# Multi-Modal Three-Stream Network for Action Recognition

Muhammad Usman Khalid[1,2] and Jie Yu[1]

[1]Computer Vision Research Lab, Robert Bosch GmbH, Hildesheim, Germany
[2]Department of Computer Science, TU Dortmund, Germany

***Abstract*—Human action recognition in video is an active yet challenging research topic due to high variation and complexity of data. In this paper, a novel video based action recognition framework utilizing complementary cues is proposed to handle this complex problem. Inspired by the successful two stream networks for action classification, additional pose features are studied and fused to enhance understanding of human action in a more abstract and semantic way. Towards practices, not only ground truth poses but also noisy estimated poses are incorporated in the framework with our proposed pre-processing module. The whole framework and each cue are evaluated on varied benchmarking datasets as JHMDB, sub-JHMDB and Penn Action. Our results outperform state-of-the-art performance on these datasets and show the strength of complementary cues.**

## I. Introduction

Human action recognition in video has attracted a lot of attention in varied application domains like autonomous driving, human-machine interaction, video surveillance and health support. It aims to understand human behavior and interaction by exploiting visual features and temporal dynamics from video. One of the major challenges of action recognition is the large variability in human actions, i.e. humans perform a single action differently or single human carries out each action in many ways. In addition, there are variations due to camera position, camera motion, occlusion and resolution.

Recently, impressive progresses in this area have been achieved [1]–[4]. Effective feature extraction from large amount of video data has proved to a very crucial factor. For example, the very successful two stream networks proposed in [1], [5] are trained individually on RGB frames and optical flows to extract complementary features, i.e. visual appearance and motion dynamics, which are fused in a late fusion manner. Nevertheless, the performance of these networks is still significantly affected by quantity and quality of data. Current datasets for action recognition in the community are still relatively limited compared to image classification tasks in the sense of diversity and sample quantity. since datasets are relatively small compared to image classification tasks. Collecting and annotating video datasets demand high amount of resources and time. To this end, human poses as high-level and compact description become an important features, as they show good performance on even relatively small datasets. The approach proposed in [6] orders and encodes human joint poses into $3D$ tensors to train a CNN network, fusing its output with a spatial attention mechanism on RGB videos, where all body joints are computed and used.

This paper presents a novel approach for exploiting complementary data sources: RGB, optical flows, and human poses as data inputs for training. In particular, an end-to-end CNN framework is proposed to train directly on body joint tensor, which can be derived from GT poses or even noisy pose detections by recent pose estimators, e.g. [7]. For the latter case, practical post processing mechanisms are employed to handle imperfect or missing joint detections in realistic videos. Finally, complementary cues for action recognition, i.e. appearance, optical and posture features are analyzed and fused to handle varied action classes. Experiments are performed on the challenging action recognition datasets, namely JHMDB, sub JHMDB and Penn Action, our results outperform state-of-the-art approaches.

The remainder of the paper is organized as follows: in Section II, recent trends of action recognition approaches are briefly reviewed. Our proposed Pose ConvNET is introduced in Section III. The following Section describes our fusion schemes to incorporate multi-modal inputs. Experiments and evaluation results are given in Section V. In the last section, the proposed approaches and results are concluded.

## II. Related Work

Many of the action recognition methods are based on high dimensional features from videos using hand crafted features. Unsupervised learning approaches like bag-of-words and fisher vectors have been proved to be a very effective way to extract discriminative and compact representation from such high dimensional data [8]. Some approaches utilized also deep learning features and combined with hand crafted features as in [4].

To capture temporal structure of actions in video, [9] stacked consecutive video frames and extended the first convolutional layer to learn the spatio-temporal features while exploring different fusion approaches, including early fusion, slow fusion and late fusion. In contrast to previous approaches which can take only fixed number of temporal inputs, [10] proposed Long Term Recurrent Convolutional Networks (LRCN) which can work with variable temporal inputs and can also incorporate long term dependencies. In [5], a novel sparse spinet concept is proposed to improve the efficiency of temporal sampling by

considering the high redundant information between neighboring frames.

Two stream network is proposed in [1] to extract visual and motion features simultaneously, which improved the classification accuracy greatly compared to each feature alone. Such an architecture improved many challenging action recognition problems significantly and become more and more popular. In [3], two stream network is exploited with different fusion schemes via 3D convolutional kernels and 3D pooling.

In addition to the successful two stream networks, human poses are also very popular features utilized to solve human action recognition problems. In [2], estimated poses are used and coded with bag-of-words approach to classify actions. Some approaches like [11] and [12] solved pose estimation and action recognition jointly, where [11] formulated pose estimation as an optimization problem over a set of action specific manifolds and performed two tasks iteratively. In order to incorporate 3D human poses in CNN, [6] proposed a novel $3D$ pose-tensor, which preserved the spatial structure of body joints and encoded pose motion in a compact manner. Along with pose CNN, a spatial attention mechanism is used to localize relevant regions for action classification. Inspired by this idea, we propose an extended framework for 2D poses with imperfect joints, so that it can be widely applied in any videos without 3D information available.

## III. Pose Stream

In this section, we introduce our technique to use $2D$ pose information of human joints in video to do action classification. $2D$ joint positions are arranged in a special formation named pose tensor, that preserves the spatial structure of pose and motion information present in the video. The pose tensor is trained directly with a CNN named Pose ConvNET. In contrast to some pose based techniques [6] that works only with ground truth poses, our technique is robust to work with both ground truth and detected poses. In our experiments, detected poses are estimated by the 2D CMU pose estimator [7]. However, estimated joint positions are often not completed due to occlusion or other issues in video. We propose two interpolation methods to complete missing joint positions, which improve the training efficiency and performance significantly. Details of the proposed 2D pose tensor and Pose ConvNET will be described in the following sections.

### A. Formation of Pose Tensor

In a video frame, a person can be represented by its $n$ corresponding joints in 2D image coordinates as shown in Figure 1(a). The joint positions can be either annotated manually, i.e. ground truth [14], [15] or estimated by a pose estimator [7], [16]. Following [13], a special joint ordering is formulated keeping their neighborhood relationship. Figure 1(b) shows a tree-like structure where each node represents a corresponding joint position. The tree is formed by starting from belly joint and branches are formed with limbs and hand joints as shown in Figure 1(b). To form a pose tensor, a path passes from the root node through all subsequent nodes in the pose tree in such a way that all nodes are traversed at least once as shown in Figure 1(c). This traversal keeps the neighborhood relationship among joints preserved in the structure. Based on this path, pose tensor is formed by concatenating all the joint positions $(x, y)$ that occurs in the path traversal in one row of pose tensor as shown in Figure 1(d). Here each row corresponds to the joint positions of one person in one frame. By keeping the same ordering of joints, joint positions in any other frames in a video sequence are stacked row-wisely to form a pose tensor. In this way, a video sequence, corresponding to one action sample, is described by a pose tensor.

More specifically, a video $V$ is divided into $K$ segments $(S_1, S_2, ..., S_K)$ of equal length to keep the dimension of pose tensor fixed for all video samples in the dataset, One snippet $T_k$ is randomly chosen from each segment $S_k$. Then a pose tensor is formed by joint positions of the corresponding person in all snippets $(T_1, T_2, ..., T_K)$ as shown in figure 1(d). The second and third channel of pose tensor are the first-order and the second order derivation of joints positions, corresponding to velocity and acceleration of joints in consecutive snippets $(T_{k-1}, T_k)$. Thus, $3D$ pose tensor is formulated which not only preserves the spatial structure of human pose but also captures motion information of joints.

*a) Pose Normalization:* As 2D joint positions in image coordinate are sensitive to camera perspectives and image resolution, which are not scale invariant. A normalization is required which keeps all the poses to be of similar size and to be centered in the image. Firstly, joint positions are normalized with respect to torso length which keeps all poses to be of same scale, given mathematically as.

$$\overline{P}_{i(x,y)} = \frac{P_{i(x,y)}}{d} \qquad \forall i = 1, ..., n \tag{1}$$

Here, $d$ is the torso length, $P_i$ are the raw joint position (x,y), $\overline{P_i}$ are the scaled joint position (x,y) for joint $i$. These scaled joint positions are then shifted such that mid point of torso shifts at origin (0,0), mathematical defined as

$$PF_{i(x,y)} = \overline{P}_{i(x,y)} - \overline{P}_{torso(x,y)} \quad \forall i = 1, .., n \tag{2}$$

Here,$\overline{P}_{torso(x,y)}$ is the midpoint between neck and belly joint positions, $PF_{i(x,y)}$ are final normalized joint positions (x,y) for joint $i$ to make 3D pose tensor.

### B. Handling of Missing Joint Positions

In practice body joints are not always visible in videos, therefore some joint positions can not be estimated correctly by a pose detector as shown in Figure 2(a). Handling of such missing joint positions is a critical part by formulating pose tensor. Simply marking these points as invalid or assigning a specific value, would corrupt the input and cause some unexpected issues by training a CNN on the pose tensor. Therefore, two interpolation techniques are implemented to estimate missing joint positions: temporal interpolation using the joint positions available in other frames, and spatial interpolation exploiting spatially neighbored existing joint in the same frame.

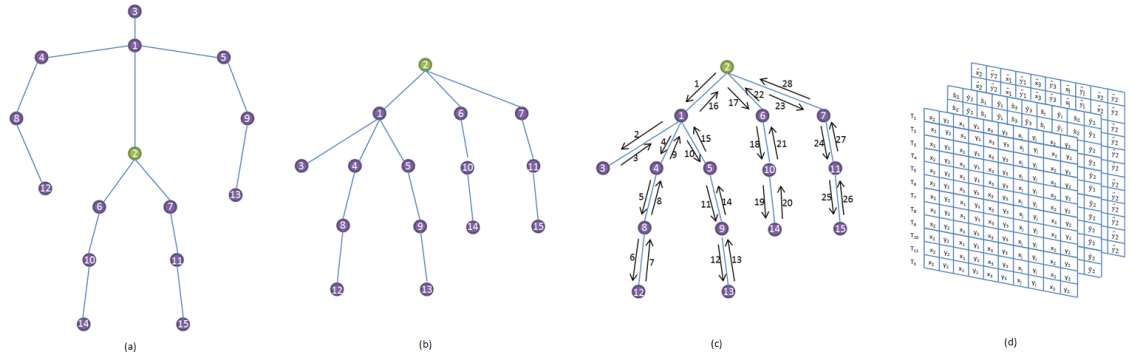


Fig. 1. Formation of pose tensor for JHMDB. (a) Full body joint position with corresponding labels (b) Tree like structure formed by starting from belly-joint node(c) Traversing of tree like structure by starting from first node and ending at the same node [13], (d) 3d pose tensor with first channel with special ordering of 2d joint positions, 2nd channel with differences of pose in 2 frames and 3rd channel differences of differences of pose in consecutive frames

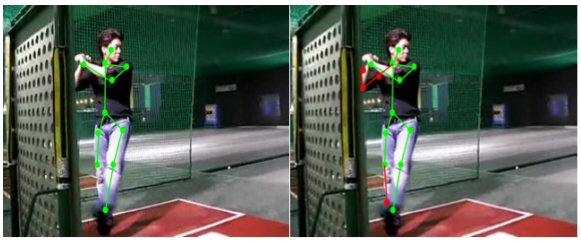


Fig. 2. Interpolation of pose (a) original pose detection (b) completed pose detection after interpolation of joint positions

*1) Temporal Interpolation of Pose:* Videos contain rich motion information covering the smooth movement of human joints. One consequence is that some invisible joints become visible in the continuous frames or inversely. By making using of the continuity of the joint movement, the position of invisible joints can be interpolated from temporally neighbored visible joints if any. We used a simple linear interpolation to estimate location of missing joint positions which achieves promising estimation for short temporal range. Temporal interpolation is especially useful by estimating missing joint positions with short-termly changing visibility.

*2) Spatial Interpolation of Pose:* For long-term occluded human joints, temporal interpolation has its limitation. If some joints are not detected for a long temporal range, the linear motion assumption made by temporal interpolation is not valid any more. Therefore, we exploit spatial context information of neighbored joints to estimate the missing joint. This idea is based on the fact that locations of joints of each pose are strongly statistically correlated, especially among neighbored joints, e.g. head and shoulder. Similar as [17], neighborhood relationships between joints are utilized to vote possible location of missing joints. A polynomial function is used to model the spatial relationship of neighbored joints. This model is learned from varied video datasets with ground-truth poses.

As directly neighbored joints provide more accurate estimation, the whole body is divided into 5 body parts keeping their tight neighborhood relations of joints as shown in Figure 3, where part 1 to part 4 have tight spatial relationship and part 5 has only a loose spatial relationship. For a missing joint position within frame, first all available joint positions of the corresponding body part with tight spatial relationship, i.e. part 1 to part 4, are selected to estimate the missing joint position. If no joint position from the corresponding body part is available, then joints from body part 5 are selected for missing upper body joints. For other cases, all the available joint positions of this pose in that frame are selected. Each selected joint position votes for the position of the missing joint and the average vote of all selected joints is considered as the final estimation of the missing joint.

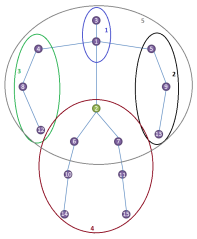

Fig. 3. Configuration of the 5 body parts for spatial interpolation of missing joint positions

## C. Pose ConvNET

Pose tensor is trained with a CNN (Pose ConvNET) in an end to end fashion. This ConvNET have two convolutional layers with a RELU function along with Max Pooling Layer. Final features are extracted via a fully connected layer and a fully connected softmax layer is used for classification. A relative shallow network is used with small filter size $3 \times 2$, as pose tensor is of highly compact data and consists of only high level features. An important advantage is that nor large amount of training data or properly pre-training are needed, therefore a flexible training for varied real-world applications is possible. The network is trained with Xavier initialization and a standard categorical cross-entropy loss function. A Pose

ConvNET trained for a video $V$ with $K$ segments can be mathematically defined as:

$$PCN(V) = F((T_1, T_2, ..., T_K); W) \quad (3)$$

Here, $F$ is the function representing Pose ConvNET with parameters $W$ which operates on pose tensor $(T_1, T_2, ..., T_K)$ formed by $K$ snippets and produces class scores for video $V$.

## IV. Three-Stream Convolutional Neural Network

Fusion of two stream CNN [1], [3], [5] based on RGB and optical flows has given promising results in the domain of human action recognition. We extend this framework by fusing an additional stream of pose tensor with the conventional two stream CNN. A three stream network is designed to capture context information from a spatial channel, motion information from a flow channel and semantic posture information using the proposed pose ConvNET.

We use the TSN framework proposed in [5] for training of RGB and optical flow streams, where warped flow fields [8] are calculated to compensate camera motion, to suppresses background motion and to make motion concentrated on actors, similar as a visual saliency map. Pre-trained spatial and temporal models on videos of UCF101 [18] are used and fine-tuned on the new datasets.

Following the sampling concept proposed in TSN framework, each video $V$ is divided into $P$ equal segments $(S_1, S_2, .., S_P)$. For each segment, one snippet $T_p$ for spatial stream and stack of consecutive snippets within segment $S_p$ for temporal stream are randomly sampled. Each CNN model is trained separately with these sampled snippets from video. The temporal segment network is defined mathematically as

$$TSN(V) = G(F(T_1; W), F(T_2; W), ....F(T_P; W)) \quad (4)$$

Here, $F(T_p; W)$ is the function representing spatial and temporal CNN models with parameters $W$ which operates on the short snippet $T_p$ and produces class scores for that snippet. $G$ represents the segmental consensus function which aggregates the scores from all the snippets within one video and gives video based score. We used average pooling of scores as consensus function $G$. During training of each of TSN streams, this aggregated video level prediction is used to minimize the loss function and errors are propagated through back propagation algorithm.

Three stream convolutional neural network (TSCNN) is formulated by fusing scores from Pose ConvNET with video based scores from spatial and temporal streams of TSN. The final score will be weighted sum of scores as given below:

$$\begin{aligned} &TSCNN(V) = \\ &w_p * PCN(V) + w_s * TSN_s(V) + w_t * TSN_t(V), \end{aligned} \quad (5)$$

where $PCN$, $TSN_s$ and $TSN_t$ are video based scores for pose ConvNET, spatial and temporal streams as shown in Figure 4. $w_p$, $w_s$ and $w_t$ are the weights accordingly, which are estimated empirically.

TABLE I
Classification accuracy for each of models trained on the RGB, optical flow, poses (ground truth, estimated) of the JHMDB, subJHMDB and Penn Action datasets

| Network | Accuracy | | |
|---|---|---|---|
| | JHMDB | subJHMDB | Penn Action |
| Spatial (RGB frame) | 57.90% | 58.76% | 86.42% |
| Temporal (Optical flow) | 73.33% | 81.14% | 96.72% |
| Pose (GT pose) | 70.84% | 75.44% | 96.25% |
| Pose (Est. pose) | 54.90% | 63.60% | 89.32% |

## V. Experiments

In this section, experiments on datasets JHMDB, sub-JHMDB [14] and Penn Action [15] are presented along with some specific implementation details. Both datasets contain varied action videos with action labels and 2D human pose annotations, which are required by the Pose ConvNET stream. We explore the performance of each stream and their fusion in terms of accuracy of action classification. Finally we compare the performance of our approach to some state-of-the-art approaches.

### A. JHMDB

JHMDB dataset [14] contains 21 action classes with total of 928 videos and 33183 frames. A subset of the JHMDB named sub-JHMDB is also provided with 316 videos and 12 action classes. Different environments, changing camera view points and high intra-variations of actions are covered in both datasets. All joints are annotated manually even under occlusion. We conducted two experiments: one is based on $2D$ annotated joint positions (GT Pose) with $n = 15$, another one is based on estimated $2D$ joint positions (Est. Pose) by [7] with $n = 14$, where 4 face key points are discarded. All joint positions are normalized and missing joint positions in case of estimated poses are estimated by interpolation (see Section III). From them the final pose tensor is formed with $K = 15$ and of size $(15 \times 58 \times 3)$ for GT Pose and $(15 \times 54 \times 3)$ for Est. Pose.

For comparison, four CNN models are trained separately on each cue, i.e. RGB, optical flow and poses (GT and Est). Pre-trained spatial and temporal models on UCF101 [18] are used with $P = 15$ snippets and their fully connected layers are fine tuned with JHMDB and sub-JHMDB datasets. According to the standard protocol, three splits are provided. Experiments are performed on all three splits and averaged results are reported in Table I. The temporal model performs best on JHMDB and sub-JHMDB. In contrast, the spatial model is much less performing. It shows that motion is much more important features than image context on both datasets, that matches our observations as well. The model trained with GT Pose shows close performance to the temporal model. However, the model trained with Est. Pose has a significant accuracy drop, especially on JHMDB, where full bodies are often not visible, which decreases the performance. It shows

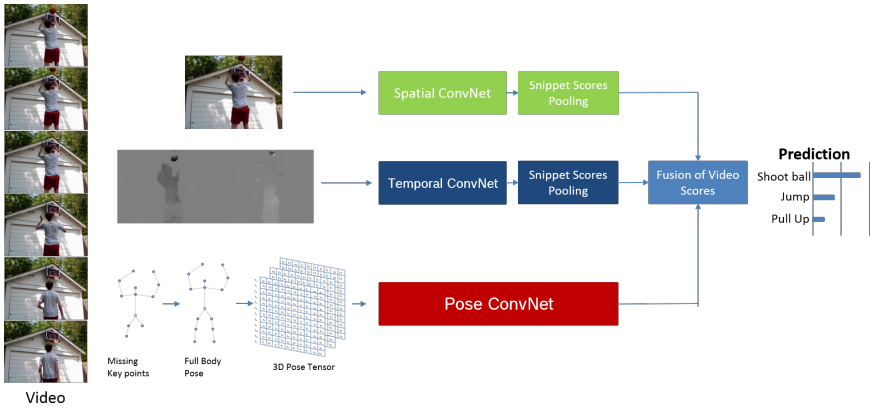


Fig. 4. Three Stream Network Architecture

that our interpolation methods have some limitations by facing lots missing joint positions in video.

### B. Penn Action Dataset

The Penn Action dataset contains 15 action categories. The dataset provides both action labels and positions of $n = 13$ human joints even under occlusion. Following the setting in [15], data are divided into 50/50 for training and testing.

The spatial, temporal and pose models are trained similarly as in Section V-A. Results of each individual stream : RGB, optical flows and pose tensor (GT and Est.) are reported in Table I. Pose tensor based on GT pose was built with $n = 13$ joints, head joint as root node and $K = 15$ snippets. Thus, the size of pose tensor with GT Pose was $(15 \times 50 \times 3)$. Similar trends can be observed as that on JHMDB, where the temporal model performs best. However, the results of the model trained on GT Pose are very close to that of temporal model. The model trained on estimated poses performs better than spatial model, despite the fact that pose tensor has more compact input. It shows the power of semantic features by learning.

TABLE II
CLASSIFICATION ACCURACY FOR EACH OF THE SPATIAL, TEMPORAL AND POSE CNNS ON THE JHMDB AND SUB-JHMDB DATASETS

| **Fusion Modularity** | **Accuracy** | | | |
|---|---|---|---|---|
| | **JHMDB** | | **sub-JHMDB** | |
| | Pose (GT) | Pose (est.) | Pose (GT) | Pose (est.) |
| RGB + flow | 75.83% | | 78.09% | |
| RGB + pose | 73.45% | 62.86% | 69.02% | 66.10% |
| flow + pose | 79.32% | 71.69% | 83.20% | 81.30% |
| RGB + flow + pose | 83.05% | 78.81% | 87.29% | 85.12% |

### C. Fusion of Multiple Cues

In this section varied fusion schemes are evaluated on the JHMDB, sub-JHMDB and Penn Action Datasets. Table II shows the performance on the JHMDB, sub-JHMDB of four different combination of three cues, RGB + optical flow (conventional two stream network), RGB + pose, flow + pose, and RGB + flow +pose, with two pose variants, GT and estimated pose. For all the experiments, we used $(w_p, w_s, w_t) = (1, 1, 1)$ as the weights for fusion of three streams. Comparing to conventional two stream fusion configurations proposed in [5], improvements of $7.2\%$ and $9.2\%$ respectively are achieved on the JHMDB and sub-JHMDB by using GT pose, while $2.98\%$ and $7.03\%$ by using estimated poses. Even the fusion of the temporal and pose models outperforms the conventional two stream. A clear benefit by fusing additional pose feature can be observed.

TABLE III
CLASSIFICATION ACCURACY FOR EACH OF THE SPATIAL, TEMPORAL AND POSE CONVNETS ON PENN ACTION DATASET

| **Fusion Modularity** | **Accuracy** | |
|---|---|---|
| | pose (GT) | pose (Estimated) |
| RGB + flow | 95.04% | |
| RGB + pose | 93.72% | 91.67% |
| flow + pose | 97.85% | 97.10% |
| RGB + flow + pose | 98.50% | 98.41% |

Similar results on the Penn action dataset are shown in Table III: the performance of the three stream network using the GT human pose is $3.46\%$ better than the RGB and optical flow fusion, proposed in [5]. Even the fusion using estimated poses is very close to three stream with GT pose. It shows that recent pose estimators are already very stable on some real world data.

### D. Comparison to State-of-the-art Approaches

A comparison of of our three stream network using GT and estimated joint positions with recent state-of-the-art deep learning and conventional hand crafted approaches for JHMDB, sub-JHMDB and Penn Action datasets are reported in Table IV. Clearly, apart from JHMDB with estimated Poses, our proposed three stream network outperforms the recent state-of-the-art approaches with significant difference for all three datasets, with GT and Estimated Poses. On JHMDB the three stream network with estimated poses has a lower performance due to frequently invisible body parts as mentioned in the previous section. These results explain that our proposed fusion scheme of three cues shows a complementary behavior.

TABLE IV
COMPARISON OF THE THREE STREAM CNN (TSCNN) WITH ESTIMATED (EST) AND GROUNDTHRUTH (GT) HUMAN JOINT POSITIONS WITH THE STATE-OF-THE-ART ON JHMDB SUB-JHMDB AND PENN ACTION DATASETS.

| State-of-the-art | JHMDB | Sub-JHMDB | PennAction |
|---|---|---|---|
| Pose [14] | 69 | 52.9 | - |
| STIP [15] | - | - | 82.9 |
| Action Bank [15] | - | - | 83.9 |
| MST [19] | - | 45.3 | 74.0 |
| AOG [20] | - | 61.2 | 85.5 |
| P-CNN [21] | 79.5 | 72.5 | - |
| Hierarchical [22] | - | 77.5 | - |
| IHLPF [23] | 80.4 | - | - |
| JDD [24] | - | 83.3 | 95.7 |
| Pose+idt-fv [12] | - | 74.6 | 92.9 |
| RPAN-(S+T) [25] | - | 81.1 | 97.4 |
| TSCNN Est pose (Our) | 78.8 | 85.1 | 98.4 |
| TSCNN GT pose (Our) | **83.1** | **87.3** | **98.5** |

| Actions | RGB | Flow | Pose | Fused |
|---|---|---|---|---|
| Catch | 0.40 | 0.60 | 0.40 | 0.80 |
| Climb_stairs | 0.00 | 1.00 | 1.00 | 1.00 |
| Golf | 1.00 | 1.00 | 1.00 | 1.00 |
| Jump | 0.25 | 0.75 | 0.63 | 0.88 |
| Kick_ball | 0.63 | 0.88 | 0.38 | 0.75 |
| Pick | 0.63 | 1.00 | 1.00 | 1.00 |
| Pullup | 1.00 | 1.00 | 0.77 | 1.00 |
| Push | 1.00 | 1.00 | 0.90 | 1.00 |
| Run | 0.43 | 0.71 | 0.43 | 0.71 |
| Shoot_ball | 0.17 | 0.33 | 1.00 | 0.83 |
| Swing_baseball | 0.14 | 0.86 | 0.86 | 0.86 |
| Walk | 0.50 | 0.50 | 1.00 | 1.00 |

Fig. 5. Classification Accuracy for each of class of subJHMDB split1 for each of the cue and their fusion

### E. Qualitative Analysis of Cues

In order to get more insights of the complementary behavior of different cues, some examples are qualitatively examined and summarized in Figure 5. It is clear that no single cue alone gets an overall good performance on varied action classes, as the fused cues do. It is observed that the flow cue works especially good on actions with fast motions, e.g. run and swing baseball, while the pose cue contributes much to actions with unique posture or significant body motion, e.g. climb stair, pick and shoot ball. The RGB cue performance worse than other two cues, however it is still very important by understanding the context information, as meadow for action "golf". It is confirmed that almost all actions are improved by fusing all cue together. However, it is not a trivial task to identify contribution of each cue on different actions empirically. How to learn the fusion scheme dynamically, is an important research topic for the future.

## VI. CONCLUSION

This paper has presented a novel framework to utilize human body poses along with RGB frames and optical flows for action recognition. Both GT and estimated poses are suported, that enables a wide range of applications in real world. In experiments, very promising results are shown in the benchmarking datasets and outperform recent state-of-the-art approaches. The complementary behavior of RGB, optical flow and pose is observed in our experiments. Dynamic adaptation of fusion scheme for different actions will be investigated in the future.

## ACKNOWLEDGMENT

This work was supported by the Computer Vision Research Lab of Robert Bosch GmbH.

## REFERENCES

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.
[2] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *CVPR*, 2013.
[3] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*, 2016.
[4] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *CVPR*, 2015.
[5] 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 *ECCV*, 2016.
[6] F. Baradel, C. Wolf, and J. Mille, "Pose-conditioned spatio-temporal attention for human action recognition," *arXiv preprint arXiv:1703.10106*, 2017.
[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.
[8] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *CVPR*, 2013.
[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.
[10] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.
[11] A. Yao, J. Gall, and L. Van Gool, "Coupled action recognition and pose estimation from multiple views," *International journal of computer vision*, 2012.
[12] U. Iqbal, M. Garbade, and J. Gall, "Pose for action-action for pose," in *FG*. IEEE, 2017.
[13] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *ECCV*, 2016.
[14] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *ICCV*, 2013.
[15] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *ICCV*, 2013.
[16] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.
[17] J. Müller and M. Arens, "Human pose estimation with implicit shape models," in *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*. ACM, 2010, pp. 9–14.
[18] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.
[19] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *CVPR*, 2014.
[20] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu, "Joint action recognition and pose estimation from video," in *CVPR*, 2015.
[21] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *ICCV*, 2015, pp. 3218–3226.
[22] I. Lillo, J. Carlos Niebles, and A. Soto, "A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets," in *CVPR*, 2016.
[23] J. Fan, Z. Zha, and X. Tian, "Action recognition with novel high-level pose features," in *ICMEW*. IEEE, 2016.
[24] C. Cao, Y. Zhang, C. Zhang, and H. Lu, "Action recognition with joints-pooled 3d deep convolutional descriptors." in *IJCAI*, 2016.
[25] W. Du, Y. Wang, and Y. Qiao, "Rpan: An end-to-end recurrent pose-attention network for action recognition in videos," in *CVPR*, 2017.