Dynamic Image Networks for Action Recognition: A Critical Review

Seby Jacob
260656219
seby.jacob@mail.mcgill.ca

***Abstract*— A dynamic image is a novel concept in abstracting the contents of a video in a single RGB image. Following a brief introduction on dynamic images, their strength as a tool for video representation is assessed by their performance in action-recognition on videos from the UCF-101 dataset. Dynamic images generated from videos are trained and tested by adopting various standard convolutional neural network architectures.**

I. Introduction

Human poses and actions have been of great interest since the 15^{th} century when Leonardo Da-Vinci began to study and teach the nuances of the subject for accurate reconstruction. Human action recognition is one of the most important problems of our time. It is integral for driving content based search in videos, home-entertainment, sports analysis, and most of all, automatic surveillance and security systems. With the towering amount of video-data recorded everyday, we need algorithms to understand and interpret this data automatically. Action recognition however, is quite challenging due to the large intra-class variations, low video resolution and the high dimensionality of video data. Currently, the state of the art in action recognition from the *UCF-101* dataset is at 91.4% accuracy [2].

Recent years have witnessed great progress in action recognition. Most of the research in action recognition can be broadly classified into two categories. The first category focuses on representing videos as handcrafted features and Bag of Visual Words (BoVWs). The second category of research employs deep convolutional neural networks to generalize video representations from raw video frames to train recognition systems end-to-end. Two-stream ConvNets [5] is the best performing action recognition method using deep learning. Nonetheless, the most effective video representations to exercise deep convolutional networks is still an open problem. With hope to address this issue, I present here a review on a novel dynamic image encoding of videos for action recognition.

II. Related Work

This section provides a brief summary on action recognition literature.

A. Types of Feature Representations

Feature representations for action-recognition tasks must ideally be robust to viewpoint, background and camera-angle variations, while still preserving enough discriminatory information to determine the action.

1) Sampling: Critical information can be sampled from videos in three main ways, namely, dense sampling, regular sampling and sparse sampling. Dense sampling gathers information from all the pixels in a video. Sparse sampling methods extract interest points or interest regions like corners, blobs, edges, etc when they occur in 3-D space-time. Harris 3D detector [6] and Hessian detector [7], [8], detect spatio-temporal corners and blobs from the raw image data by estimating local maxima of simple gradient functions. These methods perform only in datasets with repeated actions with less background variance like the KTH dataset. Camera motion also jeopardizes the recognition performance of such detectors. In order to counteract the effects of camera movement, *Wang and Schmid* proposed dense trajectories to track features of interest across frames [9]. They used *SURF* descriptors and optical flow to track interest points across the temporal dimension in order to estimate the camera motion. The matches between keypoints are used to apply homographic transformations on subsequent frames to rectify the effects of camera motion. Using the rectified frames, densely sampled key-points are encoded in fisher vectors [3] and trained in a linear SVM classifier. The method achieved state-of-the-art performance in human activity recognition (*Hollywood2*: 64.3%, *HMDB51*: 57.2%, *Olympic Sports*: 91.1%, *UCF50*: 91.2%).

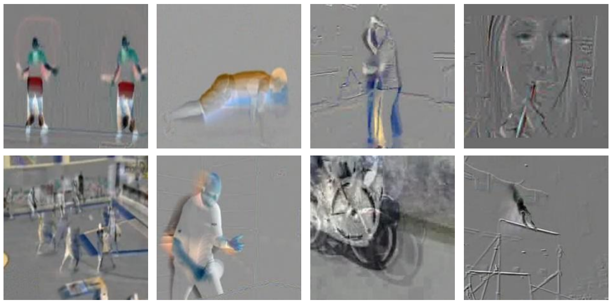

Fig. 1: A few examples of dynamic images obtained from temporal rank pooling of RGB frames in videos. ***Top row*** *(L to R): Jump Rope, Push-ups, Golf Swing, Applying Make Up.* ***Bottom Row*** *(L to R): Basketball Shooting, Table Tennis Shot, Biking, Gymnastics.*

Even though computationally intensive, the dense sampling methods outperform the sparse samplers in classification accuracy. This is due to the fact that dense sampling encodes contextual information about the background as well (eg. Basketball is usually played in a basketball court).

2) Feature Descriptors: Feature descriptors can be split into two categories, namely general primitive features and specialized primitive features. General primitive features is a direct transformation of raw video input whereas specialized primitive features requires pre-processing into auxiliary features before classification.

A variety of descriptors that encode features using intensity gradients have been implemented. HOG (Histogram of Oriented Gradients) is used in many 2-D image detection and classification tasks. HOG3D by *Klaser and Schmid* extends 2-D HOG into the spatio-temporal domain in an attempt to wrap local features and motion into a single package [11]. Although such adaptations do help in gathering crucial action information, they can be very

sensitive to variations in contrast, lighting and video quality.
Histograms of optical flow can be used to describe how pixels flow from one frame to the next. Histograms of Optical Flow (HOF) encodes local motion information by quantizing the orientations of optical flow vectors. This method succeeds in differentiating between actions in two ends of the spectrum (e.g wave and walk) but fails in discriminating between actions that appear similar (e.g box and clap). Hence, combinations of HOF to capture motion and HOG to capture appearance information performs better. Motion Boundary Histograms (MBH) describe actions by binning the gradients of optical flow vectors. This is effective in suppressing camera motion but most texture information and background context is discarded.

Since *Alex Krizhevsky* reintroduced Convolutional Neural Networks (CNNs) for image classification [15], they have been extensively adopted in the realm of action recognition. Most of the work using CNNs for action recognition consider videos as a stream of raw image pixels [16] or optical flow displacement fields [17]. It is notable that the optical flow displacements perform better than the raw image pixels of the videos. Research has then extended to using Recurrent Neural Networks and Long Short Term Memory units for enabling the network to activate on temporal triggers in the video frames.
The next checkpoint is the two-stream model [19] that separates the appearance and the motion parts of the video (much like HOG and HOF as described earlier). To increase the performance of a two-stream model, residual connections between the two streams have also been proposed. Even though CNNs are currently leading the state-of-the-art in image recognition tasks, they require enormous datasets and extensive computation time to train. To circumvent these requirements, we can use networks pre-trained for image classification. This however, could constraint the recognition power only to appearance rather than motion.

In terms of using specialized primitive features, Motion Energy Images (MEI) and Motion History Images (MHI) can be calculated using the difference between silhouettes in contiguous frames. These store the spatial locations and the recency of motion. Since these algorithms are sensitive to the object's displacement and orientation, we can use a concept of introducing motion history volumes using depth images. Nevertheless, this method is hard to use on the widely available video data we intend to perform recognition tasks on.
Since CNNs and specialized primitive features are strong contenders in action recognition, *Hakan Bilen and Andrea Vedaldi* [20] introduced dynamic images for action recognition. This idea uses temporal rank pooling on video frames to generate a single RGB image that captures the appearance and temporal motion properties of a video. This dynamic image so generated is used in classification by using a pre-trained AlexNet re-purposed for dynamic images. This method is much easier to train than 3D convolutional networks and proves to be quite efficient in action recognition while being exceptionally fast.

B. Current Status

Before the introduction of CNNs for action recognition, iFV-encoded iDT features with HOG, HOF, and MBH descriptors using a linear SVM was topping the accuracy with 57.2% and 85.9% on HMDB51 and UCF-101 datasets, respectively [4]. With more dimensions and more computational power, *Peng and Wang* [21] increased the accuracy, albeit with little improvements (3.9% and 2% respectively). Its quite interesting to note that even simple linear classifiers can achieve significant results towards good recognition.

The dual stream CNN approach using raw image data and optical flow as discussed earlier, achieved comparable results to hand-crafted features raising it to the pole-position of candidate methodologies for action recognition. The top-tier in both UCF-101 and HMDB51 is occupied by a method combining the high-dimensional hand-crafted features and the dual-stream CNNs that share residual connections [22]. The classic dual-stream method, even though accurate, takes an impractically long time to classify each sample, since calculating optical-flow between frames is quite complex. An alternative approach uses coarser optical flow encoders that slightly degrade the accuracy. This loss of accuracy is compensated by an algorithm that is 27 times faster [23]. CNNs promise even better accuracy if a high-volume dataset is provided for training. However, collecting a quality, high-volume, annotated video dataset to train CNN architectures is quite overwhelming.

Even though vision algorithms have made great strides in action recognition, its accuracy can hardly be compared to humans. Enough visual clarity in videos enable humans to easily classify actions with minimal effort, irrespective of variations in viewpoint, backgrounds, occlusions, etc. Algorithms that deliver 90% accuracy on UCF-101 may point towards robustness in viewpoint variations, but it is key to note that these algorithms only achieve around 65% in slightly less constrained datasets like HMDB51. This deems viewpoint variation still an open problem.

III. Problem Representation

Deep convolutional neural networks have demonstrated their power in many image recognition and detection tasks [15]. This power makes them the premier choice for any image related tasks in the artificial intelligence spectrum. In many recent works, researchers have tried to transfer their virtue on images to videos. This choice is obvious, since videos are streams of RGB image frames. However, to exploit CNNs for action recognition from videos, we need an effective encoding of video features valid in a CNN context. We could train the networks on raw video frames in a temporal fashion, but it is painfully slow and cannot achieve results comparable to state-of-the-art. Dynamic images [20], introduced by *Hakan Bilen* and *Andrea Vedaldi* prove to be a very potent representation of videos that can be trained easily using existing CNN architectures with minimal effort.

A. Dynamic Image

A dynamic image is a single RGB image that summarizes the complete spatio-temporal context of a video in a single RGB image. A video is converted into a dynamic image by applying temporal rank pooling on its individual frames. The authors of [20] provide in-depth discussion of temporal-rank-pooling and an approximate pooling operator that enables end-to-end training by back-propagation. The approximate rank pooling operator can be added as a layer in the network architecture as well. The choice of the optimal location of the rank-pooling layer in a network will be discussed in IV. In this work, I want to evaluate the effectiveness of dynamic image representations of videos in CNNs to solve the action recognition problem. *Figure 1* contains a few examples of dynamic images. It is key to observe that a dynamic image captures only moving elements of a video while static pixels are averaged away by the temporal rank pooling operation.

B. *Dataset*

To evaluate the performance of dynamic images on action-recognition tasks, I will use the UCF-101 dataset [24], one of the popular benchmark datasets in literature. This dataset consists of 101 human actions distributed over 13,320 video clips. All the videos are sourced from YouTube and manually clipped to include the entire action while minimizing interference from frames irrelevant to the action class. Each action class consists of 25 groups, which in-turn has 4 to 7 clips. The clips within each group are similar in some respect (e.g. background, actors, etc). The dataset has a mean clip length of 7.21 seconds. All videos have a frame-rate of 25 fps and a fixed resolution of 320 × 240. Three different train-test splits of 73-27% (approx) is used to avoid randomness in the experimental setup. As previously mentioned, the clips within each group are similar. Therefore, clips from the same group should be used exclusively either in training or testing. The videos are converted into RGB image frames at their maximum frame rate (25 fps) and re-sized from 320 × 240 to 256 × 256 to be consistent with the existing CNN architectures in a pre-processing step.

C. *Evaluation Metrics*

The dataset consists of 101 different classes of human actions in the wild. Undeviating from current practice, the recognition performance from each algorithm will be evaluated using a mean average precision *(mAP)* over all action classes. This will give us an unbiased estimate of the algorithm performance on all classes and is easily comparable to the results reported in literature.

IV. Algorithm Selection and Implementation

In this section, I discuss the various choices made for experimentation including methodology, network-architectures and the number of dynamic images per video.

A. *Experimentation Methodology*

In their work, Hakan Bilen *et. al* utilize a pre-trained AlexNet [15] fine-tuned for dynamic images for action recognition on the UCF-101 and HMDB51 datasets. A pre-trained AlexNet has been trained on 1.2 million images for image classification. The authors emphasize the power of transfer learning, where a network trained for an auxiliary task can be fine-tuned for good performance on the primary task. For fine-tuning, the gradient descent learning rate is kept very low (α = 0.001) so that the weights previously learned by the network are only adjusted ever so slightly to generalize for the main task.

The authors discuss and evaluate two methods of training the network with dynamic images, namely, Single Dynamic Images (SDI) and Multiple Dynamic Images (MDI). SDI generates a single dynamic image from rank-pooling all the frames of a video to be used as input to the AlexNet, whereas MDI divides the video into random sub-samples to generate multiple dynamic images. They experimentally determined that MDI performs better than SDI by a considerable margin. The results are reported in *Table I*. The best results achieved by the paper is an mAP of 89.1 on UCF-101 using an ensemble of 3 methods; namely MDI end-to-end, Static RGB and dense trajectories [9]. To evaluate the efficacy of dynamic images in video representation, I only experiment on MDI end-to-end training.

The approximate rank pooling method introduced in [20] enables back propagation training while being 45 times faster than the classic SVM formulation of temporal rank pooling. This increase in training speed compromises the mAP on the dataset only by a mere 2.5 points. For these reasons, I employ approximate rank pooling as a layer in the network architecture. Experiments on the placement of rank-pooling layer in the network architecture conclude that the best performance is achieved on rank-pooling the raw RGB frames. The results of their experiments are presented in *Table II*. Based on these results, my experiments place the rank-pooling layer as the first layer that succeeds the frame input layer.

Method	**mAP**
SDI	57.0
MDI	70.9
MDI end to end + Static RGB	76.9
MDI end to end + Static RGB + dense trajectories	89.1

TABLE I: mAP on various methods using Dynamic Images on UCF-101 as reported in [20].

RankPool Layer	**HMDB51**	**UCF-101**
After Raw Image Frames	35.8	70.9
After conv1	-	67.1

TABLE II: mAP from experiments on RankPool Layer Placement using MDI [20].

B. *MDI Sampling*

In the Matlab implementation of the original work [20], multiple dynamic images are generated from one video by randomly sub-sampling 10 contiguous image frames. This is only sub-optimal for the recognition task at hand since important frames that may contain the crux of the video may never be picked for pooling. From here on, this method will be referred to as random sample MDI. Intuition suggests multiple dynamic images to be generated using overlapping frame sub-samples so that important frames will not be missed. To achieve this, full length MDI sampling is used, where multiple dynamic images are generated using a moving window over the frame samples with a stride of 6 frames. Recognition precision on UCF-101 using both methods are compared.

C. *Transfer Learning*

Transfer learning is a method of employing a network pre-trained for an auxiliary task after fine-tuning it for the main task. Since videos are spatio-temporal streams of images, we can use pre-trained image classification networks for action recognition. This follows the intuition that the optimal filters and weights learned for image classification should not be far from their optimum values for action recognition. In the Dynamic Images paper, the authors use a pre-trained AlexNet [15] to evaluate the competence of dynamic images.

AlexNet is a fairly compact (5 convolutional layers, 2 fully connected layers) deep convolutional neural network that achieved state-of-the-art results in image classification in the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. To evaluate the power of transfer learning, the results from using dynamic-images on a trained AlexNet and an un-trained AlexNet are presented and compared.

Since 2012, many popular network architectures have been published. VGGNet [25] is one such deep convolutional neural network (10 convolutional layers, 3 fully connected layers). VGGNet outperforms AlexNet in ILSVRC by 9% in accuracy, and should thereby, give us better precision in UCF-101 using dynamic images. This premise is also fully tested and the results are presented in V. Both network architectures are illustrated in *Figure 2* for comparison.

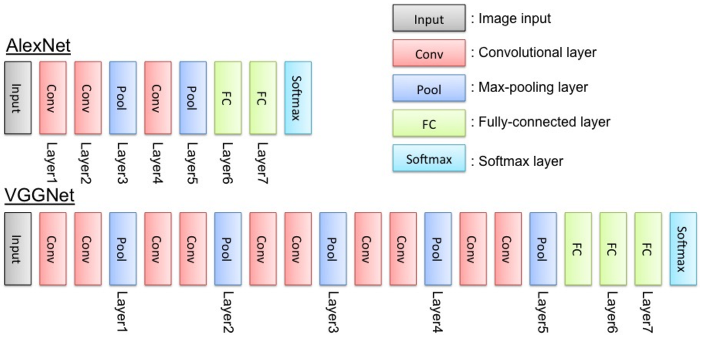

Fig. 2: Illustration of VGGNet and AlexNet architectures.

V. Experiments

This section presents experimental results from various CNN architectures using MDI. Training curves and results of all experiments are laid out.

A. Random Sampling MDI on a Pre-Trained AlexNet

We can use the results from the methodology described in the paper [20], as a base-line for the experiments that follow. For this experiment, a pre-trained AlexNet from the *MatConvNet* [26] repository was fine-tuned and optimized for the UCF-101 dataset using Random Sample MDI. The validation error saturates between training epochs 20 and 30, as seen in *Figure 3*. While the paper reports a precision of 70.9, I get a best case precision of 66.75 using the same methods and parameters.

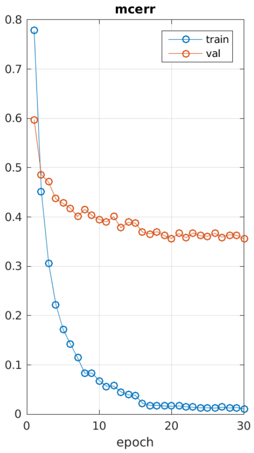

Fig. 3: Multi-class Error vs Number of epochs on MDI (Random Sampling) on pre-trained AlexNet

B. Full Length Sampling MDI on a Pre-Trained AlexNet

On a pre-trained AlexNet, Multiple Dynamic Images were generated using the full length sampling technique as discussed before in IV-B. The intuition behind full length sampling is to avoid missing important frames in training the network. However, the results are worse than the random-sampling baseline (*Table III*). It is evident from *Figure 4* that the training saturates much faster (epoch 20).

Experiment	**MI***	**Paper**
MDI (Random Sampling) on Trained AlexNet	66.75	70.9
MDI (Full Length) on Trained AlexNet	60.75	N.A
MDI (Random Sampling) on Un-trained AlexNet	66.82	N.A
MDI (Random Sampling) on Trained VGGNet	**70.3**	N.A

*TABLE III: Experiments on different algorithms and the mAP on UCF-101. ***(MI-My Implementation)***

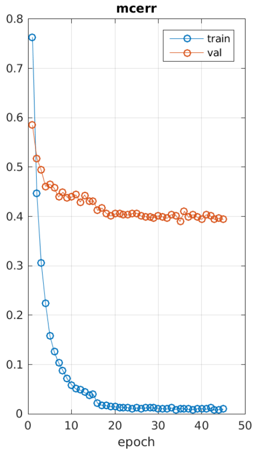

Fig. 4: Multi-class Error vs Number of epochs on MDI (Full Length Sampling) on pre-trained AlexNet

C. Random Sampling MDI on an Un-Trained AlexNet

Since the paper tries to take advantage of transfer learning, the difference between fine-tuning a pre-trained network and training the architecture from scratch needs to be evaluated. An AlexNet was trained from scratch for this task, after randomly initializing the weights and filters. The network takes longer than a pre-trained network to stabilize. While the pre-trained network saturates around epoch 25, the fresh network asymptotes at epoch 35. It is quite interesting to note that the results are very similar. In fact the previously un-trained network performs slightly better. From *Table III*, it can be observed that the trained network has an mAP of 66.75 while the un-trained network has an mAP of 66.82.

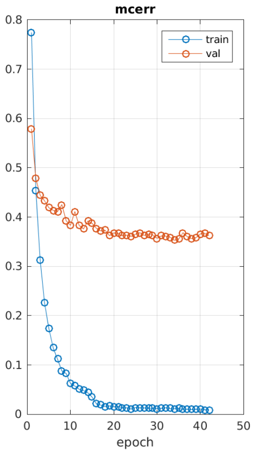

Fig. 5: Multi-class Error vs Number of epochs on MDI (Random Sampling) on an un-trained AlexNet

D. Random Sampling MDI on a Pre-Trained VGGNet

VGGNet outperforms AlexNet in image recognition tasks. The performance between a fine-tuned AlexNet and a fine-tuned VGGNet can be compared for UCF-101 as well. As opposed to a pre-trained AlexNet, the pre-trained VGGNet saturates at epoch 40. We can owe this to the fact that VGGNet is deeper and more complex when compared to AlexNet. Moreover, as expected, VGGNet outperforms AlexNet by 3.55 mAP points (*Table III*).

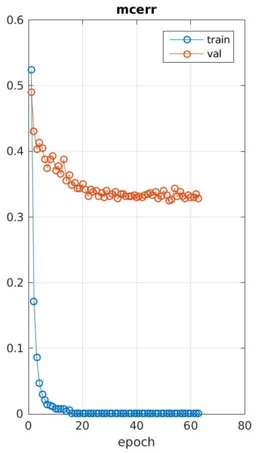

Fig. 6: Multi-class Error vs Number of epochs on MDI (Random Sampling) on a trained VGGNet

VI. Discussion and Analysis

Insights procured from the aforementioned experiments are discussed in this section.

A. Random Sampling MDI on a Pre-Trained AlexNet

The baseline for this review is established by evaluating the mAP on UCF-101 by fine-tuning a pre-trained AlexNet for 30 training epochs using multiple dynamic images randomly sampled from the raw video frames. While the original work [20] obtains an mAP of 70.9, my implementation saturates at a maximum mAP of 66.75, despite the algorithm and parameters being the same. The reason for this gap between the reported and evaluated precision is unclear, except for the possible differences in implementation hardware or the inherent randomness in weight updates.

B. Full Length Sampling MDI on a Pre-Trained AlexNet

An improvement on the baseline was expected when using full length sampling to obtain multiple dynamic images, since full length sampling using a moving window approach will not skip any frames. On the contrary, the obtained result was worse than the baseline established in V-A. The decline in mAP can be attributed to the fact that random sampling rarely picks the same dynamic images on subsequent epochs while full length sampling picks the same dynamic images on all epochs. This is evident in how the training curve of full length sampling saturates at epoch 20 and hardly fluctuates in validation accuracy, (*Figure 4*) whereas the random sampling training curve (*Figure 3*) saturates at epoch 25 and exhibits evident fluctuation in accuracy, owing to the different dynamic images constructed at each epoch.

C. Random Sampling MDI on an Un-Trained AlexNet

Transfer learning techniques have been demonstrated to have advantages over training a network for a task from scratch, since the pre-trained networks will have seen more training data beforehand. When training a fresh AlexNet for action recognition from scratch, I expected an mAP much lower than the base-line since the pre-trained network has already been trained on 1.2 million images. Naturally, the fresh network took longer to train. Surprisingly, the results obtained from the pre-trained network and the new network are very similar. In fact, the un-trained network performs slightly better (0.7 mAP points). This can be associated to the fact that dynamic images and ILSVRC images are heavily different in their characteristics, even though both are in the RGB color-space. The dynamic images resemble motion blur images while ILSVRC images used to pre-train the AlexNet are normal object class images. Hence, we can conclude that transfer learning only helps in converging the training faster.

D. Random Sampling MDI on a Pre-Trained VGGNet

In addition to image classification, a pre-trained VGGNet [25] outperforms AlexNet in UCF-101 as well. This enables us to infer that action-recognition from videos and image-classification are similar tasks. A deeper architecture learns filters that are better defined and picks up more relevant features from training images leading to better accuracy. From this, we can understand that positive strides made in image classification tasks help us move toward perfection in action recognition tasks as well.

In addition to the discussed experiments, efforts were put into analyzing the performance of a trained ResNet [27], a CNN with 101 layers (current state of the art in Imagenet recognition challenge with a 6.05% error rate in the top-5 classes). Even though I was successful in structurally re-purposing the network for the action recognition task, the training time on such a big network, with limited computational resources, made analysis impractical within the given time-frame. I expect ResNet to give results even better than VGGNet since it is demonstrated that a network that performs better on image classification performs better in action recognition tasks as well.

VII. Conclusion

Dynamic Image Networks for Action Recognition was published in 2016. By this time, AlexNet was only a surviving artifact of the efficiency of CNNs and many models with greater impact had been discussed in the literature. Their work would have obtained better results, maybe even beyond the state-of-the-art, if they had used a deeper network. In accordance with the authors' claims, it can be demonstrated that dynamic images are powerful tools for action recognition. This is verified by obtaining similar results to their claims. The potential of dynamic images is made even more important by their ability to harness the power of existing neural network architectures.

Dynamic images bridge the gap between action recognition and image classification by capturing all the motion contents of a video in a single RGB image. It is evident that a network that performs better in image classification replicates the same efficiency in action recognition. Dynamic images provide a solid direction of future research and I believe it is a promising approach for solving the action recognition problem. However, there is a lack of theoretical reasoning on the mastery of Convolutional Neural Networks on image-related tasks. An impeccable neural network classifier with low error rates can be built only if we understand and rectify their weaknesses accordingly. Action recognition algorithms still have a long road ahead. Robust, real-time algorithms with human-like intelligence, although far from our reach, can be achieved only through persistent and enduring research.

References

[1] http://fortunelords.com/youtube-statistics/

[2] Wang, Limin, et al. ”Towards good practices for very deep two-stream convnets.” arXiv preprint arXiv:1507.02159 (2015).

[3] J. Sanchez, F. Perronnin, T. Mensink, and J. J. Verbeek.Image classification with the fisher vector: Theory and practice. International Journal of Computer Vision, 105(3):222245, 2013. 1

[4] H. Wang and C. Schmid. Action recognition with improved trajectories. In ICCV, pages 35513558, 2013. 1, 4

[5] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In NIPS, pages 568576, 2014.

[6] I. Laptev. On Space-Time Interest Points. International Journal of Computer Vision (IJCV), 64(2):107123, 2005.

[7] G. Willems, T. Tuytelaars, and L. Van Gool. An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In European Conference on Computer Vision (ECCV), volume 5303, pages 650663, 2008.

[8] A.H. Shabani, D.A. Clausi, and J.S. Zelek. Salient Feature Detectors for Human Action Recognition. In Ninth Conference on Computer and Robot Vision (CRV), pages 468475, 2012.

[9] Wang, Heng, and Cordelia Schmid. ”Action recognition with improved trajectories.” Proceedings of the IEEE International Conference on Computer Vision. 2013.

[10] F. Perronnin, J. Sanchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In ECCV, 2010.

[11] Klaser, Alexander, Marcin Marszaek, and Cordelia Schmid. ”A spatio-temporal descriptor based on 3d-gradients.” BMVC 2008-19th British Machine Vision Conference. British Machine Vision Association, 2008.

[12] P. Scovanner, S. Ali, and M. Shah. A 3-Dimensional SIFT Descriptor and Its Applications to Action Recognition. In Proceedings of the 15th ACM International Conference on Multimedia, pages 357360, 2007.

[13] L. Yeffet and L. Wolf. Local Trinary Patterns for Human Action Recognition.In 12th IEEE International Conference in Computer Vision (ICCV), pages 492497, 2009.

[14] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing Action at a Distance. In IEEE International Conference on Computer Vision (ICCV), pages 726 733, October 2003.

[15] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

[16] Ji, Shuiwang, et al. ”3D convolutional neural networks for human action recognition.” IEEE transactions on pattern analysis and machine intelligence 35.1 (2013): 221-231.

[17] Simonyan, Karen, and Andrew Zisserman. ”Two-stream convolutional networks for action recognition in videos.” Advances in Neural Information Processing Systems. 2014.

[18] Donahue, Jeffrey, et al. ”Long-term recurrent convolutional networks for visual recognition and description.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[19] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In Advances in Neural Information Processing

[20] Bilen, Hakan, et al. ”Dynamic image networks for action recognition.” IEEE International Conference on Computer Vision and Pattern Recognition CVPR. 2016. Systems (NIPS), 2014.

[21] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice.Computer Vision and Image Understanding (CVIU), 150:109125, 2016.

[22] C. Feichtenhofer, A. Pinz, and R.P. Wildes. Spatiotemporal Residual Networks for Video Action Recognition. In Advances in Neural Information Processing Systems (NIPS), 2016.

[23] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time Action Recognition with Enhanced Motion Vector CNNs. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 27182726, 2016.

[24] Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah. ”UCF-101: A dataset of 101 human actions classes from videos in the wild.” arXiv preprint arXiv:1212.0402 (2012).

[25] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

[26] http://www.vlfeat.org/matconvnet/

[27] He, Kaiming, et al. ”Deep residual learning for image recognition.” arXiv preprint arXiv:1512.03385 (2015).

[28] Szegedy, Christian, et al. ”Going deeper with convolutions.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.