Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 831–838

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

YOLO based Human Action Recognition and Localization

Shubham Shinde[a,]*, Ashwin Kothari[a], Vikram Gupta[b]

[a]*Visvesvaraya National Institute of Technology, Nagpur (India)*
[b]*Awidit Systems pvt. Ltd., Gurugram (India)*

Abstract

Human action recognition in video analytics has been widely studied in recent years. Yet, most of these methods assign a single action label to video after either analyzing a complete video or using classifier for each frame. But when compared to human vision strategy, it can be deduced that we (human) require just an instance of visual data for recognition of scene. It turns out that small group of frames or even single frame from the video are enough for precise recognition. In this paper, we present an approach to detect, localize and recognize actions of interest in almost real-time from frames obtained by a continuous stream of video data that can be captured from a surveillance camera. The model takes input frames after a specified period and is able to give action label based on a single frame. Combining results over specific time we predicted the action label for the stream of video. We demonstrate that YOLO is effective method and comparatively fast for recognition and localization in Liris Human Activities dataset.

Keywords: video analytics; human action recognition; You Only Look Once (YOLO); Liris Human Activities dataset; Convolutional Neural Network; action label.

1. Introduction

In today's digital world among all the media types available, video is the most generated and consumed format but still the least processed. A recent report from IHS in 2015 indicated that the amount of memory requirement for daily video surveillance coverage worldwide is 566 petabytes. It also estimates that this visual data will rise to 2500 petabytes by the completion of 2019 [1]. Bringing analytics collected from videos into the mainstream Internet can empower endless new applications in surveillance camera, robotics, content based video search and human computer Interface.

*corresponding author. Tel.: +91 9403007237

E-mail addresses: shubham.shinde@students.vnit.ac.in (Shubham Shinde), *ashwinkothari@ece.vnit.ac.in (Ashwin Kothari), vikramg@awidit.com (Vikram Gupta).*

10.1016/j.procs.2018.07.112

Over the past few years, researchers have widely adopted Convolutional Neural Networks (CNN) for image classification problems. Due to the success of CNNs in classification of an image and its contents, researchers have started to employ CNN for video classification to a greater extent. Classifying real life videos (LIRIS dataset [3]) into arbitrary free-form activities is a challenging task mainly because of illumination conditions, occlusion, background clutter, deformation, viewpoint, scale and intra-class variation.

Research done in this field till now can be classified into two approaches:

1. Human action recognition using two-stream CNNs [5] (spatial and temporal streams). Here two-stream CNNs are mainly trained on multiframe dense optical flow. Where temporal and spatial stream deals with motion in form of dense optical flow and still video frames respectively.
2. Human action recognition based on skeleton tracking [6]. In this approach, a video stream is passed through a skeleton-tracking algorithm. Then based on the movement between selected joints and their respective angular velocities, action recognition can be done.

Even though above approaches have achieved notable results, after considering their computation time and training overhead, it seems that they might be using more information than required [19]. In our approach, we can recognize and localize actions from fewer video frames (often even from the single frame), without the need of optical flow data from frames.

The model we proposed here allocates a unique action label and confidence score to each frame with the global action label for video sequence is acquired by finding the most frequent action label. Periodic frames from the video sequence are processed and not the whole video. Also, in most of the cases, a single frame is sufficient for recognition of action present in the video. In the present work, we have utilized minimum number of frames thus reducing the computational time. In case of rapid diminishing of confidence score more frames are added.

In this project, we investigate how You Only Look Once (YOLO) [2] after training with LIRIS dataset [3] can give us action label, confidence score and the bounding box of the localized action.

2. Proposed Methodology

The workflow we implemented for this paper is explained in Figure 1. The method we used for training and testing of the model is You Only Look Once (YOLO), which will be explained in details in next section. The dataset we used to train the model is LIRIS human activities dataset.

We trained our model using frames containing appropriate action, from the training set of LIRIS dataset. For testing purpose, we selected 30 frames in a video for recognition and localization of action, from the testing video frames. These frames are chosen after a period of a certain number of frames which depends upon the total number of frames in the video. Path of these frames is stored in the text file which will be further given as input to the model. Output after Recognition and localization of each frame is stored in CSV file.

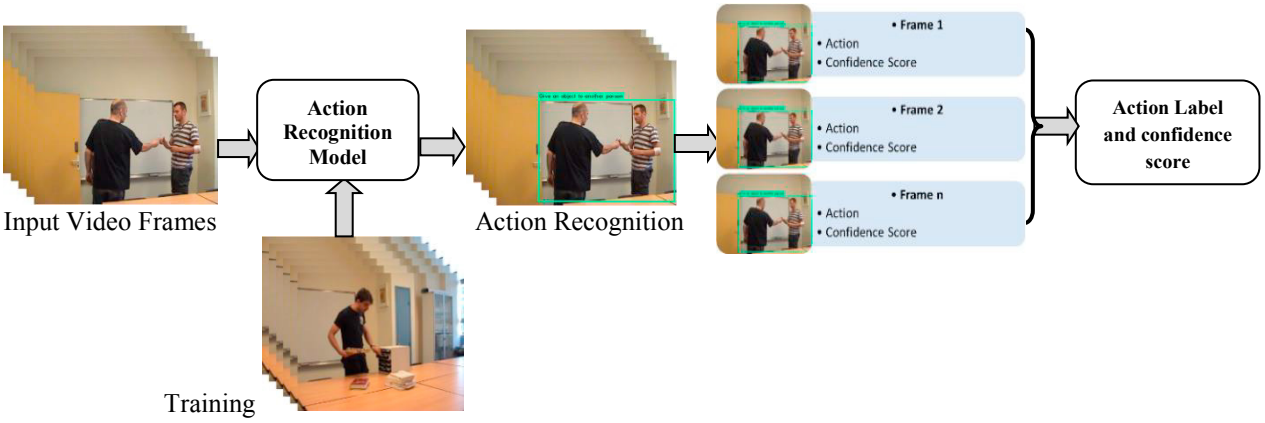

Fig. 1: Workflow of the algorithm

After performing action detection on each frame, the action label detected in more than 5 frames and having a confidence threshold of 0.5 over 30 frames is classified as concluding action label in the video. The combined confidence score of the action label overall is calculated by averaging all confidence score obtained.

3. You Only Look Once (YOLO)

3.1. Object Detection

In images, You Only Look Once (YOLO) [2] is an advanced approach object detection. YOLO applies a single CNN to the entire image which further divides the image into grids. Prediction of bounding boxes and respective confidence score are calculated for each grid. These bounding boxes are analyzed by the predicted confidence score. The Architecture of YOLO has 24 convolutional layers and 2 fully connected layers [7]. The architecture is shown in Figure 2.

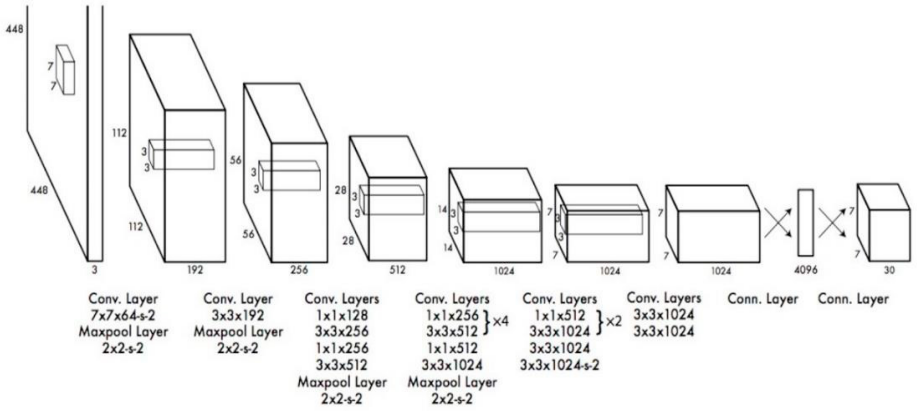

Fig. 2: The architecture of YOLO [7]

YOLO takes an input image and resizes it to 448×448 pixels. The image further goes through the convolutional network and gives output in form of 7×7×30 tensor. Tensor gives the information about 1) coordinates of bounding box's rectangle and 2) Probability distribution over all classes the system is trained for. Thresholding these confidence scores (probability) eliminates class labels scoring lesser than 30%.

3.2. Advantages over other Detectors

Most of the other CNN-based systems reuse classifiers or localizers to detect an object in the image. These model are applied to an image at multiple locations and different scales. And regions with high confidence score are considered as detections.

Compared to conventional methods of object detection, YOLO has certain advantages. 1) Rather than using two-step method for classification and localization of object, YOLO applies single CNN for both classification and localization of the object. 2) YOLO can process images at about 40-90 FPS, so it is quite fast [2]. This means streaming video can be processed in real-time, with negligible latency in a few milliseconds. The architecture of YOLO makes it extremely fast. When compared with R-CNN, it is 1000 times faster and 100 times faster than fast R-CNN [2, 8].

3.3. Training

The flowchart for training of YOLO based action recognition model is shown in Figure 3. First, we converted the LIRIS dataset labels to usable label files for YOLO. YOLO requires a .txt file for each frame with a line for each action in the frame. Further YOLO requires some files to start training which are:

- Total number of action classes.
- Text file with the path to all frames which we want to train.
- Text file with names of all action classes.
- The path to save trained weight files.
- A configuration file with all layers of YOLO architecture described in figure 2.
- Pre-trained convolutional weights.

The value of filters in the configuration file of YOLO (.cfg file) for second last layer is not arbitrary and depends on the total number of classes [9]. The number of filters can be given by:

$$filters = 5 * (2 + number\_of\_classes) \quad (1)$$

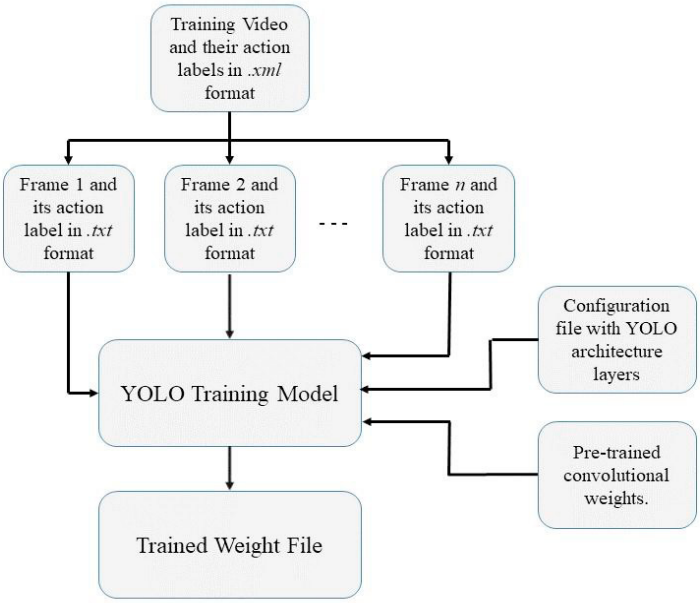

Fig. 3: The Flowchart for training of YOLO

4. Liris Human Activities dataset

Among all the video dataset present today, most of them belong to:
1) Simple periodic actions like walking, running (e.g. KTH dataset [10])
2) A more realistic dataset for human-human or human-object interactions (e.g. CASIA [11], UCF101 [12])
3) Actions in YouTube, Dailymotion videos (e.g. UCF YouTube [13], Google Ava [14])
4) RGB-D dataset containing depth information in the video (e.g. MSRC-12 [15])

But not all of these datasets give localization of action in the video. As localization of action is important, we are using LIRIS human activities dataset [3] in this paper. The dataset contains the videos in form of RGB frames. The annotations provided are in form of an XML file. The dataset is ideal for processing video based on surveillance camera. There are 10 visual annotated actions which include human-human, human-object, and human-human-object interactions. Table 1 shows the list of action categories in LIRIS dataset.

Table 1: The action categories in LIRIS dataset

Sr. No	Action classes of dataset	Abbreviation	Type of Interaction
1	Discussion between two or more people	DP	Human-Human
2	Give an object to another person	GO	Human-Human-Object
3	Put/take an object into/from a box/desk	TO	Human-Object
4	Enter/leave a room (pass through a door)	ER	-
5	Try to enter a room (unsuccessfully)	EU	-
6	Unlock and enter (or leave) a room	UR	-
7	Luggage left unattended	LB	Human-Object
8	Handshaking	HS	Human-Human
9	Keyboard typing	TK	Human-Object
10	Mobile/ Telephone conversation	TC	Human-Object

Fig. 4: Frames in LIRIS dataset (i) Try to enter a room (unsuccessfully), (ii) Discussion between two or more people, (iii) Handshaking, (iv) Leave baggage unattended, (v) Give an object to another person, (vi) Unlock and enter (or leave) a room , (vii) Put/take an object into/from a box/desk, (viii) Enter/leave a room (pass through a door), (ix) Telephone conversation

The dataset contains 367 actions from 167 videos. These 167 videos are further divided into 109 training and 58 testing videos. Per class ratio of training to test data is kept approximately 2:1. Figure 4 shows some of the frames in LIRIS dataset.

5. Result

The methodology proposed in this paper was implemented on the system having following specifications:

- OS: Ubuntu 16.04LTS (64 bit)
- CPU: Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
- CPU Ram: 32 GB
- GPU: Gigabyte GeForce GTX 1050 Ti
- Graphics card Ram size: 4 GB

5.1. Training and Testing:

In training process, each iteration took about 5.25s. We trained our model up to 40000 iterations and average loss was found to be 0.32 for the batch size of 32 with 8 subdivisions. For more accuracy and to reduce the average loss, we can train our model with more number of iterations. For testing, average recognition of action in frame takes about 61ms that is 15-16 FPS (Frame per second). Hence, Real-time recognition of action is feasible on off-the-shelf desktop PCs with a mid-level Graphics Processing Unit (GPU).

5.2. Quantitative Results:

The accuracy obtained on the test dataset is the primary evaluation metric for classification in our paper. The confusion matrix is used to calculate Precision, Recall, F-score and overall Accuracy of the model. Confusion matrix obtained on test dataset is shown in Table 2, where rows represent actual actions while column represents the predicted actions.

Table 2: The confusion matrix obtained. It shows the classification accuracy on LIRIS human activity dataset.

	DP	**GO**	**TO**	**ER**	**EU**	**UR**	**LB**	**HS**	**TK**	**TC**
DP	7	0	0	0	0	0	0	0	0	0
GO	1	5	0	0	0	0	0	1	0	0
TO	0	1	10	0	0	0	0	0	0	0
ER	0	0	0	12	0	0	0	0	0	0
EU	0	0	0	1	7	1	0	0	0	0
UR	0	0	0	0	0	11	0	0	0	0
LB	1	0	0	0	0	0	6	0	0	0
HS	0	1	0	1	0	0	0	8	0	0
TK	0	0	0	0	0	0	0	0	4	0
TC	0	0	2	0	0	0	0	0	0	6

Table 3: Precision, Recall, F-score for each action class using confusion matrix

Sr. No.	***Label***	***Precision (%)***	***Recall (%)***	***F-Score (%)***
1	DP	77.778	100	87.5
2	GO	71.429	71.429	71.429
3	TO	83.333	90.909	86.956
4	ER	85.714	100	92.307
5	EU	100	77.778	87.5
6	UR	91.667	100	95.652
7	LB	100	85.714	92.307
8	HS	88.889	80	84.211
9	TK	100	100	100
10	TC	100	75	85.714
	Mean	89.881	88.083	88.358

Most classification results are concentrated along the diagonal. The accuracy obtained by methodology proposed in this paper was found to be 88.372%. Precision, Recall, F-score for each action class is presented in Table 3. Table 4 indicates that proposed method performs better when compared F-scores of state-of-art methods. In the proposed approach, we assumed that the action takes place near to the camera and is less occluded.

Table 4: Comparisons of performance with the state-of-the-art methods (F-scores for action recognition and localization).

Sr. No.	*Method*	*F-score (%)*
1	B. Ni [16]	41.7
2	ADSC-NUS-UIUC [3]	53
3	S. Saha [17]	58.10
4	S. Mukherjee et al. [18]	81.27
5	Ours	88.358

6. Conclusion and Future Work

In this paper, we proposed a real-time model for human action recognition in video based on YOLO. The main finding of our study is that even a small number of frames can be used to recognize actions in a video. We found that even for some cases, a single frame is sufficient for recognition of action. Examples of action detection of such frames are shown in figure 5.

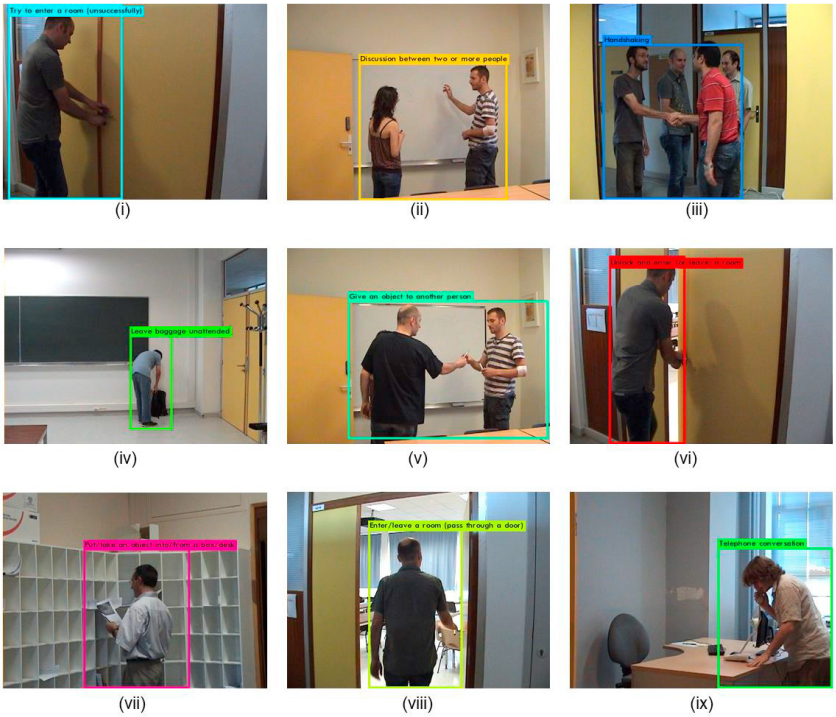

Fig. 5: Recognition and Localization of actions in single frame shown in Figure 4.

Our future work is focused on finding a metric which can help the action recognition to complete within a few frames, and thus the classification stops automatically for the particular action. We will also focus on improving action recognition with help of object detection in frames so that more complex human actions can be detected. Movement of objects or Euclidean distance between centers of moving object and human can also provide more information about action occurring in the video.

Acknowledgements

The research work presented in this paper is supported by Awidit Systems Pvt. Ltd. We gratefully acknowledge Centre of Excellence (CoE) on Commbedded Systems, Dept. of Electronics and Communication, VNIT, Nagpur for use of the GPUs used for this project.

References

[1] "Data Generated by new survellillance cameras to increase exponentially in the coming years." [Online, Accessed on Mar 12, 2018]. http://www.securityinfowatch.com/news/12160483/data-generated-by-new-surveillance-cameras-to-increase-exponentially-in-the-coming-years

[2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 779-788. doi: 10.1109/CVPR.2016.91

[3] C Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandrea, C.-E. Bichot, C. Garcia, B. Sankur, Evaluation of video activity localizations integrating quality and quantity measurements, In Computer Vision and Image Understanding (127):14-30, 2014.

[4] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In CVPR, pages 4305–4314, 2015.

[5] Simonyan, Karen & Zisserman, Andrew. (2014). Two-Stream Convolutional Networks for Action Recognition in Videos. Advances in Neural Information Processing Systems. 1.

[6] Papadopoulos G.T., Axenopoulos A., Daras P. (2014) Real-Time Skeleton-Tracking-Based Human Action Recognition Using Kinect Data. In: Gurrin C., Hopfgartner F., Hurst W., Johansen H., Lee H., O'Connor N. (eds) MultiMedia Modeling. MMM 2014. Lecture Notes in Computer Science, vol 8325. Springer, Cham.

[7] "YOLO-You only look once, real time object detection explained." [Online, Accessed on Mar 12, 2018]. Available at: https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006

[8] "YOLO- Real time object detection" [Online, Accessed on Mar 12, 2018]. Available at: https://pjreddie.com/darknet/yolo/

[9] "Start training YOLO with our own data" [Online, Accessed on Mar 12, 2018]. Available at: http://guanghan.info/blog/en/my-works/train-yolo/

[10] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: International Conference on Pattern Recognition, vol. 3, 2004, pp. 32–36.

[11] C. for Biometrics, S. Research, Casia Action Database for Recognition, 2013.

[12] A.R.Z.K. Soomro, M. Shah, A dataset of 101 human action classes from videos in the wild, in: THUMOS: The First International Workshop on Action Recognition with a Large Number of Classes, in conjunction with ICCV 2013, 2013.

[13] U. of Central Florida, Ucf youtube Action Dataset, 2013.

[14] AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions **arXiv:1705.08421v3**

[15] S. Fothergill, H.M. Mentis, P. Kohli, S. Nowozin, Instructing people for training gestural interactive systems, in: J.A. Konstan, E.H. Chi, K. Höök (Eds.), ACM Conference on Computer-Human Interaction, 2012, pp. 1737–1746.

[16] Ni, B., Pei, Y., Liang, Z., Lin, L., Moulin, P.: Integrating Multi-stage depth-induced contextual information for human action recognition and localization. In: IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, pp. 1–8 (2013).

[17] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip H. S. Torr, Fabio Cuzzolin: Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos at arXiv:1608.01529 [cs.CV].

[18] Mukherjee S., Mallik A., Mukherjee D.P. (2015) Human Action Recognition Using Dominant Motion Pattern. In: Nalpantidis L., Krüger V., Eklundh JO., Gasteratos A. (eds) Computer Vision Systems. ICVS 2015. Lecture Notes in Computer Science, vol 9163. Springer, Cham.

[19] Luc van Gool. "Action snippets: How many frames does human action recognition require?", 2008 IEEE Conference on Computer Vision and Pattern Recognition, 06/2008.