A Deep Neural Network Video Framework for Monitoring Elderly Persons

M. Farrajota(✉), João M.F. Rodrigues, and J.M.H. du Buf

Vision Laboratory, LARSyS, University of the Algarve, 8005-139 Faro, Portugal
{mafarrajota,jrodrig,dubuf}@ualg.pt

Abstract. The rapidly increasing population of elderly persons is a phenomenon which affects almost the entire world. Although there are many telecare systems that can be used to monitor senior persons, none integrates one key requirement: detection of abnormal behavior related to chronic or new ailments. This paper presents a framework based on deep neural networks for detecting and tracking people in known environments, using one or more cameras. Video frames are fed into a convolutional network, and faces and upper/full bodies are detected in a single forward pass through the network. Persons are recognized and tracked by using a Siamese network which compares faces and/or bodies in previous frames with those in the current frame. This allows the system to monitor the persons in the environment. By taking advantage of parallel processing of ConvNets with GPUs, the system runs in real time on a NVIDIA Titan board, performing all above tasks simultaneously. This framework provides the basic infrastructure for future pose inference and gait tracking, in order to detect abnormal behavior and, if necessary, to trigger timely assistance by caregivers.

Keywords: Design for aging · Design for quality of life technologies · Deep learning

1 Introduction

Deep learning methods have advanced greatly in recent years and they provide now the leading artificial vision framework for classification and categorization tasks. The deep learning architecture is inspired by the mammalian visual system, where simple processes are involved in the visual cortex through a recursive hierarchy [22]. Most deep architectures employ the multi-stage architecture studied by Hubel and Wiesel [11], composed by a hierarchy of layers where each layer consists of filtering, non-linearity and pooling stages.

In this paper we present a framework based on deep neural networks for detecting and tracking persons in a domestic environment, for telecare scenarios to aid elderly people at home, using one or multiple cameras. This framework is composed by three main tasks: (1) detection; (2) recognition; and (3) tracking. By employing a deep convolutional neural network (ConvNet) architecture for tasks (1) and (2), persons can be spotted using an algorithm for full-body

M. Antona and C. Stephanidis (Eds.): UAHCI 2016, Part II, LNCS 9738, pp. 370–381, 2016.
DOI: 10.1007/978-3-319-40244-4_36

pedestrian detection. A face recognition network allows to identify the persons in scenarios where multiple persons and their activities must be monitored.

Detection is done by using a sliding window search over the output features of the last convolutional layer in the network, where a classifier searches for persons and faces. To this end, a classifier scans for persons and faces over multiple scales of the feature map in parallel. This allows to process for arbitrary image region sizes in a single step over the network, thereby eliminating the need for feature computations over multiple scales. To recognize different persons, a Siamese network is used which compares detected faces retrieved from person detections against a database of faces belonging to all persons to be monitored in the specific domestic environment. After recognition, monitoring is reduced to tracking persons in consecutive frames by using another Siamese network which computes a binary matching classification between full-sized person detections in current and previous frames. Also, by computing image features only once and using them across the entire system networks, and by taking advantage of parallel processing of ConvNets on GPUs, the system runs in real time on a NVIDIA Titan GPU.

The main contribution of this paper is the integration into a single framework of detection, recognition and tracking tasks using a ConvNet model and to execute these tasks in a single, feed-forward pass over the network, thus reducing expensive, time-consuming computations in feature processing. This framework provides the basic infrastructure for future pose inference and gait tracking, to detect abnormal behavior and, if necessary, to trigger timely assistance by caregivers. In the next section we present the state of the art, followed in Sect. 3 by the framework, in Sect. 4 it is presented the experimental evaluation and finalizes (Sect. 5) with the conclusion and future work.

2 State of the Art

Deep convolutional neural networks (ConvNets/CNNs) have been used for many years [15] for tasks in category recognition of dominant objects in images: traffic signs [21], house numbers [19], handwritten characters [14], objects from the Caltech-101 dataset [12], or objects from the 1000-category ImageNet dataset [13,26]. These deep networks usually integrate low-,mid- and high-level features and classifiers in an end-to-end multilayer fashion [26] and the number of features (levels) can be increased by the number of stacked layers: the depth. The big advantage of ConvNets is that the entire system can be trained in an end-to-end fashion, from raw pixels to complex feature categories, therefore reducing or eliminating the need to handcraft suitable feature extractors, although this requires in practice a large amount of training data.

The latter aspect implies that accuracy on small datasets such as Caltech-101 has not been record-breaking, mainly because that such networks, with their millions of parameters, require more data in order to reduce over-fitting effects. A popular solution for such cases has been transfer learning: to use pre-trained features on much bigger datasets [17] and then to fine-tune the network on the

smaller dataset in order to increase the accuracy. Recent works have shown how well deep networks can learn features from data in a supervised [13] or unsupervised way [25], and state-of-the-art results have been obtained using optimized back-propagation learning algorithms on huge datasets [13,23].

The detection of persons and pedestrians (and faces) is a very challenging task due to the large variability caused by different poses, abundant partial occlusions, complex/cluttered backgrounds and frequent changes in illumination. In recent years, considerable progress in the development of approaches and applications has been obtained concerning object detection [8] and class-specific segmentation [20] in tracking scenarios [24], pedestrian detection being of particular interest [4]. Many existing state-of-the-art methods use a combination of bio-inspired [6] or hand-crafted features such as HoG [2], Integral Channel Features [4] and other variations and combinations [7], along with trainable classifiers such as boosted ones [4], SVMs [7] or random forests [3]. Although low-level features can be designed by hand with good success, mid-level features composed by combinations of low-level features are difficult to engineer without resorting to some sort of learning procedure. Multi-stage classifiers that learn hierarchies of features can be trained end-to-end with little prior knowledge. ConvNets are examples of such hierarchical systems.

Face recognition in unconstrained conditions has been extensively studied due to the availability of LFW [10], a very popular dataset for face recognition and an algorithm benchmark. Although recently there have been significant advances in the field of face recognition [18], implementing face recognition efficiently presents serious challenges with current approaches. The currently best performing face verification algorithms [18] use a two-stage approach that combines a multi-patch deep ConvNets and deep metric learning. They extract low-dimensional but very discriminative features for face verification and recognition. Hence, using ConvNets for feature extraction is the best strategy for face recognition. Finally, many approaches have been proposed to perform long-term visual tracking [24]. Also in this area, deep neural networks trained for general-purpose applications have been proposed for long-term tracking [5]. This typically requires scale-invariant feature extraction when the object dramatically changes in shape as it moves in the scene.

3 Framework

As mentioned in the Introduction, the framework consists of three main tasks: (1) detection; (2) recognition; and (3) tracking. In the next sections these tasks will be presented in detail.

3.1 Detection

Persons and faces are detected using a sliding window method over the image, similar to many popular methods on a ConvNet, e.g. [20]; see Fig. 1b. The method consists of two steps: (i) convolve the entire image with the feature

extraction layers of the ConvNet only once, thus avoiding expensive computations over multiple scales, and (ii) slide the classifier over the resulting feature map in multiple scales in order to detect persons and/or faces with different sizes. A fully supervised ConvNet model [13] is used for feature extraction and category (person/face/background) classification. The model is divided into two modules:

(a) *Feature extraction* is based on the convolutional features of an Alexnet. Each layer of the feature extraction consists generically of (a.1) convolution of the previous layer output (or, in the case of the 1st layer, the input image) with a set of learned filters; (a.2) passing the responses through the rectified linear function $\mathrm{ReLU}(x) = \max(x, 0)$; and (a.3) max pooling over local neighborhoods. We use only the feature layers of the Alexnet before the first fully-connected classification layer, and we also exclude the last max-pooling layer in our network.

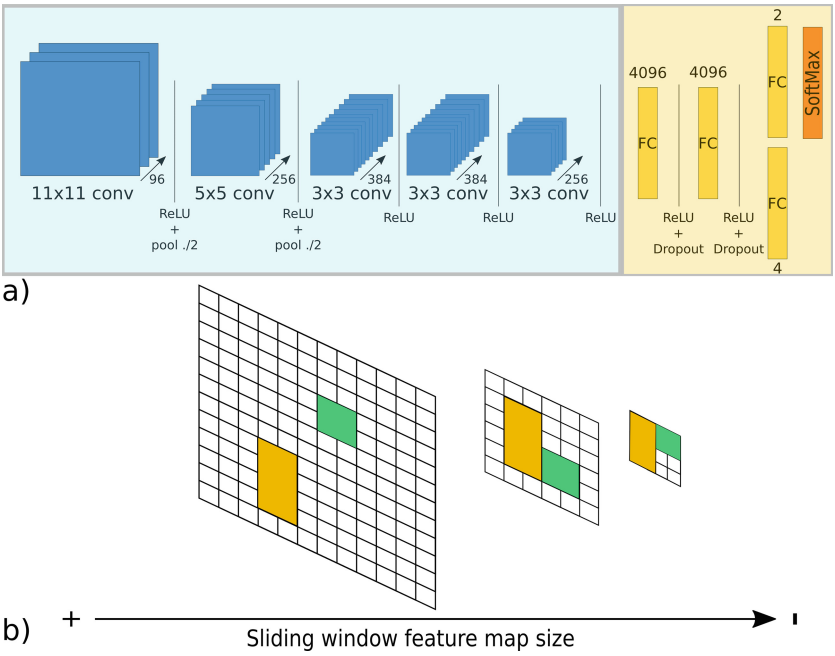

Fig. 1. Detection network architecture: (a) The network architecture used for classification/localization; (b) Sliding window scheme at multiple scales.

(b) For *Classification*, the top few layers of the network are conventional fully-connected (FC) networks and the final layer is a combination of a softmax classifier and a regression classifier during training. Those are converted to convolutional layers for normal system operation. The first two fully-connected layers have 4096

hidden units with 50 % dropout followed by the ReLU function. The output is connected to one FC layer with 2 outputs for classification, followed by a softmax layer and another FC layer with 4 outputs for bounding box estimation.

The convolutional layers accept arbitrary input sizes, and they also produce outputs of variable sizes. Since fully-connected layers require fixed-length vectors, they are replaced after training by convolutional layers. The result is that variable sized images can be processed in a single pass through the network, and computations are faster when evaluating the network.

Once the network is fine-tuned, detection amounts to little more than running a forward pass. The network takes as input a single-scale image, and several region proposals (RoI-region of interest; Fig. 1b color regions) are obtained after sliding the classifier over the last convolution layer (Fig. 1(a). The person/face ConvNet detector was designed for images of size 640×480 pixels. To classify/detect persons and faces, two classifiers with a fixed size are: one classifier window of 64×128 pixels for person detection, and 64×64 pixels for face detection. In order to classify/detect persons and faces at various sizes, the feature map is reduced and the previous classifiers are used. This is achieved by applying a 2×2 max pooling kernel over the feature map with a stride of 2 grid pixels, halving its size. This is done twice, resulting in probed regions by the classifiers of 128×256 and 256×512 pixels for person detection and 128×128 and 256×256 pixels for face detection. Feature maps smaller than the classifier are padded with zero values. This way, only one classifier needs to be trained, and scanning for larger faces and persons is quicker.

For each new image frame, the forward pass outputs a class posterior probability distribution and bounding box coordinates for each classifier, where all regions classified as background are then removed and non-maximum suppression to the output bounding boxes is applied in order to filter the strongest detections from the weakest (all bounding boxes which overlap at least 50 %). Although several classifiers are used for detection, their outputs are independent. Therefore, all classifiers can be applied in parallel when evaluating the network.

3.2 Recognition

For recognition we use two Siamese networks [1] to match detected faces and known persons. The two networks have the same architecture but are trained specifically for both tasks. Face recognition consist of matching detected faces in input images with face images from a database of known persons. Person recognition consists of matching a detected person in frames i and $i - 1$.

The Siamese network architecture (Fig. 2b) is composed of two networks with shared weights (Fig. 2(c) which are coupled at the top by two fully-connected layers to compute a binary (positive or negative) classification match between two inputs (see Fig. 2(a). Backpropagation is used to train the model using stochastic gradient descent and the negative log-likelihood loss. The networks are composed of the Alexnet feature extraction layers (Fig. 2b) as used in the previous section, followed by a spatial pyramid pooling layer (SPP) [9] with pooling

windows of size 1×1, 2×2 and 3×3, and at the end two fully-connected layers with 4096 hidden units with 50 % dropout and ReLU regularization. The two networks are coupled with a matching module composed of two fully-connected layers, one with 4096 hidden units with 50 % dropout and ReLU regularization connected to another one with 2 outputs for classification, followed by a softmax layer. Because of the SPP layer, inputs images can have different sizes and aspect ratios.

Although the architectures for face and person recognition are identical during training, their implementation is different. For person recognition, detected patches are pooled in the feature map and those are fed to the SPP layer. Thus, by using the same convolutional features as in the detection process, features can be reused in the recognition process. Extra feature computations can be avoided and only the top layers (SPP, FC's and matching layer) of the Siamese network need to be applied. Detections in previous frames are stored, and matching consists of computing a forward pass through the last layers of the network with computed features in the current frame. This results in little computational overhead.

For face recognition, detected face regions are sampled from the input image and scaled to 128×128 pixels in order to be compared with the faces in the database which have all been normalized to this size. Then, pairs of images to be compared are fed into the Siamese network and the matching is computed. However, in order to reduce additional computations, ConvNet outputs of all faces in the database have been preprocessed and stored in memory. Therefore, only an input frame must be processed by the network, and because the detection features are reused, recognition is a very fast process.

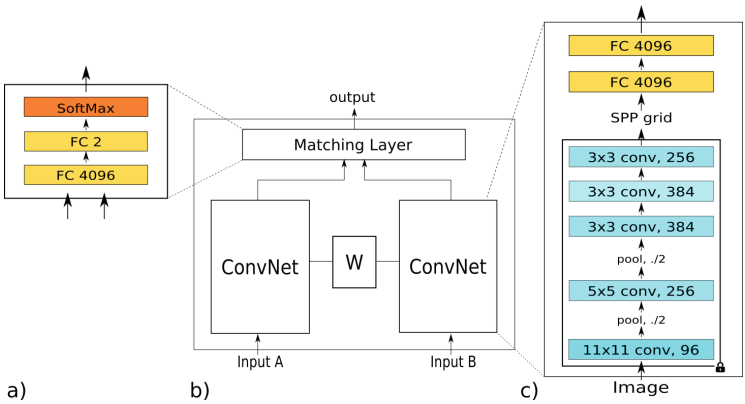

Fig. 2. Recognition architecture: (a) The Siamese matching layer; (b) The Siamese architecture for person/face recognition; c) Network architecture used for feature extraction.

3.3 Tracking

After a person has been detected and recognized, tracking only involved the detection window (bounding box). The procedure is as follows: (a) in frame i the person's bounding box is detected after the forward pass through the network and non-maximum suppression; (b) the person's features are matched with all detections in the previous frame $i-1$ using the Siamese network tuned to person matching, and the box with the highest but positive classification is selected; finally, (c) an additional constraint on the displacement between the current and the previous frames is applied. Only displacements smaller than a threshold are considered, and the average velocity and position of the detections are computed using the person's position in i, $i-1$ and $i-2$. This corresponds to the average trajectory and velocity of the person in the scene. This information also helps to keep track of occluded persons during several frames by predicting where the person may be located, assuming that the velocity remains constant. In case of mis-detections, i.e., occlusions or false negatives, the previous detections are used and the person's position is continually predicted up to a set number of frames (maximum 10), after which the detection windows stop being tracked and are discarded after that.

4 Experimental Evaluation

To train and evaluate the framework we used two datasets: (a) The Caltech pedestrian dataset [4] for pedestrian detection and tracking, and (b) the Labeled Faces in the Wild (LFW) dataset [10] for face classification and recognition.

4.1 Implementation Details

Pre-training: We used an Alexnet for feature extraction, which has been trained on Imagenet [17]. This is standard practice for deep networks, since the number of parameters is much larger than the available data for training a specific application and it provides a good starting point for the actual training. The network was trained on the ILSVR2012 [17] dataset with 1 million images of 224×224 pixels image.

Training: We used two datasets for person detection (Caltech) and face recognition (LFW). For detection, we used the Caltech typical category with 77,210 positive samples and 600 random negative samples for training, and 40,665 positive samples and 560 negative samples for testing. Of the LFW dataset, 10,000 faces were randomly sampled from the total of 13,233 faces for training, and the remaining 3,233 were used for testing. For the Caltech recognition training set, 10,000 pedestrian pairs were sampled from the dataset, 5,000 positive and 5,000 negative pairs. For testing we selected 2,000 pairs, 1,000 positive and 1,000 negative pairs. In case of the LFW dataset we followed the standard evaluation protocol defined for the "unrestricted" setting using no outside data. Here, the dataset was split into 10,586 training and 2,647 testing samples randomly.

Data generation: Data samples for training and evaluation were generated from ground truth bounding box annotations (Caltech) or images with normalized sizes (LFW). From the pedestrian detection dataset where ground truth bounding box annotations are available, sample regions with an intersect-over-union (IoU) [16] of at least 70 % overlap with the ground truth bounding box were used as positive samples. Those with overlaps ranging from 30 % to 50 % were used as negative samples. Additionally, negative samples with random sizes were randomly selected from a hard negative image set in order to scrutinize false positive rates. During training, we used a positive to negative data ratio of 1:2. Moreover, when a positive sample was selected, we applied a 50 % chance of flipping the sample label to negative. If a label was not flipped to negative (i.e., it stayed positive), we applied an additional 50 % chance of the sample having a full overlap with the ground truth bounding box plus additional padding with background pixels, or to have an overlap between 70 % and 100 % with the bounding box. Positive sample images with full overlap have a context ratio (padding) of the bounding box with the surrounding background. This context ratio ranges between 0 % and 100 % background padding: 0 % corresponds to no background pixels being added to the sampling region around the bounding box; 100 % corresponds to adding background pixels of half the height of the ground truth bounding box around the sampling region (top and bottom, left and right). In case of the LFW face detection/recognition dataset, there is no annotation information regarding ground truth bounding boxes. Since all face images are centered, the center pixels with $\{x_{min}, y_{min}, x_{max}, y_{max}\} = \{50, 50, 200, 200\}$ were used as ground truth coordinates, and the same positive and negative selection scheme as used for the pedestrian dataset was applied as well when training face classifiers. Random image crops were also used in the recognition task when generating data samples for training the Siamese network.

For face recognition, face samples were all resized to 128×128 pixels, and these were copied from images of 140×140 pixels with random shifts between 0 and 12 pixels, both horizontally and vertically. Color augmentation and image jittering were used to increase accuracy on all datasets. For color augmentation, we applied color casting to alter the intensities of the RGB channels in training data. Specifically, for each image, we randomly changed the R, G and B values up to ± 20 % with a 50 % chance. For image jittering, a 15-pixel maximum offset in the x and y directions was allowed, but this was only applied to samples with full overlap with a bounding box. For each image the global means of the R, G and B values were subtracted, after which the variances were normalized to 1.

Generated data containing bounding boxes of positive samples were used to train the bounding box estimation layer. We adopted the parameterizations of the four coordinates as in [8] for bounding box normalization: (1) $t_x = (x - x_a)/w_a$, (2) $t_y = (y - y_a)/h_a$, (3) $t_w = log(w/w_a)$, and (4) $t_h = log(h/h_a)$, where x, y, w and h denote the sampling window center coordinates and its width and height, whereas x_a, y_a, w_a and h_a denote the box's ground truth coordinates, and t_x, t_y, t_w and t_h are the normalized coordinates for regression.

Optimization Parameters: We used mini-batches of 256 samples, where positive and negative samples were randomly sampled with the same probability (50 % chance) from the training data. All network weights, which were not pre-trained on Imagenet, were randomly initialized and then updated using stochastic gradient descent with momentum of 0.9 and weight decay of 5×10^{-4}. The starting learning rate of 10^{-2} was reduced by a factor of 10 after the error converged (if it did not decrease further after 5 epochs) down to 10^{-4}. Dropout with a rate of 50 % chance was applied to the first and second fully connected layers of the classifiers. The detection classification layer was trained during 30 epochs with negative log-likelihood loss, and the regressor layer was trained during 100 epochs with mean-squared error loss. The Siamese networks for person and face matching were both trained during 100 epochs using the negative log-likelihood loss.

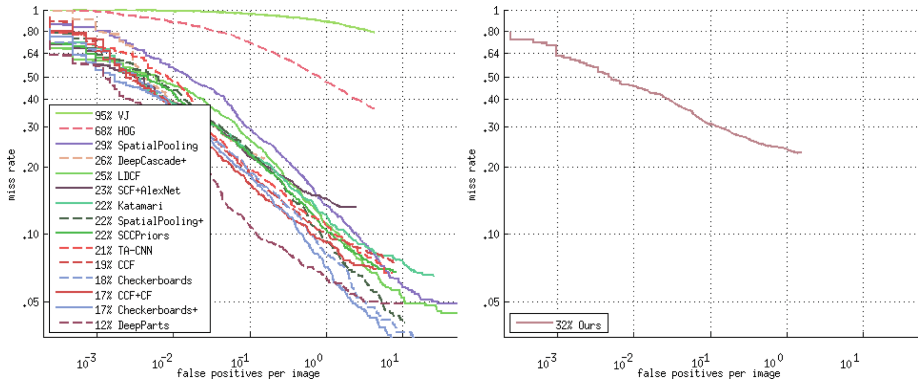

Fig. 3. Left, Available top-performing methods on the Caltech pedestrian dataset for large pedestrians benchmark [4], and (right) our method. Lower curves indicate better performance.

4.2 Results

The results were separated in pedestrian/person and face detection and recognition. Our person detection algorithm scores competitively against available top-performing algorithms specially tunned for pedestrian detection. Although not being amongst the top-performers, the trade-of between accuracy and speed necessary in our framework for fast detection of persons for tracking purposes minimizes this performance gap. In Fig. 3, a benchmark on the Caltech's pedestrian detection dataset [4] of our method ("Ours", burgundy line) against several available top-performing algorithms is shown.

In the case of face detection, our algorithm scores 93.1 % accuracy in the LFW dataset. Although this dataset was not designed for face detection benchmarking, the resulting accuracy shows our face detector's proof-of-concept. For our

recognition algorithms, our face algorithm scores 94.89 % accuracy on the LFW dataset in the unrestricted, no outside data category and the person algorithm scores 93.22 % in the adapted Caltech dataset for person comparison. Note that a comparison of our recognition methods against others authors was not possible to perform due to the lack of available benchmarking data for this particular setups. Regarding our tracker's performance, moving persons were successfully detected and matched in domestic environments with multiple users in occluded and non-occluded scenarios. In cases of heavy, but temporary occlusions, the system was able to track persons robustly and recover from most situations when users continue in the same trajectory as in previous detections.

5 Discussion

In this paper we presented a framework based on deep neural networks for detecting and tracking people in known environments using one or multiple cameras. Deep neural networks (ConvNets) provide very expressive features for vision tasks. By feeding video frames into a ConvNet and using a sliding window detector, faces and upper/full bodies can be detected in a single forward pass through the network with good accuracy and speed. This is an important step: by computing a frame's features only once, and sliding a classifier on top of the last feature layer, persons and faces can be detected at about 10 fps in 640×480 pixel images using a NVIDIA Titan GPU. Moreover, the same features can be used for face/person recognition by computing a binary matching classification between two faces or persons using a Siamese network, without any considerable impact on performance. The developed framework provides the basic infrastructure for additional improvements to be added for a timely assistance and aid for elderly people in case of an accident or other problem.

Therefore, future work will focus on pose prediction and gait estimation. This will enable an early detection of health symptoms related to the gait and pose of a person.

Acknowledgements. This work was supported by the FCT project LARSyS: UID/EEA/50009/2013 and FCT PhD grant to author MF (SFRH/BD/79812/2011).

References

1. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Conference CVPR, vol. 1, pp. 539–546 (2005)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. IEEE Conf. CVPR **1**, 886–893 (2005)
3. Dollár, P., Appel, R., Kienzle, W.: Crosstalk cascades for frame-rate pedestrian detection. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 645–659. Springer, Heidelberg (2012)
4. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral Channel Features, pp. 1–11. BMVC Press, Cambridge (2009)

5. Dundar, A., Bates, J., Farabet, C., Culurciello, E.: Tracking with deep neural networks. In: 47th Annual Conference CISS, pp. 1–5 (2013)
6. Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H.: Bio-Inspired pedestrian detection and tracking. In: 3rd International Conference on Advanced Bio-Informatics, Bio-Technology Environments, pp. 28–33 (2015)
7. Felzenszwalb, P.F., Girshick, R.B., Mcallester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. PAMI **34**, 1–20 (2009)
8. Girshick, R.: Fast R-CNN. In: IEEE Proceedings of the ICCV, June 2015
9. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visualrecognition. IEEE Trans. PAMI **37**, 346–361 (2015). IEEE
10. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07–49, Uni. Massachusetts, Amherst, 49(07–49), 1–11 (2007)
11. Hubel, D.H., Wiesel, T.N.: Receptive fields of single neurones in the cat's striate cortex. J. Physiol. **148**, 574–591 (1959)
12. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: IEEE Proceedings of the ICCV, pp. 2146–2153 (2009)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1–9 (2012)
14. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: NIPS, pp. 396–404 (1990)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. IEEE Proc. **86**, 2278–2323 (1998)
16. Nowozin, S.: Optimal decisions from probabilistic models: the intersection-over-union case. In: IEEE Proceedings of the CVPR, pp. 548–555. IEEE (2014)
17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV **115**(3), 211–252 (2015)
18. Schroff, F., Dmitry, K., Philbin, J.: FaceNet : a unified embedding for face recognition and clustering. In: IEEE Proceedings of the CVPR, pp. 815–823 (2015)
19. Sermanet, P., Chintala, S., LeCun, Y.: Convolutional neural networks applied to house numbers digit classification. In: Proceedings of the ICPR, pp. 3288–3291 (2012)
20. Sermanet, P., Eigen, D., Zhang, X., Mathieu, C., Fergus, R., LeCun, Y.: OverFeat : Integrated Recognition, Localization and Detection using Convolutional Networks. arXiv preprint, pp. 1–15 (2013). arXiv:1312.6229
21. Sermanet, P., Lecun, Y.: Traffic sign recognition with multi-scale convolutional networks. In: Proceedings of the International Joint Conference on Neural Networks, pp. 2809–2813 (2011)
22. Serre, T., Poggio, T.: A neuromorphic approach to computer vision. Commun. ACM **53**(10), 54–61 (2010)
23. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv, pp. 1–13 (2014)
24. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. IEEE Trans. PAMI **36**, 1442–1468 (2014)

25. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the International Conference on Machine Learning, ICML 2008, pp. 1096–1103 (2008)
26. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part I. LNCS, vol. 8689, pp. 818–833. Springer, Heidelberg (2014)