Moving Towards in Object Recognition with Deep Learning for Autonomous Driving Applications

Ayşegül Uçar, Yakup Demir
Department of Mechatronic Engineering, Elazığ, Turkey
Department of Electrical Electronics Engineering, Elazığ, Turkey
{agulucar,ydemir}@firat.edu.tr

Cüneyt Güzeliş
Department of Electrical Electronics Engineering, Izmir, Turkey
{cuzelis.guzelis}@yasar.edu.tr

***Abstract*—Object recognition and pedestrian detection are of crucial importance to autonomous driving applications. Deep learning based methods have exhibited very large improvements in accuracy and fast decision in real time applications thanks to CUDA support. In this paper, we propose two Convolutions Neural Networks (CNNs) architectures with different layers. We extract the features obtained from the proposed CNN, CNN in AlexNet architecture, and Bag of visual Words (BOW) approach by using SURF, HOG and k-means. We use linear SVM classifiers for training the features. In the experiments, we carried out object recognition and pedestrian detection tasks using the benchmark the Caltech 101 and the Caltech Pedestrian Detection datasets.**

Keywords— Convolutions Neural Networks; Bag of visual Words; Support Vector Machines

I. Introduction

Object detection and recognition are the most important research topics in autonomous driving applications because a control action is applied with respect to the object firstly being detected and later being recognized. Recently, the applications of object recognition to vehicles in the real life have rapidly grown. Some examples include a lane departure warning and lane-keeping assist system detecting white line, detection of obstacles in front of the vehicle using stereo images, a pedestrian detection warning system using infrared images [1-4], and the detection of vehicles on the road ahead using a combination of laser radar and single lens camera system [5].

The main aim of autonomous vehicle applications is to detect, track and recognize static and dynamic objects such as animals, cars, trucks, motorbikes, and pedestrians. Object recognition is one of the challenges in the field of computer vision. In the literature, it was developed several kinds of feature extraction algorithms such as Histogram of Gradient (HOG), [6], Scale Invariant Feature Transform (SIFT) [7,8], The Speeded up Robust Feature (SURF) [9], Binary Robust Independent Elementary Feature (BRIEF) [10], the Oriented Fast and Rotated BRIEF (ORB)[11], and the Binary Robust Invariant Scalable Keypoints [12] (BRISK), and Fast Retina Keypoint (FREAK) [13].

On the other hand, it was proposed many classification algorithms such as Spherical/Elliptical classifiers [14], Support Vector Machines (SVMs) [14], Extreme Learning Machines (ELMs) [15, 16], Adaboost, Naïve Bayes, and Decision Forests [17-18] to detect and recognize objects. In the recent years, deep learning methods have emerged as a powerful machine learning method for object detection and recognition [19-21]. Deep learning methods are different from all traditional approaches. They automatically learn features from raw pixels directly and in a fast way more complex models comparing to shallow ones using the manually designed features [19]. In deep learning methods, local receptive fields grow in a layer by layer manner. The low-level layers extract fine features such as line, border, and corner while higher-level layers exhibit higher features such as object portions, like pedestrian parts, or the whole object, like car, traffic signs. In other words, they allow for representing an object in different granularities. Their successes are presented on the challenging ImageNet classification task across thousands of classes [22, 23] by using Convolutional Neural Networks (CNNs) called as a kind of deep neural networks. It was presented that CNNs outperform recognition performances of conventional feature extraction methods.

The contributions of this work are to develop powerful recognition and detection algorithm applying a decision fusion over the obtained features by using the proposed CNN architectures. The main objective of the proposed deep networks is to provide high accuracy and up seep.

In this study, we developed two CNN architectures. First one is architecture with ten layers, and the other is one like Alexnet architecture with nine layers. We performed detailed experiments to evaluate the proposed CNNs. To comparatively exhibit the success of deep learning, we applied also BOW approach by using SURF, HOG, and K-means.

In the rest of this paper is organized as follows. In Section 2, the proposed CNN architecture and Bag of visual Words (BoW) approach are introduced. In Section 3, the proposed algorithm is introduced. The experimental results are shown in Section 4. Finally, the paper is concluded in Section 5.

II. Methodology

A. Deep Convolutional Neural Networks

CNN is a kind of feed-forward neural networks consisting of multi layers. The outputs and inputs of each layer are

978-1-4673-9910-4/16/$31.00 ©2016 IEEE

represented by sets of image arrays. CNNs are composed in terms of various combinations of the convolutional layers, local or global pooling layers combining the outputs of neuron clusters, and fully connected layers with pointwise nonlinear activation function. CNNs use spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers as depicted in Fig.1. The layer building blocks of CNN are briefly explained below as:

Convolutional Layer: The layer performs the main workhorse operation of CNN [19, 20]. The input image is convoluted by using a set of learnable filters or kernels. Each produces one feature map in the output image. The feature maps are fed as input data of the second convolutional layer.

Pooling Layer: The layer reduces the feature dimension. The input images are partitioned into a set of non-overlapping rectangles. Each region is down–sampled by a non-linear down-sampling operation such as maximum or average. The layer achieves faster convergence, better generalization, small invariance to translation and distortion. The layers are usually located between successive convolutional layers to reduce the spatial size.

Rectified Linear Units (ReLU) Layer: The layer includes units employing the rectifier. Given a neuron input, x, the rectifier is defined as $f(x)=\max(0,x)$ in the neural networks literature. The rectifier function increases nonlinearity of the decision function. The gradient of its active unit is 1, the others are zero. There is not a gradient vanishing problem in backpropagation. The most features are enhanced, and the others are suppressed. The sparse models are obtained. A smooth approximation to the rectifier is $f(x)=\ln(1+e^{x})$ that its derivative is logistic function. This function is used to accelerate learning.

Fully Connected Layer: The layers are put through after the convolutional layers, pooling layers, and ReLU layers are located. The neurons of these layers are fully connected to all activations in the previous layer. The layer is accepted as final pooling layer feeding the feature to classifier. Matrix multiplication and a bias addition are used to calculate their activations.

Loss layer: The trade-off between predicted and real class values is determined the defined function in this layer. The layer is usually a final layer of the network. There are different loss functions for different applications. Euclidean loss function is usually used for applying regression by using real valued outputs in range from $-\infty$ and ∞. Softmax loss function is applied for determine the label assigned with a single class of M mutually exclusive classes. Sigmoid cross-entropy loss function is used for estimating M independent probability outputs in the interval of 0 and 1.

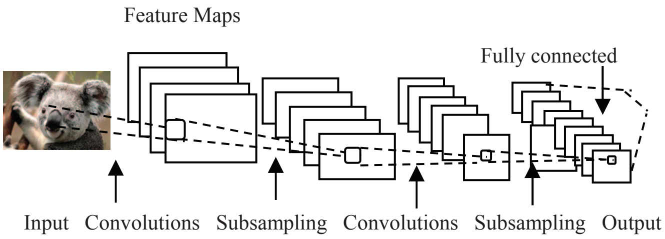

Fig. 1. A typical Conventional Neural Networks architecture

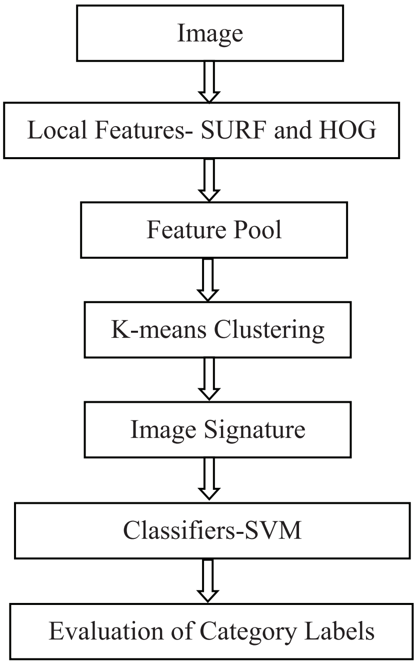

Fig. 2. Schematic of Bag of Word algorithm

B. Bag of Visual Words

In object recognition applications, the BoW approach has provided several advantages such as fast run time, high recognition accuracy, and less computational load. In the BoW approach, the image is considered as a text. Hence ‘words’ defining the text are determined. The steps of approach are performed as follows. The image features are detected, and feature keypoints are extracted. In this step, a feature detector can be used or can be defined a grid to extract feature descriptors. Then, it is obtained a visual word dictionary called as codebook by reducing the number of features through quantization of feature space using a clustering algorithm. Fig. 2 gives the block schema of BOW approach. There are a few descriptor methods [6-13]. In this paper, we used SURF and HOG to obtain feature keypoints and feature descriptors, respectively. We used multiclass linear SVM classifier to recognize the encoded features from each image category.

III. The Proposed Algortihm

The block schema of the proposed CNN detection and recognition algorithm is given in Fig. 3. The algorithm is applied at eight folds:

(1) Divide to nine patches all images as in Fig. 4,

(2) Up/down resize to 64x64x3 each patch and convert to grey image,

(3) Construct a CNN with ten layers consisting of input, convolutional, max pooling, convolutional, average pooling, convolutional, convolutional, and softmax regression,

(4) Apply stochastic gradient descent to the proposed CNN for applying to each image patch.

(5) Extract the features from full convolution layer,

(6) Determine distinctive feature sets by applying Principal Component Analysis (PCA)

(7) Use linear multi class SVM classifier for training the features,

(8) Perform the decision fusion to the outputs of nine SVM classifiers.

In addition, we constructed a network with nine layers like AlexNet and applied to the proposed algorithm. Moreover, we performed BOW approach to each path with 8x8 grid and 800 code. We applied also then it 5-7 steps of the algorithm.

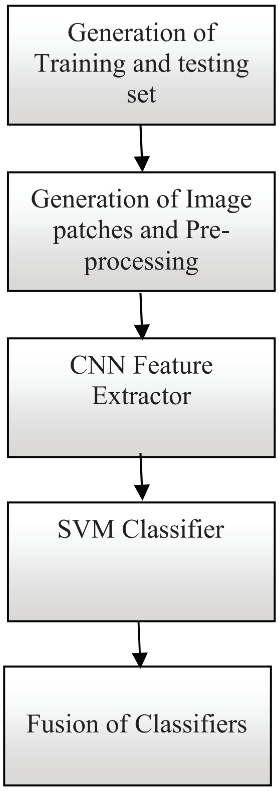
Generation of Training and testing set
Generation of Image patches and Pre-processing
CNN Feature Extractor
SVM Classifier
Fusion of Classifiers

Fig. 3. The block schema of the proposed algorithm.

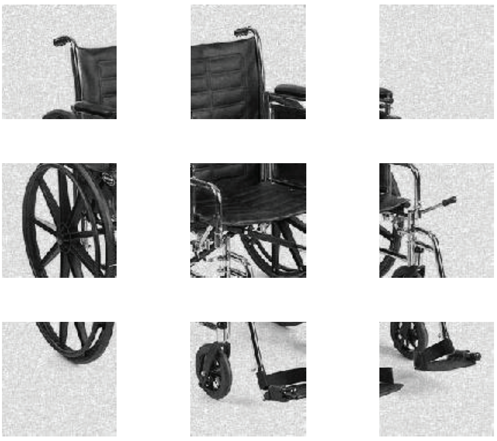

Fig. 4. Some crop examples of a chair image from Caltech-101 dataset.

IV. Experiments

In this section, the proposed framework is evaluated on two datasets: Caltech-101 [24] and Caltech pedestrian [25]. We carried out experiments by using MatConvNet and Vlfeat toolboxes in MATLAB [26, 27]. MatConvNet is computer vision toolbox allowing Graphical Processing Unit (GPU) by achieving CUDA support. In experiments, we used both CPU and GPU thanks to NVIDIA GTX 550.

A. Experimental Results on the Caltech-101 Dataset

In the first experiments, the Caltech-101 database is used since it has the most diverse object database and large inter-class variability [24]. The dataset includes 101 object categories of 9144 images. The image number in each class range contains from 31 to 800. The database contains both rigid objects such as cars, wheel chairs and bikes and non-rigid such as lions, cats, and flowers. Some images from the database are shown in Fig. 5. Most of them are about 300 × 300 pixels. We adjust training and testing set with respect to experimental setup procedure of Fei and Fergus et al. [24, 28] for a fair comparison. We conducted the training set by randomly selecting 15 or 30 images per class and the test set on up to 50 images per class since some categories are very small. We separated into 9 patches from corners and centre each of images and then up/down sampled them to the image size 64x64x3. Fig. 4 shows an example of 9 crops. All images were converted to grey scale. All of the images were added as an array into a matrix. Each image was subtracted from the per-pixel mean across all images to get its normalized image.

We tried two different network architectures. In the first architecture, a network with 10 layers consisting of layers called as input, convolutional, max pooling, convolutional, average pooling, convolutional, convolutional, and softmax regression was constructed. In the second architecture, we constructed a network similar to that of the architecture of AlexNet [21,29]. Alexnet has five convolutional, some of which are put through by max-pooling layers, and three fully connection layers. We used 64x64 input patches. We applied stochastic gradient descent with minimum batch size of 236. We selected learning rate for weights and biases as 0.001 and 0.02, respectively. After being trained in CNN, we applied Principal Component Analysis (PCA) to the outputs of full convolution layer. We fed the reduced features as the input to the SVM classifier with linear kernel. We applied one–against-rest method for multi-class classification. We used 5-fold cross validation for determining regularization parameter in the range from 2^{-10} to 2^{10}. The outputs of SVM classifiers for each path were combined a decision fusion rule by using a weighted majority voting rule and recognized an object.

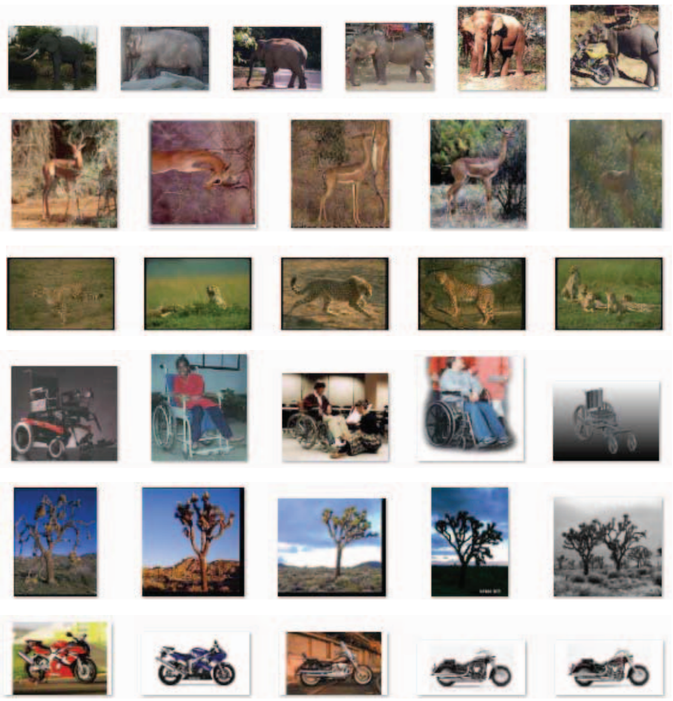

Fig. 5. Some examples from Caltech-101 dataset [24].

All experiments were repeated ten times with different randomly selected training and test images, and the average of per-class recognition rates is recorded for each run. Table I gives comparatively classification rates with the best results in the literature for different algorithms. In Tables, Proposed Single Network-1 presents the proposed CNN architecture with 224x224 of input size without any patch. Single AlexNet-SVM-2 presents the CNN architecture with 224x224 of input size without any patch.

Our best result is 92.80±0.43%. Especially pre-trained ALexNET-SVM architecture achieves to converge faster.

TABLE I. PERFORMANCE COMPARISON ON CALTECH 101 DATASET

Method	Accuracy (%) 15/Class	Accuracy (%) 30/Class
Bo et. al [30]	-	81.40±0.33
Yang et. al [31]	73.20	84.30
Zeiler &Fergus [32]	83.80±0.50	86.50±0.5
He et. al [33]	-	91.40±0.70
Chatfield et. al [34]	-	88.35 ± 0.56
Proposed Single Network-1	84.93±0.50	91.13±0.92
Proposed Fusion-1	89.80±0.50	92.80±0.43
Single AlexNet-SVM-2	84.80±0.45	86.80±0.43
Proposed Fusion-2	82.01±0.22	88.66±0.43
Proposed BOW	75.60±0.55	82.80±0.41

B. *Experimental Results on the Caltech- Pedestrian Dataset*

In the second part of experiments, we used Caltect pedestrian dataset for pedestrian detector. The dataset was collected over 11 sessions being used 640x480 30Hz video taken from a vehicle in an urban environment. We used for subsets set00-set05 for training and subsets set6-set10 for testing. The labels and evaluation code in [25] were used. As in [25], we used log-average miss rate to summarize the performance of detector. Table II gives the comparison of the log-average miss rate with state-of-the-art models [6, 36-38]. The experiments demonstrate that the proposed deep model outperforms the state-of-the-art algorithms.

TABLE II EXPERIMENTAL COMPARISON ON CALTECH PEDESTRIAN DATASET

Method	Log average missing rate (%)
Dalal &Triggs [6]	66
Walk et.al [35]	48
Hoang et. al [36]	54
Yan et. al [37]	37
Angelova et. al [38]	26.1
Proposed Single Network-1	41
Proposed Fusion-1	36.12
Single AlexNet-SVM-2	39.4
Proposed Fusion-2	37.2
Proposed BOW	47

V. Conclusions

This paper presents a new object recognition and pedestrian detection algorithm. The important steps of this algorithm are to divide into nine patches the image and to extract features relating to each patch. Feature extraction is carried out by using both different CNN architectures and BOW approach. Pre-trained AlexNet with nine layers and a large CNN architecture with ten layers are proposed for recognition and detection algorithm. In BOW, feature detectors and descriptors are obtained by SURF and HOG. The visual codes are obtained by using K-means. In the algorithm, the reduced features are fed to SVM classifiers. The outputs of SVM classifiers are fused by using majority voting rule. The recognition and detection performances were obtained on Caltech 101 and pedestrian datasets. The results have been shown to significantly better those in the literature.

REFERENCES

[1] "Mobileye Pedestrian Collision Warning System www.mobileye.com/technology/applications/pedestriandetection/pedestrian-collision-warning/."

[2] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection - a practical example of automatic emergency braking," in Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2010.

[3] "BMV Driving Assistance Package http://www.bmw.com/."

[4] "VW Emergency Assistance System http://safecarnews."

[5] "http://www.toyota-global.com/innovation/safety_technology/toyota-safety-sense/."

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[7] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," 2004.

[8] D. Lowe, "Object recognition from local scale-invariant features," In, in Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157, 1999.

[9] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in Proceedings of the (ECCV), 2006, pp. 404–417, 2006.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in Proceedings of the (ECCV), pp 778–792, 2010.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," 2011.

[12] S. Leutenegger, M. Chli, and R. Siegwart. Brisk: "Binary robust invariant scalable keypoints," 2011.

[13] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[14] A. Uçar, Y. Demir, and C. Güzeliş, A penalty function method for designing efficient robust classifiers with input–space optimal separating surfaces," Turk J Elec Eng & Comp Sci; vol. 22, pp. 1664–1685, 2014.

[15] A. Uçar, Y. Demir, and C. Güzeliş, "A new facial expression recognition based on curvelet transform and online sequential extreme learning machine initialized with spherical clustering," Neural Computing and Applications, vol. 27. pp. 131-142, 2016.

[16] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," Neurocomputing, vol. 70, pp.489–501, 2006.

[17] T. K. Ho, "The Random Subspace Method for Constructing Decision Forests" IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, pp. 832–844, 1998.

[18] G. I. Webb, J. Boughton, and Z. Wang, "Not So Naive Bayes: Aggregating One-Dependence Estimators," Machine Learning (Springer), vol. 58, pp. 5–24, 2005.

[19] L. Deng and D. Yu, "Deep Learning: Methods and Applications," Foundations and Trends in Signal Processing vol.7. pp. 3–4, 2014.

[20] S. Christian, A. Toshev, and D. Erhan. "Deep neural networks for object detection," Advances in Neural Information Processing Systems, 2013.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Neural Information Processing Systems, pp. 1097–1105, 2012.

[22] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. ILSVRC-2012, http://www.image-net.org/challenges/LSVRC/2012/.

[23] A. Berg, J. Deng, and L. Fei-Fei. "Large scale visual recognition," www.imagenet.org/challenges. 2010.

[24] L. Fei-Fei, R. Fergus, P. Perona, "One-Shot learning of object categories," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, pp. 594–611, 2006.

[25] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection:an evaluation of the state of the art," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, pp. 743–761, 2012.

[26] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 4

[27] A. Vedaldi and K. Lenc, "MatConvNet – Convolutional Neural Networks for MATLAB. In ACMMM, 2015.

[28] L. Fei-Fei, R. Fergus, and P. Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," In IEEE CVPR Workshop of Generative Model Based Vision, 2004.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S., Satheesh, S. Ma, and L. Fei-Fei, "Imagenet large scale visual recognition challenge. International Journal of Computer Vision, pp.1-42, 2014.

[30] L. Bo, X. Re, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[31] Y. Jianchao, Y. Kai, G. Yihong, and H. Thomas, "Linear spatial pyramid matching using sparse coding for image classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," CoRR, vol. abs/1311.2901, 2013.

[33] K K. He, A. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in Proceedings of the (ECCV), 2014.

[34] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in Datails: Delving Deep into Convolutional Nets," 2014

[35] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[36] V. D. Hoang, M. Hale, and K. H. Jo, "Hybrid cascade boosting machine using variant scale blocks based HOG features for pedestrian detection," Neurocomputing vol. 135, pp. 357–366, 2014

[37] J. Yan, X. Zhang, Z. Lei, S.Liao,and S.Z.Li, "Robust multi-resolution pedestrian detection intraffic scenes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[38] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, and D. Ferguson, "Real-time pedestrian detection with deep network cascades," in Proceedings of the (BMVC), 2015.