Temporal Hockey Action Recognition via Pose and Optical Flows

Zixi Cai[†] Helmut Neher Kanav Vats David A. Clausi John Zelek

[†]Tsinghua University University of Waterloo

caizx15@mails.tsinghua.edu.cn {hneher, k2vats, dclausi, jzelek}@uwaterloo.ca

Abstract

In this paper, a novel two-stream architecture has been designed to improve action recognition accuracy for hockey using three main components. First, pose is estimated via the Part Affinity Fields model to extract meaningful cues from the player. Second, optical flow (using LiteFlownet) is used to extract temporal features. Third, pose and optical flow streams are fused and passed to fully-connected layers to estimate the hockey players action. A novel publicly available dataset named HARPET (Hockey Action Recognition Pose Estimation, Temporal) was created, composed of sequences of annotated actions and pose of hockey players including their hockey sticks as an extension of human body pose. Three contributions are recognized. (1) The novel two-stream architecture achieves 85% action recognition accuracy, with the inclusion of optical flows increasing accuracy by about 10%. Thus, demonstrating the complementary nature of pose estimation and optical flow. (2) The unique localization of hand-held objects (e.g., hockey sticks) as part of pose increases accuracy by about 13%. (3) For pose estimation, a bigger and more general dataset, MSCOCO, is successfully used for transfer learning to a smaller and more specific dataset, HARPET, achieving a PCKh of 87%.

1. Introduction

Vision-based human action recognition has gained increasing attention in the past few years because of broad applications in smart surveillance systems, smart elderly assistance, human-computer interaction, and sports as examples. Many challenges, such as lack of data, small human size due to camera position, and motion blur from high speed human actions exist in many applications. Other challenges known to sports are noisy data from bulky clothing and equipment and similarities between foreground and background. One application that emulates all aforementioned challenges is ice hockey.

This paper focuses on incorporating pose information and optical flow for action recognition in a unified two-stream architecture (shown in Fig. 1) to provide high-level features unique to pose estimation and optical flow to depict motion, thus, improving the overall accuracy of action recognition. It also demonstrates the complementary nature of pose estimation and optical flow in improving action recognition accuracy. The two-stream architecture analyzes pose and temporal features via a convolutional neural network (CNN), then the outputs of the two streams are concatenated via fully-connected layers.

Although many works explore action recognition in videos on large benchmark datasets such as UCF101 and HMDB, few focus on sport videos [3, 22, 23, 28]. To date, there are no publicly available temporal action recognition datasets in hockey considering individual players; one dataset explores multiple players temporally [40], while, another dataset only considers still images with no temporal information considered [11]. To solve this problem, a novel publicly available dataset, known as HARPET (Hockey Action Recognition Pose Estimation Temporal), comprised of hockey image sequences (three images per sequence) captured by a single RGB camera are used, with annotations including pose (comprised of 18 joints including the hockey stick) and actions, is generated. Four types of actions are considered: skating forward, skating backward, passing and shooting. The dataset contains around 100 sequences per class, with around 1200 images in total.

Testing on HARPET dataset, the two-stream architecture obtains around 85% end-to-end accuracy. For pose estimation model, due to small number of examples in dataset, transfer learning is leveraged to reduce overfitting and is demonstrated to be effective with around 87% PCKh@0.5 [1]. It is also demonstrated that localization of hand-held objects can improve the accuracy of sports action recognition, which to the best of our knowledge, has not been explored in previous works.

The rest of the paper is organized as follows. In Section 2, we view papers on action recognition, highlighting

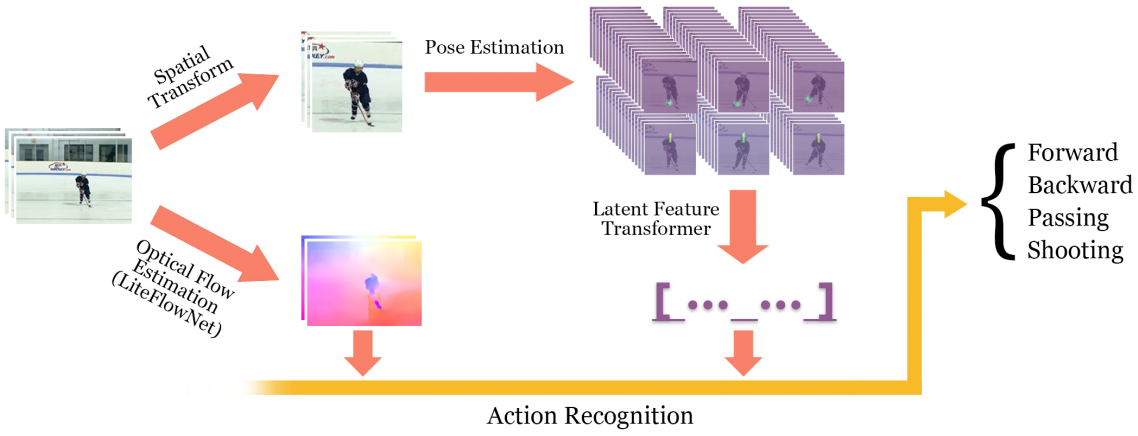


Figure 1: Overall pipeline. Our method takes sequence of 3 images as input. Part confidence maps and part affinity fields are predicted for each spatially transformed image, and converted into a latent joint feature vector. Optical flows are generated in the second stream. Optical flows and the latent feature vector are used as the input of the action recognition component, which predicts probabilities of skating forward, skating backward, passing, and shooting.

two-stream-based and pose-based frameworks, and discuss works on hockey action recognition. The architecture, comprising of pose estimation and action recognition models, is illustrated and implementation details are explained in Section 3. We evaluate accuracy of both pose estimation and end-to-end action recognition on HARPET in Section 4.

2. Background

Action recognition is a widely researched topic which, before the advent of deep networks, employed hand-crafted features, dense trajectories [48] and improved dense trajectories [50]. State-of-the-art action recognition models incorporate these features in action recognition [8, 9, 19, 20, 49]. Recently, deep networks have shown promising action recognition accuracy through the use of 3D convolutions in demonstrating better capability of capturing spatiotemporal latent structure in videos than 2D convolutions [7, 21, 42, 44–46]. The major downside of 3D CNNs is the large number of parameters, making it easy to overfit on small datasets which is common in many practical applications. Also, the use of recurrent neural networks, which are manifested to be adept at modeling sequential data are explored [3, 10, 34]. Li *et al*. [25] introduce spatio-temporal attention networks for action recognition, enabling identification of the key video frames and spatial focus in frames for recognizing actions. To summarize, mainstream methods improve action recognition using several overlapping categories including: hand-crafted features [41,48,50], two-stream neural networks [7, 12, 38, 52, 55], 3D convolutional networks [7,21,42,44–46], recurrent neural networks [3,10, 34], and pose-based methods [2, 9, 11, 14, 32, 54].

Besides the techniques mentioned above, pose features are widely used in works on action recognition. Pose estimation and action recognition are two problems that leverage information from each other. Yao *et al*. [2,54] claim that pose-level features are useful for action recognition and introduce an architecture for coupled 3D pose estimation and action recognition. Gall *et al*. [14] also use action recognition for 3D pose estimation. Luvizon *et al*. [32] use a multi-task framework for joint action recognition and 2D/3D pose estimation. Wang *et al*. [47] develop action representations based on 2D human poses. Fani *et al*. [11] use 2D pose from stacked hourglass network to infer action from still images. Iqbal *et al*. [18] introduce a framework to help estimate pose with action priors and then improve action priors with updated pose information and hence, oscillate between pose estimation and action recognition. Nie *et al*. [35] combine action recognition and video pose estimation in a unified framework with a spatial-temporal And-Or Graph model. Chéron *et al*. [8] use a pose-based CNN as a descriptor for action recognition.

Pose is a high-level spatial feature, while optical flows represent temporal information. Two-stream networks, [7, 12, 15, 16, 53, 55] is one of the prominent category of the state-of-the-art approaches in recent years, first proposed in [38]. A spatial stream analyzes a single video frame and a temporal stream uses multi-frame optical flow, both via a series of convolutions and fully-connected layers. Classification scores predicted by the two streams with softmax are fused via averaging or linear SVM. One of the advantage of

separate streams is that they can be trained independently, and thereby spatial stream can be pre-trained on large still image classification datasets (such as ImageNet).

Plenty of variants of two-stream networks exist. Wang *et al*. [52] model long-range temporal structure by uniformly segmenting videos and selecting a snippet from each segment. Two-stream networks are applied to each snippet and results are fused. Zhu *et al*. [55] learn to estimate optical flow with an unsupervised architecture. It is done by minimizing the difference between the first frame and the frame reconstructed from the second frame by inverse warping according to predicted optical flow. Carreira *et al*. [7], use 3D convolutions in a two-stream architecture by pre-training original 2D filters on ImageNet and inflating them into 3D by repeating weights.

Our work is similar to these two-stream-based methods in the sense that we extract pose information and temporal information (optical flow parsed by CNN) in two separate streams before combining them. The pose stream is trained using transfer learning using pre-trained weights of the Part Affinity Fields model [5] trained on MSCOCO dataset.

In the context of hockey, tracking is a major research focus [4,24,26,33,36,37]. Most of the works on action recognition in hockey look at the game with a wider perspective, keeping event detection as the main focus [6, 40, 43]. Action recognition, paying attention to individual players, is explored in very few works [11, 29–31]. In these works, hand-crafted HOG features are first computed for tracking multiple individuals, and then a probabilistic framework is devised to model the action. They do not leverage high-level unique-to-human feature such as pose. In Fani *et al*. [11], pose is considered but temporal information is neglected. Analyzing spatial and temporal information in two streams of CNNs is a powerful technique in understanding spatiotemporal structure, and pose features can provide valuable information for analyzing actions. In our work, two-stream-based architecture, combining pose and optical flow, applied to hockey action recognition concentrated on individual player, is presented.

3. Methodology

3.1. Overview

The overall network architecture, as shown in Fig. 1, illustrates the proposed approach of implementing a two-stream network incorporating pose estimation, via the model using part affinity fields (PAFs) [5], and optical flow estimation, via LiteFlowNet [17]. The network takes a sequence of three images as an input, which is then used in the first stream by spatially transforming and cropping the image to a pixel size of 368×368, centering the person and applying the pose estimation model, which is described in Section 3.2. Afterwards, the pose features are then concatenated in a latent feature vector layer (Section 3.3). The second stream then applies optical flow estimation to extract features at a macroscopic level (Section 3.4). From both the streams, the action of the given sequence is then classified and the output of the network determines whether a hockey player is skating forward, skating backwards, passing or shooting. The training details are illustrated in Section 3.5.

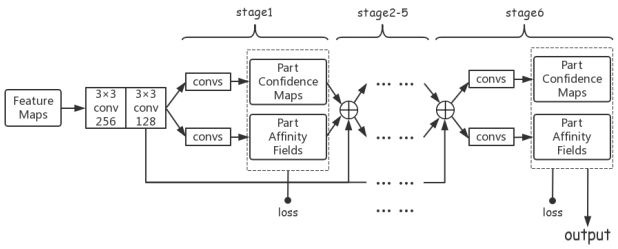


Figure 2: Multi-stage pose estimation architecture. Each stage predicts part confidence maps and part affinity fields through a series of convolutions. Prediction is iteratively refined and loss is computed at the end of each stage.

3.2. Pose Estimation

Cao *et al*. [5] propose a novel feature representation called part affinity fields, which evaluates association between two joints. In PAFs the, 2D vector at each pixel indicates position and orientation for a certain limb [5]. Fig. 2 shows the network generating part confidence maps and PAFs.

The feature maps extracted by VGG-19 [39], after two 3×3 convolutions, are passed through six stages. Each stage is split into two branches predicting part confidence maps and part affinity fields via a series of convolutions. Then part confidence maps and part affinity fields as well as the aforementioned feature maps (passed through two convolutions) are concatenated together and taken as input by the next stage. Stage 1 has five convolutions, where the first three employ a kernel size of 3×3 and the last two employ a kernel size of 1×1. Stage 2-6 each has seven convolutions, where the first three employ a kernel size of 7×7 and the last two employ a kernel size of 1×1. Strides of all convolutions are 1, and paddings are all set to keep the size of the feature maps same. The prediction is refined iteratively, and loss is calculated for maps and fields output by every stage.

3.3. Latent Feature Transformer

Fig. 3 briefly shows the pipeline for transforming part confidence maps and part affinity fields to a latent joint feature vector. To obtain the full pose of a single person, an existing algorithm [5] is modified, which first obtains limb connection candidates and then assembles them into pose of

multiple persons. For each joint, we reserve two peaks with the highest score in corresponding part confidence map, instead of filtering candidates with threshold. This ensures no joint will be lost. The joint with the highest score is not selected because the best location cannot be determined merely according to part confidence maps because the network sometimes makes mistakes, and that we want to leverage information provided by PAFs.

Then, a single candidate is selected for each joint. We start from the candidate of head top with the higher value, and expand it into full pose by iteratively selecting joint candidates which are most probable to associate with determined joints. Head top, being a relatively easier joint to detect as compared to limbs, is set to be the starting point since the network is less likely to make mistakes on it. The score of association between joint candidates is determined by calculating the line integral over the corresponding PAF along the limb, formally shown by Eq. (10) and (11) in Cao *et al*. [5]. Other joints that are easy to predict, such as the pelvis, were tried as the starting point, however, the accuracy is nearly the same.

After locations of all joints in three images are obtained, the procedure mentioned in Fani *et al*. [11] is applied to each one of them. In Fani *et al*. [11], joints identified in all images are scaled by the average head segment length (distance between head top and upper neck) in all training images. We normalize joints of each image with the head segment length in order to eliminate the impact of discrepancy in human's size between different images. Angles between certain limbs are also calculated (Table 1). Scaled joint locations and computed angles are concatenated to form a feature vector for each image. We concatenate vectors for three images into a one dimensional feature vector of size 156 which is fed to an action recognition component.

3.4. Action Recognition Component

LiteFlowNet (Hui *et al*. [17]) is a state-of-the-art network for optical flow estimation. In their work, pyramidal features are received by cascaded flow inference and flow regularization modules, which iteratively increase resolution of flow fields. Pre-trained LiteFlowNet is used in our pipeline. Since LiteFlowNet takes two images as input, two optical flows are generated from three images.

The action recognition component leverages information provided by joint locations and optical flows. The architecture is illustrated in Fig. 4. The optical flow fields obtained are concatenated into a 4-channel map and resized to 56×56 pixels. Then, the map is passed through several convolutional and max-pooling layers followed by two fully-connected layers and converted into a flat feature vector. Relu activation is used for all convolutional and fully-connected layers in this part. The feature vector generated from optical flows is concatenated with the latent joint feature vector. The flow feature vector concatenated with latent joint feature vector is passed through four fully-connected layers, the first three of them with sigmoid activation and

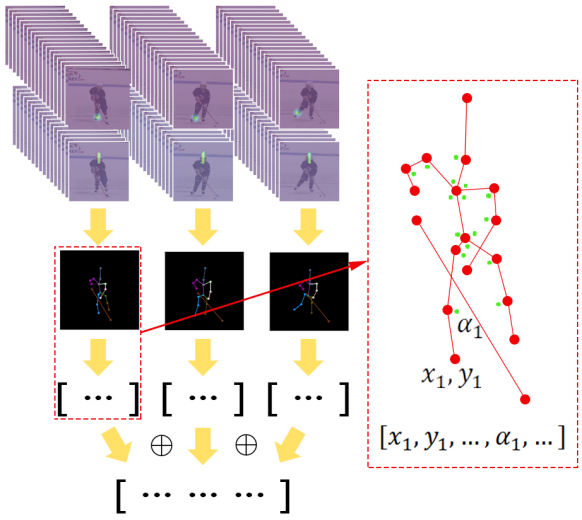


Figure 3: Latent feature transformer. Pose is obtained from part confidence maps and part affinity fields for each image, and transformed into a flat latent joint feature vector. Dashed box on the right shows details of this transformation. The vector contains coordinates of all joints and angles between some limbs (green dots indicate angles between limbs that are to be calculated, which are shown more clearly in Table. 1). Finally, latent feature vectors of 3 images are concatenated.

head top	upper neck	thorax
upper neck	thorax	left shoulder
pelvis	thorax	left shoulder
thorax	left shoulder	left elbow
left shoulder	left elbow	left wrist
upper neck	thorax	right shoulder
pelvis	thorax	right shoulder
thorax	right shoulder	right elbow
right shoulder	right elbow	right wrist
thorax	pelvis	left hip
pelvis	left hip	left knee
left hip	left knee	left ankle
thorax	pelvis	right hip
pelvis	right hip	right knee
right hip	right knee	right ankle
left hip	pelvis	right hip

Table 1: Angles calculated in latent feature transformer. Each row in the table indicates an angle. A row whose items are A, B, C from left to right represents $\angle ABC$.

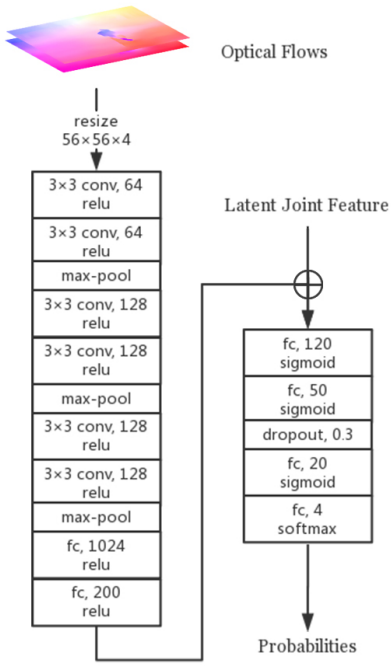


Figure 4: Action recognition architecture. Optical flows are resized and passed through interwoven convolutions and max-pooling, followed by fully-connected layers, and converted into a vector. It is concatenated with latent joint feature vector. Fully-connected network predicts probabilities of each class from the vector.

the last with softmax to output probabilities of four classes. A dropout layer is added after the second fully-connected layer (50 units) to reduce overfitting.

3.5. Training Details

Since our dataset is small, various methods are employed to reduce overfitting such as dropout, dataset augmentation and early stopping.

As a basic method of transfer learning, pose estimation network is fine-tuned based on weights pre-trained on MSCOCO dataset [27]. The dataset contains over 100K person instances and covers various real-world scenarios, which can provide relevant knowledge for transfer learning. In order to avoid overfitting, weights of all layers, except last 3 layers of the stage 5 and all layers of the stage 6 are frozen. However, joints we want to learn here are different from that in the MSCOCO dataset. In order to perform transfer learning, only the last two stages of the pose estimation network are trained such that, the loss of the rest of the stages is not computed and the last two stages output 18 new joints.

A variety of data augmentation is performed in training to make our dataset appear more diverse. For the pose estimation network, original images are randomly flipped, scaled, rotated, similar to Cao *et al*. [5]. For the action recognition network, in addition to the methods applied to pose estimation network, we perturb the location of each joint. Note that whenever flipping is applied to joints while training action recognition component, it is also applied to optical flows, because direction of background movement and orientation of person, which is represented by pose, are tied together when telling the direction of person's movement.

The pose estimation network and the action recognition network are trained separately. Validation loss of the pose estimation network decreases with training loss, so the network is trained until convergence and the last model checkpoint is selected. However, the action recognition part starts to overfit after 30 epochs. So an early stopping technique of training 30 epochs and picking up the checkpoint with highest validation accuracy was adopted. We select three models with highest validation accuracy at test time, which will be explained in Section 4.3.

In addition, we found that the action recognition network is difficult to train if the input is the prediction of the pose estimation network, so we instead train the pose estimation network with augmented ground truth of joint locations and validate it with the prediction. The network generalizes well to the case where joints are not precisely localized. This is because augmentation applied to joints eliminates the impact of possible discrepancy between distribution of ground-truth and predicted joint locations, which makes the network able to tolerate joint errors.

The training hyperparameter configurations are as follows. Weights are learned using mini-batch stochastic gradient descent with batch size set to 2 and momentum set to 0.9 for both two sub-networks. For the pose estimation network, L2 regularization is added to the convolution kernel weights with a regularization coefficient of 5×10^{-4}. In every epoch, all training images are fed once, so the number of iterations per epoch is $\frac{N}{2}$ (N training images). Training lasts 300 epochs. The learning rate is initially set to 10^{-2} and changed to 10^{-3} after 200 epochs. For the action recognition network, learning rate is 10^{-2} throughout the 30-epoch training. The dropout ratio is set to 0.3. The training process takes about 14 hours for the pose estimation network and about 13 minutes for the action recognition network, on a TITAN X GPU.

4. Testing and Results

4.1. Dataset Preparation

The model is trained and tested on the HARPET dataset which is composed of sequences of 3 images with time interval of $\frac{1}{6}$ seconds between any two successive frames in

30 frames per second video. Sequence length is set according to previous work on temporal modeling [13,51,52]. The sequences are collected from video clips scraped from the internet and from several instructional DVDs about hockey. Video segments were extracted from these videos and broken up into consecutive frames. Next, the sequences are classified into one of 4 classes: forward, backward, passing and shooting. Finally, 18 joints (16 human joints and 2 stick joints) are annotated in all images.

The dataset has 106 sequences for forward, 104 for backward, 113 for passing and 101 for shooting. There are a total of 1272 images of which joints are annotated respectively. The HARPET dataset is randomly split into three sets: 70% for training, 15% for validation and 15% for testing. The pose estimation component and the action recognition component are both trained on training set. The validation set is used to pick the best model.

4.2. Accuracy of Pose Estimation

To evaluate the pose estimation network, PCKh@0.5 [1] metric is used. According to the PCKh@0.5 metric, a joint is localized correctly if distance between prediction and ground truth is less than one-half of head segment length (distance between top of head and upper neck), and percentage of correctly-localized joints is computed. Results are illustrated in Table. 2.

The results demonstrates that the network trained on MSCOCO dataset can be transferred to the hockey domain with good accuracy (86.95% overall accuracy). Stick prediction has the worst precision (75.40%), which, several reasons account for poor precision of the hockey stick. (1) Joints, minus the hockey stick, are inferred from joints considered in the MSCOCO dataset which makes it easier for the network to transfer those joints, however, the stick is a new concept which takes more effort to learn. (2) The current model does not have a large enough receptive field to capture the whole stick that can be very long in images. (3) In many images, the stick is occluded or moves too quickly, adding difficulties to recognition. Prediction of elbows and wrists is also unsatisfactory, due to frequent occlusions.

Common failure cases are shown in Fig. 5. Left-and-right error and stick mislocalization due to occlusion and high-speed motion are typical.

4.3. Accuracy of Action Recognition

To show that the hand-held object is a strong cue for action recognition in hockey, coordinates of stick top (butt end) and stick end (stick blade) are purposely ignored by latent feature transformer (denoted by **-ST**) for a comparison with the original method which takes all joints including the stick into consideration (denoted by **+ST**). Besides, the temporal stream which analyzes optical flows is removed so that the action recognition network only looks at the latent joint feature vector (denoted by **-OF**), for a comparison with the original two-stream architecture (denoted by **+OF**). Hence, we have four combinations.

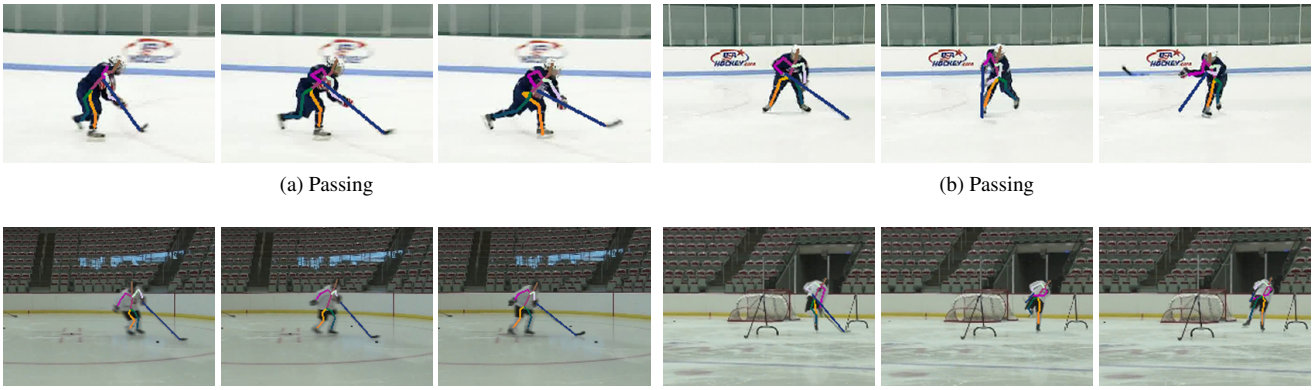
Parts	PCKh@0.5 (left/right, top/end)
Head	94.18%
Upper Neck	97.25%
Thorax	96.30%
Shoulder	85.19%/89.42%
Elbow	78.31%/80.95%
Wrist	76.72%/80.42%
Pelvis	**97.35%**
Hip	92.06%/87.83%
Knee	91.53%/91.00%
Ankle	89.42%/86.24%
Stick	71.96%/78.84%
Overall	86.95%

Table 2: Results of pose estimation. Values of left and right shoulder/elbow/wrist/hip/knee/ankle, as well as stick top and stick end, are averaged to shorten the table.

Figure 5: Common failure cases of pose estimation.

As mentioned above, since the validation accuracy does not steadily increase throughout training, selecting a checkpoint with the highest validation score is appropriate compared to simply picking up the last checkpoint. Moreover, due to lack of data, validation accuracy fluctuates drastically throughout training and there is an inevitable gap between validation and test accuracies. To evaluate the overall performance of each combination more appropriately, we test on 3 checkpoints of action recognition model with top 3 validation accuracy.

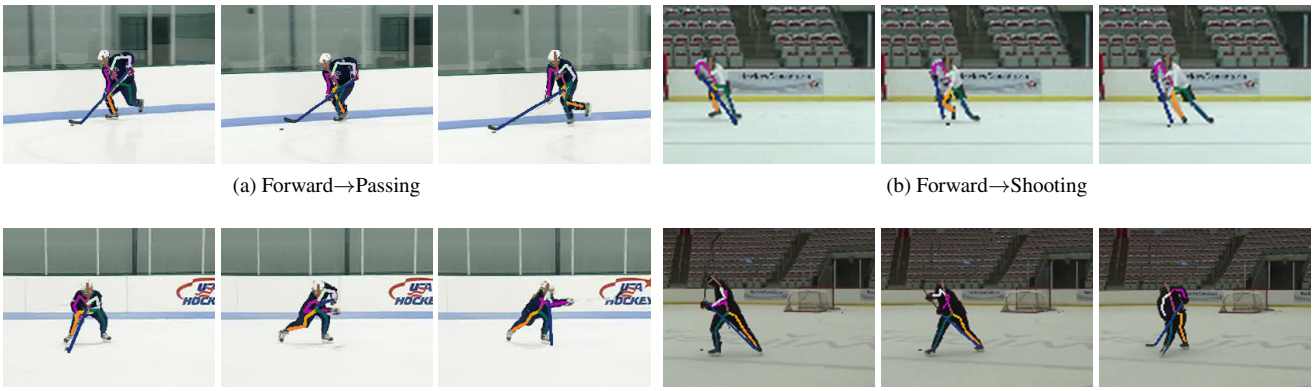
Table 3 shows the precision and recall for each



(a) Passing (b) Passing

(c) Backward (d) Shooting

Figure 6: Some examples of correct classification. Action recognition network can tolerate joint localization errors.



(a) Forward→Passing (b) Forward→Shooting

(c) Shooting→Forward (d) Shooting→Passing

Figure 7: Common failure cases of action recognition (ground truth→prediction).

class of each combination (**Fw.**=Forward, **Bw.**=Backward, **Ps.**=Passing, **St.**=Shooting for convenience). Here values of 3 checkpoints of every combination is averaged. Overall accuracy of each combination is illustrated in Table 4. In this table, the results of all selected checkpoints are shown ($\mathbf{1}^{st}$, $\mathbf{2}^{nd}$, $\mathbf{3}^{rd}$ refer to validation accuracy ranking) along with average values.

From the macro view, both stick information and optical flow, complementary to each other, help improve the overall accuracy as well as precision and recall rate for most classes. Exceptions lie in precision and recall rate of shooting and recall rate of skating forward. The stick plays a more important role than optical flow, as indicated by the combination of **+ST, -OF** outperforms **-ST, +OF**. When leveraging both stick and optical flow, end-to-end action recognition accuracy can be boosted to about 85%.

From the micro view, precision rate for passing is unsatisfying while the recall rate for passing is comparable to other classes, which means in many cases, other actions are mistaken for passing. This can also be seen from confusion matrices (Fig. 8). From comparison between **-ST, -OF** and **+ST, -OF** as well as **-ST, +OF** and **+ST, +OF**, stick increases accuracy of other 3 classes except shooting. This can be justified by observing that shooting is the only action among the 4 that is likely to lead to drastic change in pose so that joints are sufficient to recognize it. From comparison between **-ST, -OF** and **-ST, +OF** as well as **+ST, -OF** and **+ST, +OF**, it is demonstrated that optical flows improve results under most circumstances.

Some examples of correct classification are displayed in Fig. 6. In Fig. 6 (a) and (c), pose is correctly obtained thus leading to correct classification. In contrast, Fig. 6 (b) and (d), joints (sticks, more specifically) failed to be localized accurately but still produced correct results in the action

Methods	Precision			
	Fw.	Bw.	Ps.	St.
-ST, -OF	69.25%	81.82%	46.18%	82.63%
-ST, +OF	77.73%	82.05%	49.26%	77.22%
+ST, -OF	79.12%	88.08%	53.36%	80.65%
+ST, +OF	**96.06%**	**95.83%**	**56.67%**	**87.22%**
Methods	Recall			
	Fw.	Bw.	Ps.	St.
-ST, -OF	71.67%	50.00%	66.67%	80.39%
-ST, +OF	70.00%	50.00%	73.33%	**88.24%**
+ST, -OF	**80.00%**	62.50%	83.33%	72.55%
+ST, +OF	**80.00%**	**87.50%**	**90.00%**	80.40%

Table 3: Precision and recall rate of each combination, for 4 classes. Each value in the table is the average of corresponding values of 3 checkpoints.

Methods	Accuracy			
	1^{st}	2^{nd}	3^{rd}	Avg.
-ST, -OF	71.43%	68.25%	63.49%	67.72%
-ST, +OF	68.25%	68.25%	74.60%	70.37%
+ST, -OF	74.60%	73.02%	74.60%	74.07%
+ST, +OF	84.13%	**85.71%**	80.95%	**83.60%**

Table 4: Overall accuracy of each combination. Results of all selected checkpoints are shown as well as average values. $\mathbf{1}^{st}$, $\mathbf{2}^{nd}$, $\mathbf{3}^{rd}$ refer to validation accuracy ranking

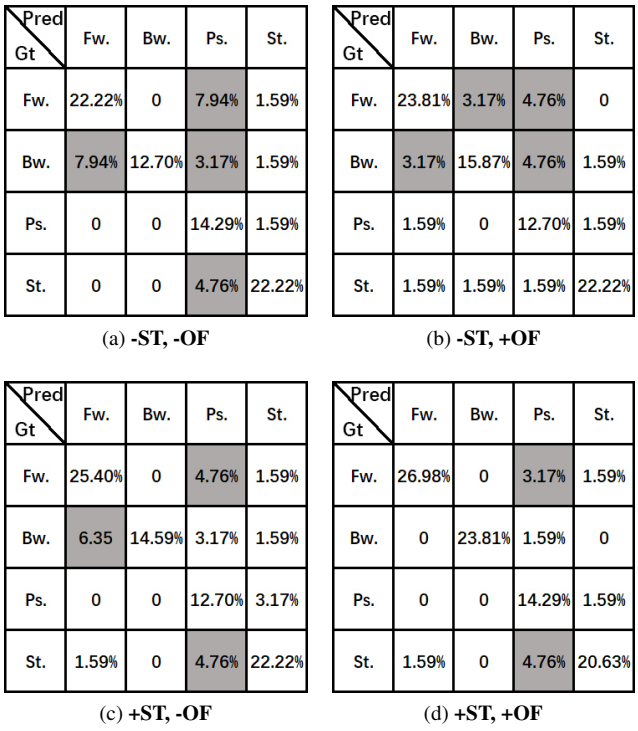
recognition network, thus indicating that the action recognition network can tolerate some joint localization errors.

Fig. 7 shows some failure cases. It can be seen that accuracy of action recognition is limited by accuracy of pose estimation. Misclassification of Fig. 7 (c) and (d) is due to the failure in predicting stick top and stick end. This is common in shooting case because the stick is likely to move too fast, or be lifted too high (lifting stick too high is a rare case in training images, so the network cannot recognize the stick well in this situation). Fig. 7 (a) and (b) reveal a major inherent downside of the method. Even if pose is predicted precisely, correctness cannot be guaranteed. Under many circumstances, contextual information is helpful, such as movement of the puck, position of the goal, action of surrounding players. Pose does not contain these factors, and it is also difficult for the network to learn to capture crucial detailed information from optical flows, especially when the dataset is too limited.

5. Conclusion

In this paper, we propose a novel two-stream architecture for action recognition. The two streams estimate pose and parse optical flows via CNN, which are then concatenated and passed through fully-connected layers to output classification scores. The architecture extends general two-stream networks by leveraging pose, which is a high-level feature that is shown to be suitable for action recognition, achieving 85% end-to-end accuracy. Experimental results demonstrate that pose and optical flows, as different-level features, are complementary to each other. It is also demonstrated that hand-held objects, sticks in the context of ice hockey, play an important role in analyzing the sport actions. In addition, we transfer the information from the pose estimation model pre-trained on MSCOCO dataset to our small hockey dataset achieving 87% overall accuracy measured by PCKh@0.5.

There is room for improvement. (1) Although three sparsely-sampled images are generally adequate to depict an action, considering additional images can be more reliable and accurate. (2) Sometimes, a joint in an image that is difficult, even for a human to localize, can be better inferred by utilizing the neighboring frames i.e., temporal information can also be leveraged in pose estimation. (3) High-level activities such as puck location and goal scored, can also be taken into consideration.



(a) **-ST, -OF** (b) **-ST, +OF**

(c) **+ST, -OF** (d) **+ST, +OF**

Figure 8: Confusion matrices for 4 combinations. In each cell is percentage of sequences which belong to a certain class and are mistaken for a certain class. Cells indicating misclassfication with ratio higher than 3% are highlighted.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, June 2014.

[2] G. F. Angela Yao, Juergen Gall and L. V. Gool. Does human action recognition benefit from pose estimation? In *Proceedings of the British Machine Vision Conference*, pages 67.1–67.11. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.67.

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

[4] Y. Cai, N. de Freitas, and J. J. Little. Robust visual tracking for multiple targets. In *European Conference on Computer Vision - ECCV 2006*, pages 107–118. Springer, 2006.

[5] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, July 2017.

[6] M. Carbonneau, A. J. Raymond, . 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)*, pages 2808–2811, May 2015.

[7] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017.

[8] G. Ch'eron, I. Laptev, and C. Schmid. P-CNN: Pose-based CNN Features for Action Recognition. In *International Conference on Computer Vision - ICCV 2015*, 2015.

[9] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid. Potion: Pose motion representation for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, April 2017.

[11] M. Fani, H. Neher, D. A. Clausi, A. Wong, and J. Zelek. Hockey action recognition via integrated stacked hourglass network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 85–93, July 2017.

[12] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, June 2016.

[13] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, page 1, 2013.

[14] J. Gall, A. Yao, and L. Van Gool. 2d action recognition serves 3d human pose estimation. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, pages 425–438, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[15] R. Gao, B. Xiong, and K. Grauman. Im2flow: Motion hallucination from static images for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3165–3174, July 2017.

[17] T.-W. Hui, X. Tang, and C. C. Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8981–8989, 2018.

[18] U. Iqbal, M. Garbade, and J. Gall. Pose for action - action for pose. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 438–445, May 2017.

[19] M. Jain, H. Jgou, and P. Bouthemy. Better exploiting motion for better action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2555–2562, June 2013.

[20] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *2013 IEEE International Conference on Computer Vision*, pages 3192–3199, Dec 2013.

[21] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, Jan 2013.

[22] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, June 2014.

[23] Y. Kong, X. Zhang, Q. Wei, W. Hu, and Y. Jia. Group action recognition in soccer videos. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, Dec 2008.

[24] B. Li, C. Yang, Q. Zhang, and G. Xu. Condensation-based multi-person detection and tracking with hog and lbp. In *2014 IEEE International Conference on Information and Automation (ICIA)*, pages 267–272, July 2014.

[25] D. Li, T. Yao, L. Duan, T. Mei, and Y. Rui. Unified spatio-temporal attention networks for action recognition in videos. *IEEE Transactions on Multimedia*, 21(2):416–428, Feb 2019.

[26] F. Li and R. J. Woodham. Video analysis of hockey play in selected game situations. *Image and Vision Computing*, 27(1-2):45–58, 2009.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[28] Z. Liu, Z. Miao, and Y. Huo. A realtime human action recognition method based on single view key poses in sports video. 2015.

[29] W.-L. Lu and J. J. Little. Simultaneous tracking and action recognition using the pca-hog descriptor. In *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, pages 6–6, June 2006.

[30] W.-L. Lu and J. J. Little. Tracking and recognizing actions at a distance. In *Proceedings of the ECCV Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE06), Graz, Austria*, volume 1, 2006.

[31] W.-L. Lu, K. Okuma, and J. J. Little. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, 27(1-2):189–205, 2009.

[32] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[33] Z. Musa, M. Z. Salleh, R. A. Bakar, and J. Watada. Gbln-pso and model-based particle filter approach for tracking human movements in large view cases. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(8):1433–1446, Aug 2016.

[34] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, June 2015.

[35] B. X. Nie, C. Xiong, and S. Zhu. Joint action recognition and pose estimation from video. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1293–1301, June 2015.

[36] K. Okuma, D. G. Lowe, and J. J. Little. Self-learning for player localization in sports video. *arXiv preprint arXiv:1307.7198*, 2013.

[37] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European conference on computer vision*, pages 28–39. Springer, 2004.

[38] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[40] K. Sozykin, S. Protasov, A. Khan, R. Hussain, and J. Lee. Multi-label class-imbalanced action recognition in hockey videos via 3d convolutional neural networks. In *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 146–151, June 2018.

[41] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1390–1399, 2018.

[42] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *European conference on computer vision*, pages 140–153. Springer, 2010.

[43] M. R. Tora, J. Chen, and J. J. Little. Classification of puck possession events in ice hockey. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 147–154, July 2017.

[44] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, Dec 2015.

[45] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[46] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, June 2018.

[47] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, June 2013.

[48] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[49] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[50] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[51] L. Wang, Y. Qiao, and X. Tang. Latent hierarchical model of temporal structure for complex activity classification. *IEEE Transactions on Image Processing*, 23(2):810–822, 2014.

[52] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[53] Y. Wang, L. Zhou, and Y. Qiao. Temporal hallucinating for action recognition with few still images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5314–5322, 2018.

[54] A. Yao, J. Gall, and L. van Gool. Coupled action recognition and pose estimation from multiple views. *International Journal of Computer Vision*, 100(1):16–37, Oct. 2012.

[55] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann. Hidden two-stream convolutional networks for action recognition. *arXiv preprint arXiv:1704.00389*, 2017.