Little-Big Deep Neural Networks for Embedded Video Surveillance

Cătălin Alexandru Mitrea[1], Mihai-Gabriel Constantin[1], Liviu-Daniel Ștefan[1], Marian Ghenescu[2], Bogdan Ionescu[1]
[1] University Politehnica of Bucharest, Multimedia Lab, CAMPUS, Bucharest, Romania
[2] UTI Grup, Softrust Vision Analytics, Bucharest, Romania
Contact author e-mail: camitrea@gmail.com

***Abstract*—Embedded systems are under continuous development of innovative technological trends, such as adoption of smart devices which are becoming capable of running complex video analytics tasks locally. Lately, deep neural networks have been successfully applied in the field of computer vision achieving state-of-the-art results. These techniques are not yet suitable for resource limited deployments due to high memory footprint and computational cost, factors that affect the inference time. To tackle these constraints, we propose a person re-identification architecture based on the DarkNET Deep Learning Neural Network architecture for person detection and segmentation, which is combined with SIFT algorithm for feature extraction and SVM for the classification task. The algorithm is implemented on a low processing embedded hardware system, namely Raspberry PI. The experiments were conducted on the proposed SPOTTER dataset. The results are conclusive to continue further investigation of applying specialized algorithms for real-time applications which can run on resource limited embedded systems.**

Keywords—embedded; deep neural networks; video person re-identification; video surveillance

I. Introduction

The embedded systems market is under an increased accelerating trend. This growth is partially a result of the consistent development of technology, such as adoption of the Internet of Things (IoT) and smart devices with applications that addresses scenarios in which the input data is in form of videos. Video is also used for evidence by law enforcement or advanced analytics in retail environments. However not all the video generated locally on acquisition devices (e.g., CCTV camera) must be transmitted and stored, instead the processing could take place locally using the device hardware. This is currently an issue as the hardware capabilities are low in terms of processing power on embedded devices.

There has been a growing trend in recent years in deploying deep learning models on edge devices. Deep Neural Networks (DNN) have become an established technology for image and video objects detection, retrieval and classification tasks [1, 2]. Various deep learning architectures such as deep neural networks (DNN), convolutional deep belief neural networks (CDNN), and deep belief networks (DBN) have been applied to many fields, where they have been shown to produce state-of-the-art results [3]. Although obtain high performance, Deep Neural Networks are high processing consuming, hence are difficult to implement on embedded systems due to low hardware capabilities. A consistent number of techniques and strategies have been proposed to reduce the network size in order to adapt it for embedded system. Weight pruning [4] is an effective approach, in which weights are pruned to achieve high compression ratios. Other techniques such as threshold setting and biased weight decay [5] could be integrated to the weight pruning procedure. Another approach to DNN compression is the low rank approximation of the weight matrix [6]. Lowering the precision of weights is also a straightforward technique to reduce both the model size and computation cost of DNNs. However, model compressing is not sufficient, other features need to be taken into consideration such as computing intensity (number of operations during network forward step).

To address all these limitations (e.g., algorithms performance on low processing embedded hardware) we propose a two-stage pipeline for person re-identification. The first stage consists of a DNN architecture based on DarkNET [7] for person detection and segmentation and the second stage consists of SIFT [8] with SVM [9] for feature extraction and person classification. The algorithm is deployed on a low processing embedded hardware system, namely Raspberry PI with the purpose of person re-identification. The experiments are conducted over a new proposed surveillance database named SPOTTER[1].

The main contributions of this paper are twofold: i) a new dataset for person re-identification that includes part occlusion and misalignments to allow the study of the effectiveness of the algorithms in a real case scenario; ii) a practical deep neural network architecture which can be deployed on a low processing embedded hardware system.

The rest of the article is structured as following. In section II we present our proposed two-stage pipeline. The proposed architecture contains a set of optimized operator kernels with a minimal runtime for the target device. In section III we present the newly re-id dataset and the experimental results. Given a dataset that allows simultaneous evaluation of detection and re-identification we show that localization ability of detectors plays a critical role in re-identification. The paper is concluded in Section IV with a summary of the key points of the work.

[1] http://www.campus.pub.ro/lab7/spotter/index.php

978-1-5386-2350-3/18/$31.00 ©2018 IEEE

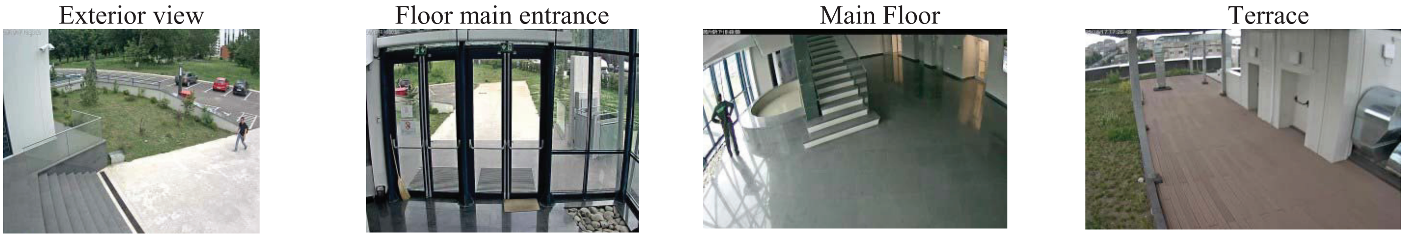


Figure 1. SPOTTER surveillance database samples.

II. Proposed System Arhitecture

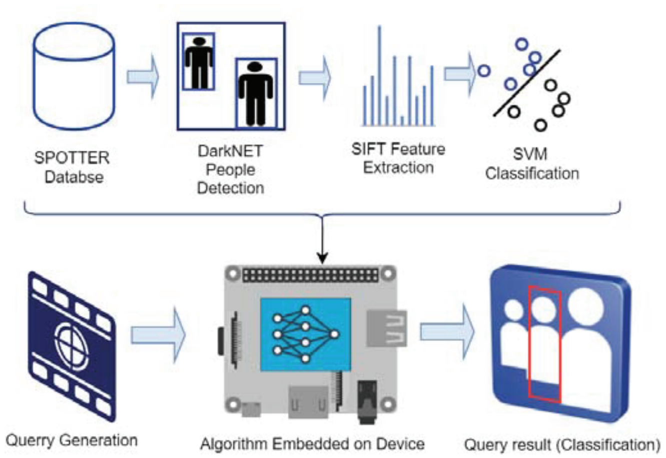


Figure 2. Illustration of the two-stage pipeline for person re-id deployed on embedded devices.

The proposed two-stage pipeline is depicted in Figure 2. The entire processing task takes place on Raspberry PI embedded platform. At the first stage, the DarkNET DNN is used for pedestrian detection and segmentation. Opposed to traditional pedestrian detection system (e.g., motion detection), this method provides superior performance as the network should retrieve all humans from images and exclude other moving objects. In the second stage we extract relevant representations from the body images using SIFT along with SVM for classification. Our proposed architecture achieved high performance running on Raspberry PI embedded platform that will be further discussed in the next subsections.

A. *Raspberry PI 3*

The embedded hardware employed to run our proposed architecture is a Raspberry Pi board. The unit is equipped with a Broadcom system on a chip (SoC) and an integrated ARM CPU with on-chip graphics processing unit (GPU). Processor speed for Raspberry PI ranges from 700 MHz to 1.2 GHz and on-board memory ranges from 256 MB to 1 GB RAM.

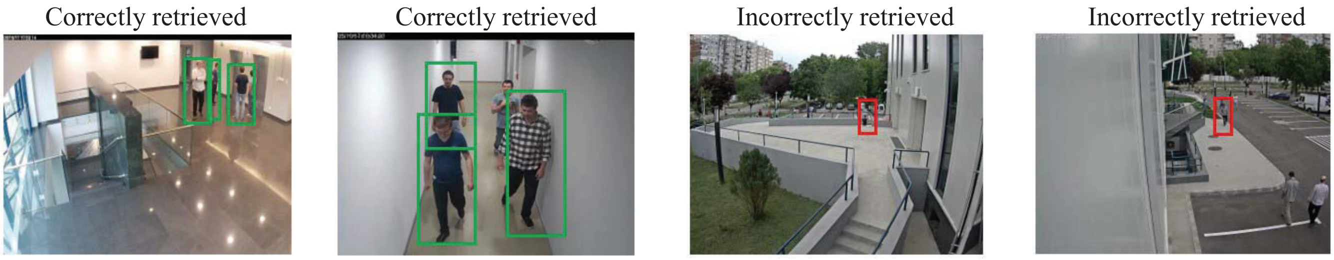
Secure Digital (SD) cards are used to store the operating system and programs memory. For current tests we adopted the Raspberry Pi 3 Model B as being the latest single-board computer from the Raspberry Pi Foundation which is running a 1.2Ghz 64-bit quad-core ARM processor with 1GB or RAM. The Raspberry Pi processing platform was selected as having a low hardware profile and low-powered, similar with CCTV video camera hardware available in surveillance industry. The Raspberry PI platform is running Raspbian OS and to make it faster the graphical interface was disabled during experiments.

B. *DarkNET Deep Neural Network*

The DarkNET architecture is composed of 9 convolutional layers followed by 2 fully connected layers. The network takes as input a full image which is then subdivided into a 7 × 7 grid. Each grid cell represents a classifier which is responsible for generating bounding boxes around potential pedestrians and class probabilities for those bounding boxes. Finally, we set a threshold over the detections and select the high scoring ones. The DarkNET model has several advantages over classifier-based systems. It looks at the whole image at test time, so its predictions are informed by global context in the image. It also makes predictions with a single network evaluation, unlike systems like R-CNN which require thousands for a single image. This makes it fast, more than 1000 times faster than R-CNN and 100 times faster than Fast R-CNN [10]. Neural networks such as state-of-the-art AlexNet [1] and SqueezeNET [11] are difficult to deploy on embedded systems, as they depend on many 3rd party libraries and processing platforms such as Caffe [12], Torch [13], OpenCV [14], etc. DarkNET on the other hand is entirely implemented in C/C++ programming which makes it attractive to use on embedded systems in terms of fast running time and lower hardware requirements.

C. *SIFT*

Color and texture information are important features for describing person images. The scale-invariant feature transform (SIFT) is used in this work to describe local features of extracted people by DarkNET. The algorithm uses invariant key-points which usually lie on high-contrast regions of the image, such as people body parts or cloths edges. The SIFT features are local and based on the appearance of the object at interest points and are invariant to a certain degree to image scale and rotation. Key-points are also robust to changes in illumination, noise, and minor changes in viewpoint. Also, they are relatively easy to extract and allow for correct object identification with low probability of mismatch.



Correctly retrieved Correctly retrieved Incorrectly retrieved Incorrectly retrieved

Figure 3. Query retrieval results including overlapping scenes and different camera views.

D. SVM

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In this work we selected a linear SVM kernel as is fast, low demanding in terms of processing power and achieves stat-of-the-art results in many classification-based tasks.

III. Experimental Results

A. SPOTTER Database

We conducted experiments on the new video re-id dataset named SPOTTER. The dataset contains 137,000 manually annotated frames summing more than 100 minutes of video at different image resolutions (800 × 600, 1280 × 720, 1280 × 800, 1280 × 960 pixels), which makes it feasible to evaluate both pedestrian detection and person re-identification. The videos are captured from different camera views with multiple positive samples for each probe in the gallery. The bounding boxes are not biased towards ideal situations, where pedestrians are well-aligned. The dataset includes part occlusion and misalignments to allow the study of the effectiveness of the algorithms in an actual real case scenario. The dataset is split into two parts: 70,000 images for training and 5,565 images for testing. We report the Mean Average Precision score for this dataset.

B. Parameters Tunning

For optimization we used the NNPACK [15] pack which is an acceleration package for neural network computations. NNPACK aims to provide high-performance implementations of convnet layers for multi-core CPUs and provides low-level performance primitives leveraged in leading deep learning established frameworks, such as Caffe and Torch. NNPACK is used to optimize DarkNET without using a GPU. It is useful for embedded devices using ARM CPUs and NEON architecture. We found that compressing the network model is not enough to run it efficiently on embedded platforms. Computing intensity needs to be taken into consideration. For example, in [9] the parameters are optimized to generate weights size at only 5 MB, however a forward pass through entire network performs 2.2 billion operations. The original work in [1] who started a revolution on adaptation of DNN performs 2.3 billion operations per forward pass. DarkNET reference model selected into our work performs only 800 million floating point operations, making it almost 3 times faster than previous ones.

The running time comparison of the proposed two-stages pipeline deployed on the Raspberry PI platform is shown in Table 1 (including subspace learning time). The analysis includes the DARKNET and Tiny DARKNET models. Tiny DarkNET represents the smallest model of the network that can be built while still being able to retrieve pedestrians (with respect to the precision). In processing 448 × 448 person images, the first stage, namely pedestrian detection and segmentation requires 46 seconds per image on average using the DarkNET model and 15 second using the Tiny Darknet model. After the optimization of the models using the NNPACK, the inference time decreased to 15 and 1.3 seconds for the DarkNET and Tiny DarkNET, respectively. After adding the second stage, the inference time slightly increases with 1.8 seconds per frame for the DarkNET model and 0.7 seconds for the Tiny DarkNET model.

Table 1. Processing time with and without optimization.

Model	**Running time per frame (seconds)**	
	Without optimization	***With Optimization***
Normal DarkNET	46 sec	7 sec
Tiny DarkNET	15 sec	1.3 sec
Model	**Running time per frame (seconds) of above with SIFT extraction and SVM classification added**	
	Without optimization	***With Optimization***
Normal DarkNET	47.6 sec	8.5 sec
Tiny DarkNET	16.8 sec	2 sec

C. Results

To assess retrieval performance, we used a global measure of performance, the Mean Average Precision (MAP), which is computed as the mean of the average precision scores for each query.

Table 2. Results obtained using the proposed architecture to classify 12 distinct people (classes) from SPOTTER Database.

	Class											
	1	2	3	4	5	6	7	8	9	10	11	12
MAP	0.42	0.41	0.46	0.87	0.19	0.41	**0.93**	0.81	0.22	0.92	0.27	0.52

In Table 2 we report the performance for each probe in the gallery. The top score is obtained for class 7 with a MAP score of 93% followed closely by class 10 and class 4 with 92% and 87% respectively. We observed that the top results were obtained by the classes with the color and texture information well defined. Several result samples returned by the proposed system are depicted in Figure 3.

Compared with the classical systems, the entire detection chain is based on DNN (detection, segmentation). In typical approaches, the humans are detected based on motion (background subtraction) which fails when many objects are moving as it generates significant more targets to be analyzed, therefore reducing overall system performance (load, time to process, complexity).

IV. Conclusions

In this article we proposed an architecture specialized for people classification which can be deployed on embedded hardware. The processing system is composed from the DarkNET DNN architecture which is used as pedestrian detector, combined with SIFT for feature extraction and SVM for final pedestrian classification. The entire processing system is implemented on an embedded device, namely Raspberry PI while obtaining high results.

We found that most of the established DNN (e.g., AlexNet, SqueezeNET, etc) are based on powerful processing libraries and frameworks such as Caffe, Torch, etc., that tend to be difficult to be deployed on embedded devices, in many situations even unable to run on Raspberry PI embedded platform due to high memory footprint and computational cost (RAM and CPU). Except the model compression, we found that the DNN architectures need to be optimized by reducing the convolution layers (which are one of the most demanding in terms of computing requirements) and optimized to be split and handled in parallel on all available CPU cores.

During our experiments we also trained DarkNET from scratch as well in in an end-to-end manner, however the results are unsatisfactory at this point (maximum MAP of 14.6 % was obtained for class 11).

Finally, we show that the efforts towards increasing the inference time by compressing the model of the network, degrade the accuracy of the system. In this regard, we plan to examine this aspect to establish better tradeoffs to allow the detection and re-identification to be processed in an end-to-end manner.

V. Acknowledgment

This work was supported by the Ministry of Innovation and Research, UEFISCDI, project SPOTTER, grant PN-III-P2-2.1-PED-2016-1065, agreement 30PED/2017 and partially supported by project SPIA-VA, grant PN-III-P2-2.1-SOL-2016-02-0002, agreement 2SOL/2017.

References

[1] K Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, CoRR, abs/1409.1556, 2014.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, CoRR, abs/1409.4842, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks, *In NIPS*, pages 1106-1114, 2012.

[4] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint arXiv:1510.00149, 2015.

[5] L. Y. Pratt. Comparing biases for minimal network construction with back-propagation. *Morgan Kaufmann Pub*, 1989, vol. 1.

[6] M. Denil, B. Shakibi, L. Dinh, N. de Freitas et al. Predicting parameters in deep learning. *In Advances in Neural Information Processing Systems*, 2013, pp. 2148–2156.

[7] Joseph Redmon. Darknet: Open Source Neural Networks in C, http://pjreddie.com/darknet/.

[8] Lowe, David G. (1999). Object recognition from local scale-invariant features (PDF). *Proceedings of the International Conference on Computer Vision*. 2. pp. 1150–1157. doi:10.1109/ICCV.1999.790410.

[9] S Cortes, Corinna; Vapnik, Vladimir N. (1995). Support-vector networks. *Machine Learning*. 20 (3): 273–297.

[10] Joseph Redmon, Ali Farhadi. YOLO9000: Better, Faster, Stronger. In *Computer Vision and Pattern Recognition* (cs.CV), arXiv:1612.08242v1.

[11] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. In *Computer Vision and Pattern Recognition*, arXiv:1602.07360v4.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the 22nd ACM international conference on Multimedia (MM '14). ACM*, New York, NY, USA, 675-678, 2016.

[13] Ronan Collobert and Koray Kavukcuoglu and Clément Farabet. Torch7: A Matlab-like Environment for Machine Learning in BigLearn. In *NIPS Workshop*, 2011.

[14] G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.

[15] Marat Dukhan. NNPACK: an acceleration package for neural network. computations. https://github.com/Maratyszcza/ NNPACK, 2016.