Journal Homepage: - www.journalijar.com
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)
ISSN NO. 2320-5407

Article DOI: 10.21474/IJAR01/7082
DOI URL: http://dx.doi.org/10.21474/IJAR01/7082

RESEARCH ARTICLE

INVESTIGATION OF CONVOLUTIONAL NEURAL NETWORKS FOR VISUAL TRACKING OF PEDESTRIANS.

Artūras Jonkus, Paulius Tumas and Artūras Serackis.
Department of Electronics, Vilnius Gediminas Technical University.

Manuscript Info

Manuscript History

Received: 12 March 2018
Final Accepted: 14 April 2018
Published: May 2018

Keywords:-
pedestrian detector, YOLO.

Abstract

The problem of human detection in an image or video sequence is still a hot topic nowadays. It has been actively researched and still, the accurate and fast detection remains an issue. This paper aims to provide additional insights into existing solutions for pedestrian detection. The proposed method is to only use a part of video frames for object detection showed that it is possible to receive 88 % processing speed increase without accuracy lost when using every second frame. However, skipping more frames introduces tracking latency of approximated location of a pedestrian.

Introduction:-

It seems natural to focus on images captured by one's eyes or cameras because they provide a lot of information about the surrounding world. It can be both the goods on the shelf or people walking down the street. Our brain processes such images without much mental effort, but it is a much more difficult task to be transferred to computers. In recent years, solutions to such problem have been greatly improved by Convolutional Neural Networks (CNN). Every year state-of-the-art results are achieved in object detection tasks, but it comes with a price - there are much more calculations to be performed than in classical object detection algorithms.

In the recent years, excellent results have been achieved by quantizing the weights of CNN or by measuring them using low precision numbers (Park et al. 2017; Hubara et al. 2016) as well as changing the corresponding number of layers making the CNN model smaller capable to work even with mobile devices, but only marginally less effective, such as *SqueezeNet* (Iandola et al. 2016). Our proposed method is to only use part of video frames for object detection in order to gain object tracking speed. Pedestrians are selected as an experiment object.

An overview of tools, data and methods used, analysis of the results of the system with the changes made and the resulting conclusions are also presented.

Materials and Methods:-

The chosen parameter to define the speed of object tracking algorithm's is a number of processed frames per second (FPS).

Objects in the image can be marked rectangular called bounding boxes. If this rectangle defines an object accurately, it can be called a ground truth (GT). An estimate (E) is a bounding box surrounding rectangle which is produced by an object detection algorithm. Two parameters describing the recognition quality independently of the ground truth

Corresponding Author:- Artūras Jonkus.
Address:- Department of Electronics, Vilnius Gediminas Technical University.

(GT) and measurement (E) shape, are recall and precision (Smith et al. 2005). A recall is used to describe how much of the measuring rectangle covers the ground truth area. It might happen that, despite the high parameter value, object tracking results will not be satisfactory. A whole reference rectangle could be covered by a measuring rectangle, but only a part of the measuring rectangle would be used to cover the object (Fig. 1). The formula for this coefficient $p_{i,j}$:

$$p_{i,j} = \frac{|E_i \cap GT_j|}{|GT_j|}, \quad (1)$$

here E_i – an area covered by a measure, GT_j– area covered by ground truth.

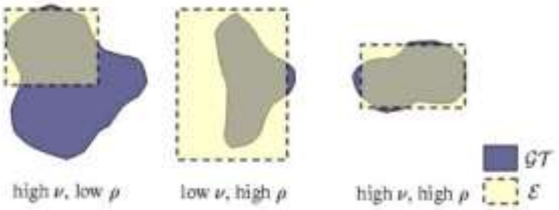

Fig. 1:- Graphical representation of precision (v) and recall (p)

The precision describes how well the measured part of the rectangle covers the reference rectangle. It is highly probable that a high coefficient would not guarantee high tracking quality (Fig. 1). Although the entire measuring rectangle might be used to cover the object, it would not cover the entire object. This coefficient $v_{i,j}$ can be calculated as:

$$v_{i,j} = \frac{|E_i \cap GT_j|}{|E_i|} \quad (2)$$

Developers of MOTChallenge (Milan et al. 2016) distinguish two requirements for object tracking in a video sequence. The first one – for each recognizable object, it is necessary to determine whether the recognized object is classified as true positive (TP) or false positive (FP). It should also be noted if the object in the image was not detected where necessary (false negative, FN). The second requirement is that if the object is detected after a video frame where an object hasn't been detected, tracking algorithm should make sure it receives the same unique identifier that was used before. The loss of an identifier would increase the number of incorrectly guessed (FP) and non-recognized objects (FNs).

The aforementioned parameters (TP, FP, and FN) can be used to calculate mean average precision (mAP). The ratio of intersection over union in a successful object detection is considered to be at least 0.5.

Object tracking quality is described by multiple object tracking precision (MOTP) (Milan et al. 2016; Bernardin et al. 2006). It is evaluated as the average of the object's localization error:

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t}, \quad (3)$$

here c_t – correctly guessed number of objects in the frame, $d_{i,t}$ is the intersection between the bounding box that defines the object and the real bounding box presented in the data.

Multiple object tracking accuracy (MOTA) is one of the most commonly used tracking parameters. It evaluates three types of errors and is defined by the following formula (Milan et al. 2016; Bernardin et al. 2006):

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t}, \quad (4)$$

here *t* is the frame number, GT_t is the number of all objects to be detected, $IDSW_t$ is the number of tracking identifier switches. It is possible for error count to be bigger than the count of objects to be recognized, so this parameter is considered to be in the range (-∞, 100) when using percentages.

Due to a large number of samples it is chosen to use the dataset that has already been developed and proven in applications - Caltech Pedestrian Dataset (Dollár et al. 2009; Dollár et al. 2012). The entire dataset consists of as many as 250 thousand pictures where 350 thousand bounding boxes of pedestrians are annotated. It has been decided to use an incomplete set of data, taking into account the technical limitations of the equipment used for training.

Multi-Object Detection Benchmark (2D MOT-15) video files are used for testing (Milan et al. 2016). Experiments are therefore performed for images of a different size.

A usual object tracking system consists of three parts - object detection, filtering used to dynamically update tracking coordinates and tracking coordinate assignments to relevant objects. Each of these three parts may consist of different algorithms which would allow the whole system to achieve the desired result. However, for the optimized system a particular set of them is used. Detection is based on the YOLOv2 machine training model (Redmon & Farhadi 2016; Redmon et al. 2015), Kalman filter is used for object tracking, and the problem of identifier assignment is solved by the Kuhn-Munkres algorithm (the complexity of calculations $O(n^3)$) also known as Hungarian algorithm (Munkres 1957). Thus the detector indicates the coordinates of a detected object, the Kalman filter ensures that the object found between the frames is more resistant to high-frequency noise, while the Hungarian algorithm tries to ensure that a particular object retains its originally assigned identifier.

A paper describing a combination of a Kalman filter and Hungarian algorithm called SORT (Simple Online Realtime Tracking) has recently been published and showed promising results (Bewley et al. 2016). The advantage of this combination is a high speed - tracking is not loaded with attempts to solve extreme cases that slow down the algorithm, even if they make them more resistant to errors. This problem is considered to be self-solving because there is a steady improvement in the quality and speed of recognition algorithms. In order to make the object detection more resistant to noise, the results of tracking are stored for some time before they are deleted.

The experimental research uses a computer with eight core AMD Ryzen 7 1700 3.7 GHz processor, 32 GB of RAM, ASUS ROG Strix GeForce GTX1080Ti 11 GB RAM GAMING graphic processor. The Python programming language and the OpenCV library have been used to write the code, and some parts of it are written in C and C ++ to achieve higher algorithm performance.

Training is terminated after 45 thousand iterations. Then the best results are returned to the validation data. The training is carried out in 416 × 416 images, and then the resolution of an image is increased to 608 × 608. This allows better detection of small objects and a higher detection result. Training is done five times, 6000 images of the same database are used for validation. The best achieved average detection accuracy is 77.81%.

Table 1:- Multiple object tracking metrics for a reference algorithm

Video name	Resolution	Object detection speed, FPS	Tracking speed, FPS	Speed of a whole tracking pipeline, FPS	*Recall*	*Precision*	MOTA	MOTP
PETS09-S2L1	768 × 576	15.4	733	15.1	25.7	71.2	13.2	66.1
TUD-Campus	640 × 480	16	707	15.6	57.4	81.2	42.6	64.4
TUD-Stadtmitte	640 × 480	16.3	612	15.9	67.3	94.3	62.1	63
ETH-Sunnyday	640 × 480	16.2	539	15.7	58.6	76.5	38.3	71.7
ETH-Pedcross2	640 × 480	16.3	838	16	23.1	92.7	20.3	73.5
KITTI-17	1224 × 370	20.5	538	19.8	41.9	57.3	6.9	67.7
ADL-Rundle-6	1920 × 1080	10.3	484	10.1	45.3	78	30.7	70.9
Venice-2	1920 × 1080	10.3	424	10.1	34.3	60	10.2	69
Total		15.2	609	14.8	44.2	76.4	28	68.3

The initial parameters for pedestrian tracking are given in Table 1. It is seen that the most of calculation time is used for the detection of the object and the difference between it and other parts is measured several tenths of times. It should also be noted that a relatively high tracking quality is available for small images (640 × 480). Lower quality is obtained by tracking subjects in large video frames or where one side of the video frame is longer than the other (KITTI-17). It can also be seen (according to the MOTP) that the accuracy of the object localization is about 70%, but there is a rather high number of errors that the MOTA parameter evaluates.

Results of the Experimental Investigation:-

Experiments start by using only every second, third or fourth frame for pedestrian detection. Metrics of multiple object detection and their standard deviations are represented in Table 2. Fig.2-3 show how a speed of YOLOv2 object tracking algorithm is dependent on using only a part of frames.

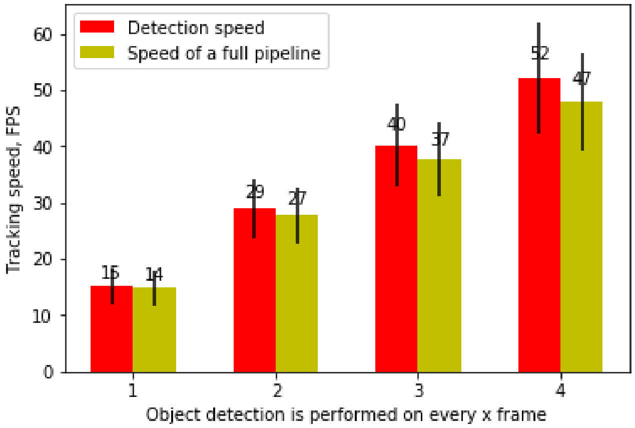

Fig 2:- Comparison of speed when only one of several frames is used.

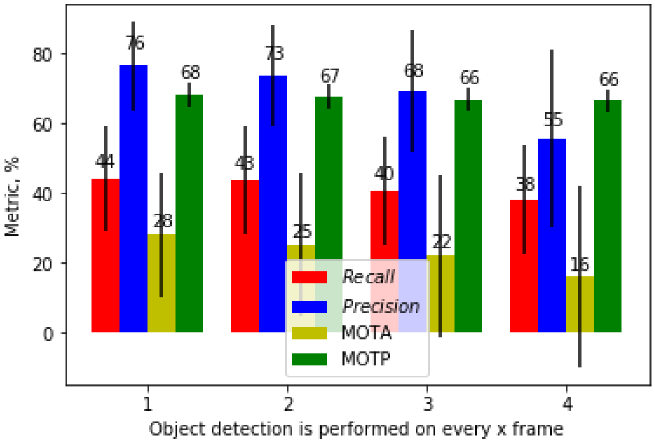

Fig 3:- Comparison of object tracking parameters when only one of several frames is used

Aforementioned results show that using only one of several frames for object detection greatly decreases computational complexity. However, it is worth mentioning that every skipped frame has its cost - pedestrian tracking algorithm gets less robust, localization is performed slightly worse (lower recall and precision), more detection and identification mistakes are made (lower MOTA metric).

A different approach to skipping frames for detection can be used. It is possible to only skip part of the frames. Experiments skipping every third and fourth have been carried out and their results are represented in Table 3 and Fig. 4-5.

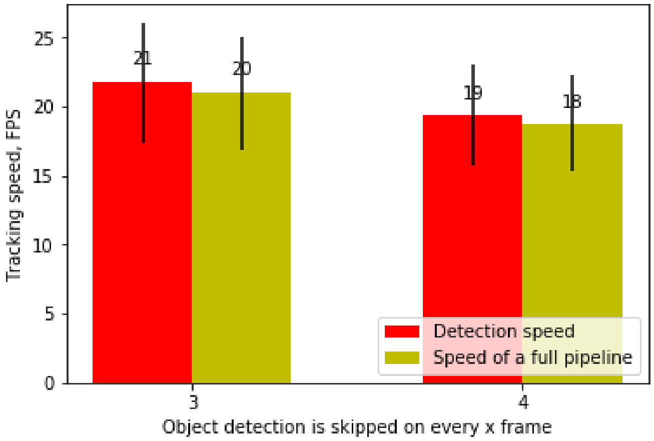

Fig.4. Comparison of speed when dropping a fraction of video frames for detection.

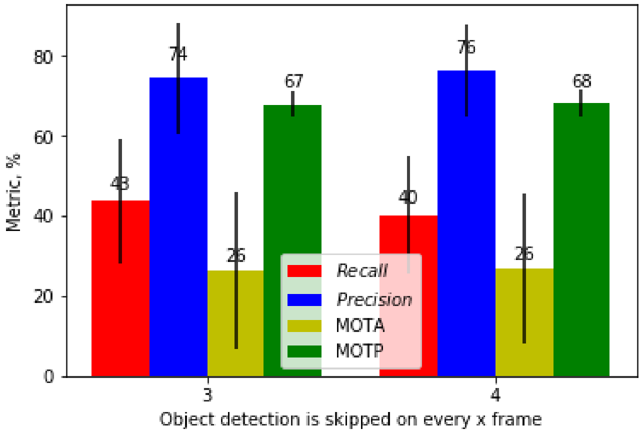

Fig. 5:- Comparison of object tracking parameters when dropping a fraction of video frames for detection

Table 2:- Multiple object tracking results after only performing object detection for a fraction of the frames

Video name	Resolution	Object detection speed, FPS	Tracking speed, FPS	*Recall*	*Precision*	MOTA	MOTP
Detection is performed in every second frame							
PETS09-S2L1	29.7	756	28.5	24	65.5	9.2	65.3
TUD-Campus	30.6	742	29.4	55.7	79.7	40.1	64.2
TUD-Stadtmitte	30.8	621	29.3	68.3	94.6	63	62.5
ETH-Sunnyday	30.5	573	29	59.1	73.8	35.8	71.2
ETH-Pedcross2	30.8	875	29.7	23.4	89.9	19.6	72.5
KITTI-17	37.8	560	35.4	38.4	48.6	-6.1	65.7
ADL-Rundle-6	20.8	526	20	46.1	78	31.3	70.1
Venice-2	21	463	20.1	34.4	59.2	9.3	69.4
In total:	**29**	640	**27.8**	**43.7**	**73.7**	**25.3**	**67.6**
Detection is performed in every third frame							
PETS09-S2L1	42.5	733	40.2	19.7	56.9	2.6	64
TUD-Campus	44.2	746	41.7	56.3	79.2	40.4	63.2

TUD-Stadtmitte	44.2	616	41.3	58.8	94.6	63.8	62.2
ETH-Sunnyday	42.7	530	39.5	59.4	59.1	29.6	70.5
ETH-Pedcross2	41.4	812	39.4	23.4	88.7	19.5	71.5
KITTI-17	50	548	45.8	26.6	39.1	-17.9	65.2
ADL-Rundle-6	28.3	517	26.8	45.6	75.9	29.4	69.5
Venice-2	28.2	442	26.5	34.9	58	9.3	69
In total:	**40.2**	**618**	**37.7**	**40.6**	**68.9**	**22.1**	**66.9**
Detection is performed every fourth frame							
PETS09-S2L1	54.3	716	50.5	14.5	41.6	-7.5	64
TUD-Campus	56.7	677	52.3	54.6	74.5	34.5	63.2
TUD-Stadtmitte	56.2	584	51.3	57.6	92	60.7	62
ETH-Sunnyday	55.3	547	50.2	52.7	64.3	21.3	70.2
ETH-Pedcross2	56.1	857	52.7	22.4	8.1	16	70.8
KITTI-17	66.4	538	59.1	22.9	30.7	-31.5	64.8
ADL-Rundle-6	36.1	514	33.7	45.4	74.6	28.1	69.4
Venice-2	36.1	450	33.5	34.6	57.6	7.8	68.4
In total:	52.2	610	47.9	38.1	55.4	16.2	66.6

Table 3:- Multiple object tracking results after skipping part of the frames for object detection

Video name	Resolution	Object detection speed, FPS	Tracking speed, FPS	*Recall*	*Precision*	MOTA	MOTP
Detection is not carried out every third frame							
PETS09-S2L1	22.3	737	21.6	24.3	67	10.3	65.7
TUD-Campus	22.7	718	22.0	56.8	80.3	41.5	64.1
TUD-Stadtmitte	23.4	607	22.5	67.5	94.1	62.3	62.9
ETH-Sunnyday	23	546	22.1	59.8	75.1	37.8	71.3
ETH-Pedcross2	23.2	847	22.6	23.2	91.4	20.1	72.8
KITTI-17	29.2	556	27.7	37.6	51.1	-2	67.1
ADL-Rundle-6	15	494	14.6	45.7	77.5	30.7	70.5
Venice-2	15.2	438	14.7	34.5	59.9	10.1	69.2
In total:	21.8	618	21	43.7	74.6	26.4	68
Detection is not carried out every fourth frame							
PETS09-S2L1	19.8	729	19.3	25	68.5	11.3	65.8
TUD-Campus	20.8	725	20.2	56.3	80.5	41.2	64.7
TUD-Stadtmitte	21	625	20.3	67.5	94.4	62.3	62.7
ETH-Sunnyday	20.9	550	20.1	28.9	74.8	36.9	71.5
ETH-Pedcross2	21.1	836	20.6	23.1	91	19.8	72.9
KITTI-17	24.5	536	23.5	40.3	53.4	1.2	67.1
ADL-Rundle-6	13.2	491	12.9	45.8	77.9	31.1	70.5
Venice-2	13.9	440	13.5	34.2	59.6	9.7	69.4
In total:	19.4	617	18.8	40.1	76.3	26.7	68

The aforementioned results in Table 3 and Fig.4-5 show that skipping some frames enable a speedup of an object tracking algorithm. A decrease of computational complexity introduces virtually no penalty on multiple object tracking metrics.

Conclusions:-

An increase of speed for convolution neural network can be done by changing parts of network's architecture but finding an optimal solution could be a highly complex task. Skipping an object detection part of an algorithm can be used to speed multiple object tracking algorithm without much loss of tracking accuracy.

The speedup of a tracking algorithm is linearly dependent on skipped frames, but using only every second frame doesn't make algorithm two times faster. Around 10-12 percent of initial computation power is spent on parts other than pedestrian detection (Kalman filter and Hungarian algorithm).

References:-

1. Bernardin, K., Elbs, A. & Stiefelhagen, R., 2006. Multiple object tracking performance metrics and evaluation in a smart room environment. *Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV*, 90, p.91. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.7070&rep=rep1&type=pdf.
2. Bewley, A. et al., 2016. Simple Online and Realtime Tracking. *Proceedings - International Conference on Image Processing, ICIP*, 2016–Augus, pp.3464–3468.
3. Dollár, P. et al., 2009. Pedestrian detection: A benchmark. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pp.304–311.
4. Dollár, P. et al., 2012. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4), pp.743–761.
5. Hubara, I. et al., 2016. Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. , pp.1–29. Available at: http://arxiv.org/abs/1609.07061.
6. Iandola, F.N. et al., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. , pp.1–13. Available at: http://arxiv.org/abs/1602.07360.
7. Milan, A. et al., 2016. MOT16: A Benchmark for Multi-Object Tracking. , pp.1–12. Available at: http://arxiv.org/abs/1603.00831.
8. Munkres, J., 1957. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1), pp.32–38. Available at: http://epubs.siam.org/doi/10.1137/0105003.
9. Park, E., Ahn, J. & Yoo, S., 2017. Weighted-Entropy-Based Quantization for Deep Neural Networks. *Cvpr*, pp.5456–5464. Available at: http://openaccess.thecvf.com/content_cvpr_2017/papers/Park_Weighted-Entropy-Based_Quantization_for_CVPR_2017_paper.pdf%0Ahttp://openaccess.thecvf.com/content_cvpr_2017/html/Park_Weighted-Entropy-Based_Quantization_for_CVPR_2017_paper.html.
10. Redmon, J. et al., 2015. You Only Look Once: Unified, Real-Time Object Detection. Available at: http://arxiv.org/abs/1506.02640.
11. Redmon, J. & Farhadi, A., 2016. YOLO9000: Better, Faster, Stronger. Available at: http://arxiv.org/abs/1612.08242.
12. Smith, K. et al., 2005. Evaluating Multi-Object Tracking.