Multi-label Class-imbalanced Action Recognition in Hockey Videos via 3D Convolutional Neural Networks

Konstantin Sozykin, Adil Khan *Member, IEEE,* Stanislav Protasov, Rasheed Hussain *Member, IEEE*

***Abstract*—Automatic analysis of the video is one of most complex problems in the fields of computer vision and machine learning. A significant part of this research deals with (human) activity recognition (HAR) since humans, and the activities that they perform, generate most of the video semantics. Video-based HAR has applications in various domains, but one of the most important and challenging is HAR in sports videos. Some of the major issues include high inter- and intra-class variations, large class imbalance, the presence of both group actions and single player actions, and recognizing simultaneous actions, i.e., the multi-label learning problem. Keeping in mind these challenges and the recent success of CNNs in solving various computer vision problems, in this work, we implement a 3D CNN based multi-label deep HAR system for multi-label class-imbalanced action recognition in hockey videos. We test our system for two different scenarios: an ensemble of k binary networks vs. a single k-output network, on a publicly available dataset. We also compare our results with the system that was originally designed for the chosen dataset. Experimental results show that the proposed approach performs better than the existing solution.**

***Index Terms*—deep learning, computer vision, 3D convolutional neural networks, activity recognition**

I. Introduction

Automatic recognition of human activities in a video is an exciting and challenging research area. Robust solutions of this problem can be applied in various fields, such as video retrieval and searching [1], robotics[2], healthcare [3], sport analytics[4], and security etc. G. Johansson [5] pioneered this area by developing the first method for modeling and analysis of human locomotion in visual data. Since then a significant amount of work has been done in this regard, details of which can be found in [6].

It should be noted that although there is no universal definition for the terms action and activity, some scientists define action as a single movement pattern of a human body, and activity as a composition or series of such actions [6]. However, in this work, these terms are used as synonyms.

Human action recognition in visual data is quite challenging due to interpersonal differences, and variations in motion patterns and recording settings. For a better understanding, it would be reasonable to describe these challenges in detail.

Variations in motion or movement patterns may result because the same activity may be performed differently by different individuals as well as by the same person [7]. There are many reasons for this. For example, stress, time of the day, health and emotional states. From machine learning point of view, it is called high intraclass variability or variance problem.

On the other hand, there is interclass similarity; it is the case when two or more different classes have similar characteristics, but they are fundamentally different. Good and straightforward examples of this case are activities like walking and running (jogging). They have a higher visual similarity, but they are definitely from different action categories. [8]

Recording settings and sensor characteristics also play a significant role. Different recording settings sometimes require different algorithmic approaches e.g. static vs. moving camera-based human action recognition systems [6]. Similarly, sensor characteristics may influence the quality of data acquisition.

In addition to the problems mentioned above, the class imbalance is a typical problem in action recognition task. It is the case when the classes are not represented equally. This could lead to a problem since many machine learning approaches (especially complex algorithms like neural networks) work well only if the number of observations for all classes are roughly equal [9]. There are a number of methods in machine learning literature that can be used to handle this problem. For example, balancing the training data by means of oversampling or under-sampling, and class weight adjustment [10].

Furthermore, in real-life, it is common to have situations when at any given moment more than one action may happen. It happens because in the case of videos often multiple persons are present, and they may simultaneously interact with each other or with different objects. This transforms the automatic understanding of video to a more complex problem. From machine learning point of view, it means that an observation may belong to multiple classes. Therefore, human action recognition problem may lead to the multi-label learning problem. Multi-label learning is a generalization of supervised learning with the assumption that observed instances can belong to more than one class simultaneously. As a large field of research, it has its own issues and associated methods. For example, it can require special loss functions or algorithms to work in k-output mode. More details can be found in [11], [12].

K. Sozykin, A.Khan, S. Protasov and R. Hussain are with Faculty of Computer Science and Engineering, Innopolis University (e-mail: k.sozykin@innopolis.ru, a.khan@innopolis.ru, s.protasov@innopolis.ru, r.hussain@innopolis.ru).

The authors would like to thank *Marc-Andre Carbonneau et. al.* for providing well annotated dataset and source code for correct validation of this paper.

Also we would like to thank Moscow Institute of Physics and Technology *Fluid Dynamics and Seismoacoustics Lab* and especially *Ivan Tsybulin* for providing Nvidia Tesla computational cluster.

Last but not the least, the domain of the chosen activities, such as home activities or sports activities, can further add to the complexity of the recognition task. In the previous paragraphs, we discussed the class imbalance and multi-label learning problems. In the case of sports action recognition from video, they are both strongly present. For example, in many active team sports, such as hockey or soccer, *Goal* is a very rare action compared to *Running*. Therefore, even if we have many hours of video data it is difficult to collect enough samples of the *Goal* class. On the other hand, in team sports, players may perform different actions at the same time. A combination of such factors makes the analysis of sports video more complicated.

Accordingly, in this work, we implement a deep learning approach for multi-label action recognition in hockey videos in the presence of class imbalance problem. Hockey is a fast-paced team-based sports, which may present all of the problems that are mentioned above. The experimental results show that our approach is capable of providing better recognition rates than existing work on a state-of-the-art video dataset of the hockey game.

We chose neural networks, especially deep networks, for building our recognition system, since they offer real advantages. Firstly, deep learning and convolution neural networks (CNNs) have recently shown excellent performance in different complicated visual tasks. Examples of such visual tasks include but are not limited to image recognition [13], [14], [15], object detection and recognition [16], [17], [18], object tracking [19], image segmentation [20], [21], style-transferring [22], self-driving cars and robotics [23], etc.

Next, with convolution-based feature extraction, we can learn not only the classification models but also the class representations [24]. By learning representations, we mean learning a set of abstract features that can efficiently represent each class. In general, if we have a better representation of some data, especially visual, we can do a better learning for related or similar tasks using these representations. In other words, we can do transfer learning [25] to save time and computational resources. In our case (multi-label learning), this point is very important, because it will be interesting to see whether CNNs can learn shared representations and can discover correlations among different action classes.

To summarize, we make the following contributions in this work:

1) We implement, test and compare two deep approaches for multi-label activity learning having class imbalance problem in hockey videos. The two approaches are: (i) An ensemble of k binary networks, and (ii) A single multi-label k-output network. Also, we compare the results with a state-of-the-art existing solution [26].
2) After implementing two different approaches along with their own strategies for handling the class imbalance problem, and performing a series of experiments to properly evaluate each approach, in the end, we we are providing a lightweight 3D CNN architecture for activity recognition in hockey videos. By light-weightiness, we mean that our architecture has less than 1M learnable parameters, which gives about 3.5 Mb sized models. It is important because such models could be integrated into hardware, like FPGA, which usually have less than 10 Mb of memory [27], or mobile platforms.
3) We make 3D CNN our baseline and provide F_1 measure scores for a publicity available dataset [26]. We do it for 11 activities, instead of just three activities as was done in the previous work [26]. It will be useful for any researcher who is working on the same or similar problems. We believe that there are lots of areas where this dataset, with provided F_1 baseline scores, will be helpful.

II. Related Work

We discuss the related work on action recognition in sports video as two separate sections.

A. Traditional Approaches

Most of the existing works on action recognition in sports video are based on traditional machine learning and computer vision methods. For example, [28] proposed a method for learning and recognizing activities in a volleyball game. The authors concentrated on single player activity recognition and got 77.8 % recognition accuracy. Their main idea was to build a context descriptors based on Histogram of Oriented Gradients (HOG), and Histogram of Optical Flow (HOF) features and employ Support Vector Machines (SVM) and Gaussian Mixture Models (GMM) as classifiers, for seven classes in six video in public datasets.

In [29], Perse et al. proposed a trajectory-based method for multi-player template action recognition of the basketball game. Their approach was based on trajectory templates matching using GMMs along with some game-specific knowledge. The system was evaluated using a private dataset, which consisted of up to 34 action-templates , and a recognition accuracy of up to 93% was achieved.

For action recognition of the hockey game, Carbonneau et al. [26] presented a solution for play-break detection using STIP[30] detectors and SVMs. Although their system achieved a good performance, up to 90 % recognition accuracy, their analysis was limited to only three activities. There are no baseline recognition scores for nine other action classes that are present in their dataset.

B. Deep Approaches

Nowadays deep learning has shown excellent performance, especially in visual tasks, such as object recognition ([18], image classification[31], and sports action recognition. For example, [32] presents a CNN and Long Short-term Memory (LSTM) based architecture for learning hierarchical group activities in volleyball video dataset, which was collected from YouTube. The key idea in their work was to use fine-tuned AlexNet features (fc7 layer) as input to a two-staged LSTM classifier for person and group activity recognition. The approach yielded a recognition accuracy of 63 - 86% for six activities.

In [33], Karpathy et al. presented a Sport-M1 dataset collected by Standford Vision Lab, and multi-resolution CNN

architecture that achieved 41.3 - 64.1% average accuracy. The dataset was about various sports, and consisted of 487 activities.

Recently, Kay et al. from Google DeepMind team presented the Kinetics [34] dataset. It is a large-scale publicly available Youtube-based dataset that includes various sets of human actives, approximately over 400 activities within 300,000 videos. In their work three deep baseline approaches were presented, including 3D CNN, 2-stream CNN (with RGB and optical flow inputs), and CNN+LSTM models with performance in the range of 56 - 79 % on the presented new dataset.

There are not many works about the application deep learning models for action recognition of the hockey game, especially for the case of multi-label action recognition problem. Furthermore, as mentioned above, [26] presented a video dataset of a hockey game, which consists of 11 activities; however, the paper reported recognition results of a traditional pipeline-based action recognition approach for only three activities. The dataset is well-annotated, and the source code of the proposed system is also available online. Therefore, the aim of our work is to implement a deep learning end-to-end system for multi-label action recognition of the hockey game having class imbalance problem, and evaluate it for all activities recorded in the said dataset and compare our results with the original method.

III. Methodology

Let us first define the multi-label learning problem in the context of action recognition in hockey videos. Let $D = \{(\mathbf{x}_i, \mathbf{y}_i)|1 \leq i \leq m\}$ be the multi-label training data. For the i-th multi-label instance $(\mathbf{x}_i, \mathbf{y}_i)$, $\mathbf{x}_i$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, ..., x_{id})$ of real values, and $\mathbf{y}_i$ is the associated k-dimensional label vector $(y_{i1}, y_{i2}, ..., y_{ik})$ of binary values for k possible classes (actions). For an unseen instance $\mathbf{x}$, the classifier $h(.)$ predicts $(y_1, y_2, ..., y_k)$ as a vector of labels for $\mathbf{x}$.

To do such kind of multi-label action recognition, we need to do two things: feature extraction and classification. A significant advantage of deep learning approaches over traditional methods is that deep learning methods work as end-to-end systems. In other words, such methods can perform both tasks at the same time.

One way to automatically extract features from video data is to apply CNNs, with typical convolution and pooling layers. However, In our work, we use 3D convolution and 3D pooling, which is a generalization of CNN operations, to perform feature extraction not only from a single image but also from a slice of frames, making it possible to learn dynamics between frames. Another way to learn time relations in some continuous data is by using recurrent neural networks, and especially LSTMs [35]. Such networks can memory about the data during propagations, and use this memory to find temporal patterns in the data. In this work, we also test the idea of connecting 3D CNN and LSTM networks in application to our task.

At the classification stage, it is common to use multi-layer perceptrons, which in deep learning community are called dense layer. We use the same dense layers, with batch normalization and drop-out blocks to avoid over-fitting.

As for how all of this is implemented as an end-to-end system, we implement and test two different strategies. First is an ensemble of k independent single-label learning networks. It is a simple general idea for multi-label learning, where we split k multi-label problem into k binary learning problems, training and evaluating k classifiers independently for each of the k classes. In literature, this method often is called binary relevance [11]. However, in this work we prefer to call it an ensemble of k networks, which has nothing to do with ensemble learning theory [36]. To get multi-label prediction we just concatenate individual predictions into one vector.

The second is a single multi-label k-output network. In this case, we train one neural network that can predict multi-label answer directly.

In both cases, we have a vector of real numbers as model output that can be interpreted as class probabilities. To make the final prediction, we calibrate the real-valued output on each label against a threshold, common value for which is 0.5 [11].

Now that we have explained how feature extraction and classification is done in our system and the two strategies that we implement, we provide a graphical illustration of how the whole system is implemented and evaluated in Fig. 1. Concrete details of each component are as follows.

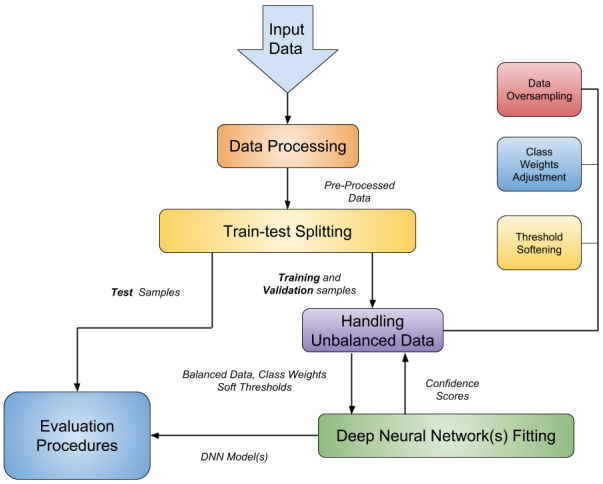

Fig. 1. Graphical illustration of the research methodology that we followed in this work.

A. Data Preprocessing

Preprocessing of the video data contains the following stages:

1) Resizing: Main reason for resizing is the hardware limitations, since even a small batch of video data may require a lot of memory for processing. We resize each frame of a video. We found empirically that resizing by four times is optimal for our case. That is, it is good in terms of learning time and it has no negative effect on the quality of predictions.

2) Data normalization: This is a necessary step in neural network training since without normalization the loss gradient values could go unproportionate and could negatively

affect the training process. In this work we perform Z-score normalization[37], also called standardization.

3) Windowing: This means splitting the data, both the instances and their associated labels, into fixed-size units/sequences. We use overlapping sliding window protocol with a window size of 15, and an overlap of five frames. Doing so helps us in producing more samples.

4) Sequence Labeling: Each sequence of frames, produced in the previous step, must be associated with a single label vector. We apply the majority rule over each element of the associated 15 label vectors (for 15 frames) to produce the final label vector.

B. Training-test Splitting

After the data preprocessing, we divide the entire data into two parts: training and test datasets, using a 70:30 split. Using the same split, we further divide the training data into training and validation datasets, which are used for training the models and selecting the appropriate values of the hyper-parameters. The test dataset is used in the end to evaluate the learned models.

C. Handling of Unbalanced data

By this, we mean handling the class imbalance problem. In the case of the ensemble of k binary networks, we apply a simple technique called oversampling [38]. In this approach, when training each of the k networks, we achieve balance by randomly adding copies of instances of the under-represented class. However, oversampling is not an optimal method to use for solving the class imbalance problem in the case of single multi-label k-output network. The reason: an instance, in this case, may be associated with multiple labels, and randomly adding copies of such instances may affect the correlation among different labels. Therefore, for this case, we implement the following two-staged approach based on the concepts that are described in [10], [11].

At the first stage, we use a technique called class weight adjustment, where the weight of a class is determined as

$$w_i = log(\mu \frac{m}{m_i}) \tag{1}$$

where w_i is the weight of the i-th class, m is the total number of instances in the training dataset, m_i is the number of instances that are associated with the i-th class, and μ is some constant in the range of $0 \cdots 1$. In our case, we set its value to 0.7, which is found empirically. It should also be noted that if (1) returns a weight that is less than one, its value is set back to one. Thus the minimum possible weight of a class is one.

At the second stage, we perform threshold softening for under-represented classes. By threshold, we mean the value against which the real-valued model output is going to be calibrated. To do this, we perform an initial training of the model using the calculated class weights, and a k-dimensional threshold vector whose elements are assigned to the default threshold $\alpha = 0.5$. After the initial training, the model is tested on the validation dataset to obtain the confidence scores (real-valued model output) for each class over all instances. The new threshold for the i-th class is then computed as

$$th_i = \alpha \frac{1}{w_i} c_i \tag{2}$$

where th_i is the new threshold, α is the default threshold, w_i is the class weight, and c_i is the maximum confidence score obtained for the class over all instances during the validation step.

To conclude, in the case of single multi-label k-output network, class imbalance problem is resolved by assigning higher class weights to under-represented classes and using softer thresholds for the same.

D. Network structure and Training Settings

The network structure that is used in this work is similar to other convolutional networks for computer vision tasks, such as VGG [13]. It is a well-known chain of convolutions, max-pooling, and rectified linear units activations [39]. To prevent over-fitting and making training a little bit faster we add drop-out blocks. The batch-norm layer has the same purpose. To decide the action category, we use two dense layers with sigmoid activation. The network structure is summarized in Fig. 2

It is important to mention that we found this structure using a series of incremental experiments on training and validation data for the *Play* class from the chosen dataset, which will be explained in the next section, in a one-against-all setting. During this search, we always balance between the performance and the number of parameters. Once found and validated for the *Play* class, we fix this structure as the basis for all other cases. It should also be noted that for training our models, we use binary cross-entropy as the loss function, which is calculated as

$$L = -t \log(p) - (1 - t) \log(1 - p) \tag{3}$$

where t is the target value, and p is the predicted value. For network optimization, we use Adam [40] optimizer.

IV. Experiments and Results

A. Dataset

The dataset that is used in this work was presented in [26]. The paper presented a two-staged hierarchical method, based on classical computer vision, for play-break detection in non-edited hockey videos. The dataset consists of 36 gray scale videos having a 480 * 270 pixels resolution captured at 30 frames per second. Alls videos were recorded using a static camera. Each video was named as $P - C - N - gray.avi$, where P is the period number ($1 \cdots 3$), C is the camera (left or right), and N is the sequence number for the video. Fig. 3 presents frames from two point of views.

There are 12 types of events in this dataset. Full list of events include: *Celebration, Checking, Corner Action, End of Period, Face-Off, Fight, Goal, Line Change, Penalty, Shot, Save*, and *Play*. Detailed explanation can be found in the original paper [26]. Every frame of a video is labeled with a binary string. For example, a frame having a label of 00000000101 means that this frame is associated with classes *Shot* and *Play*.

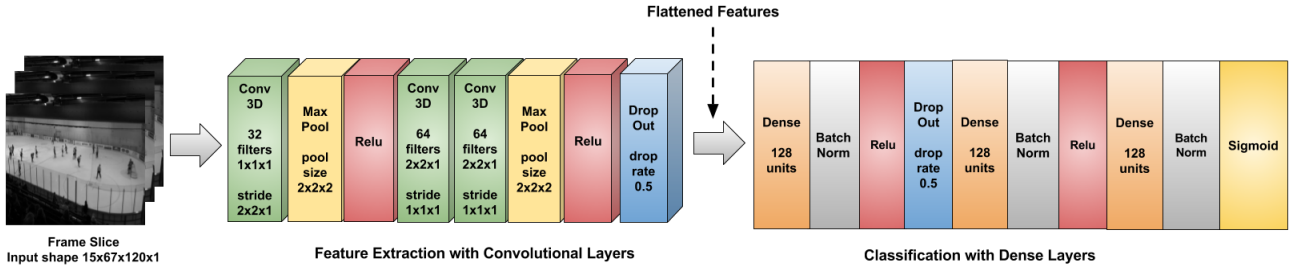

Fig. 2. General Structure of the network.

Fig. 3. Left View and Right View

B. Implementation Details

The entire work is implemented in Python3 using the Keras library, which is a deep learning library for Theano [41] and Tensorflow [42]. It is designed to enable fast experimentation with deep networks. We also employ Sklearn [43] for system performance evaluation and OpenCV [44] library for image and video processing.

C. Metrics

We choose F_1 score as the evaluation metric for our experiments, since F_1 score is one of the recommended metrics to be used in the case of unbalanced data [11]. We can define F_1 score as:

$$F_1 = \frac{2PR}{P+R} \tag{4}$$

where $R = \frac{T_p}{T_p+F_n}$ is the recall, $P = \frac{T_p}{T_p+F_p}$ is the precession, T_p mean true positives, F_p means false positives, and F_n means false negatives.

D. Experiments

To evaluate our work, we perform a series of experiments, which is as follows.

1) Ensemble Model vs. Single Multi-label k-output Model: The purpose of this experiment is to understand which strategy works better. For this, we take the basic structure, which we have described previously, and apply it to both strategies. The results of this experiment for all activities are summarized in second and third columns of Table I, respectively.

2) LSTM units: The purpose of this experiment is to evaluate the idea of merging CNN and LSTM layers. The aim is to understand whether LSTM layers will help in learning temporal information in the data and improve the recognition accuracy. To perform this experiment, we add LSTM layers to both models from the previous experiment right after the flattened CNN feature layers and before the dense layers. The results for all activities are summarized in the last two columns of Table I, respectively.

3) Comparison with Original work: The purpose of this experiment is to compare our deep learning approach for action recognition of the hockey game with the original work on the chosen dataset [26]. It is important to mention that for this experiment we used our models without LSTM units, based on the performance that we got in the previous experiments. The results of this experiment for the three activities (as was done in [26]) are summarized in Table II.

4) Evaluating the Use of Data Normalization: The purpose of this experiment is to evaluate our claim data normalization is important for achieving high recognition accuracy. To perform this experiment, we use only the single multi-label k-output model, as it provided the best results in the previous experiments. We remove the data normalization component and repeat the same settings as in the first experiment. The results for all activities are summarized in column (B) of Table III.

5) Evaluating the Use of Class Weights Adjustment: The purpose of this experiment is to evaluate our claim that handling the data unbalancing problem is important for achieving high recognition accuracy. We repeat the same settings as in the fourth experiment, but this time we do the data preprocessing and remove the class weights adjustment part instead. The results for all activities are summarized in column (A) of Table III.

6) Evaluating the Use Threshold Softening: The purpose of this experiment is to evaluate our claim that threshold

TABLE I
F_1 SCORES FOR EVERY ACTION. ENSEMBLE MODEL (EM), SINGLE MULTI-LABEL k-OUTPUT MODEL (SMKO), ENSEMBLE MODEL WITH LSTM (EML), SINGLE MULTI-LABEL k-OUTPUT MODEL WITH LSTM (SMKOL)

Event	EM	SMKO	EML	SMKOL
Celebration	0.60	0.62	**0.63**	0.58
Checking	0.20	**0.38**	0.19	0.28
End of period	0.90	**0.98**	0.90	0.95
Fight	0.74	**0.85**	0.60	0.83
Goal	0.38	0.38	0.36	**0.62**
Penalty	0.54	**0.79**	0.70	0.75
Shot	**0.46**	0.30	0.16	0.18
Save	**0.60**	0.46	0.36	0.21
Line change	0.68	**0.78**	0.67	0.73
Face off	0.78	**0.86**	0.81	0.80
Play	0.93	**0.95**	0.92	0.93
Average F_1	0.62	**0.67**	0.57	0.62

TABLE II
COMPARISON WITH ORIGINAL WORK [26] IN TERMS OF F_1 SCORE. ENSEMBLE MODEL (EM), SINGLE MULTI-LABEL k-OUTPUT MODEL (SMKO)

Event	Original paper	EM	SMKO
Line change	0.52	0.68	**0.78**
Face-off	0.36	0.78	**0.86**
Play	0.86	0.93	**0.95**

softening is important for achieving high recognition accuracy. We repeat the experiment, keeping everything except the threshold softening part of the system and use a constant threshold, instead. The results are summarized in column (C) of Table III.

7) Evaluating the Use of All: The purpose of this experiment is to study how the system would perform if we removed all of the above components. We repeat the experiment, but we do not perform any data normalization, weights adjustment as well as threshold softening. The results are summarized in column (D) of Table III.

V. DISCUSSION

Our network can process six minutes of video from the dataset in approximately 15 seconds during the forward-propagation on Tesla K40 level GPU. Therefore, it is possible to adapt our solution to real-time settings, e.g., for streaming videos. Fig. 4 shows visual real-time performance of our system in four situations: *end of a period, line change, face-off,* and *checking-play* (multi-label case).

Furthermore, based on the results of ensemble model versus the single model, we can see that the single model approach is better (F_1 score of 0.62 vs. 0.67 on average, respectively). Another natural advantage of using the single model is that it can be trained k-times faster since it has the same number of weights as one model in the ensemble of k-networks (980,000).

If we take a look at the results of the experiments with LSTM units, we can see that adding LSTM units decreases the performance. It is important to understand why did it happen. One reason can be that we are using a short sequences of frames, and LSTM units cannot fit the data in such a setting. Further experiments with sequences of different lengths is needed to confirm this point. However, in the case of *Goal*, the single model with LSMT units gave the best results (F_1: 0.6) than all other models. We think that the main reason for this is that the *Goal* is the quickest/shortest action among others and in this case the chosen sequence length is enough for LSTM units to learn temporal information.

If we take a look at the results of the experiments that evaluate the importance of data normalization, class weights adjustment and threshold softening, we may see that when applied all together these steps increase the average F_1 performance significantly compared to the case where we use none of them (0.67 vs. 0.17). Furthermore, solving the class imbalance problem seems to have the most positive influence on the most under-represented classes, such as *Checking*; see its associated rows in Table I column SMKO and Table III columns (A) and (C). On the other hand pre-processing seems to be important for almost all classes, e.g., *Play*; see its associated row in Table III columns (A) and (B). As for the threshold calibration, it is also important for all cases. In some cases, such as *Celebration*, it can increase individual performance of an action by up-to 35%. However we need to pay further attention to improving these procedures, such as implementing new algorithms for balancing in multi-label case or estimation of callibration thresholds in a different way, e.g., by learning or maximizing some criteria.

According to the obtained results, presented in Table I, it can be seen that the system performed well for actions like *Play*, *Face off*, and *End of Period*, with scores of 0.78 - 0.95, However, for other actions, such as *Shot*, *Save*, and *Celebration*, the performance is not in the same range, having scores of 0.38 - 0.62.

We think that we got good results for *Play*, *Face off*, and *End of Period* because we can consider these actions as group actions - interaction among players. Thus we may conclude that for group situations, our system is capable of learning adequate features to achieve an optimum recognition accuracy. Although *Celebration* can also be classified as a group action; it has its own specific challenges. For humans, primary signal of celebration is raising of the hockey sticks by players. However, the same phenomena may confuse the model. Maybe we can get better performance by increasing the frame resolution, but this would require more computational and memory resources.

In case of *End of Period* we have good result because the

TABLE III
F_1 SCORES FOR EVERY ACTION CATEGORY FOR SINGLE MULTI-LABEL k-OUTPUT MODEL (F_1 SCORE OF 0.67) AFTER REMOVING: (A) WEIGHTS ADJUSTMENT, (B) DATA NORMALIZATION, (C) THRESHOLD SOFTENING, (D) ALL OF THE PREVIOUS STEPS

Event	(A)	(B)	(C)	(D)
Celebration	0.0	0.35	0.26	0.0
Checking	0.1	0.16	0.09	0.0
End of period	0.83	0.87	0.93	0.66
Fight	0.48	0.43	0.83	0.0
Goal	0.003	0.22	0.0	0.0
Penalty	0.6	0.33	0.63	0.0
Shot	0.12	0.07	0.03	0.0
Save	0.23	0.16	0.14	0.0
Line change	0.64	0.51	0.74	0.07
Face off	0.73	0.62	0.79	0.27
Play	0.91	0.85	0.95	0.81
Average F_1	0.42	0.42	0.49	0.17

main characteristics in this case is moving spectators, which in other cases did not happen, see Fig. 4 as an example.

As for the *Save*, *Shot*, *Goal* and *Checking*, one probable reason for low recognition accuracy could be the fact that these events are related to the movements of specific players. Since there is no player level tracking and segmentation being done, this may result in not learning adequate deep representations for these classes. Also, for *Shot* and *Save*, the ensemble model has shown better performance than the single model, but for other events, we have opposite results. We believe that it is because these are rare events, and in the case of the single model, it is difficult to balance these events, even with the presented techniques.

Therefore, there are many directions for the future works. To improve the described issues, we need to add to our pipeline tracker and detection stages, which would enable the system to localize the players and help in improving recognition rates of actions that are related to specific players. Furthermore, it would be reasonable to make evaluations on other hockey video datasets, which are labeled in a similar way to the original dataset. However, it is difficult to find such datasets that are also publicly available. Therefore, we plan to collect our own dataset in future.

Another interesting question would be to see how our architecture will learn representations on non-gray-scale videos. In this case, it would also be important to understand how to ensure real-time properties of the system within RGB videos, since it can significantly increase the time for learning and prediction.

Finally, another very important question is to understand how different deep architectures, especially non-VGG like, will work for the chosen problem. It may be reasonable to try to use different combinations of recurrent layers or to try to use multi-input networks to provide more information and learn better representations. When testing different architectures, it would be useful to employ and evaluate various loss functions. For example, we can consider the ranking loss methods [11] for multi-label learning. Nevertheless, the results obtained in this work show the feasibility of employing the proposed system for multi-label action recognition of the hockey game in the presence of class imbalance problem. For the chosen dataset, the system is not only better than the traditional machine learning-based recognition methods, but with some additional support it can also be used for real-time action recognition in a live game.

VI. CONCLUSION

In this paper, we present deep learning based solution for hockey game action recognition in multi-label learning settings having class imabalance problem. The proposed system achieved good performance for several action categories, and it can be adapted for real-time use, although this might require the use of a specific hardware. As a part of our contributions, we present baseline F_1 scores for all action categories in a publicly available hockey videos dataset. Our results are better than the existing solution, and it can be a starting point for further research using this dataset.

REFERENCES

[1] D. Gernimo and H. Kjellstrm, "Unsupervised Surveillance Video Retrieval Based on Human Action and Appearance," in *Proceedings of the 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 4630–4635.

[2] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ser. ICRA'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 3644–3649. [Online]. Available: http://dl.acm.org/citation.cfm?id=1703775.1704035

[3] A. M. Khan, Y. Lee, S. Lee, and T. Kim, "Accelerometer's position independent physical activity recognition system for long-term activity monitoring in the elderly," *Med. Biol. Engineering and Computing*, vol. 48, no. 12, pp. 1271–1279, 2010. [Online]. Available: https://doi.org/10.1007/s11517-010-0701-3

[4] C. Huang, H. Shih, and C. Chao, "Semantic analysis of soccer video using dynamic bayesian network," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 749–760, 2006. [Online]. Available: https://doi.org/10.1109/TMM.2006.876289

[5] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[6] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, Apr. 2011. [Online]. Available: http://doi.acm.org/10.1145/1922649.1922653

[7] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 33:1–33:33, Jan. 2014. [Online]. Available: http://doi.acm.org/10.1145/2499621

[8] L. Pang, J. Cao, J. Guo, S. Lin, and Y. Song, "Bag of spatio-temporal synonym sets for human action recognition," in *Proceedings of the 16th International Conference on Advances in Multimedia Modeling*, ser. MMM'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 422–432. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-11301-7_43

Fig. 4. Real-time Demo: Green labels are the ground truth, Red labels are the predictions

[9] Y. L. Murphey, H. Guo, and L. A. Feldkamp, "Neural learning from unbalanced data," *Applied Intelligence*, vol. 21, no. 2, pp. 117–128, Sept. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:APIN.0000033632.42843.17

[10] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. on Knowl. and Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sept. 2009. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2008.239

[11] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, 2014. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2013.39

[12] M. S. Sorower, "A literature survey on algorithms for multi-label learning," Oregon State University, Tech. Rep., 2010.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. [Online]. Available: http://arxiv.org/abs/1512.02325

[17] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[18] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[19] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," *CoRR*, vol. abs/1704.06036, 2017. [Online]. Available: http://arxiv.org/abs/1704.06036

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *CoRR*, vol. abs/1511.00561, 2015. [Online]. Available: http://arxiv.org/abs/1511.00561

[22] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *CoRR*, vol. abs/1508.06576, 2015. [Online]. Available: http://arxiv.org/abs/1508.06576

[23] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep learning," 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org

[25] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *CoRR*, vol. abs/1411.1792, 2014. [Online]. Available: http://arxiv.org/abs/1411.1792

[26] M. Carbonneau, A. J. Raymond, E. Granger, and G. Gagnon, "Real-time visual play-break detection in sport events using a context descriptor," in *2015 IEEE International Symposium on Circuits and Systems, ISCAS 2015, Lisbon, Portugal, May 24-27, 2015*, 2015, pp. 2808–2811. [Online]. Available: http://dx.doi.org/10.1109/ISCAS.2015.7169270

[27] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[28] G. Waltner, T. Mauthner, and H. Bischof, "Indoor activity detection and recognition for sport games analysis," *CoRR*, vol. abs/1404.6413, 2014. [Online]. Available: http://arxiv.org/abs/1404.6413

[29] M. Perse, M. Kristan, J. Pers, and S. Kovacic, "A template-based multi-player action recognition of the basketball game," in *Proceedings of the ECCV Workshop on Computer Vision Based Analysis in Sport Environments*, 2006, pp. 71–82.

[30] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2009, pp. 124.1–124.11, doi:10.5244/C.23.124.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[32] M. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," *CoRR*, vol. abs/1511.06040, 2015. [Online]. Available: http://arxiv.org/abs/1511.06040

[33] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 1725–1732. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.223

[34] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *CoRR*, vol. abs/1705.06950, 2017. [Online]. Available: http://arxiv.org/abs/1705.06950

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[36] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, no. 1-2, pp. 1–39, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1007/s10462-009-9124-7

[37] E. Kreyszig, H. Kreyszig, and E. J. Norminton, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ: Wiley, 2011.

[38] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *The Data Mining and Knowledge Discovery Handbook.*, 2005, pp. 853–867.

[39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, J. Frnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814. [Online]. Available: http://www.icml2010.org/papers/432.pdf

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[41] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[42] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[44] G. Bradski, "Opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

Konstanin Sozykin received the Engineering Diplom (Master degree equivalent) in automatic control systems from the Kazan National Research Technological University, Russia, in 2015, the short-track Bachelor degree in computer science and engineering from Innopolis University, Russia, in 2017. During 2014 and 2015 he was a research assistant in Heat and Mass Transfer Lab, KNRTU, where he developed algorithms for distillation process control. During 2015 and 2017 he was a research intern at Machine Learning and Knowledge Representation lab, Innopolis University, where he worked on face-detection and activity recognition problems. His current research interests include but not limited to computer vision, image and speech processing and deep learning with the applications on video and biomedical image analysis.

Dr. Adil Khan is an Associate Professor in the Department of Computer Science at Innopolis University, Russia. He is also the head of Machine Learning and Knowledge Representation Lab at Innopolis. He has received his Ph.D. in Computer Engineering from Kyung Hee University, South Korea. He has over 11 years of experience in academic research and teaching. His primary research interests include machine learning, computer vision, data analytics, wearable computing, and context-aware computing. His research work, comprising over 50 articles, is published in various international conferences and journals. He is currently participating in several international research projects. He is an IEEE member and is also a reviewer for numerous IEEE, ACM, Elsevier and other international journals.

Stanislav Protasov was born in Russia in 1987. He received his Candidate of Science (Mathematics and Physics) degree in Voronezh State University in 2013. His thesis was devoted to computer sterevision. Since 2015 he works as a postdoctoral researcher in Machine Learning and Knowledge Representation lab in Innopolis University.

Rasheed Hussain received his B.S. in Computer Software Engineering from University of Engineering and Technology, Peshawar, Pakistan in 2007 and MS and PhD degrees in Computer Engineering from Hanyang University, South Korea in 2010 and 2015, respectively. He also worked as a Postdoctoral Research Fellow at Hanyang University, South Korea from March 2015 to August 2015. Furthermore, he worked as a Guest Researcher in the Department of Informatics at University of Amsterdam (UvA), Netherlands from August 2015 to May 2016. He is currently working as Assistant Professor in the Institute of Information Sciences at Innopolis University, Kazan, Russia. His main research interests include information security and privacy, applied cryptography, Vehicular Ad Hoc NETworks (VANET), Vehicular Social Networks (VSN), VANET applications and services, cloud computing, smart grid security, location-based services, VANET-based clouds, Big data, and Internet of Things (IoT). He is currently working on emergent VANET-based clouds, VSN security and services, and Named Data Networking (NDN).