Action Recognition by Single Stream Convolutional Neural Networks: An Approach using Combined Motion and Static Information

Sameera Ramasinghe, Ranga Rodrigo
Department of Electronic and Telecommunication and Engineering
University of Moratuwa, Sri Lanka
samramasinghe@gmail.com, ranga@uom.lk

Abstract

We investigate the problem of automatic action recognition and classification of videos. In this paper, we present a convolutional neural network architecture, which takes both motion and static information as inputs in a single stream. We show that the network is able to treat motion and static information as different feature maps and extract features off them, although stacked together. We trained and tested our network on Youtube dataset. Our network is able to surpass state-of-the-art hand-engineered feature methods. Furthermore, we also studied and compared the effect of providing static information to the network, in the task of action recognition. Our results justify the use of optic flows as the raw information of motion and also show the importance of static information, in the context of action recognition.

1. Introduction

Recently, there has been a growing interest in applying semi-supervised or unsupervised learning methods for extracting and defining features for a wide variety of machine-learning and computer vision related tasks. These methods often rely on learning multiple layers of feature hierarchies to extract increasingly abstract representations at each stage. These deep learning methods have shown promising results, over hand crafted features, in areas such as speech (Mohamed *et al.*, Dahl *et al.* [5]) and vision (Jarrett *et al.*[1], Ciresan *et al.* [4], Rifai *et al.* [17], Krizhevsky *et al.* [13]).

However, applying deep-learning methods for the task of activity recognition and classification still remains an open and interesting research problem, mainly due to the temporal and high dimensional data. Recognizing human actions in real world environment has many applications, commercial as well as non-commercial. Some examples are, security surveillance, sports domain, military applications and patient monitoring. Action recognition in real world environment entails challenges of its own, as it is difficult to extract robust features to represent videos, which contains cluttered backgrounds, occlusions, and viewpoint variations.

Most of the existing approaches, which tackle this problem, such as [9], [19], [16], are based on hand-engineered features. Although these hand crafted feature-methods have shown promising results over many challenging datasets, the fact remains that deciding on a global feature, which is able to compensate for cluttered back grounds, occlusions, and viewpoint variations, is a cumbersome task. The advantage of unsupervised or semi-supervised learning techniques is its adjustability to the dataset. If designed correctly, these methods are able to learn and generate complex feature detectors at higher levels by itself, which can accurately classify a particular dataset. In other words, the network itself is able to decide on which features are the best for the task of classifying that particular dataset. This is extremely advantageous when we work with real-world videos. Therefore, we construct a deep convolutional neural network, and test it with the popular dataset, Youtube Action Dataset [15]. This contains 11 action categories, entailed with the challenges mentioned above.

In our method, we apply stacked data frames as inputs to the network. We follow the philosophy that in the context of action recognition, providing hand-calculated motion information as an input to the network actually helps rather than letting it extract motion information from raw RGB values, in contrast to still image classification. We treat optic flows as the basic input which contains motion information. We also provide it with few still images, and let it extract static information about scene context. We do this because scene context holds vital information about the action which is performed on it, although not as important as motion. Therefore, we create a stack of optic flows and still video frames to represent each video and use it as an input to the network.

Once trained, we compare it with the available state-of-the-art methods, on the same dataset, and show that our

978-1-4799-6100-9/15/$31.00 ©2015 IEEE

model is able to surpass the best results that have been reported over this dataset by a significant margin. For classification, we use *one against the rest* method, as done by the other authers. Then, we also study and compare the effect of adding static information to the network, in terms of the accuracy.

2. Related Work

Hand-engineered feature based methods have been heavily used in action recognition in the past. Most of these methods are based on the popular static-image features like HOG [6], SURF [2] *etc.*. In some cases, extensions of these static image features to the temporal axis, or features like HOF , MBH [7] and HOG3D [12] also provide basis for these hand-engineered methods. These methods give reasonably accurate results over standard datasets. In this section, we will focus only on unsupervised or semi-supervised feature learning approaches. We will discuss a few approaches that are closely related to our work.

Quoc *et al.* [14] use an extension of Independent Subspace Analysis (ISA) algorithm to learn invariant spatio-temporal features from unlabeled video data. First they learn features with small input patches and then they use convolutional layers to learn more complex features from larger input patches, using previously learned features as inputs, which are suitable to be applied for video recognition.

Shuiwang *et al.* [10] perform 3D convolution across both spatial and temporal dimensions of stacked video frames, and capture motion information encoded in multiple adjacent frames. They hardwire the first input layer, which provides 5 channels: grayscale, gradient-x, gradient-y, optic flow-x, and optic flow-y channels. Although follow a similar procedure, we use only optic flows as motion information, and provide those information as a stack of frames. Also, we avoid using 3D convolution, which makes ours less computationally complex. Baccouche *et al.* [1], is quite similar to Shuiwang *et al.* [10]. Thier method is based on two steps: First, they use a fully automated process, extend the convolutional neural networks to 3D and learn spatio temporal features from videos. But, instead of applying hardwired inputs, they act directly on raw inputs, which are pixel intensities of video frames. Then, they use those learned features to train a recurrent neural network, for the purpose of classifying the entire sequence. In contrast, we provide our network, optic flows maps as motion information, rather than letting it extract those information directly from RGB values. Kim *et al.* [11] operate on slightly high level hand-crafted inputs (spatio-temporal outer boundaries volumes). As a second step, it uses a convolutional network to further improve those features.

Simonyan (*et al.* [18] uses a two stream approach for motion and static information, where they do late fusion of the outputs to produce the final result. As opposed to that, we use a single stream approach, and found that the network itself is able to identify still images and optic flows as different feature maps and extract features from them, even though stacked together. Therefore, using two separate streams for static and motion information and late fusion of both, seems not required.

3. Architecture

The network we present contains five convolutional layers, five max pooling layers, one fully-connected layer and one 2-way softmax layer. Each convolutional layer is immedieately followed by a max pooling layer, with a kernal size of 2×2 and a stride of 2. The overall architecture of our network is shown in figure 1.

As the activation function, we use leaky rectified linear units. The motivation behind this choice is that, unlike still image classifcation, which deals with RGB values, motion information contains negative values. These negative values are needed to be treated with importance, rather than cutting them off as in normal rectified linear units. We apply a small negative slope of 0.001 to the leaky rectified linear units. The activation function is,

$$f(x) = 1(x < 0)(0.001x) + 1(x \geq 0)(x). \qquad (1)$$

3.1. Optimization of the Network

We use stochastic gradient descent and back propagation to train our network. We use batch training instead of online training and use mini-batches of 64 samples. A momentum of 0.05 and weight decay of 0.0005 is used in training. All models are initialized with learning rates of 0.001.

4. Methodology

4.1. Enlargement of the Dataset.

The YouTube Action Dataset [15] consists of 1,168 sports and home videos from YouTube with 11 types of actions: *basketball shooting, cycle, dive, golf swing, horse back ride, soccer juggle, swing, tennis swing, trampoline jump, volleyball spike*, and *walk with a dog*. Each of the action sets is subdivided into 25 groups sharing similar environment conditions. This is a challenging dataset with camera jitter, highly cluttered backgrounds and variable illumination settings. The spatial resolution is 320×240.

Convolutional neural networks work best with larger datasets. Network is better trained and the accuracy generally improves if the available training dataset is large. Therefore we split each video into videos of 51-frames with 25-frames overlaps. By following this procedure, we also reduce the amount of data representing a single video and also we get a fixed, equal amount of data representing each video. After splitting, the dataset consists of 7669 videos.

4.2. Stacked Motion and Static Information for Representing Video Segments

Both motion information and static information contain information about the activities performed in a video. An action maybe highly correlated with the environment in which it is performed. For example, the diving action is highly associated with swimming pools and basketball shooting is highly probable to occur in a basket-ball court. Therefore, when we provide information to the network for classifying this data, it is important to provide both motion and information to increase the classification accuracy.

In this work, the network is trained using two approaches. In the first approach, videos are represented with both static and motion information, where input is a stack of optic flow maps and still images. In the second aproach, videos are represented with only motion information.

4.3. Calculation of Dense Optic Flows

For representing motion information, we choose dense optic flow maps. For generating optic flow maps, we use the efficient and accurate method presented by Brox *et al.* [3]. Their method is based on a theory for warping, and is based on 3 assumptions: a brightness constancy assumption, a gradient constancy assumption, and a discontinuity-preserving spatio-temporal smoothness constraint. Prior to calculating dense optic flows, we resample each video frame to the size 128×128 and convert each frame to grayscale. Then we calculate optic flows for the whole 51 frames, with a space of two frames between each used two frames, as

$$[O_{xj}, O_{yj}] = f(v_i, v_{i+3}) \tag{2}$$

where $f(\cdot)$ is the optic flow calculation function, v_i is the i^{th} video frame, and O_{xj} is the j^{th} optic flow map containing magnitude of the *x*-values of the optic flows, and O_{yj} is the j^{th} optic flow map containing magnitude of the *y*-values of the optic flows. This procedure produces a total of 26 frames which contain the optic flow information, 12 frames for the *x*-values and 12 frames for the *y*-values. Then we stack these frames in the order,

$$S_j = O_{x1}, O_{y1}, O_{x2}, O_{y2}, ..., O_{x13}, O_{y13} \tag{3}$$

where, S_j is the stack representing j^{th} video segment.

4.4. Stacking of Static Information

As mentioned in section 3.2 , we follow procedure mentioned in this section, only for approach 1. We choose 5 still frames from each video, with an interval of 9 frames between each. And we stack this with the optic flow frames which are calculated. Total stack representing a particular video segment is as follows:

$$S_j = O_{x1}, O_{y1}, O_{x2}, O_{y2}, ..., O_{x13}, O_{y13}, s_1, s_2, ..., s_5 \tag{4}$$

where, s_i are still frames sampled from each video.

4.5. Data Augmentation

Data augmentation is a method used as a counter measure for over fitting. It is a process of artificially enlarging the available dataset using label-preserving transformations. Also, it provides an amount of transformation invariance to the test data. In our work, we use four transformations: 1) rotation, 2) reflection, 3) translation and 4) scaling. In rotational transformation, each sample is rotated randomly between 0 and 360 degrees. Rotation algorithm rotates the each training stack of frames around its center points while preserving the label. It also preserves the size of the each input frame, which is 128×128, by cutting out edges and filling the blank areas with zero values. In reflection, we simply get the reflection of each frame in the stack. Then each stack is translated randomly between 0% and 50%, in both *x*-direction and *y*-direction. Scaling algorithm scales each stack between the factors -1.5 and 1.5, randomly. It is important to notice that in each of these transformations, the original stack dimensions are preserved.

4.6. Initialization of Weights

As a particular network gets deeper, the importance of the initialization scheme of the network becomes important. Stochastic gradient decent may work poorly with random initialization of weights if the network is too deep. The main two reasons for this are: 1) if the weights are too small, then the signals which are travelling through the network may shrink as it goes through layers of neurons and would eventually die out before it crosses the entire network, 2) if the weights are too large, the signals will grow as it passes through each layer of network until they are too large to be meaningful. As our network consists of 7 layers (not counting the max-pooling layes), we decided to use the method described in Glorot *et al.* [8], instead of random initialization of weights. Therefore, following Xavier initialization, our network weights in each layer are distributed as follows:

$$W \sim U\Big[\frac{-\sqrt{6}}{n_j + n_{j+1}}, \frac{\sqrt{6}}{n_j + n_{j+1}}\Big] \tag{5}$$

Where $U(\cdot)$ is the uniform distribution and n_j is the j^{th} layer size.

5. Results and Comparison

5.1. Approach 1

In our first approach, our data is a stack of both motion and static information. The summery of our results is shown

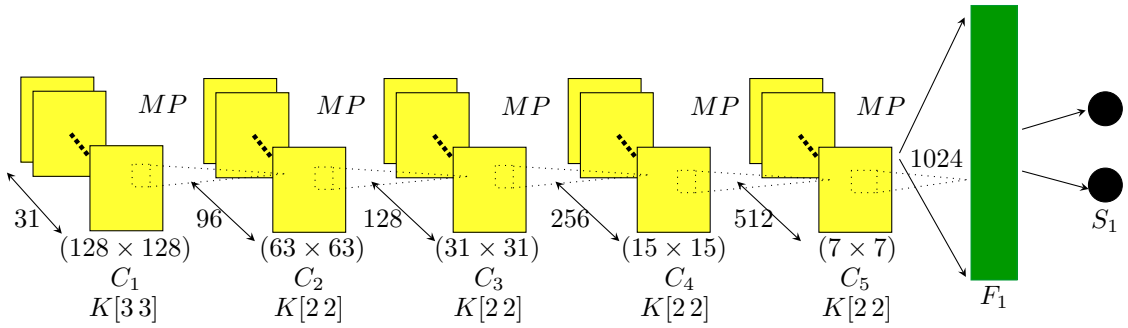

Figure 1. Network architecture consists of five convolutional layers, five max pooling layers, one fully connected layer and one softmax layer. Each convolutional layer is followed by a max pooling layer with a kernal size of 2×2 and a stride of 2.

in Table 1. We were able to achieve accuracies over 90% for all the classes, except in *soccor juggling*, which is 87.8% . We achieve an overall accuracy of 93.13%. We compare our results with three algorithms: 1) Wang *et al.* [19], 2) Lucas *et al.* (KLT) [16] and 3) Ikzler *et al.* [9].

Our method surpasses the other methods in all the classes except in *diving*, *golf swinging* and *volleyball spiking* . Overall, our network surpassed the currently reported best accuracy by a significant 8.93% margin.

5.2. Approach 2

In this approach, we trained and tested the network only with stacked motion information. The results and the comparison between approach 1 and 2 are presented in Table 2. Network still was able to classify the videos with a high accuracy. As expected, the accuracy over each class is reduced slightly compared to approach 1. This proves our initial assumption: providing static information about the environment increases the action classification accuracy, as the environment may hold vital information about a particular action. Also it is worth noting that the network is able to treat optic flow frames and static image frames as different feature maps and extract features from them, even when they are provided in a single stream. This increment of accuracy is unpredictable and depends greatly upon the scene's correlation with the action and the uniqueness of the motion information generated from the action.

Class	Ours	KLT[16]	Wang et. al [19]	Ikizler-Cinbis [9]
B_shooting	**95.6%**	34.0%	43.0%	48.48%
Biking	**93.1%**	87.6%	91.7%	75.17%
Diving	92.8%	**99.0%**	**99.0%**	95.0%
G_swinging	95.0%	95.0%	**97.0%**	95.0%
H_riding	**94.3%**	75.0%	85.0%	73.0%
S_juggling	**87.8%**	65.0%	76.0%	53.0%
Swinging	**92.4%**	86.0%	88.0%	66.0%
T_swinging	**94.9%**	71.0%	71.0%	77.0%
T_jumping	**94.0%**	93.0%	**94.0%**	93.0%
V_spiking	93.2%	**96.0%**	95.0%	85.0%
W_dog	**91.4%**	76.4%	87.0%	66.67%
Accuracy	**93.13%**	79.9%	84.2%	75.21%

Table 1. Comparison of our network with state-of-the-art algorithms. Accuracies reported over each class are compared.

6. Conclusion and Discussion

In this work, we have proposed a deep convolutional network which operates on stacked motion and static information, to classify actions in videos. We use a single stream of motion and static information, and our network model is able to learn to treat static and motion frames as different feature maps and accurately calculate features off them. The process of using separate streams for motion and static information, and late fusion is, therefore, not required. The network is tested on the popular and challenging dataset, Youtube Action Dataset, which contain 11 classes. Our network classified each class with a high accuracy. We also compared our method with three hand-crafted feature methods and showed that our method is superior in terms of classification accuracy.

It is also seen that using optic flows as the input containing motion information is justified, as the network was able to accurately calculate features off optic flow maps. Static information also is important in classifying videos as the environment may have a correlation with the actions. Therefore, adding static information to the motion information is useful and the output accuracy is consistantly increased. However, this increment of accuracy may vary depending on the correlation between the action and the environment.

In future, it would be interesting to see the effect of reducing background motion prior to calculating the optic flows. Also, a slightly higher level and a more accurate representation of motion, e.g., dense trajectories may provide different results. Also, it is fair to assume that applying 3D convolution across feature maps, may recognize more useful features distributed across the temporal axis, and would

Class	Approach 1	Approach 2	Diff
B_shooting	**95.6%**	95.2%	+0.4%
Biking	**93.1%**	91.4%	+1.7%
Diving	**92.8%**	90.5%	+2.3%
G_swinging	**95.0%**	92.6%	+2.4%
H_riding	**94.3%**	**94.3%**	0.0%
S_juggling	87.8%	**88.0%**	−0.2%
Swinging	**92.4%**	90.7%	+1.7%
T_swinging	**94.9%**	92.1%	+2.8%
T_jumping	**94.0%**	93 %	+1.0%
V_spiking	**93.2%**	92.8%	+0.4%
W_dog	**91.4%**	**91.4%**	0.0%
Accuracy	**93.1%**	92.0%	+1.1%

Table 2. Comparison of results: approach 1 vs approach 2. Providing static motion has increased the overall accuracy by +1.1%.

increase the classification results.

7. Acknowledgement

We would also like to thank the National Research Counsil of Sri Lanka, for supporting us for this work, through the grant 12-018.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *Human Behavior Understanding*, pages 29–39. Springer, Berlin Heidelberg, Germany, 2011. 1, 2

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, Graz, Austria, 2006. 2

[3] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, pages 25–36. Springer, Prague, Czech Republic, 2004. 3

[4] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649, Providence, Rhode Island, 2012. IEEE. 1

[5] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2012. 1

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893, San Diego, CA, USA., 2005. IEEE. 2

[7] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision (ECCV)*, pages 428–441. Springer, Graz, Austria, 2006. 2

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 2010. 3

[9] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. *European Conference on Computer Vision (ECCV)*, pages 494–507, 2010. 1, 4

[10] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013. 2

[11] H.-J. Kim, J. S. Lee, and H.-S. Yang. Human action recognition using a modified convolutional neural network. In *Advances in Neural Networks–ISNN 2007*, pages 715–723. Springer, Berlin Heidelberg, Germany, 2007. 2

[12] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1, University of Leeds, Leeds, UK, 2008. British Machine Vision Association. 2

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, Lake Tahoe, Nevada, 2012. 1

[14] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368, Colorado Springs, Colorado, 2011. IEEE. 2

[15] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1996–2003, Fontainebleau Miami Beach, Florida, 2009. IEEE. 1, 2

[16] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Conference on Artificial Intelligence (IJCAI)*, volume 81, pages 674–679, Vancouver, BC, Canada, 1981. 1, 4

[17] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 833–840, Bellevue, Washington, USA, 2011. 1

[18] 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. 2

[19] 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, Colorado Springs, Colorado, 2011. IEEE. 1, 4