A Deep Recurrent Convolutional Neural Network based Subtraction model for Video

[1]Dr.C.Ravichandran, [2] Dr.S.Venkatesan, [3]Dr.P.Rajendran, [4]Dr.G.Paramaguru,

[5]Dr.Neetha Delphin Mary Kulandaiswamy.

[1]Professor, Department of Electronics and Communication Engineering,
[2]Assistant Professor, Department of Science and humanities,
[3]Assiatant Professor, Department of Mechanical Engineering,
[4]Associate Professor, Department of Mechanical Engineering,
[5]Assistant Professor, Department of Civil Engineering,
[12345]Tagore Institute of Engineering and Technology, Deviyakuruchi, Salem, India – 636112
[*]Corresponding Author: Dr.C.Ravichandran

Abstract: In this work, we present a novel background subtraction system that uses a deep Recurrent Convolutional Neural Network (CNN) to perform the segmentation. With this approach, the network parameters can be learned from data by training a single CNN that can handle various video scenes with RNN for understanding the context of a video frames by passing the output of one training step to the input of the next training step, along with the new frames. This gives a new approach to estimate background model from video. A spatial-median filtering is used as the post-processing of the network outputs. This method is tested on different data-sets, and the network outperforms the existing algorithms with respect to the average ranking over different evaluation metrics in real time processing.

Keywords: Background subtraction, Recurrent CNN, Video Segmentation

1. Introduction

Video data processing is very common in variety of video based applications where it is important to extract and process the relevant information. Since the background data in video streams is redundant information and it consumes large amount of storage and computing power during any process, it is essential to extract only the relevant meaningful information. Hence, segmentation in video sequences becomes an active research in the recent days. Many applications such as surveillance of videos, vehicular traffic analysis, object tracking, human activity recognition and capturing optical motion uses this technique to find the moving objects in the picture. The main process in segmentation is separation of fore ground from back ground. In order to model the background, factors like illumination changes, weather conditions in the scene and subtle changes in the backgrounds need to be analyzed critically which demands for more robust and adaptive background representation. Two issues in background subtraction are Change detection and motion detection.

Background subtraction is a widely used method to detect the moving objects in the videos obtained by static camera. The detection of moving objects in the source frame and current frame is called “background model” or “background image”. Background subtraction is done if the image would be a part of a video

stream. The main goal of the background subtraction process, hence, can be given in a nutshell as the detection of the objects in the foreground in a frame sequence obtained from one or more camera. Here, the detection of the foreground objects is by calculating the difference between the static background and current frame. Thus, sophisticated background subtraction algorithms that assure robust background subtraction under various conditions must be employed. Hence, the background subtraction is a pre-processing stage in all video data processing to remove the redundant data. The main difficulties in the background subtraction process are Illumination changes, Dynamic background, Camera jitter and Ghosts/intermittent object motion.

Deep neural networks like CNN become prominent player in learning visual representations [1]. There are more number of recent works have been proposed with CNNs in image classification [2,3,4,5,6,7], detection [8,9,10,11,12], face recognition [13, 14] etc. M. Babaee et al discussed novel background subtraction system that uses a deep Convolutional Neural Network (CNN) to perform the segmentation [15]. Most of these improvements focus on designing with an aim to learn features based on datasets [16]. The most important layers in CNNs are convolution layer and pooling layer. The convolutional layers convolve local image frames independently with multiple filters, and the responses are combined according to the coordinates of the image regions. The pooling layers summarize the feature responses. Both convolutional layers and pooling layers are computed not by considering other regions.

CNN is mainly implemented for feature xtraction, but it does not integrate the context information. This can be done in neural networks with recurrent connections. This combination is extensively studied in online handwriting recognition, speech recognition and machine translation. Recurrent neural networks (RNN), dated back late 80's, use the long-range context information captured by a fixed number of recurrent weights. RNNs have been used to applications in natural language processing [17], speech processing [18] and image processing [19]. A RNN is a Processing element with feedback connections among itself. Its hidden layer state is a function of all the previous states and hence it will keep all the past inputs inn its memory. Because of this, RNN is considered a network with deep in time and it can find the correlations between the input data at different states of the video sequence.

CNN models for video processing have been used for learning of 3-D spatio-temporal filters over raw sequence data and also on video segments for learning the frame representations with either optic flow or trajectory-based models. These models learn either a fully general time-varying weighting, or temporal pooling. In line with these CNN based models, we proposed a model with combination of CNN with RNN. Recurrent Neural Networks with long-range learning augments hidden state with nonlinear mechanisms to process by using simple memory-cell like neural gates.

The outline of this paper is as follows: In Section 2, Fundamentals of CNN and RNN are explained. It also explains the existing algorithm with CNN. In Section 3, the proposed approach for background subtraction with a combination of CNN and RNN. In Sections 4, obtained results are discussed with

existing methods followed by detailed discussion and analysis. Finally, in Section 5, conclusion of this work is presented.

2. Convolutional Recurrent Neural Network

In this proposed work, Recurrent convolution Neural Network (RCNN) is taken with five convolution layers, one recurrent layer and two fully connected layers. Here, CNN and RNN are to learn middle-level visual patterns and spatial dependencies between the middle level visual patterns respectively. In the last stage, fully connected layers are used to learn a global image representation. When the convolutional layers are more, abstract and robust patterns can be extracted more. With back propagation, the global representations will be transmitted back to RNN for improving the spatial dependency encoding. This will feed CNNs further to learn better middle-level and low-level features. Since RNN is suitable for one dimensional time sequences, the 2D images will be first scanned in four different scanning directions

(Left to right, right to left, top to bottom and bottom top). Then these four sequences will be given to RNN for spatial dependencies. The summation of the results of RNN will be processed by fully connected layers for global image representation. This is shown in Figure 1.

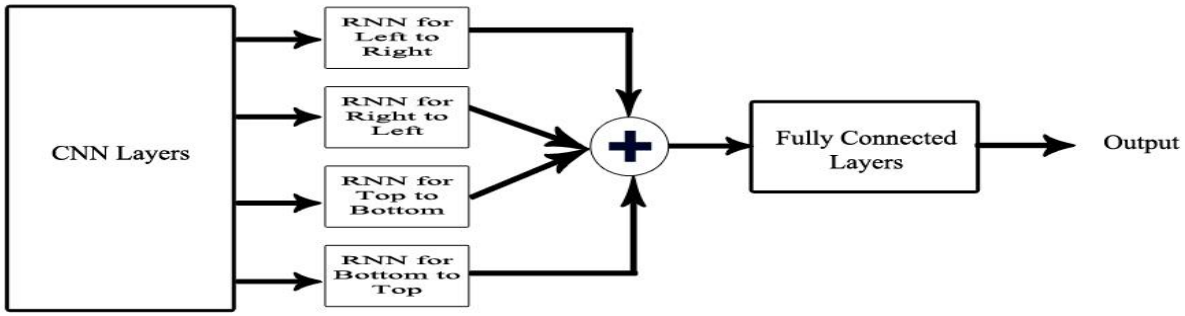

Figure 1: Block diagram of RCNN Framework

Recurrent Neural Network

In RNNs, the output of the previous hidden states is connected to the current states in the form of feedback loops to encode the contextual information of the video sequences.

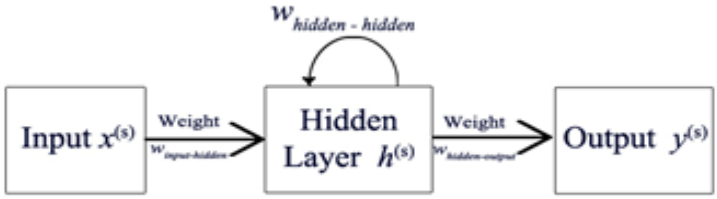

Fig. 2: General RNN structure

A typical RNN [20] is shown in Figure 2 with input layer of length S, hidden layer $\mathbf{h(s)}$, and the predicted output $\mathbf{y(s)}$ at the state $\mathbf{S \in [1, \dots, S]}$. The
Processing at hidden layer and the output are defined as

$$\boldsymbol{h^{(s)} = f_h\left(W_{hh} h^{(s-1)} + W_{ih} x^{(s)} Wihs(s) + b_h\right)} \quad (1)$$

$$y^{(s)} = f_o\left(W_{ho} h^{(s)} + b_o\right) \quad (2)$$

where $\mathbf{x^{(s)}}$ - the input data, $\mathbf{h^{(s)}}$ - the hidden layer units, $\mathbf{y^{(s)}}$ - the output
$\mathbf{W}_{ih}$, $\mathbf{W}_{hh}$ and $\mathbf{W_{ho}}$ - the transformation matrices
between $\mathbf{x^{(s)}}$ & $\mathbf{h^{(s)}}$, $\mathbf{h^{(s-1)}}$ &$\mathbf{h^{(s)}}$ and $\mathbf{h^{(s)}}$ & $\mathbf{y^{(s)}}$.
b_h and b_o - the constant bias terms, f_h and f_o - the non-linear activation functions.

RNNs will remember all the previously processed data by updating$\mathbf{W_{ih}}$, $\mathbf{W_{hh}}$ and$\mathbf{W_{ho}}$, in this work, RNN is used to learn the connections between image regions at different spatial positions called spatial dependencies. As RNN is more suitable for one dimensional data, image from CNN layer is converted into quad-directional spatial sequence to collect the context in all image regions.

Referring to (1) the quad-directional RNN is defined as:

$$\boldsymbol{h^{(s)}_{\rightarrow} = f_h\left(W_{hh\rightarrow} h^{(s-1)}_{\rightarrow} + W_{ih\rightarrow} x^{(s)} + b_{h\rightarrow}\right)} \qquad \boldsymbol{h^{(s)}_{\leftarrow} = f_h\left(W_{hh\leftarrow} h^{(s-1)}_{\leftarrow} + W_{ih\leftarrow} x^{(s)} + b_{h\leftarrow}\right)}$$

$$\boldsymbol{h^{(s)}_{\downarrow} = f_h\left(W_{hh\downarrow} h^{(s-1)}_{\downarrow} + W_{ih\downarrow} x^{(s)} + b_{h\downarrow}\right)} \qquad \boldsymbol{h^{(s)}_{\uparrow} = f_h\left(W_{hh\uparrow} h^{(s-1)}_{\uparrow} + W_{ih\uparrow} x^{(s)} + b_{h\uparrow}\right)} \quad (3)$$

$$\boldsymbol{h^{(s)} = h^{(s)}_{\rightarrow} + h^{(s)}_{\leftarrow} + h^{(s)}_{\downarrow} + h^{(s)}_{\uparrow}} \quad (4)$$

where $\boldsymbol{h^{(s)}_{\rightarrow}}$ - the left-to-right hidden layer units
$\boldsymbol{h^{(s)}_{\leftarrow}}$ -the right-to-left hidden layer units
$\boldsymbol{h^{(s)}_{\downarrow}}$ -the top-to-bottom hidden layer units and $\boldsymbol{h^{(s)}_{\uparrow}}$ - the bottom-to-top hidden layer units. The summation of these four is$\boldsymbol{h^{(s)}}$.

The weights of each unfolded step in forward and backward procedures of four directional RNN sequences are updated step by step as follows:

$$W^{(\Delta+1)}_{ih\rightarrow} = W^{(\Delta)}_{ih\rightarrow} + x^{(\Delta)} e^{(\Delta)}_{h\rightarrow} \propto \qquad W^{(\Delta+1)}_{hh\rightarrow} = W^{(\Delta)}_{hh\rightarrow} + h^{(\Delta-1)}_{\rightarrow} e^{(\Delta)}_{h\rightarrow} \propto \qquad W^{(\Delta+1)}_{ih\leftarrow} = W^{(\Delta)}_{ih\leftarrow} + x^{(\Delta)} e^{(\Delta)}_{h\leftarrow} \propto$$

$$W^{(\Delta+1)}_{hh\leftarrow} = W^{(\Delta)}_{hh\leftarrow} + h^{(\Delta-1)}_{\leftarrow} e^{(\Delta)}_{h\leftarrow} \propto \qquad W^{(\Delta+1)}_{hh\downarrow} = W^{(\Delta)}_{hh\downarrow} + h^{(\Delta-1)}_{\rightarrow} e^{(\Delta)}_{h\downarrow} \propto \qquad W^{(\Delta+1)}_{hh\downarrow} = W^{(\Delta)}_{hh\downarrow} + h^{(\Delta-1)}_{\rightarrow} e^{(\Delta)}_{h\downarrow} \propto$$

$$W^{(\Delta+1)}_{ih\uparrow} = W^{(\Delta)}_{ih\uparrow} + x^{(\Delta)} e^{(\Delta)}_{h\uparrow} \propto \qquad W^{(\Delta+1)}_{hh\uparrow} = W^{(\Delta)}_{hh\uparrow} + h^{(\Delta-1)}_{\rightarrow} e^{(\Delta)}_{h\uparrow} \propto \quad (5)$$

Where $e^{(\Delta)}_{h}$ the gradient of error propagated
From The output layer to the hidden layer
Δ - step size α - learning rate
In order to collect all the hidden units in the RNN layers, two fully connected layers which

are defined as follows:

$$g = f_g(W_{hg}H + b_g) \qquad y = f_y(W_{gy}g + b_y)$$

$$H = \left[(h^{(1)})^T, \ldots, (h^{(s)})^T, \ldots, (h^{(S)})^T\right]^T \quad (6)$$

where W_{hg} transfer matrix for the concatenated RNN outputs H to the global hidden layer g . W_{gy} transfer matrix for g to the predicted class label y H - the concatenation of all sequential states $h^{(s)}$ $(s = 1, \ldots, S)$

bg & b_y – the bias value fg & fy nonlinear activation function and softmax

3. Proposed method

The proposed method consists of an algorithm for background image generation, a novel RCNN for background subtraction and a median filter. The background image is used to perform background subtraction from the incoming frames. The foreground separation is done by combining the segmentation mask from SuBSENSE algorithm with Flux Tensor algorithm for detecting the motion changes in the video frames by the temporal variation of the optical flow field [15]. Post-processing is done with spatial-median filtering. This calculates the median over a neighborhood for each pixel in an image for the given kernel size. Finally, each pixel is mapped to {0, 1} by the thresholding.

$$f(x;T) = \begin{cases} 1 & if\ x > T \\ 0 & otherwise \end{cases} \quad (7)$$

where T is the threshold level.

The proposed algorithm is depicted in Fig. 3.

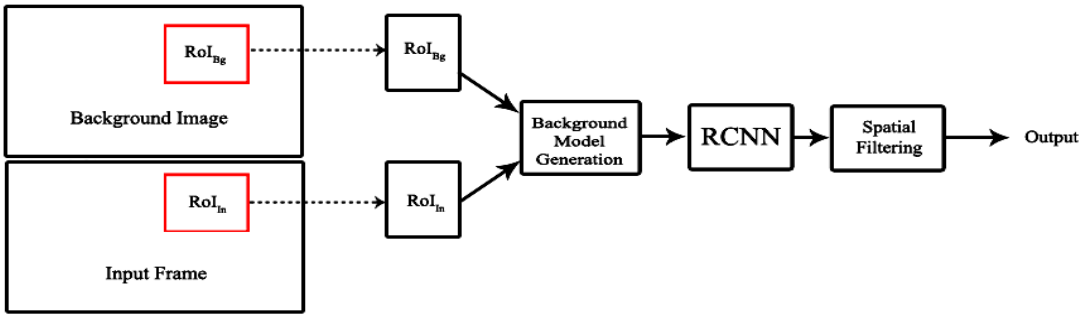

Fig. 3: Proposed method of Background subtraction with RCNN

4. Results and Discussion

The data-sets used in this proposed method is taken from CDnet 2014 [21], Wallflower [22] and PETS 2009 [23]. Here CDnet 2014 and Wallflower data set contain video sequences from different categories meant for the background subtraction task. The training batch size was 256, learning rate started from 0.01, momentum weight was 0.9, and both the RNN layer and the

fully connected layers were applied with dropout rate of 0.5. Figure 4 shows the video frames tested in this work.

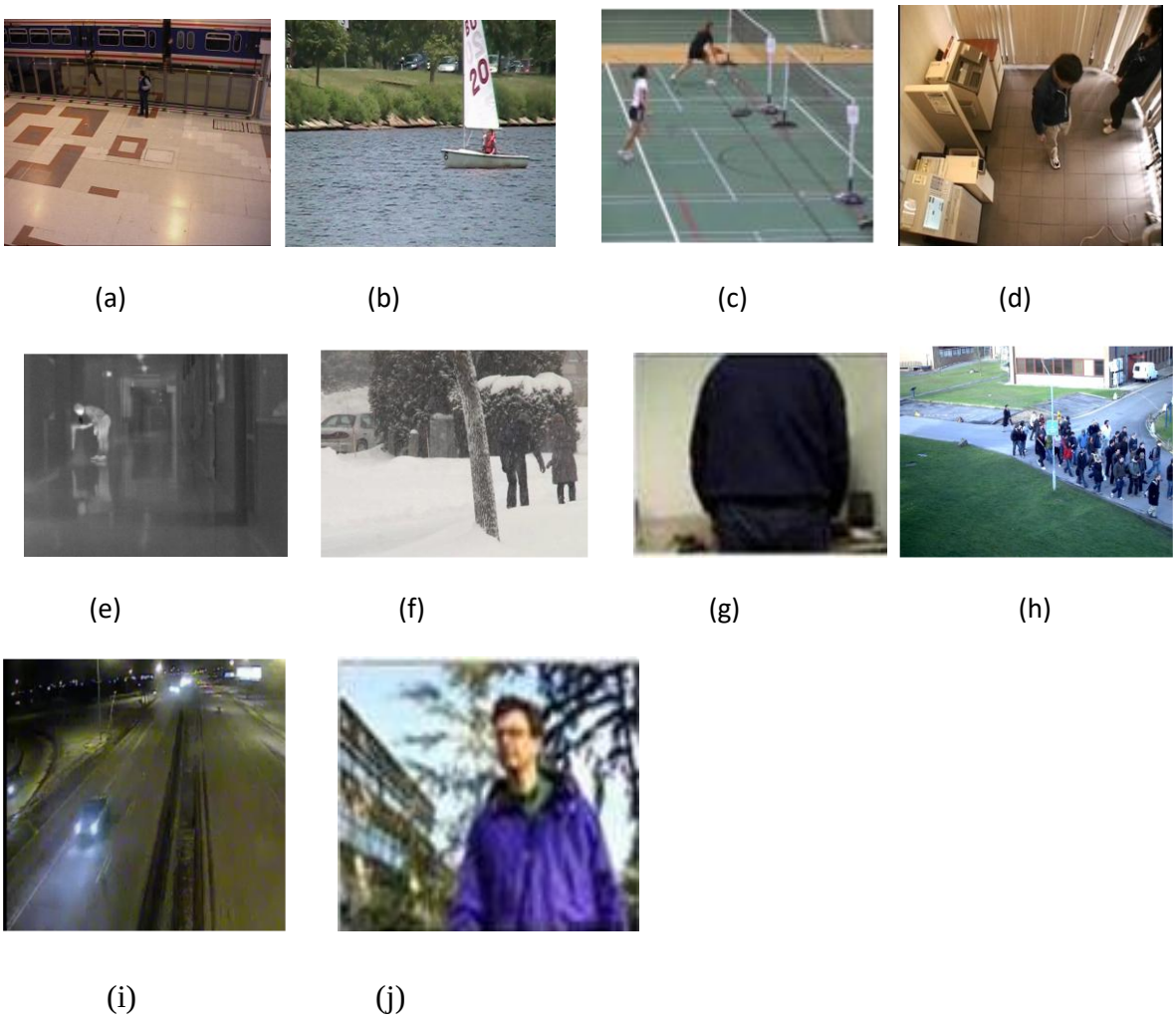

Fig. 4: Sample frame of test video sequence

(a) baseline (b) dynamic background (c) camera jitter (d) shadow (e) thermal (f) bad weather

(g) Camouflage (h) Foreground Aperture (i) night videos (j) Waving Trees.

Performance Metrics

The quality of a background subtraction algorithm is evaluated in this work by using standard F Measure. F Measure (FM) is defined as

$$F\ Measure\ (FM) = \frac{2PR}{(P+R)} \quad (8) \qquad P = \frac{TP}{TP+FP} \text{ and } R = \frac{TP}{TP+FN} \quad (9)$$

Where TP = True Positive, FP = False Positive, FN = False Negative

Table 1 Comparison of different Background Subtraction Algorithms

FM image **Method**	**FM** baseline	**FM** dynamic	**FM** camera jitter	**FM** Shadow	**FM** Thermal
RCNN	0.9623	0.8901	0.9045	0.9428	0.8210
CNN	0.9580	0.8761	0.8990	0.9304	0.7583
PBAS	0.9242	0.6829	0.7220	0.8143	0.7556
SuB SENSE	0.9503	0.8177	0.8152	0.8986	0.8171
GMM	0.8245	0.6330	0.5969	0.7156	0.6621

Table 2 Comparison of FM for the Wallflower Dataset

Method $\mathbf{FM_{Video}}$	**RCNN**	**CNN**	**PBAS**	**SuB SENSE**	**GMM**
Bad weather	0.8373	0.7071	0.2534	0.4090	0.5205
Camou flag	0.9989	0.9857	0.8922	0.9535	0.8307
Foreground Aperture	0.7120	0.6583	0.6459	0.6635	0.5778
Night video	0.6724	0.6339	0.4216	0.5211	0.4323
Waving Trees	0.9683	0.9546	0.8421	0.9597	0.9767

Table 1 and 2 shows the F Measure value of the segmented video frames coming from different algorithms for CDnet 2014 data sets and Wallflower Dataset. In Table 1, five categories of the video sequences are tested with existing algorithms GMM (Gaussian Mixture Model), SuBSENSE (Self-Balanced SENsitivity SEgmenter), PBAS (Pixel-Based Adaptive Segmenter), and CNN (Convolutional Neural Network) along with the proposed RCNN. Table 2 shows the comparison of FM values for five categories of Wallflower Dataset from GMM, SuBSENSE, PBAS, CNN and proposed RCNN.

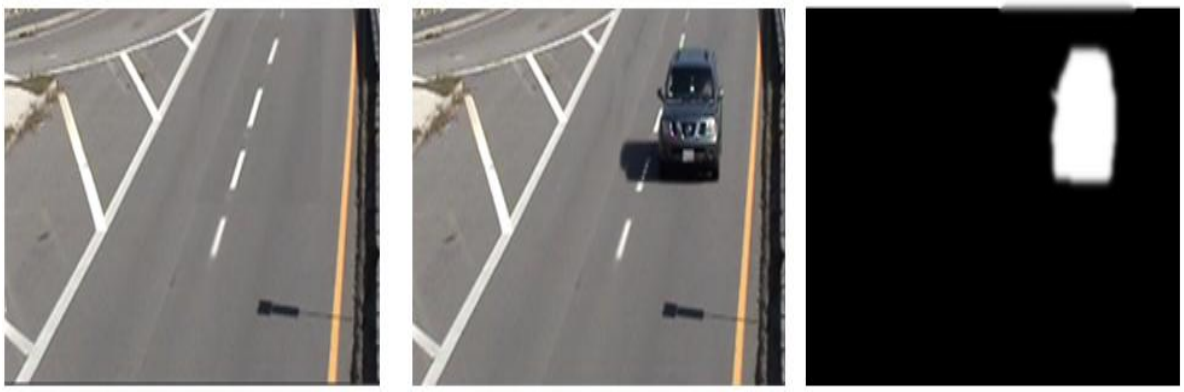

Figure 5: (a) Background image (b) Input frame (c) Network output

The output segmentations are more precise and not prone to any outliers as evident from Fig 5 and 6 for various data sets of different category. The FM values of RCNN are the best compared to other algorithms in various categories. For some categories like ‘Foreground Aperture’, the CNN yields poor results due to poor background subtraction but RCNN gives better than all. For Wallflower data-set, the proposed algorithm outperforms in terms of segmentation quality and FM especially for bad weather and foreground aperture category.

5. Conclusion

A Novel background subtraction algorithm based on Recurrent CNN is proposed in this paper for segmenting video in order to track the motion in video sequences for surveillance applications. The proposed RCNN based background subtraction algorithm outperforms compared to all existing algorithms GMM, SuBSENSE, PBAS and CNN in terms of segmentation quality.

Fig. 6: Comparison of the segmentation outputs: (a) the input images (b) Ground truth images (c) outputs of the CNN (d) outputs of RCNN

Reference

1. Babaeea M., Dinha D and Rigolla G.: *A Deep Convolutional Neural Network for Background Subtraction*, arXiv:1702.01731v1 [cs.CV] 6 Feb 2017.
2. Chatfield K., Simonyan K., Vedaldi A and Zisserman A.: *Return of the devil in the details: Delving deep into convolutional net,* In BMVC, 2014.
3. Ciresan D., Meier U. and Schmidhuber J.: *Multi-column deep neural networks for image classification,* In CVPR, 2012.
4. Deng J., Dong W., Socher R., Li L. J, Li K. and Fei-Fei L.,: *Imagenet: A large-scale hierarchical image database,* In CVPR, 2009.
5. Ferryman J and Shahrokni A.: *An overview of the pets 2009 challenge,* Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2009.
6. Girshick R., Donahue J., Darrell T and Malik J.: *Rich feature hierarchies for accurate object detection and semantic segmentation,* In CVPR, 2014.
7. Gong Y., Wang L., Guo R and Lazebnik S.: *Multi-scale order less pooling of deep convolutional activation features,* In ECCV, 2014.
8. Graves A and Schmidhuber J.: *Offline handwriting recognition with multidimensional recurrent neural networks,* In Advances in Neural Information Processing Systems (NIPS), 2008.
9. He K., Zhang X., Ren S and Sun J.: *Spatial pyramid pooling in deep convolutional networks for visual recognition,* In ECCV, 2014.
10. Le Cun B. B., Denker J., Henderson D., Howard R. E., Hubbard W and Jackel L. D.: *Handwritten digit recognition with a back-propagation network,* In NIPS, 1990.
11. Ouyang W., Luo P., Zeng X., Qiu S., Tian Y., Li H., Yang S., Wang Z., Xiong Y., Qian C., et al.: *Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection,* arXiv preprint arXiv:1409.3505, 2014.
12. Robinson T.: *An application of recurrent nets to phone probability estimation,* IEEE Transactions on Neural Networks, 5:298–305, 1994.
13. Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., and LeCun Y.: *Overfeat: Integrated recognition, localization and detection using convolutional networks,* arXiv preprint arXiv:1312.6229, 2013.
14. Simonyan K. and Zisserman A.: *Very deep convolutional networks for large-scale image recognition,* arXiv preprint arXiv:1409.1556, 2014.
15. Stoianov. I., Nerbonne. J and Bouma. H.: *Modelling the phonotactic structure of natural language words with simple recurrent networks,* In Computational Linguistics in the Netherlands, 1997.
16. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V and Rabinovich A.: *Going deeper with convolutions,* arXiv preprint arXiv:1409.4842, 2014.
17. Sun Y., Wang X. and Tang X.: *Deep learning face representation from predicting 10,000 classes,* In CVPR, 2014.
18. Taigman Y., Yang M., Ranzato M and Wolf L.: *Deepface: Closing the gap to human-level performance in face verification,* In CVPR, 2014.
19. Toyama K., Krumm J., Brumitt B and Meyers B.: *Wallflower: Principles and practice of background maintenance,* In Computer Vision (1999), The Proceedings of the Seventh IEEE International Conference on IEEE 1999, vol. 1, p. 255–261.
20. Wang Y., Jodoin P.M., Porikli F., Konrad J., Benezeth Y and Ishwar P.: *Cdnet 2014: an expanded change detection benchmark dataset,* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, p. 387–394.
21. Zeng X., Ouyang W., Wang M and Wang X.: *Deep learning of scene-*

specific classifier for pedestrian detection, In ECCV, 2014.
22. Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang and Yushi Chen.: *Convolutional Recurrent Neural Networks: Learning Spatial Dependencies for Image Representation*, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015.