Methods for Generic Object Detection (December 2018)

JunYu Wang

I. Methods

A. Hog feature

Hog is a classic methods using Histogram of Oriented Gradients, playing the role of a feature descriptor in the computer vision and image processing, for the purpose of objecct detection

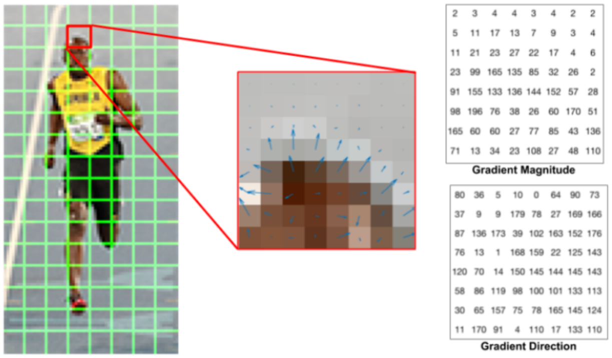
1) Calculate the Gradient Images: At every pixel, the gradient has **a magnitude and a direction**. For color images, the gradients of the three channels are evaluated (as shown in the figure above). The magnitude of gradient at a pixel is the maximum of the magnitude of gradients of the three channels, and the angle is the angle corresponding to the maximum gradient.

Now, we can calculate the magnitude and direction of gradient using the following formula:

$g = \sqrt{g_x^2 + g_y^2}$

$\Theta = arctan\frac{g_y}{g_x}$

After calculation, each pixel has two values. After this, we divided the whole pictures into 8x8 blocks and each block we get two charts. *(Fig.1)*



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

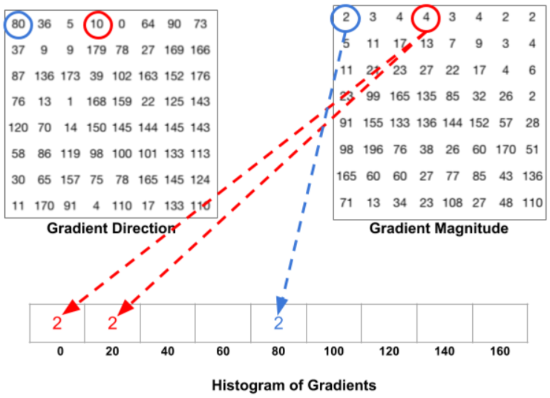
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

Fig. 1. Left : The original image devided into blocks Center : The RGB patch and gradients represented using arrows. Right : The gradients in the same patch represented as numbers

2) create a histogram of gradients in these 8×8 cells: We set the number of bins is 9 corresponding to angles 0, 20, 40 ... 160, and allocate the magnitude of gradient to its own histogram.*(Shown in Fig.2)*

Let's first focus on the pixel encircled in blue. It has an angle (direction) of 80 degrees and magnitude of 2. So it adds 2 to the 5th bin.The gradient at the pixel encircled using red has an angle of 10 degrees and magnitude of 4. Since 10 degrees is half way between 0 and 20, the vote by the pixel splits evenly into the two bins.



80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

0	20	40	60	80	100	120	140	160
2	2			2				

Histogram of Gradients

Fig. 2. This fig shows how to allocate the magnitude via direction

3) 16×16 Block Normalization: Ideally, we want our descriptor to be independent of lighting variations. In other words, we would like to “normalize” the histogram so they are not affected by lighting variations.

So we use a kernel with the size of 16x16 and a stride of 8 to slide the whole image. Suppose there are **7 horizontal and 15 vertical** positions making a total of **7 x 15 = 105** positions. Each 16×16 block is represented by a **36×1 vector**. So when we concatenate them all into one gaint vector we obtain a **36×105 = 3780 dimensional vector**.

4) Use SVM to classifier: Using feature array to train the SVM to judge if their is a item we want!

5) From the perspective of codes: Suppose we are doing Pedestrian Detection

First we get training data, which contains Pedestrian and Non-Pedestrian. We use hog extractor to get the feature from the entire image. At the same time, we get the outcome ”0” or ”1”. After finish training, we get a real image(this image is bigger than the training image) and use kernel to slide over it, the hog feature got from the kernel will be put into SVM to predict.

B. ICF Feature

ICF means **”integral channel features”**, which is a unique word first demonstrated by Piotr Dollar in [5], where the experiments are focusing on **Pedestrian Detection**.

The combination of diverse, informative channels along with the integral image trick for fast feature computation

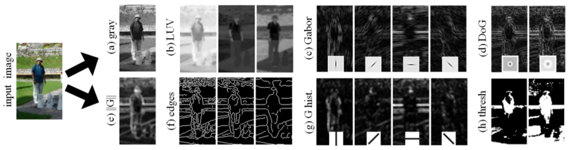


Fig. 3. Multiple registered image channels are computed using various transformations of the input image; next, features such as local sums, histograms, and Haar wavelets are computed efficiently using integral images. Such features, which we refer to as integral channel features, naturally integrate heterogeneous sources of information, have few parameters, and result in fast, accurate detectors.

opened up the door to a much broader class of very effective features

1) Advantages: When designed properly, ICF can outperform other features such as HOG. What's more, it integrates heterogeneous information with few parameters. Besides, it allows more accurate spatial detection and results in fast detectors when coupled with cascade classifiers.

2) Features: The main conribution of ICF is combining different features to boost the efficient of the detection. (see in Fig.3) And for example, the visualization of the most outstading combination(LUV+Grad+Hist) is shown in Fig. 4.

Gray and Color This is the most simple one. Gray vision is panel a. And the color channels can also be used via CIE-LUV in panel b.

Linear Filter Linear filters are a simple and effective method for generating diverse channels. Panel c captures 4 orintation's texture and panel d with Difference of Gaussian (DoG) records the 'texturedness' of the image.

Nonlinear Transformations Gradient magnitude (panel e) captures the unoriented strength while Canny edges (panel f) focuss on edge information. Panel g using gradient Histogram obtains similiar information with panel d. Panel h uses two different thresholds.

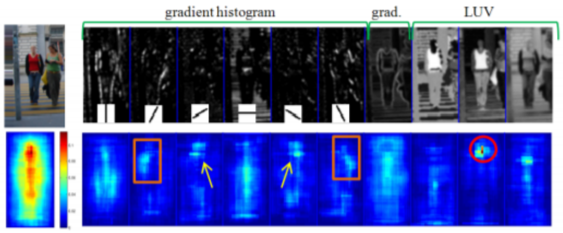


Fig. 4. Top: Example image and computed channels. Bottom: Rough visualization

3) Experiment and Outcome: The details in the experiment is in "dollarBMVC09ChnFtrs.pdf". The outcoms evaluates different combinations of features and parameters(such as pre-smoothing, numbers of classifier) (Fig. 5).

4) Conclusion: The advantages of ICF feature is that multiple registered image channels are computed using linear and non-linear transformations of the input image, and then features such as local sums, histograms and Haar features and their various generalizations are efficiently computed using integral images. And the authors expresses their willing on exploring additional families of channels.

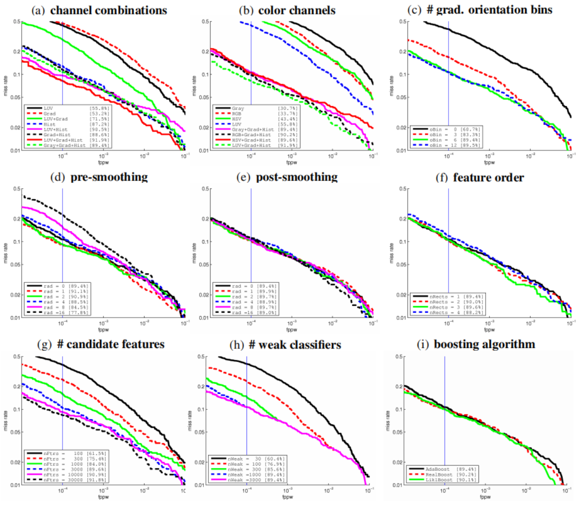


Fig. 5. The outcomes of experiments.

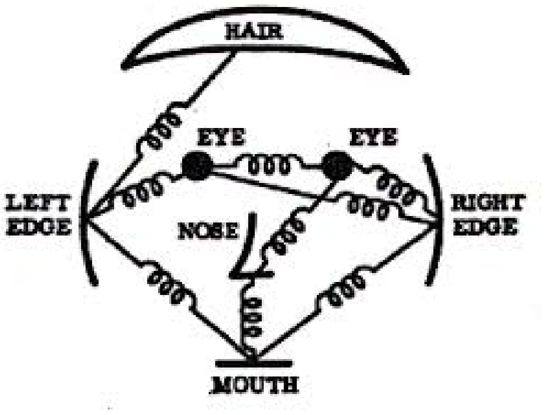


Fig. 6. A face is divided into several parts. The detector not only detect whether this is a face in general but also calculate the relative distances between them to testify its accuracy

C. Deformable Part Model

1) Advantages: Object recognition is hard to sovle because the objects in the image not only changes in illumination and color, but also in the aspect. So Pedro F. Felzenszwalb [2] improve the Dalal-Triggs detector [1] by demonstrating pictorial structures framework [3] combined with latent parameters just like in Fig. 6.

2) Structure: DPM not only detects through the main part but also depends on the other parts. Since that the training data is only labeled with bouding boxes around the objects of interest, adding bounding boxes for the small parts can be inaccurate and time consuming, which can block the effect of training the classifier(SVM).

So here uses latent SVM(LSCM), a formulation of MI-SVM [4], to treat part locations as latent(hidden)variables. This automatic part labeling has been proved to achieve great performance [2].

Responses from the root and part filters are computed a different resolutions in the feature pyramid. The resolution for part filters is twice as much as root filter.(show in Figure 7) The score of a hypothesis is given by the scores of each filter at their respective locations (the data term) minus a deformation cost that depends on the relative position of each part with respect to the root (the spatial prior). The whole calculation process has been shown in Figure 8.

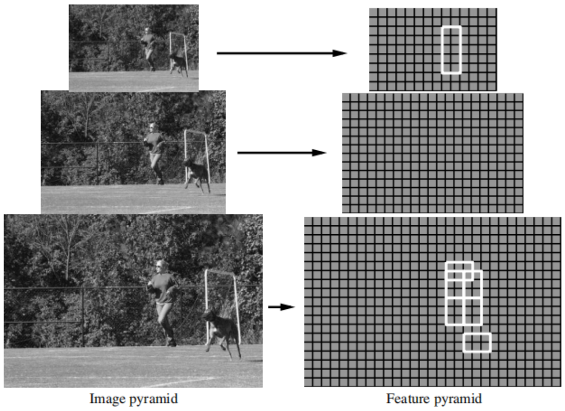


Fig. 7. This shows the feature pyramid

D. Fast to faster rcnn

1) R-CNN: This is the first version of this group of algorithm, which is demonstrated by Ross Girshick et al. in [6]. In this method, they use selective search method to extract about 2000 regions from the image which are called region proposals. The process of calculation is shown in Figure 9. For example, given a region proposal, the algorithm would predict whether there is a person in this area but make sure the face isn't cut in half. So it can also help adjusting the bounding box of the region proposals.

But F-RCNN has some fatal flaws. For example, about 2000 region proposals need to be processed through CNN per image, which takes a large amout of time. And it cannot be used in real time as it takes aroud 47 seconds for each test image.

2) Fast R-CNN: The same author of the [8] combined SPP-net in [7] (Figure 10)with R-CNN to accelerate it's calculation spped via sharing the convolutional layers. As we can see in Figure 11, the whole image is put into a Deep ConvNet. After that, the author use ROI(Region of Interest) [8] projection to find ROI layers followed by a ROI pooling, a variant of SPP-net, which means that different sizes of region proposals can be processed into same length of array to share the fully connected layers.

However, region proposals become bottlenecks in Fast R-CNN algorithm affacting its performance.

3) Faster R-CNN: Both R-CNN and Fast R-CNN apply selective search to find out the region proposals, becoming one of important factors to slow it down. So Shaoqing Ren et al. designed a cleverer method to let the system learn how to find region proposals itself in [9].

In paper, Shaoqing Ren et al. use Rigion Proposal Network(RPN) after the sharing conv layers to find region proposals for Fast R-CNN detector. This architecture is natually implemented $n \times n$ convolutional layers followed by two sibling 1×1 convolutional layers(one for regression and the other for classification respectively).

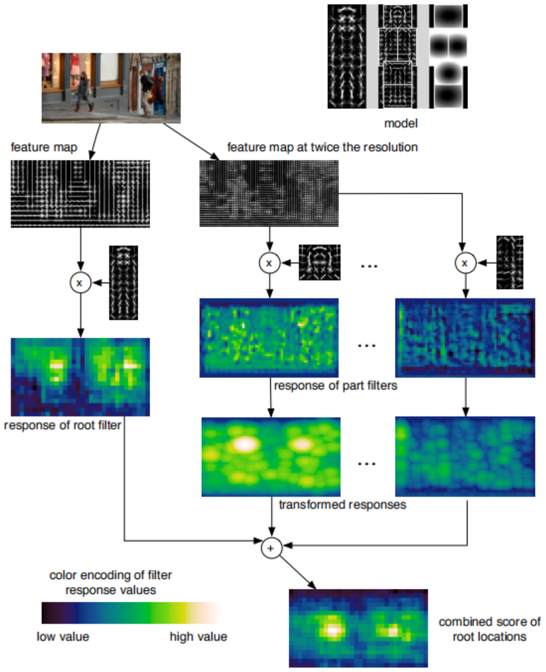


Fig. 8. The transformed responses are combined to yield a final score for each root location. Heads are more obvious in the picture helping users to find there are two people.

Suppose we have k different anchors($m \times n$ m scales and n aspect ratios). So the cls layers get $2k$ outputs while reg layers get $4k$ outputs. The loss function is

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*)+\lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*))$$

where p_i^* and t_i^* means ground truth, i is the index, p is the predicted probability of anchor t is a vector representing 4 parameters. What's more, to improve its efficiency, the system selects the region proposal with high IOU and probability.

Besides, it uses alternative training for several iterations between RPN and Fast R-CNN detector.

The development of the series of R-CNN gradually improves the region proposal quality and the overall detecting accuracy.

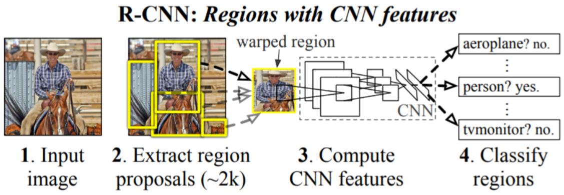


Fig. 9. This system(1) takes an original image, (2) extract region proposals, (3) uses Convolutional Neuron Network(CNN) to compute features, and then (4) classifies the features using linear SVMs

E. YOLO–You Only Look Once

YOLO [10], using a single neuron network with predictions of bounding boxes and class probabilities, is an extemely fast method for object detection.

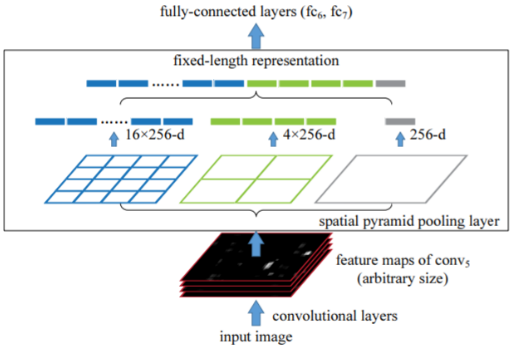

fully-connected layers (fc$_6$, fc$_7$)
fixed-length representation
16×256-d
4×256-d
256-d
spatial pyramid pooling layer
feature maps of conv$_5$
(arbitrary size)
convolutional layers
input image

Fig. 10. A network structure with a spatial pyramid pooling layter

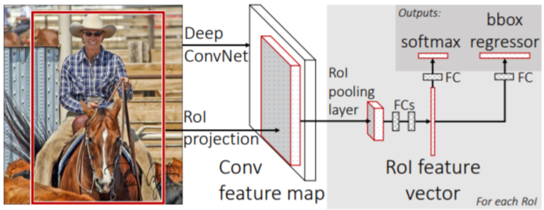


Fig. 11. The structure of Fast R-CNN. An input image and multiple ROI(regions of interest) are put into ConvNet. Each ROI is pooled into a fixed size array for the fully connected layers(FCs). FCs helps us to classify via softmax and adjucting offsets via bbox regression.

1) Unified Detection: YOLO uses deep convolution neuron network to extract features from the whole image to predict. What's more, it predicts all the anchor boxes with different classes simultaneously, which makes it possible for YOLO to achieve end-to-end training and real-time speed.

The system divides the image into an $s \times s$ grids in the end, each grid contains the information of confidence, classes and positions.

2) Architecture: The image is processed with several blocks of convolutional network, followed by the fully connected layer. In [10], Joseph et al. uses PASCAL VOC(20 classes and 2 anchors) to evaluate the algorithm, so that in Figure12, we can see that the size of outcome is $7 \times 7 \times 30(5 \times 2 + 20)$

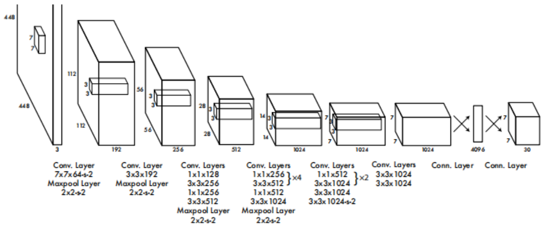


Fig. 12. This detection network has 24 convolutional network and two fully connected layers. The part of CNN is pretrained on ImageNet classification task [15].

3) Training: Authors use sum-squared error as loss function to train the network. To avoid to many grids without object, which push the confidence towards zero, two parameters are added to the loss function, which are $\lambda_{coord} = 5$ and $\lambda_{noobj} = .5$ respectively. The λ makes the grids without object make less contribution to training than before to make the network more stable. Besides, their are other strategies to boost training, such as data augmentation and dropout.

4) Comparison: Compared to DMP [2], YOLO replace deformable part with neuron network, which contains the functions of extract features, predict bounding boxes, etc. When it comes to fast R-CNN, the speed of YOLO is much higher because YOLO avoids making region proposals much less and independently training the two parts of network. Nonetheless, YOLO still inherits many good methods of fast R-CNN, such as adjusting bouding boxes and calculating scores.

F. SSD–Single Shot Detector

In [11], Liu et al. proposed SSD, the single shot detection, which is faster than the previous state-of-art method for detection and more accurate. Compared to YOLO and Faster R-CNN, SSD has deeper convolutional network and applies anchor boxes on several different feature map with different resolution.

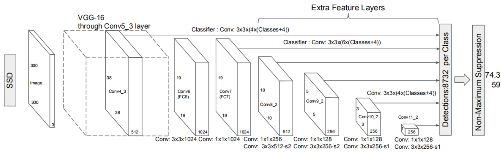


Fig. 13. Every layers can be used to extract features

1) Model Architecture: The SSD detection model is based on a feed forward convolution network with several anchor boxes to predict categories and offset, followed by a non-maximum suppression. Based on [12] and [13], we can see that lower layers in CNN capture more detailed semantic segmentation and adding gloabal context pooled from a feature map has the ability to smooth the segmentation results. Motivated by this, the author of SSD use both the lower and upper feature maps for object detection. (The structure is in Figure 12)

2) Training: The target of training is helping default boxes to find correct classes and positions of each object. The loss function is

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)) \quad (1)$$

$$L_{loc} = \sum_{i \in Pos}^{N} \sum_{m \in cx,cy,w,h} x_{ij}^{k} smooth_{L1}(l_i^m - g_j^m) \quad (2)$$

$$\begin{aligned} L_{conf}(x,c) &= -\sum_{i \in Pos}^{N} x_{ij}^{p} log(\tilde{c}_i^p) - \sum_{i \in Neg} log(\tilde{c}_i^0) \\ where \quad \tilde{c}_i^p &= \frac{exp(c_i^p)}{\sum_p exp(c_i^p)} \end{aligned} \quad (3)$$

where N is the number of mathed default boxes, α is the weight for localization and $x_{ij}^k$is 0 when no object detected

Feature map from different levels have their own receptive field sizes [14]. The scale of the default boxes for each feature map is computed as:

$$s_k = s_min + \frac{s_{max} - s_{min}}{m-1}(k-1), \quad k \in [1, m] \tag{4}$$

References

[1] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[2] Pedro F. Felzenszwalb Ross B. Girshick David McAllester and Deva Ramanan ”Object Detection with Discriminatively Trained Part Based Models,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.

[3] ——, “Pictorial structures for object recognition,” International Journal of Computer Vision, vol. 61, no. 1, 2005.

[4] S. Andrews, I. Tsochantaridis, and T. Hofmann, “Support vector machines for multiple-instance learning,” in Advances in Neural Information Processing Systems, 2003.

[5] Dollár, Piotr and Tu, Zhuowen and Perona, Pietro and Belongie, Serge ”Integral Channel Features,” in Proceedings of the British Machine Vision Conference, 2009

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik ”Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587

[7] Kaiming HeXiangyu ZhangShaoqing RenJian Sun ”Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” in Computer Vision – ECCV 2014 pp 346-361, 2004

[8] Ross Girshick ”Fast R-CNN,” The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448

[9] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, ”Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in NIPS, 2015

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi ”You Only Look Once: Unified, Real-Time Object Detection,” in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg ”SSD: Single Shot MultiBox Detector,” in ECCV 2016: Computer Vision – ECCV 2016 pp 21-37

[12] Long, J., Shelhamer, E., Darrell, T. ”Fully convolutional networks for semantic segmentation,” In: CVPR. (2015)

[13] Liu, W., Rabinovich, A., Berg, A.C. ”ParseNet: Looking wider to see better,” In: ILCR. (2016)

[14] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. Torralba, A. ”Object detectors emerge in deep scene cnns,” in: ICLR. (2015)

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, ”ImageNet Large Scale Visual Recognition Challenge,” in International Journal of Computer Vision (IJCV), 2015. 3