the number of actions or videos is increased in [9], there might be a fall in accuracy value due to overlapping similar actions. Alternatively, there might be a modification of architecture to incorporate new videos. The architecture proposed in [16] commits a very low accuracy for our dataset. The architecture of [16] is shallow, which might be causing this low accuracy value. By increasing the depth of the model, we can expect an increase in accuracy for our dataset.

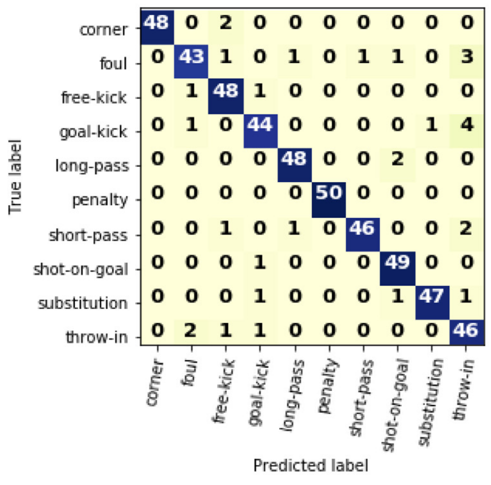
4. Conclusion

In this paper, we proposed hybrid VGG–GRU architectures to classify 10 different soccer actions. To the best of our knowledge, this is the highest number of distinct soccer actions to be classified in a single paper. A new dataset termed as SoccerAct10 is composed of various illumination conditions, various camera angles, continuous moving cameras, and overlapping patterns. Videos of varying lengths are considered, and the sequence length is determined empirically. We leverage pre-trained VGG models for feature extraction. GRU incorporates temporal features of videos and eliminates the vanishing



	corner	foul	free-kick	goal-kick	long-pass	penalty	short-pass	shot-on-goal	substitution	throw-in
corner	49	0	1	0	0	0	0	0	0	0
foul	0	45	0	0	1	0	0	1	0	3
free-kick	0	1	45	0	0	4	0	0	0	0
goal-kick	0	1	0	45	0	0	0	1	1	2
long-pass	0	1	0	0	49	0	0	0	0	0
penalty	0	0	0	0	0	50	0	0	0	0
short-pass	1	0	0	3	0	1	44	0	0	1
shot-on-goal	0	0	0	0	0	0	0	49	0	1
substitution	0	0	0	1	0	0	0	1	47	1
throw-in	1	2	0	0	1	0	0	0	1	45

(a)



	corner	foul	free-kick	goal-kick	long-pass	penalty	short-pass	shot-on-goal	substitution	throw-in
corner	48	0	2	0	0	0	0	0	0	0
foul	0	43	1	0	1	0	1	1	0	3
free-kick	0	1	48	1	0	0	0	0	0	0
goal-kick	0	1	0	44	0	0	0	0	1	4
long-pass	0	0	0	0	48	0	0	2	0	0
penalty	0	0	0	0	0	50	0	0	0	0
short-pass	0	0	1	0	1	0	46	0	0	2
shot-on-goal	0	0	0	1	0	0	0	49	0	0
substitution	0	0	0	1	0	0	0	1	47	1
throw-in	0	2	1	1	0	0	0	0	0	46

(b)

Fig. 5. Confusion matrix for (a) *Model-4*, (b) *Model-6*.

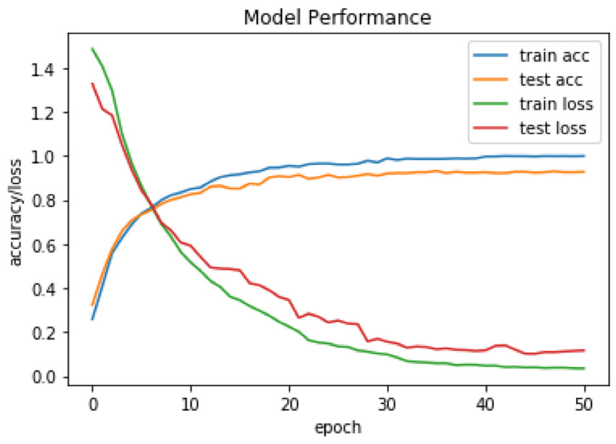


Fig. 6. Performance of *Model-4* after 1st early stopping.

gradient problem. Various experiments were conducted to determine the layers and hidden units. Our experimental results show that VGG–GRU based proposed method outperforms than the other existing models by achieving highest of 94% accuracy. However, this paper has made a few extensions for additional examination. By scaling the dataset, including more complex scene actions, this false detection rate can be minimized further. We commit to including more soccer actions. Moreover, another upcoming challenge will be considering the sudden appearance of spectators in between the main action. We also plan to explore other transfer learning models like Inception, Resnet, Mobilenet, and Xception in more detail and look over if accuracy can be further increased. By combining deep learning algorithms with the conventional feature extraction approach, some experiments will be executed. However, GPU memory is a first rate impediment to cooperate larger model.

CRediT authorship contribution statement

Anik Sen: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Visualization, Investigation, Writing - review & editing. **Kaushik Deb:** Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Soccer fan youtube behavior statistics - think with google, google. (n.d.), 2020, Accessed: 2020-08-07, https://www.thinkwithgoogle.com/data/soccer-fan-youtube-behavior-statistics.

[2] H. Liu, Highlight extraction in soccer videos by using multimodal analysis, in: 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), IEEE, 2017, pp. 2169–2173.

[3] A. Ekin, A.M. Tekalp, R. Mehrotra, Automatic soccer video analysis and summarization, IEEE Trans. Image Process. 12 (7) (2003) 796–807.

[4] W. Zhao, Y. Lu, H. Jiang, W. Huang, Event detection in soccer videos using shot focus identification, in: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2015, pp. 341–345.

[5] X. Qi, T. Wang, J. Liu, Comparison of support vector machine and softmax classifiers in computer vision, in: 2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE), IEEE, 2017, pp. 151–155.

[6] L. Ballan, M. Bertini, A. Del Bimbo, G. Serra, Action categorization in soccer videos using string kernels, in: 2009 Seventh International Workshop on Content-Based Multimedia Indexing, IEEE, 2009, pp. 13–18.

[7] P. Shi, X.-q. Yu, Goal event detection in soccer videos using multi-clues detection rules, in: 2009 International Conference on Management and Service Science, IEEE, 2009, pp. 1–4.

[8] Y. Kong, X. Zhang, Q. Wei, W. Hu, Y. Jia, Group action recognition in soccer videos, in: 2008 19th International Conference on Pattern Recognition, IEEE, 2008, pp. 1–4.

[9] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, A. Baskurt, Action classification in soccer videos with long short-term memory recurrent neural networks, in: International Conference on Artificial Neural Networks, Springer, 2010, pp. 154–159.

[10] E. Okafor, P. Pawara, F. Karaaba, O. Surinta, V. Codreanu, L. Schomaker, M. Wiering, Comparative study between deep learning and bag of visual words for wild-animal recognition, in: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2016, pp. 1–8.

[11] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: International Conference on Artificial Neural Networks, Springer, 2018, pp. 270–279.

[12] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015, CoRR, abs/1409.1556.

[13] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: International Conference on Machine Learning, 2013, pp. 1310–1318.

[14] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[15] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.

[16] M.A. Russo, A. Filonenko, K.-H. Jo, Sports classification in sequential frames using CNN and RNN, in: 2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT), IEEE, 2018, pp. 1–3.