Date of publication JUN-28, 2021, date of current version MAY-18, 2021.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.20.2.2176

Sports Recognition using Convolutional Neural Network with Optimization Techniques from Images and Live Streams

SHAKIL AHMED REJA, MOHAMMED MAHMUDUR RAHMAN
Faculty of Science & Engineering, International Islamic University Chittagong (IIUC), Kumira, Chittagong-4318, Bangladesh
(e-mail: sfshakilahmed@gmail.com, mmr.cse@iiuc.ac.bd)

Corresponding author: Shakil Ahmed Reja (e-mail: sfshakilahmed@gmail.com).

ABSTRACT This paper deals with the issue of automated image and video recognition of sports. It is a category of appreciation of human behavior, which is a very difficult task in the present day to classify images and video clips into a categorized gallery. This research paper proposes a sports detection system using a deeper CNN model that combines a fully connected layer with fine-tuning. It is applied to classify five individual sports groups through images and videos. In this work, we use a video classification method based on the image. Extended Resnet50 and VGG16 two pre-trained Deep CNNs are applied to build this sports detection system. RMSProp, ADAM & SGD optimizers are used to train the extended CNN models for five Epochs on the proposed 5sports dataset by handpicking thousands of sports images from the internet to very smoothly classify the five different types of sports. Training accuracy of approximately 83% is observed for ResNet50 with an SGD optimizer for 5 sports classes and 95% is observed for 3 sports classes.

KEYWORDS Sport classification; Multimedia content analysis; Deep learning; Pre-trained models; Convolutional Neural Networks; VGG16; Resnet50; Model-Optimizer.

I. INTRODUCTION

SPORTS is a major section in various broadcast media such as the internet, television, live streaming services, etc., and daily millions of videos of sports overflow the data servers. To further process them in order to carry out post-game analysis, tactics training for coaches, it is important to index each sport according to its category. Broadcasting firms can now monitor their operations more effectively when looking for different categories of videos without repetitive manual labor. Although a video arrangement is assessed to order it, this work is identified with the examination of scene-setting; along these lines, it can cause a further critical commitment to make a significant level away from gadgets.

Deep neural networks have made considerable progress in speech recognition, robotics, computer vision and NLP (Natural Language Processing) [1-4]. Such networks are quite effective in extracting complex, high-level abstractions of input data. Researchers applied several strategies to scene context analysis and visual information classification, according to which deep learning-based models have become progressively popular in recent times for complex computer vision tasks and fields of signal processing. Human beings often use a set of actions to describe any athletic activity and also consider surrounding areas.

Automatic recognition of sport is a part of a study of the multifunctional content analysis, and this method may be classified by algorithms or by data from image or images/video. The system collects data as a single image so that it concentrates on methods of classification of the image. Deep convolution neural Networks displayed advanced performance. One thousand different classes of images are stored in the ImageNet database, which is pre-trained to both ResNet50 & VGG16 CNN models [2, 5, 6].

II. LITERATURE REVIEW

Various researchers made many contributions to developing a system that can detect sport from images. In [7], the authors used ResNet50 and VGG16 for automatic player detection in broadcast sports videos and accuracy of up to 96% in NBA basketball clips. But there is still much more scope for improvement; more data should be collected and other CNN models should be used, i.e., ResNet50, to improve the accuracy more [2]. The VGG16 CNN model was used in [8] to detect 15 individual classes of sports and up to 92% accuracy was gained. For future improvement, other deep learning-based techniques, i.e., fine-tuning and data augmentation, etc. will be applied to increasing the accuracy more [9, 10].

The VGG16 CNN model was also used in [11], to identify various varieties of flowers with an accuracy of up to 71.5%. There is already a great deal of room to boost; other CNN models will be applied to increasing the accuracy more. In [12], the authors used ResNet50, Inception-ResNet-v2 and VGG16 with SGD optimizer to detect Live-Sports and they gained up to 96.8% accuracy in five types of sports. More data should be collected with the use of Multi-Stage-TL in the future. In [13], the authors used different types of CNN models like Xception, DenseNet201, DenseNet169, NASNetLarge, NASNetMobile, InceptionResNetV2, InceptionV3, VGG19, VGG16, MobileNet and ResNet50 to detect Lung Nodule and they gained up to 89.91% accuracy. For improvement, more data should be collected in the future. Xception, InceptionV3 and OverFeat various deep neural models are used in [14] to detect flower species from images and show the comparison among the CNN models. To increase accuracy, more data can be collected, and other different data related to many service providers can be classified to cover more domains of fields.

VGG16 and Resnet50 models with Adam optimizer were used in [15] to fire detection from images and up to 92% accuracy was gained. More data should be obtained to improve accuracy and use different optimizers like SGD and RMSProp optimizers [16]. VGG16 and Resnet50 models were also used in [17] to detect facial emotions from images and up to 92.4% accuracy was gained. A further large dataset may be obtained to improve accuracy. In paper [18] SGD optimizer was used to identify fire in videos and it was achieved 97.9% accuracy over 1427 fire images, 1758 smoke images and 2399 negative images.

Some problems are attempted to tackle in this paper and a system is developed that can identify multiple groups of sports based on two different Convolutional Neural Networks, which have very powerful extractors of spatial features with three different optimizers. To put together the proposed model, a dataset is built and data is personally categorized. This dataset is named as 5Sports. In the next section, the intention of using such a difficult model is elaborated.

III. METHODOLOGY

In this research, a sports recognition system has been proposed that can identify multiple groups of sports based on two different Convolutional Neural Networks with three different optimizers. The data flow diagram of our proposed sports recognition system is shown in Figure 1. In this diagram, six different processes have been used.

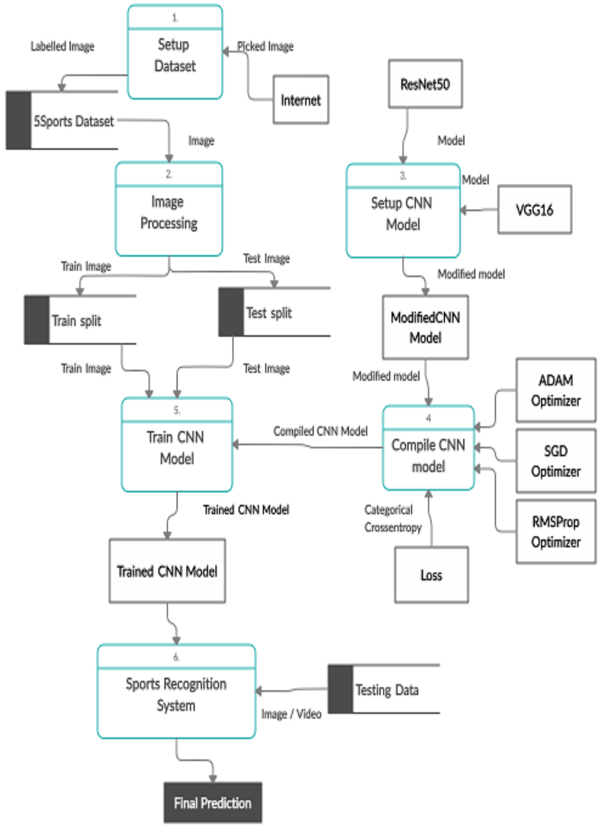


Figure 1. Data flow diagram of the sports recognition system

The first process created a data set called the 5Sports dataset. In this research, those images are collected from the internet and the data are individually labeled. After the setup of the dataset, the image processing process has been started. In this process, all the images of the dataset are converted to the grayscale image because of reduced memory size and that is why, they do not affect accuracy as well; resize is fixed 224*224 pixels because of two reasons: (1) The measurements of the input image are slightly smaller than what the CNN was trained on and increasing their scale adds too many objects and hurts loss/accuracy drastically. (2) High-resolution images contain tiny objects that are difficult to spot. It hurts precision to resize to the original input measurements of the CNN and increasing resolution would help boost our model, and image augmentation is used to generate a set of images in different properties like rotation, zoom, horizontal flip, etc. Next, our data is broken into splits for training and testing. The third process uses two different convolutional neural networks, called VGG16 and ResNet50, and compares them. Specific image classes such as 'ballplayer', 'baseball player', 'baseball', 'basketball', 'croquetball', 'cricket', 'football helmet', 'tennis ball' and more than 1,000 image classes are stored in the ImageNet database, which is pre-trained to both ResNet50 and VGG16 CNN models [2, 5, 6]. These ImageNet datasets were used as class weights in this study. Using this process needed to fine-tune the FC layers by AveragePolling2D, Dense and Dropout function. To integrate Transfer learning with fine-tuning, we remove the last predicting layer of the pre-trained model and replace it with our own predicting layers. These types of modification of CNN models are broadly discussed in CNN MODELS AND ARCHITECTURE section. After completing the modification of CNN models, it is necessary to compile the modified CNN models and do this in the 4th process.

To compile the CNN model, three different model optimizers were used, named Adam, SGD, RMSProp optimizer with categorical cross-entropy loss and the comparison among them was showed. Adam, SGD and RMSProp optimizer are broadly discussed in the MODEL OPTIMIZATION TECHNIQUES section. After completing the 1st, 2nd, 3rd and 4th process, we started to train CNN model process. Using train and test splits of the dataset files, the CNN model was trained by this method. Then newly created trained CNN models were ready to predict the image or video of the sports. 6th and the last process name is sports detection system process. In this process, we give our testing data as input and show the predicted label as output.

A. CNN MODELS AND ARCHITECTURE

VGG16 (Modified): In 2014, VGG 16 was introduced by Andrew Zisserman and Karen Simonyan from the Oxford University Visual Geometry Group Lab [6]. This model achieved 92% top-5 accuracy in the 2014 ILSVRC challenge and took the 1st and 2nd position. In the VGG model, there are two types of architecture layers. One is VGG16 and the other is VGG19. VGG19, which has 19-layer architecture, won the ImageNet competition in 2014, but VGG16, which has 16-layer architecture, achieved precision comparable to VGG19. There is not much difference between VGG16 and VGG19 models except convolutional filters. To pick the VGG16 over the VGG19 because it requires less time to train. This model is mainly pre-trained on the ImageNet dataset, which belongings are 1000 different classes of images [5]. Those Photos with RGB channels have a fixed size of 224 * 224. This model takes (224, 224, 3) as input and gives the output a vector with 1000 values. VGG16 model has 7 different arguments as follows: input_tensor, input_shape, pooling, include_top, weights, classes and classifier_activation. In this research, the FC layer of the VGG16 model is modified. So, ensure the previous head FC layer sets are left off. Table 1 shows a summary of the VGG16 model without the fully connected layer head.

Table 1. Summary of VGG16 model without the head FC layer

Layer (type)	Output Shape	Param #
. . .	. . .	. . .
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

Then, freshly prepare the head of the model basis of AveragePooling2D, Dense and Dropout [5, 19]. Then the head fully connected model was put on top of the base model. Table 2 shows a summary of the new ResNet50 model with the FC layer head.

Table 2. Summary of New VGG16 model with the head FC layer

Layer (type)	Output Shape	Param #
. . .	. . .	. . .
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
average_pooling2d_4 (Average	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
dense_8 (Dense)	(None, 256)	131328
dropout_4 (Dropout)	(None, 256)	0
dense_9 (Dense)	(None, 3)	771

This modified model has total parameters: 14,978,883, trainable parameters: 264,195 and non-trainable parameters: 14,714,688.

Figure 2(a) shows the original architecture of VGG16, and Figure 2(b) shows the modified architecture of VGG16.

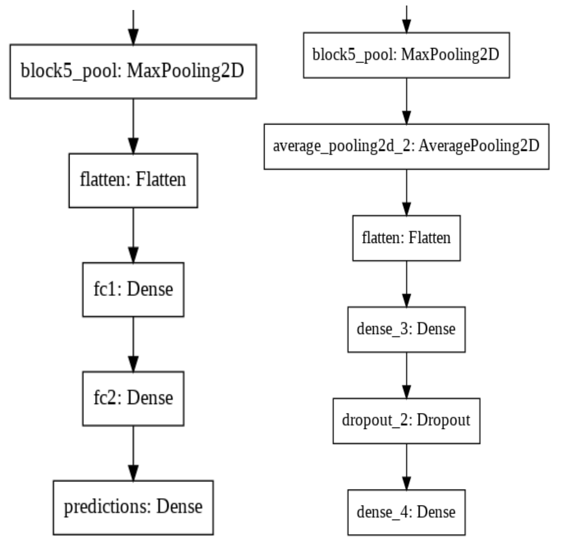


Figure 2. (a) Original Architecture of VGG16, (b) Modified Architecture of VGG16

ResNet50 (Modified): After VGG architecture has succeeded, deeper models were established which outperform shallower networks. There is a significant problem to train the deeper models because of the model complexity increase. Microsoft proposed extremely large, less complex model architectures to solve this significant problem, called ResNet [2]. ResNet model introduced the concept of skip connection [20]. To understand the skip connection, it is necessary to consider the following diagram.

Figure 3(a) is stacking the convolution layer sequentially, and Figure 3(b) is still stacking convolution layers as the previous one but now also adding the original input to the convolution block output.

(a) Without skip (b) With skip

Figure 3. (a) Without skip connection, (a) Without skip connection

The ResNet architecture consists of different depths of networks: 18, 34, 50, 101 and 152 layers. This research used ResNet architecture with 50 layers. ResNet50 model has 6 different arguments as follows: weights, include_top, input_tensor, input_shape, pooling and classes. In this research, the FC layer of the ResNet50 model is modified. So, ensure the former head FC layer sets are left off. Table 3 shows a summary of the ResNet50 model without the fully connected layer head and the final activation layer is 7*7*2048.

Table 3. Summary of ResNet50 model without the head FC layer

Layer (type)	Output Shape	Param #	Connected to
. . .	. . .	. . .	. . .
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	conv5_block3_2_relu[0][0]
conv5_block3_3_bn (BatchNormali	(None, 7, 7, 2048)	8192	conv5_block3_3_conv[0][0]
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_add[0][0]

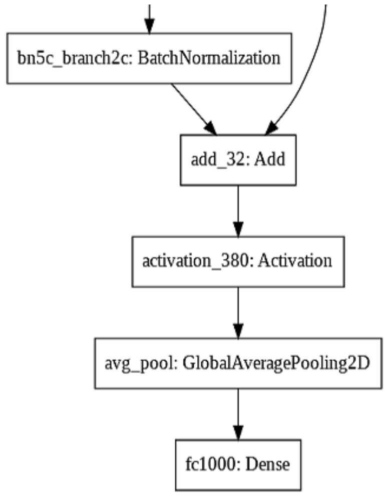
Then, freshly prepare the head of the model basis of AveragePooling2D, Dense and Dropout. Then the head fully connected model was put on top of the base model. Table 4 shows a summary of the new ResNet50 model with the FC layer head.

Table 4. Summary of New ResNet50 model with the head FC layer

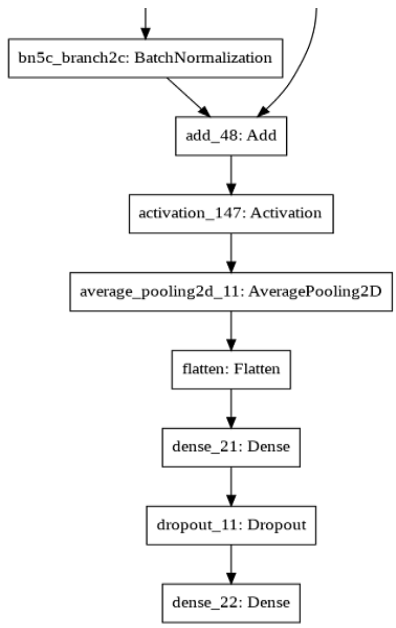
Layer (type)	Output Shape	Param #	Connected to
. . .	. . .	. . .	. . .
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	conv5_block3_2_relu[0][0]
conv5_block3_3_bn (BatchNormali	(None, 7, 7, 2048)	8192	conv5_block3_3_conv[0][0]
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_add[0][0]
average_pooling2d (AveragePooli	(None, 1, 1, 2048)	0	conv5_block3_out[0][0]
flatten (Flatten)	(None, 2048)	0	average_pooling2d[0][0]
dense (Dense)	(None, 256)	524544	flatten[0][0]
dropout (Dropout)	(None, 256)	0	dense[0][0]
dense_1 (Dense)	(None, 2)	514	dropout[0][0]

This modified model has total parameters: 24,638,339, trainable parameters: 1,050,627 and non-trainable parameters: 23,587,712.

Figure 4(a) shows the original architecture of ResNet50, and Figure 4(b) shows the modified architecture of ResNet50.



(a) Original Architecture of ResNet50



(b) Modified Architecture of ResNet50

Figure 4. (a) Original Architecture of ResNet50, (b) Modified Architecture of ResNet50.

Both pre-trained models (VGG16 and ResNet50) learn very generic features from the initial lower layers of the network. The weights of pre-trained models are frozen and not changed during training to achieve this initial layer. For learning task-specific characteristics, higher layers are used. Higher layers of pre-trained designs are trainable or fine-tuned. Output enhances with less time to practice.

B. MODEL OPTIMIZATION TECHNIQUES

Adam (Adaptive Moment Estimation): Adam optimizer is originated from the "Adaptive Moments"[16, 21, 22]. It is an update to the RMSProp optimizer. It includes bias corrections to estimates of both moments in the first order and moments in the second order to account for initialization at the origin. Moments of the gradients of the first-order and second-order are given below:

$$\begin{cases} \beta_t \leftarrow \rho_1\beta_{t-1} + (1-\rho_1)g_t \\ \gamma_t \leftarrow \rho_2\gamma_{t-1} + (1-\rho_2)\, g_t^2 \end{cases} \tag{1}$$

In equation no. 1, ρ_1 and ρ_2 maintain the balance between first and second-order moments of the gradients and the historical effects. This optimizer has given a number of parameters: epsilon, beta_1, beta_2, learning_rate, amsgrad, name, **kwargs.

SGD (Stochastic Gradient Descent): This technique is an iterative method for optimizing. This optimizer is used broadly in machine learning and deep learning [16, 22, 23]. In this optimizer learning rate is a critical factor. At each iteration, the training samples are randomly selected by SGD.

$$W_{i+1} = W_i - \eta\frac{\partial L}{\partial W_i}, \tag{2}$$

where W denotes weight parameter, L denotes loss function, *i* denotes the number of iterations, and η represents the learning rate. This optimizer has given a number of parameters: learning_rate, momentum, nesterov, name, **kwargs.

RMSProp (Root Mean Square Propagation): This technique is used to solve the problem of AdaGrad optimizer by using exponentially decaying averaging technique [16, 22]. It maintains the balance between second-order moments of the gradients and the historical effects. The running average is calculated in terms of means square,

$$v(w,t) := \gamma v(w,t-1) + (1-\gamma)\left(\nabla Q_i(w)\right)^2, \tag{3}$$

where γ denotes the forgetting factor, this optimizer has given the number of parameters: learning_rate, rho, momentum, epsilon, centered, name, **kwargs.

IV. EXPERIMENTS

A. THE DATASET

To train the proposed model, a dataset is prepared that consists of many images in different classes of sports pictures gathered from the internet, and the details were individually labeled. These datasets are named as 5Sports, which consist of five types of sports, as shown in Figure 5. We have 3,870 images in five types of sports.

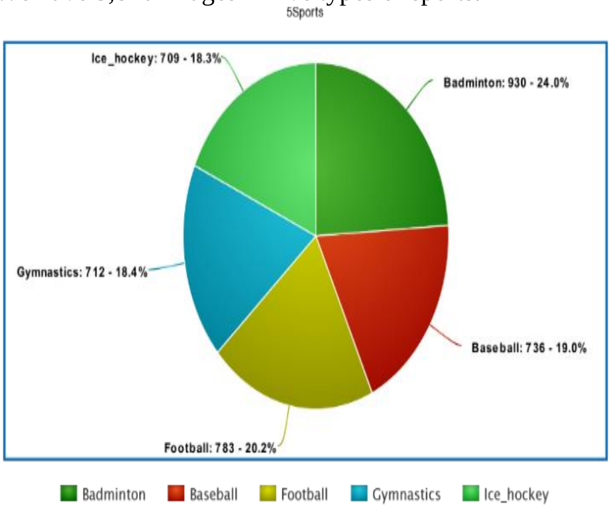


Figure 5. 5sports dataset

Some examples of datasets in various sport classes are shown in Figure 6(a) to (e).


(a) Badminton (b) Baseball


(c) Football (d) Gymnastics

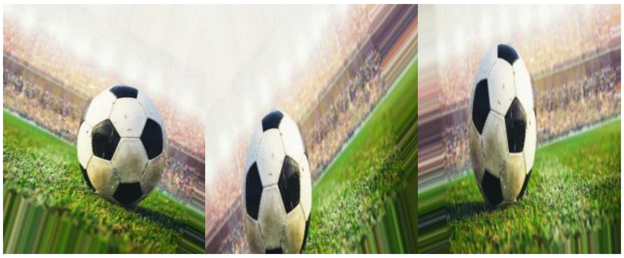
(e) Ice_hockey

Figure 6. Examples of different sport classes.

B. EXPERIMENTAL SETTINGS

To train our modified CNN models, all the images are resized to a particular size. For both VGG16 & ResNet50 this size is 224*224. After resizing the images data augmentation was applied to produce a modified example of images from the primary images by applying ImageDataGenerator class from Keras [10]. Figure 7 displays an example of data augmentation.

(a) After Augmentation



(b) Before Augmentation

Figure 7. (a) After Augmentation; (b) Before Augmentation

To calculate better performance, 5Sports datasets were divided into two subsets. The first set contains three sport classes, which are baseball, gymnastics and ice_hockey. The second set contains five sport classes, which are baseball, gymnastics, ice_hockey, badminton and football. While training, our CNN models split the datasets into 75% for training and 25% for testing purposes. With the aid of Adam, SGD, RMSProp optimizer, respectively, both sets were studied with VGG16 and Reset50 deep CNNs model. Five epochs train both models. In SGD and RMSProp optimizer, learning_rate = 0.01 and momentum = 0.9 were placed wherein Adam optimizer, placed its default parameters. Experimental results are shown in the next subsection.

C. RESULTS

Table 5 represents the result of VGG16 with the RMSProp optimizer to classify three sport classes.

Table 5. Results of VGG16 model with RMSProp optimizer

	Precision	Recall	F1-Score	Support
baseball	0.88	0.96	0.92	183
gymnastics	0.90	0.91	0.91	178
ice_hockey	0.97	0.88	0.92	177
accuracy			0.91	538
macro avg	0.92	0.91	0.91	538
weighted avg	0.92	0.91	0.91	538

Here, the precision score of baseball, gymnastics, and ice-hockey is 0.88, 0.90 & 0.97, respectively, out of 1. Also, the recall score of baseball, gymnastics, and ice-hockey is 0.96, 0.91 & 0.88, respectively, out of 1. The F1 score of baseball, gymnastics, and ice-hockey is 0.92, 0.91 & 0.92, respectively, out of 1. Now, the precision score of the macro average and weighted average is 0.92 out of 1. The recall score of the macro average and weighted average is 0.91 out of 1. The F1 score of the macro average and weighted average is 0.91 out of 1.

Table 6 represents the result of VGG16 with Adam optimizer to classify three sport classes.

Table 6. Results of VGG16 model with Adam optimizer

	Precision	Recall	F1-Score	Support
baseball	0.87	0.95	0.91	183
gymnastics	0.92	0.87	0.90	178
ice_hockey	0.96	0.93	0.95	177
accuracy			0.92	538
macro avg	0.92	0.92	0.92	538
weighted avg	0.92	0.92	0.92	538

Here, the precision score of baseball, gymnastics, and ice-hockey is 0.87, 0.92 and 0.96, respectively, out of 1. Also, the recall score of baseball, gymnastics, and ice-hockey is 0.95, 0.87 and 0.93, respectively, out of 1. The F1 score of baseball, gymnastics, and ice-hockey is 0.91, 0.90 and 0.95, respectively, out of 1. Now, the precision score of the macro average and weighted average is 0.92 out of 1. The recall score of the macro average and weighted average is 0.92 out of 1. The F1 score of the macro average and weighted average is 0.92 out of 1.

Table 7 represents the result of VGG16 with the SGD optimizer to classify three sport classes.

Table 7. Results of VGG16 model with SGD optimizer

	Precision	Recall	F1-Score	Support
baseball	0.92	0.92	0.92	183
gymnastics	0.85	0.96	0.90	178
ice_hockey	0.97	0.85	0.91	177
accuracy			0.91	538
macro avg	0.91	0.91	0.91	538
weighted avg	0.91	0.91	0.91	538

Here, the precision score of baseball, gymnastics, and ice-hockey is 0.92, 0.85 and 0.97, respectively, out of 1. Also, the recall score of baseball, gymnastics, and ice-hockey is 0.92, 0.96 and 0.85, respectively, out of 1. The F1 score of baseball, gymnastics, and ice-hockey is 0.92, 0.90 and 0.91, respectively, out of 1. Now, the precision, recall and F1 score of macro average and weighted average is 0.91 out of 1. Table 8 shows a summary of Table 5-7.

Table 8. Results of VGG16 model with RMSProp, ADAM and SGD optimizers for 3 sport classes

3 Sport Classes		
	VGG16(modified)	
Optimizer	**Accuracy**	**Time (hour)**
RMSProp	91%	1.26
Adam	92%	0.97
SGD	91%	1.24

Here, VGG16(modified) model achieved the highest 92% accuracy with the ADAM optimizer.

Table 9 represents the result of ResNet50 with the RMSProp optimizer to classify three sport classes.

Table 9. Results of ResNet50 model with RMSProp optimizer

	Precision	Recall	F1-Score	Support
baseball	0.93	0.95	0.94	183
gymnastics	0.89	0.92	0.91	178
ice_hockey	0.98	0.92	0.95	177
accuracy			0.93	538
macro avg	0.93	0.93	0.93	538
weighted avg	0.93	0.93	0.93	538

Here, the precision score of baseball, gymnastics, and ice-hockey is 0.93, 0.89 and 0.98, respectively, out of 1. Also, the recall score of baseball, gymnastics, and ice-hockey is 0.95, 0.95 and 0.92 respectively out of 1. The F1 score of baseball, gymnastics, and ice-hockey is 0.94, 0.91 and 0.95, respectively, out of 1.Now, the precision, recall and F1 score of macro average and weighted average is 0.93 out of 1.

Table 10 represents the result of ResNet50 with Adam optimizer to classify three sport classes.

Table 10. Results of ResNet50 model with Adam optimizer

	Precision	Recall	F1-Score	Support
baseball	0.89	0.97	0.92	183
gymnastics	0.98	0.88	0.93	178
ice_hockey	0.96	0.96	0.96	177
accuracy			0.94	538
macro avg	0.94	0.94	0.94	538
weighted avg	0.94	0.94	0.94	538

Here, the precision score of baseball, gymnastics, and ice-hockey is 0.89, 0.98 and 0.96, respectively, out of 1. Also, the recall score of baseball, gymnastics, and ice-hockey is 0.97, 0.88 and 0.96, respectively, out of 1. The F1 score of baseball, gymnastics, and ice-hockey is 0.92, 0.93 and 0.96, respectively, out of 1.Now, the precision, recall and F1 score of the macro average and weighted average is 0.94 out of 1.

Table 11 represents the result of ResNet50 with the SGD optimizer to classify three sport classes.

Table 11. Results of ResNet50 model with SGD optimizer

	Precision	Recall	F1-Score	Support
baseball	0.93	0.99	0.96	183
gymnastics	0.98	0.89	0.93	178
ice_hockey	0.95	0.98	0.96	177
accuracy			0.95	538
macro avg	0.95	0.95	0.95	538
weighted avg	0.95	0.95	0.95	538

Here, the precision score of baseball, gymnastics, and ice-hockey is 0.93, 0.98 and 0.95, respectively, out of 1. Also, the recall score of baseball, gymnastics, and ice-hockey is 0.99, 0.89 and 0.98, respectively, out of 1. The F1 score of baseball, gymnastics, and ice-hockey is 0.96, 0.93 and 0.96, respectively, out of 1.Now, the precision, recall and F1 score of macro average & weighted average is 0.95 out of 1. Table 12 shows a summary of Table 9-11.

Table 12. Results of ResNet50 model with RMSProp, ADAM and SGD optimizers for 3 sport classes

3 Sport Classes		
	ResNet50(modified)	
Optimizer	**Accuracy**	**Time (hour)**
RMSProp	93%	1.37
Adam	94%	1.31
SGD	95%	1.30

Here, ResNet50(modified) model achieved the highest 95% accuracy with the SGD optimizer.

Table 13 represents the result of VGG16 with the RMSProp optimizer to classify five sport classes.

Table 13. Results of VGG16 model with RMSProp optimizer

	Precision	Recall	F1-Score	Support
badminton	0.87	0.79	0.83	232
baseball	0.52	0.78	0.62	183
football	0.80	0.78	0.79	196
gymnastics	0.88	0.73	0.80	178
ice_hockey	0.99	0.80	0.88	177
accuracy			0.78	965
macro avg	0.81	0.77	0.78	965
weighted avg	0.81	0.78	0.79	965

Here, the precision score of badminton, baseball, football, gymnastics and ice-hockey is 0.87, 0.52, 0.80, 0.88 and 0.99 respectively out of 1. Also, the recall score of badminton, baseball, football, gymnastics and ice-hockey is 0.79, 0.78, 0.78, 0.73 and 0.80 respectively out of 1. The F1 score of badminton, baseball, football, gymnastics and ice-hockey is 0.83, 0.62, 0.79, 0.80 and 0.88, respectively, out of 1. Now, the precision score of the macro average and weighted average is 0.81 out of 1. The recall score of the macro average and weighted average is 0.77 and 0.78, respectively, out of 1. The F1 score of the macro average and weighted average is 0.78 and 0.79, respectively, out of 1.

Table 14 represents the result of VGG16 with Adam optimizer to classify five sport classes.

Table 14. Results of VGG16 model with Adam optimizer

	Precision	Recall	F1-Score	Support
badminton	0.75	0.83	0.79	232
baseball	0.78	0.75	0.76	183
football	0.85	0.68	0.75	196
gymnastics	0.78	0.80	0.79	178
ice_hockey	0.85	0.93	0.89	177
accuracy			0.80	965
macro avg	0.80	0.80	0.80	965
weighted avg	0.80	0.80	0.80	965

Here, the precision score of badminton, baseball, football, gymnastics and ice-hockey is 0.75, 0.78, 0.85, 0.78 and 0.85 respectively out of 1. Also, the recall score of badminton, baseball, football, gymnastics and ice-hockey is 0.83, 0.75, 0.68, 0.80 and 0.93 respectively out of 1. The F1 score of badminton, baseball, football, gymnastics and ice-hockey is 0.79, 0.76, 0.75, 0.79 and 0.89, respectively, out of 1.Now, the precision, recall and F1 score of the macro average and weighted average is 0.80 out of 1.

Table 15 represents the result of VGG16 with the SGD optimizer to classify five sport classes.

Table 15. Results of VGG16 model with SGD optimizer

	Precision	Recall	F1-Score	Support
badminton	0.82	0.87	0.84	232
baseball	0.82	0.78	0.80	183
football	0.77	0.88	0.82	196
gymnastics	0.81	0.72	0.77	178
ice_hockey	0.95	0.88	0.91	177
accuracy			0.83	965
macro avg	0.83	0.83	0.83	965
weighted avg	0.83	0.83	0.83	965

Here, the precision score of badminton, baseball, football, gymnastics and ice-hockey is 0.82, 0.82, 0.77, 0.81 and 0.95 respectively out of 1. Also, the recall score of badminton, baseball, football, gymnastics and ice-hockey is 0.87, 0.78, 0.88, 0.72 and 0.88 respectively out of 1. The F1 score of badminton, baseball, football, gymnastics and ice-hockey is 0.84, 0.80, 0.82, 0.77 and 0.91, respectively, out of 1. Now, the precision, recall and F1 score of the macro average and weighted average is 0.83 out of 1. Table 16 shows a summary of Table 13-15.

Table 16. Results of VGG16 model with RMSProp, ADAM and SGD optimizers for 5 sport classes

5 Sport Classes		
	VGG16(modified)	
Optimizer	**Accuracy**	**Time (hour)**
RMSProp	78%	1.95
Adam	80%	2.03
SGD	83%	1.95

Here, VGG16(modified) model achieved the highest 83% accuracy with the SGD optimizer.

Table 17 represents the result of ResNet50 with the RMSProp optimizer to classify five sport classes.

Table 17. Results of ResNet50 model with RMSProp optimizer

	Precision	Recall	F1-Score	Support
badminton	0.63	0.97	0.76	232
baseball	0.98	0.25	0.39	183
football	0.57	0.89	0.69	196
gymnastics	0.99	0.46	0.63	178
ice_hockey	0.96	0.92	0.94	177
accuracy			0.71	965
macro avg	0.82	0.70	0.68	965
weighted avg	0.81	0.71	0.69	965

Here, the precision score of badminton, baseball, football, gymnastics and ice-hockey is 0.63, 0.98, 0.57, 0.99 and 0.96 respectively out of 1. Also, the recall score of badminton, baseball, football, gymnastics and ice-hockey is 0.97, 0.25, 0.89, 0.46 and 0.92 respectively out of 1. The F1 score of badminton, baseball, football, gymnastics and ice-hockey is 0.76, 0.39, 0.69, 0.63 and 0.94 respectively out of 1. Now, the precision score of the macro average & weighted average is 0.82 and 0.81, respectively, out of 1. The recall score of the macro average and weighted average is 0.70 and 0.71, respectively, out of 1. The F1 score of the macro average and weighted average is 0.68 and 0.69, respectively, out of 1.

Table 18 represents the result of ResNet50 with Adam optimizer to classify five sport classes.

Table18. Results of ResNet50 model with Adam optimizer

	Precision	Recall	F1-Score	Support
badminton	0.78	0.93	0.85	232
baseball	0.77	0.67	0.72	183
football	0.70	0.93	0.80	196
gymnastics	0.96	0.67	0.79	178
ice_hockey	0.99	0.84	0.91	177
accuracy			0.81	965
macro avg	0.84	0.81	0.81	965
weighted avg	0.84	0.81	0.81	965

Here, the precision score of badminton, baseball, football, gymnastics and ice-hockey is 0.78, 0.77, 0.70, 0.96 and 0.99 respectively out of 1. Also, the recall score of badminton, baseball, football, gymnastics and ice-hockey is 0.93, 0.67, 0.93, 0.67 and 0.84 respectively out of 1. The F1 score of badminton, baseball, football, gymnastics and ice-hockey is 0.85, 0.72, 0.80, 0.79 and 0.91, respectively, out of 1. Now, the precision score of the macro average and weighted average is 0.84 out of 1. The recall and F1 score of the macro average and weighted average is 0.81 out of 1.

Table 19 represents the result of ResNet50 with the SGD optimizer to classify five sport classes.

Table 19. Results of ResNet50 model with SGD optimizer

	Precision	Recall	F1-Score	Support
badminton	0.81	0.93	0.87	232
baseball	0.86	0.68	0.76	183
football	0.70	0.96	0.81	196
gymnastics	0.99	0.60	0.75	178
ice_hockey	0.95	0.95	0.95	177
accuracy			0.83	965
macro avg	0.86	0.83	0.83	965
weighted avg	0.86	0.83	0.83	965

Here, the precision score of badminton, baseball, football, gymnastics and ice-hockey is 0.81, 0.86, 0.70, 0.99 and 0.95 respectively out of 1. Also, the recall score of badminton, baseball, football, gymnastics and ice-hockey is 0.93, 0.68, 0.96, 0.60 and 0.95 respectively out of 1. The F1 score of badminton, baseball, football, gymnastics and ice-hockey is 0.87, 0.76, 0.81, 0.75 and 0.95 respectively out of

1. Now, the precision score of the macro average & weighted average is 0.86 out of 1. The recall and F1 score of the macro average and weighted average is 0.83 out of 1. Table 20 shows a summary of Table 17-19.

Table 20. Results of ResNet50 model with RMSProp, ADAM & SGD optimizers for 5 sport classes

5 Sport Classes		
	Resnet50(modified)	
Optimizer	Accuracy	Time (hour)
RMSProp	71%	3.18
Adam	81%	3.15
SGD	83%	3.17

Here, ResNet50(modified) model achieved the highest 83% accuracy with the SGD optimizer.

Table 21 and Table 22 show the difference between the CNN model for 3 and 5 sport classes, respectively. In Table 21, VGG16(modified) achieved the highest 92% accuracy with Adam optimizer in our proposed datasets. But all other optimizers also performed well and gained 91% accuracy in SGD & RMSprop, respectively. Also, ResNet50(modified) achieved the highest 95% accuracy with the SGD optimizer, where Adam and RMSProp optimizer gained 94% and 93% accuracy, respectively.

Table 21. Difference between CNN models for three-sport classes

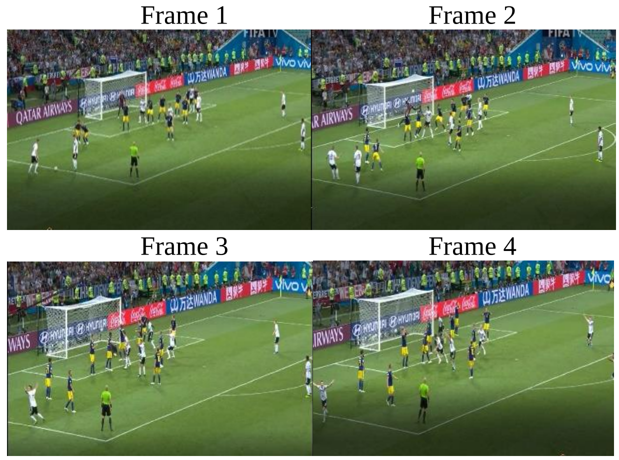
3 Sport Classes				
	VGG16(modified)		ResNet50(modified)	
Optimizer	Accuracy	Time (hour)	Accuracy	Time (hour)
RMSProp	91%	1.26	93%	1.37
Adam	92%	0.97	94%	1.31
SGD	91%	1.24	95%	1.30

In Table 22, VGG16(modified) achieved the highest 83% accuracy with the SGD optimizer in our proposed datasets. But all other optimizers also performed well and gained 80% and 78% accuracy in Adam and RMSprop, respectively. Also, ResNet50(modified) achieved the highest 83% accuracy with the SGD optimizer, where Adam and RMSProp optimizer gained 81% and 71% accuracy, respectively.

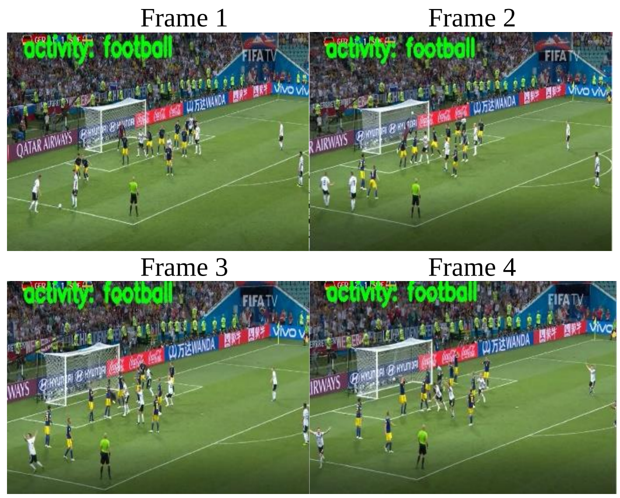
Table 22. Difference between CNN models for five sport classes

5 Sport Classes				
	VGG16(modified)		ResNet50(modified)	
Optimizer	Accuracy	Time (hour)	Accuracy	Time (hour)
RMSProp	78%	1.95	71%	3.18
Adam	80%	2.03	81%	3.15
SGD	83%	1.95	83%	3.17

Fig. 8(a) is an input video frame to detect which sport has been played in this video, and Figure 8(b) shows the result of the input video frame with green writing "activity: football".



(a) Input video sequence example.



(a) Output video sequence example.

Figure 8 (a) Input video sequence example; (b) Output video sequence example

V. CONCLUSION

This research introduced two pre-trained Deep CNNs: Resnet50 and VGG16 and with three model optimizers: Adam, SGD and RMSProp to classify five individual sports groups through images and videos. Both CNN models are applied to our datasets with different types of optimizers. To verify it effectiveness, this approach was applied to the real-world environment and got a positive vibe as shown in Figure 8. In the future, to increase the accuracy it will be applied to the new CNN models with different types of optimizers.

References

[1] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, *et al.*, "Dual learning for machine translation," in *Advances in neural information processing systems*, 2016, pp. 820-828.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, 2016, pp. 770-778, https://doi.org/10.1109/CVPR.2016.90.
[3] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, pp. 705-724, 2015, https://doi.org/10.1177/0278364914549607.
[4] D. Amodei, S. Ananthanarayanan, R., Bai, J. Anubhai, "Deep Speech 2: End-to-end speech recognition in English and Mandarin," Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 173-182. [Online]. Available at: http://proceedings.mlr.press/v48/amodei16.html.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks, " *Advances in Neural Information Processing Systems*, 2012, pp. 1097-1105.
[6] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," *Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 169-175, https://doi.org/10.1109/CCWC.2018.8301729.
[7] A. Senocak, T.-H. Oh, J. Kim, and I. So Kweon, "Part-based player identification using deep convolutional representation and multi-scale pooling," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1732-1739, https://doi.org/10.1109/CVPRW.2018.00225.
[8] M. A. Russo, L. Kurnianggoro, and K.-H. Jo, "Classification of sports videos with combination of deep learning models and transfer learning," *Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2019, pp. 1-5, https://doi.org/10.1109/ECACE.2019.8679371.
[9] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 1655-1668, 2018, https://doi.org/10.1109/TPAMI.2018.2846566.
[10] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," *Proceedings of the 2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 117-122, https://doi.org/10.1109/IIPHDW.2018.8388338.
[11] N. R. Gavai, Y. A. Jakhade, S. A. Tribhuvan, and R. Bhattad, "MobileNets for flower classification using TensorFlow," *Proceedings of the 2017 International Conference on Big Data, IoT and Data Science (BID)*, 2017, pp. 154-158, https://doi.org/10.1109/BID.2017.8336590.
[12] T. Bi, D. Jarnikov, and J. Lukkien, "Supervised two-stage transfer learning on imbalanced dataset for sport classification," *Proceedings of the International Conference on Image Analysis and Processing*, 2019, pp. 356-366, https://doi.org/10.1007/978-3-030-30642-7_32.
[13] R. V. M. da Nóbrega, S. A. Peixoto, S. P. P. da Silva, and P. P. Rebouças Filho, "Lung nodule classification via deep transfer learning in CT lung images, " *Proceedings of the 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)*, 2018, pp. 244-249, https://doi.org/10.1109/CBMS.2018.00050.
[14] I. S. Jayasinghe, D. Wijesekara, C. L. Senanayake, N. Kodagoda, S. I. Kahawandala, and K. Suriyawansa, "Video classification using pre-trained models in the convolutional neural networks," Pr*oceedings of the International Conference on Data Mining*, 2019.
[15] J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," *Proceedings of the International Conference on Engineering Applications of Neural Networks*, 2017, pp. 183-193, https://doi.org/10.1007/978-3-319-65172-9_16.
[16] R. Poojary and A. Pai, "Comparative study of model optimization techniques in fine-tuned CNN models," *Proceedings of the 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2019, pp. 1-4, https://doi.org/10.1109/ICECTA48151.2019.8959681.
[17] P. Dhankhar, "ResNet-50 and VGG-16 for recognizing facial emotions," *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 13, pp. 126-130, 2019.
[18] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," *Proceedings of the 42nd IEEE Annual Conference of the (IECON'2016)*, 2016, pp. 877-882, https://doi.org/10.1109/IECON.2016.7793196.
[19] D. Wang, L. Zhang, K. Xu, and Y. Wang, "Acoustic scene classification based on dense convolutional networks incorporating multi-channel features, " *Journal of Physics: Conference Series*, p. 012037, 2019, https://doi.org/10.1088/1742-6596/1169/1/012037.
[20] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119-133, 2019, https://doi.org/10.1016/j.patcog.2019.01.006.
[21] Z. Zhang, "Improved Adam optimizer for deep neural networks," *Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1-2, https://doi.org/10.1109/IWQoS.2018.8624183.
[22] A. M. Taqi, A. Awad, F. Al-Azzo, and M. Milanova, "The impact of multi-optimizers and data augmentation on TensorFlow convolutional neural network performance," *Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018, pp. 140-145, https://doi.org/10.1109/MIPR.2018.00032.
[23] J. Yang and G. Yang, "Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer," *Algorithms*, vol. 11, p. 28, 2018, https://doi.org/10.3390/a11030028.

SHAKIL AHMED REJA *is a Bachelor student in Faculty of Computer Science and Engineering, International Islamic University Chittagong (IIUC), Reza researches image processing and pattern recognition.*

MOHAMMED MAHMUDUR RAHMAN*, currently works at the Faculty of Computer Science and Engineering, International Islamic University Chittagong (IIUC), as Assistant Professor. Rahman researches pattern recognition, computing, system design and ontology.*