ODN: OPENING THE DEEP NETWORK FOR OPEN-SET ACTION RECOGNITION

Yu Shu[1], Yemin Shi[1], Yaowei Wang[2], Yixiong Zou[1], Qingsheng Yuan[3], Yonghong Tian[1*]*

[1] National Engineering Laboratory for Video Technology, School of EE&CS,
Peking University, Beijing, China
[2] School of Information and Electronics, Beijing Institute of Technology, Beijing, China
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

In recent years, the performance of action recognition has been significantly improved with the help of deep neural networks. Most of the existing action recognition works hold the *closed-set* assumption that all action categories are known beforehand while deep networks can be well trained for these categories. However, action recognition in the real world is essentially an *open-set* problem, namely, it is impossible to know all action categories beforehand and consequently infeasible to prepare sufficient training samples for those emerging categories. In this case, applying closed-set recognition methods will definitely lead to unseen-category errors. To address this challenge, we propose the Open Deep Network (ODN) for the open-set action recognition task. Technologically, ODN detects new categories by applying a multi-class triplet thresholding method, and then dynamically reconstructs the classification layer and "opens" the deep network by adding predictors for new categories continually. In order to transfer the learned knowledge to the new category, two novel methods, Emphasis Initialization and Allometry Training, are adopted to initialize and incrementally train the new predictor so that only few samples are needed to fine-tune the model. Extensive experiments show that ODN can effectively detect and recognize new categories with little human intervention, thus applicable to the open-set action recognition tasks in the real world. Moreover, ODN can even achieve comparable performance to some closed-set methods.

Index Terms— Open Set, Action Recognition, ODN, Open Deep Network

1. INTRODUCTION

Deep neural networks have made remarkable achievements in many visual recognition problems. Specifically, the performance of human action recognition has been improved significantly in recent works [1, 2]. However, most of these recognition systems are designed for a static closed world, where

Corresponding author: Yaowei Wang (email: yaoweiwang@bit.edu.cn) and Yonghong Tian (email: yhtian@pku.edu.cn).
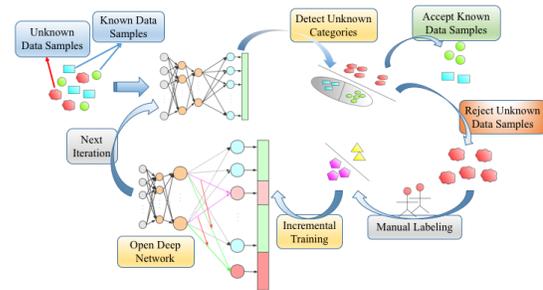
Fig. 1. The workflow of ODN. ODN first trains an initial model on the knowns. In the open-set scenario, ODN detects the unknowns. Then a strong teacher is invoked to label the unknowns. Next, ODN dynamically incorporates the new category in the model. Finally, ODN fine-tunes the model with only few samples to make the unknowns "known".

the primary assumption is that all categories are known as priori. However, action recognition in the real world is essentially an *open-set* problem, namely, it is impossible to know all action categories beforehand and consequently infeasible to prepare sufficient training samples to train the deep networks for those emerging categories. In this case, applying closed-set recognition methods will definitely lead to unseen-category errors. Therefore, an open-set method is more desirable for the real-wolrd action recognition applications.

Technologically speaking, many methods on incremental learning can be used to handle new instances of known categories [3, 4]. However, most of these approaches cannot handle unknowns or dynamically add new categories to the system. In [5], new categories are added according to the mean features. This approach, however, assumes that the number of initial categories is relatively large. An alternative multi-class incremental approach based on least-squares SVM has been proposed by Kuzborskij *et al.* [6]. Each time a category is added, the whole set of the hyperplanes will be updated, which is too expensive as the number of categories grows.

In particular, researches on open set recognition [7, 8, 9] have established formulations of rejecting unknown objects.

Among them, Abhijit *et al.* [9] adapted the concept of Meta-Recognition [10, 11] to deep networks, and proposed an OpenMax method. However, the issue of incremental learning with additional category remains an unsolved problem. Meanwhile, Abhijit *et al.* [12] also proposed a SVM-based recognition system that could continuously learn new object categories by extending the NCM-like algorithms [13] to a NNO algorithm, but it is not applicable in deep networks. Therefore, the open-set recognition with deep networks is still a challenging problem and needs more efforts to be devoted.

In this study, we extend the open world recognition problem in [12] to video action recognition task. Let the knowns labeled by positive integers $\mathcal{N}^+$ and the zero label (0) label the unknowns. And let $\mathcal{K}_t^k \subset \mathcal{N}^+$ be the set of knowns which has k categories at time t. Let the known category number be N in the initial. Thus a typical network for the open-set action recognition problem should include the following four steps:

1. **Multi-class unknown category detection:** A detector $\gamma(\varphi): \mathcal{R}^d \mapsto [0, 1]$ is needed to determine whether the current data sample φ, which corresponds to features $x \in \mathcal{R}^d$, is from an unknown (0) category or not.

2. **Unknown category labeling:** $\mathcal{R}^d \mapsto \mathcal{N}^+$. A strong teacher (human in general) is invoked to identify the unknown categories, which are detected in the previous process, by assigning a new label. Assuming the detection process finding m new categories up to time t, the set of known categories becomes $\mathcal{K}_{t+1} = \mathcal{K}_t \cup \{N+1, N+2, \ldots, N+m\}$.

3. **Open Deep Network updating:** Recognition process : $\mathcal{P}(\varphi): \mathcal{R}^d \mapsto \mathcal{K}^k$ at time t. Then the network is updated by adding one category in the classification layer at time $t + 1$. In this case, the recognition process is expressed as $\mathcal{R}^d \mapsto \mathcal{K}^{k+1}$.

4. **Incremental training:** The network with $\mathcal{P}(\varphi): \mathcal{R}^d \mapsto \mathcal{K}^{k+1}$ is incrementally fine-tuned with few labeled samples for the new category.

We can see that in the open-set action recognition problem, the major challenges lie in discovering unknown categories and dynamically opening and updating the deep network.

To address these challenges, this paper proposes the Open Deep Network (ODN) for open-set action recognition with deep networks. Fig. 1 illustrates the workflow of ODN which gradually makes each "unknown" to "known". Extensive experiments on the HMDB51, UCF50 and UCF101 benchmark datasets show that ODN can effectively detect and recognize new categories with little human intervention.

The main contributions are summarized as follows:

1. A network dynamical reconstruction method (ODN) is proposed to support the task of open-set recognition. The novel architecture, ODN, applies a multi-class triplet thresholding method to detect unknown

categories and simultaneously transfers the knowledge from known categories to new categories to quickly make "unknown" to "known".

2. An evaluation protocol is developed for the open-set action recognition task. The experiment results show that ODN is able to achieve comparable performance to the closed-set setting while few new samples are needed.

The rest of this paper is organized as follows: Section 2 simply reviews the related works on incremental learning, open-set learning and action recognition. Section 3 presents our ODN and Section 4 describes its implementation details and experiments. Finally, Section 5 concludes this paper.

2. RELATED WORKS

Incremental learning. Many incremental methods based on SVM were proposed in these years. Cauwenberghs *et al.* [14] proposed an incremental binary SVM by means of saving and updating KKT conditions. Yeh *et al.* [4] extended the incremental binary SVM to object recognition and demonstrated multi-class incremental learning. Pronobis *et al.* [15] proposed a memory-controlled online incremental SVM which combined an approximate technique [16]. Some other multi-class incremental learning methods [17, 18] are trying to increase the number of training samples instead of categories.

Open-set learning. Open-set recognition assumes there are both known and unknown data in the test phase, considering that the knowledge of categories in the real world is not available. Recent works on open-set recognition [7, 8, 9] formalized processes of rejecting unknown objects in open-set recognition. In [12], Abhijit *et al.* proposed a recognition system that suits the open world scenario but is not applicable in deep networks.

Action recognition with deep networks. In recent years, CNNs have achieved state-of-the-art performance on various tasks (e.g. [19, 20]) and it has been proven that features learned from CNNs are much better than the hand-crafted features [21]. Many works [22, 23] have been done to transfer CNNs to video tasks and have made significant progress. The two-stream network [24] is the most widely used baseline video classification model, which has both spatial and temporal networks, ultimately achieves the optimal performance. The latest works [1, 2] have achieved fantastic performance. However, these works are designed for the closed-set scenario, and how to open the deep networks and dynamically add new categories still remain unsolved.

3. OPEN DEEP NETWORK

In this section, we will describe the Open Deep Network (ODN) for open-set recognition. The methodology is composed of four parts, which will be elaborated in the following subsections.

3.1. Multi-class unknown category detection

The first step of ODN is to detect the unknowns, considering as rejecting unknowns during the open-set recognition. Scheirer *et al.*[10] defined the open space risk. When adopting deep networks to handle the open-set recognition, it is necessary to make sure that the open space risk is minimized. But it only gives a general definition and does not specify the method of measuring the distance or which space to measure in. In this paper, we measure the distance in the classification layer of the network. And in the appendix, we proof that ODN can manage the open space risk.

A natural approach for detecting unknown categories is to apply a threshold on the SoftMax's output. However, we found it not effective in the exploration experiment. Because different categories will have different activation range, a single threshold is not enough for handling these differences. Human action is a composite concept and is the composition of a series of atomic behavioral characteristics. Different human actions may contain several same atomic behavioral characteristics, and these relations may still retain in the activation level. Digging the connected information among the activation values can helps to detect unknown categories.

Based on above discussion, we propose a multi-class triplet thresholding method combining the inter-class association for detecting unknown categories, which captures the way in which different categories vary in the activation level.

We calculate a triplet threshold $[\eta, \mu, \delta]$ per category, i.e. accept threshold η, reject threshold μ and distance-reject threshold δ. Define $x_{i,j} \in \mathcal{X}$, where $\mathcal{X} = \{x_{i,j} | x_{i,j}$ is the jth correctly classified sample of category i} and the total samples number of $\mathcal{X}$ is X. Let $F_{i,j} = v_i(x_{i,j})$, $S_{i,j} = \textbf{Second}(\vec{v}(x_{i,j}))$, where v represents the activation vector of the data sample and the function $\textbf{Second}(\vec{v}(x))$ returns the second maximal confidence value of the input vector $\vec{v}(x)$, ε and ρ are empirical parameters. Then the triplet threshold $[\eta_i, \mu_i, \delta_i]$ is calculated as

$$\eta_i = \frac{1}{X} \sum_{j=1}^{X} F_{i,j} \tag{1}$$

$$\mu_i = \varepsilon * \eta_i \tag{2}$$

$$\delta_i = \rho * \frac{1}{X} \sum_{j=1}^{X} (F_{i,j} - S_{i,j}) \tag{3}$$

During the detecting phase, we accept a data sample as category label l only if the index of its top confidence value is l and the value is greater than η_l. We consider a sample unknown when all of its confidence value is below μ. And the statistical properties of δ include correlation information between the two categories, which is a simple way of using the inter-class relation information in the activation level. We use δ to help detect unknown category in hard samples, which lie between η and μ. If the distance is large enough, then we accept the data sample as category label l.

3.2. Unknown category labeling

A labeling process is applied to the unknown samples detected in the former process. We propose several methods to transfer the learned knowledge to the new categories, hence, only a few new samples will be needed. For some difficult unknown categories, detection process can be repeated to detect more training samples. On average, we only need several manually annotations.

3.3. Open Deep Network updating

The open-set recognition requires to update the system to add these new categories found in the multi-class unknown categories detection. One can always retrain the entire system using the known data and new samples, while this solution not only consumes a lot of time and computational resources, but also is easy to be over-fitting because of lacking enough training samples. We propose an updating method by transferring knowledge from the trained model. Our approach helps speed up the training stage and need very few manually annotations.

$$\begin{bmatrix} w_1 & \mathbf{w_2} & w_3 & \dots & \mathbf{w_t} & \dots & w_N \end{bmatrix} \qquad (I)$$

$$\begin{bmatrix} w_1 & \mathbf{w_2} & w_3 & \dots & \mathbf{w_t} & \dots & w_N & w_{N+1} \end{bmatrix} \qquad (II)$$

As shown in (I), the model at time t has N categories corresponding to N columns in the transfer weight. The weight contains the knowledge learned in training phase. To open the deep network, we need to add a new column corresponding to the unknown category which we choose to add at this iteration. As show in (II) above, in order to transfer knowledge from the current model, we initialize the w'_{N+1} :

$$w'_{N+1} = \frac{1}{N} \sum_{n=1}^{N} w_n \tag{4}$$

where w_n is the weight column of the nth category.

Considering that human action is a composite concept, the actions are interrelated. Though the action itself is different, some of the action elements which constitute the action may be the same. So the similar categories should play a more critical role in the initialization of w_{N+1}. We choose the weight columns of M highest activation values as w_m to form the set $\mathcal{H}$. And we define:

$$w''_{N+1} = \frac{1}{M} \sum_{m=1}^{M} w_m \tag{5}$$

The w''_{N+1} emphases the similar known categories. Though the unknown category has low activation value for each known category, the weight columns corresponding to higher activation value should be emphasized in initialization. Finally, we blend the overall knowledge and similar emphasis

from the current model as:

$$w_{N+1} = \alpha \frac{1}{N} \sum_{n=1}^{N} w_n + \beta \frac{1}{M} \sum_{m=1}^{M} w_m \qquad (6)$$

The w_{N+1} is then concatenated to the transfer weight W to support the new category. And we call this initialization method **Emphasis Initialization**.

3.4. Incremental training

After opening the deep networks, ODN fine-tunes the model using both known and new category samples. As knowledge is transfered from the trained model, it's still feasible in large scale application. To reduce the influence on accuracies of knowns, we adopt a balanced training strategy. We guarantee that each of the known categories has the same number of samples as the new category for fine-tuning, and the known samples can be get from the training set. Because the number of new category samples is small, the entire fine-tuning process finishes quickly.

As for the knowns, the system has been trained before. It is more efficient to train the model at different rates for known and new categories. ODN adopts a different learning rate decay matrix, which varies between known categories and new categories to force new categories learn much faster than known categories during the fine-tuning. We call this method as **Allometry Training**. It is quite helpful to reduce performance jitter on accuracies of knowns recognition. We define allometry factor α as:

$$\alpha_i = \begin{cases} 0.1, & i <= N \\ 1, & i > N \end{cases} \qquad (7)$$

When the modified weight transfer matrix is updated with the back propagation algorithm, we implement the allometry factor as learning rate matrix.

The Allometry Training is inspired by the fine-tuning strategy of CNNs. The main difference is that Allometry Training applies different learning rate for different elements in the same matrix.

4. EXPERIMENTS

This section will first introduce the details of datasets. Then, we describe the implementation details. Finally, we report the experimental results and give the analysis of results.

4.1. Datasets

To verify the effectiveness of our methods, we conduct experiments on three benchmark datasets: HMDB51 [25], UCF50 [26] and UCF101 [27].

The HMDB51 dataset contains 6849 clips divided into 51 action categories, each containing a minimum of 101 clips.

The UCF50 dataset is an action recognition dataset with 50 action categories, consisting of realistic videos taken from youtube. The UCF101 dataset is one of the most popular action recognition benchmarks. It contains 13,320 video clips from 101 action categories and there are at least 100 video clips for each category. We conduct experiments on the first split of HMDB51 and UCF101.

4.2. Experiments setting

Training Phase: The training procedure of ODN has two phases. First, in the initial training phase, known data samples are provided for the CNN to train a classifier with the knowns. Once the initial training phase is completed, the system will use the trained model to get activation values for detecting unknown-category sample process. Second, in the incremental training phase, labeled new category is added to the system one-by-one. And to balance the incremental training phase, we ensure that the sample number of each known category is the same as that of the unknown category.

Testing Phase: To ensure proper evaluation of ODN, we test it on a set of known and unknown samples which none of them appear in the training phase.

Also, to compare with the traditional closed-set action recognition work, we carried out the experiments on HMDB51 split1 and UCF101 split1, which means we will use much fewer training samples than any other closed-set methods while testing on standard test set.

4.3. Implementation details

To focus on Open Deep Network and simplify the training process, we use a very simple CNN implementation, GoogLeNet [19], to give a set of exploration experiments. Finally, we give the results on Inception-resnet-v2. In multi-class open set recognition process, we set some empirical parameters. And in our method **Emphasis Initializion**, we choose 5 corresponding weight columns of similar known categories to emphasize the initiation of the unknown category weight. The details are described in the appendix.

4.4. Evaluation protocol

Testing on initial model: After the initial training phase, we test the initail model on the known samples. And the accuracy on unknown test samples should be 0.

Testing during the incremental training phase: After each iteration of the incremental training phase, we test on the entire set of both known and unknown categories. To make the evaluation more rigorous, we treat all samples in the unknown categories by this stage as misidentified.

Testing phase for the whole system: Finally, we give the overall performance of the model on the entire test set of both known and unknown categories, and report the accuracy of each category.
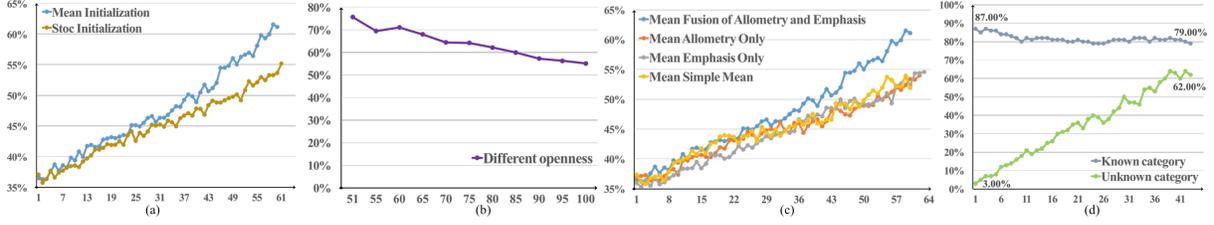
Fig. 2. (a)The comparison between Mean Initialization and Stochastic Initialization during the updating phase. (b)The accuracy of both known and unknown samples with different openness. (c)The illustration of testing accuracy on both known and unknown categories at each iteration. (d)The comparison between the accuracies of known categories and unknown categories at each iteration of the incremental training phase.
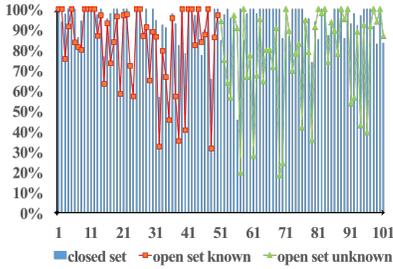
Fig. 3. The comparison between the closed-set and open-set recognition for each category.

4.5. Results and Analysis

In this section, we will report the results on three datasets, HMDB51, UCF50 and UCF101, and give a detailed analysis.

We carry out a set of exploration experiments on dataset UCF101, as Fig.2, to prove the effectiveness of ODN. We compare the mean initialization method and stochastic initialization method while initializing the new transfer weight. As Fig.2(a), we can see the mean initialization method is better for almost all the time, especially when the categories grow larger, for the stochastic initialization would drop lots of knowledge which the system has learned. In the rest of this paper, we will use mean initialization. Fig.2(b) shows the system's performance of different openness. We keep the initial model number be 50, half of the category number of UCF101, then let the open world have 1, 5, 10 unknown categories and so on. Along with the unknown categories grow larger, the openness is larger, which means the task is harder. In our evaluation protocol, we should choose half of the dataset categories as known and the other half as unknown. Because this is the most fair setting to evaluation open set models, reducing unknown categories will make the task easier, and increasing lots of unknown categories, one can easily reject most of the input samples as unknown to achieve better performance.

In Fig.2(c) we verify our Emphasis Initialization method and Allometry Training method separately, Emphasis only and Allometry only each improve the performance of the base

Table 1. Comparison of ODN to the state-of-the-art closed-set methods. Half of the training set is used to train ODN while full training set is used to train other models.

Model	UCF50	Model	HMDB51	UCF101
Var0[28]	81.03%	STIP[25]	23.0%	43.9%
bIDT[22]	88.57%	CNN[29]	26.9%	63.3%
iDT[30]	91.2%	RGB[24]	40.5%	73.0%
ODN	**93.73%**	ODN	**46.01%**	**76.07%**

method Simple mean. In Fig.2(d), we show the accuracy of known categories and unknown categories separately. The accuracy of known is decreased, for adding new categories increases the jitter of model. But the accuracy of unknown categories achieves great improvement as the model knowing more categories.

We also compare with the closed-set recognition. A detailed comparison between the closed-set recognition and ODN is showed in Fig.3, we test the accuracy of each category both on the closed set and open set. It's amazing that ODN is able to achieve comparable performance with the closed-set recognition, while we only need a teacher to label 5.39 samples on average for each unknown categories, which are detect by ODN. It is much less than averagely 94.4 samples in the closed set recognition. The details of calculating the number of labeling samples are described in the appendix.

4.6. Comparison with the state-of-the-art methods

To prove the effectiveness of ODN, we also compare it with several closed-set state-of-the-art action recognition methods in Table1. Even though using much less training data, our ODN is still able to outperform these closed-set methods.

5. CONCLUSION

In this paper, we propose an Open Deep Network (ODN) for open-set action recognition. In ODN, we propose a multi-class triplet thresholding method for unknown category de-

tection, and Emphasis Initialization and Allometry Training methods to transfer the learned knowledge from the trained model and speed up the fine-tuning phase. Experimental results show that, our ODN can effectively detect and recognize new categories with little human intervention. In the future work, we will further improve the steadiness of our network, and explore its application in a wider range of applications such as abnormal event detection.

6. REFERENCES

[1] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*. Springer, 2016, pp. 20–36.

[2] Yemin Shi, Yonghong Tian, Yaowei Wang, Wei Zeng, and Tiejun Huang, "Learning long-term dependencies for action recognition with a biologically-inspired deep network," in *CVPR*, 2017, pp. 716–725.

[3] Amund Tveit and Magnus Lie Hetland, "Multicategory incremental proximal support vector classifiers," in *KES*. Springer, 2003, pp. 386–392.

[4] Tom Yeh and Trevor Darrell, "Dynamic visual category learning," in *CVPR*. IEEE, 2008, pp. 1–8.

[5] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *ECCV*, pp. 488–501. Springer, 2012.

[6] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo, "From n to n+ 1: Multiclass transfer incremental learning," in *CVPR*. IEEE, 2013, pp. 3358–3365.

[7] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult, "Toward open set recognition," *PAMI*, vol. 35, no. 7, pp. 1757–1772, 2013.

[8] Walter J Scheirer, Lalit P Jain, and Terrance E Boult, "Probability models for open set recognition," *PAMI*, vol. 36, no. 11, pp. 2317–2324, 2014.

[9] Abhijit Bendale and Terrance E Boult, "Towards open set deep networks," in *CVPR*, 2016, pp. 1563–1572.

[10] Walter J Scheirer, Anderson Rocha, Ross J Micheals, and Terrance E Boult, "Meta-recognition: The theory and practice of recognition score analysis," *PAMI*, vol. 33, no. 8, pp. 1689–1695, 2011.

[11] Peng Zhang, Jiuling Wang, Ali Farhadi, Martial Hebert, and Devi Parikh, "Predicting failures of vision systems," in *CVPR*, 2014, pp. 3566–3573.

[12] Abhijit Bendale and Terrance Boult, "Towards open world recognition," in *CVPR*, 2015, pp. 1893–1902.

[13] Marko Ristin, Matthieu Guillaumin, Juergen Gall, and Luc Van Gool, "Incremental learning of ncm forests for large-scale image classification," in *CVPR*, 2014, pp. 3654–3661.

[14] Gert Cauwenberghs and Tomaso Poggio, "Incremental and decremental support vector machine learning," in *NIPS*, 2001, pp. 409–415.

[15] Andrzej Pronobis, Luo Jie, and Barbara Caputo, "The more you learn, the less you store: Memory-controlled incremental svm for visual place recognition," *Image and Vision Computing*, vol. 28, no. 7, pp. 1080–1097, 2010.

[16] Nadeem Ahmed Syed, Syed Huan, Liu Kah, and Kay Sung, "Incremental learning with support vector machines," 1999.

[17] Fayin Li and Harry Wechsler, "Open set face recognition using transduction," *PAMI*, vol. 27, no. 11, pp. 1686–1697, 2005.

[18] Zhuang Wang, Koby Crammer, and Slobodan Vucetic, "Multiclass pegasos on a budget," in *ICML*. Citeseer, 2010, pp. 1143–1150.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al., "Going deeper with convolutions," CVPR, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.

[21] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *CVPR*. IEEE, 2005, vol. 1, pp. 886–893.

[22] Yemin Shi, Wei Zeng, Tiejun Huang, and Yaowei Wang, "Learning deep trajectory descriptor for action recognition in videos using deep neural networks," in *ICME*. IEEE, 2015, pp. 1–6.

[23] Shengxin Zha, Florian Luisier, Walter Andrews, Nitish Srivastava, and Ruslan Salakhutdinov, "Exploiting image-trained cnn architectures for unconstrained video classification," *arXiv*, 2015.

[24] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014, pp. 568–576.

[25] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre, "Hmdb: a large video database for human motion recognition," in *ICCV*. IEEE, 2011, pp. 2556–2563.

[26] Mubarak Shah Kishore K. Reddy, "Recognizing 50 human action categories of web videos," in *MAVP*, 2012.

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv*, 2012.

[28] Sinisa Todorovic, "Human activities as stochastic kronecker graphs," in *ECCV*, pp. 130–143. Springer, 2012.

[29] Sreemanananth Sadanand and Jason J Corso, "Action bank: A high-level representation of activity in video," in *CVPR*. IEEE, 2012, pp. 1234–1241.

[30] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *ICCV*. IEEE, 2013, pp. 3551–3558.