Look and Think Twice: Capturing Top-Down Visual Attention with Feedback Convolutional Neural Networks

Anonymous ICCV submission

Paper ID 1344

Abstract

While feedforward deep convolutional neural networks (CNNs) have been a great success in computer vision, it is important to remember that the human visual context contains generally more feedback connections than forward connections. In this paper, we will briefly introduce the background of feedbacks in the human visual cortex, which motivates us to develop a computational feedback mechanism in the deep neural networks. The proposed networks perform inference from image features in a bottom-up manner as traditional convolutional networks; while during feedback loops it sets up high-level semantic labels as the goal to infer the activation status of hidden layer neurons. The feedback networks help us better visualize and understand on how deep neural networks work as well as capture visual attention on expected objects, even in the images with cluttered background and multiple objects.

1. Introduction

> *"What did you see in this image?"*
> *"Panda, Tiger, Elephant, Lions."*
> *"Have you seen the Gorilla?"*
> *"Oh! I even didn't notice there is a Gorilla !"*

Visual attention typically is dominated by *"goals"* from our mind easily in a top-down manner, especially in the case of object detection. Cognitive science explains this in the "Biased Competition Theory" [1, 5, 6], that human visual cortex is enhanced by top-down stimuli and non-relevant neurons will be suppressed in feedback loops when searching objects. By "looking and thinking twice", both human recognition and detection performances increase significantly especially in images with cluttered background [3]. This leads to the selectivity in neuron activations [16], which reduces the chance of recognition being interfered with either noises or distractive patterns.

Inspired by the above evidences, we present a novel *Feedback Convolutional Neural Network* architecture in this paper. It achieves this selectivity by jointly reasoning the outputs of class nodes and the activations of hidden layer neurons during the feedback loop. As shown in Figure 1, during the feedforward stage, the proposed networks perform inference from image features in a bottom-up manner as traditional Convolutional Networks; while in feedback loops, it sets up high-level semantic labels (*e.g.*, outputs of class nodes) as the "goal" in visual search to infer the activation status of hidden layer neurons. The networks are powerful enough to apply for class model visualization [25, 34] and object localization even in cluttered scenes with multiple objects.

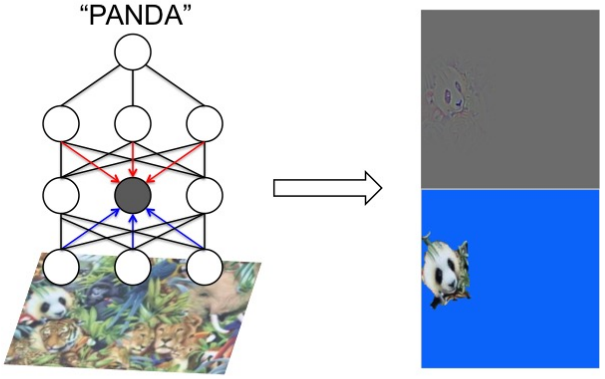


Figure 1. We propose a novel Feedback Convolutional Net model for captuing visual attention by infering the status of hidden neuron activations. The feedback net is designed to utilize both bottom-up image features and top-down semantic labels to infer the hidden neuron activations. The salient area captured by feedback often matches the correponding "target" object, even in the images with cluttered background and multiple objects.

1.1. Optimization in a Feedback Loop

From a machine learning perspective, the proposed feedback networks *add extra flexibility to Convolutional Networks, to help in capturing visual attention and improving feature detection*. Convolutional Neural Networks [18, 15, 26] have achieved great success in both machine learning

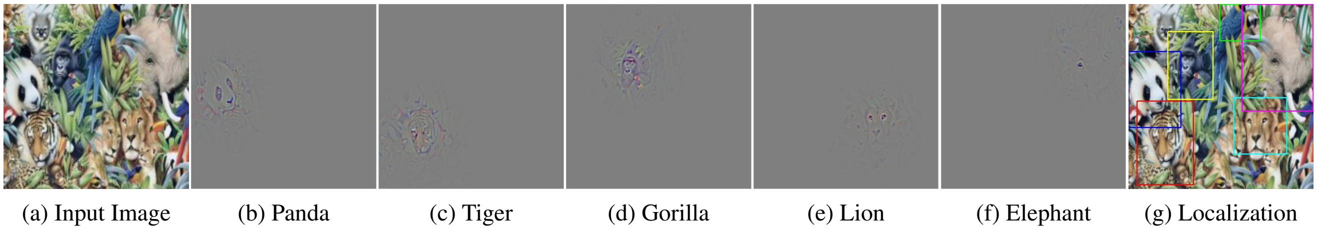


Figure 2. We illustrate the localization power of the feedback net on a multi-object image with cluttered background. (a) shows the original input image which both VggNet and GoogleNet recongize as ”comic book”. (b) - (f) illustrate our feedback modelon understanding the image given different class labels as a prior. We visualize the gradient of each class node with respect to image after the feedback net finish its inference. (g) shows the final localizations for different objects based on the gradients. Better viewed in color.

and computer vision recent years. Benefit from large scale of training data, (*e.g.,* ImageNet [4]), CNNs are capable of learning filters and image compositions at the same time. Various approaches have been adopted to further increase ability of CNN, by either adding regularization in training [11, 13], or going deeper [26, 29]. Inspired by the Deformable Part-Based Models (DPM) [8] that model middle level part locations as latent variables and search for them during object detection, we utilize a simple yet efficient method to optimize image compositions and assign neuron activations given “goals” in visual searching. The algorithm maximizes the posterior response of network given target high-level semantic concepts, in a top-down manner. Compared with traditional bottom-up strategies [11, 13] which aim to regularize the network training, the proposed feedback framework adds flexibilities to the model inference from high-level concepts down to receptive fields.

As the example shown in Figure 1, given a high-level semantic stimulus “PANDA”, only the neurons in hidden layers related with the concept “PANDA” will be activated by iterative optimization in a feedback loop. As a result, only salient regions related with the concept “PANDA” are captured in visualizations. Figure 2 also shows the visualizations of saliencies given different semantic concepts for the same input image. As suggested by those results, the feedback networks achieve certain level of selectivity and provide non-relevant suppression during the top-down inference, allowing the model to focus on the most importatn image regions that improve the class confidence.

1.2. Weakly Supervised Object Localization

Given the gradient visualizations shown in Figure 2, we further develop an algorithm for weakly supervised object localization. Instead of using large amount of supervision (*e.g.*, bounding box positions) in traditional methods such as R-CNN [9] or using regression model [7, 26], we don’t require any localization information in the training stage. In this case, we utilize *a unified network performing both recognition and localization tasks*, to answer questions of “what” and “where” simultaneously, which are the two most important tasks in computer vision. Experimental results suggest that our weakly supervised algorithm using feedback network could achieve similar performance on ImageNet object localization task as GoogLeNet [29] and VGG [26].

The remainder of this paper is organized as follows: Section 2 introduces the related work, while we formulate our algorithm in Section 3. Experiments of visualization and object localization are demonstrated in Section 4. We conclude this work and future directions in Section 5

2. Related Work

2.1. Feedforward and Feedback Mechanism

Deep Neural Network takes a *feedforward-Back Error Propagation* strategy to learn features and classifiers simultaneously, from large scale of training samples [15, 26, 20, 24, 2]. Various approaches have been proposed to further improve the discriminative ability of deep neural network, either by 1) adding regularization to improve the robustness of learnt model and get rid of overfitting, such as Dropout [27], PReLU [11], Batch Normalization [13]; or 2) making the network deeper [29, 26].

Despite great successes achieved by applying Feedforward Networks to image recognition and detection, evidences accumulate from cognitive studies and point to the feedback mechanism that may dominant human perception processes [3, 23, 16, 19]. Recently, tentative efforts have been made to involve feedback strategy into Deep Neural Networks. Deep Boltzmman Machines (DBM) [24] and Deconvolutional Nerual Networks [35] try to formulate the feedback as a reconstruction process within the training stage. Meanwhile, Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM) [12] are utilized to capture the attention drifting in a dynamic environment and learn the feedback mechanisms via reinforcement learning [28, 21]. DRAW from Google DeepMind [10] combine above two into a generative model, to synthesis the image generation process.

As formulated in *Biased Competition Theory* [1, 6], feedback, which passes the high-level semantic information down to the low-level perception, controls the selectivity of neuron activations in an extra loop in addition to the feedforward process. This results in the "Top-Down" attention in human cognition. Hierarchical probabilistic computational models [19] are proposed to characterize feedback stimuli in a top-down manner, which are further incorporated into deep neural networks, for example, modeling feedback loops as latent variables in DBM [31], or using selectivity to resolve fine-grained classification [21], *et al.*. However, due to the computational efficiency and capacity limitations of generative models used in [21, 31], they are hardly used in large scale datasets.

2.2. Visualization, Detection, and Localization

Feedback is always related with visualization of CNN and object localization since both of these aim to project the high-level semantic information back to image representations. To visualize neuron responses and class models, various approaches are proposed either using deconvolution [34] or optimization based on gradients [25, 17]. As demonstrated in [25], visualization of Convolutional Neural Network is showing semantically meaningful salient object regions and helps understand working mechanism of CNNs.

Object detection and localization are more about feedback, by treating detection / localization as a searching process with clear "goals." To localize and detect objects in images, typical approaches use supervised training, which relies on large amount of supervision, *e.g.*, ground-truth bounding boxes, or manually labeled segmentation in training samples [7]. To behave "searching", sliding window is used [7], or instead region proposals detected by image segmentations [30] in R-CNN [9]. However, both of these approaches are computational intensive and naturally bottom-up: selecting candidate regions, performing feedforward classification and making decisions.

Inspired by visualizations of CNNs [34, 25], a more feasible and cognitive manner for detection / localization could be derived by utilizing the saliency maps generated in feedback visualizations. Moreover, an ideal approach should unify the recognition and detection in a single feedforward-feedback network architecture. However, if possible, the challenge lies on how to obtain semantically meaningful salience maps with high quality for each concept. That's the ultimate goal of our work presented in this paper.

3. Model

We first review the current state-of-the-art feedforward Deep Convolutional Neural Networks (CNNs) architecture and then propose our feedback model on top of that.

3.1. Review of Convolutional Neural Networks

The most recent state-of-the-art deep CNNs [26] consist of many stacked feedforward layers, including convolutional, rectified linear units (ReLU) and max-pooling layers. For each layer, the input $\mathbf{x}$ can be an image or the output of a previous layer, consisting of C input channels of width M and height N: $\mathbf{x} \in \mathcal{R}^{M \times N \times C}$. The output $\mathbf{y}$ consists of a set of C' output channels of width M' and height N': $\mathbf{y} \in \mathcal{R}^{M' \times N' \times C'}$.

Convolutional Layer: The convolution layer is used to extract different features of the input. The convolutional layer is parameterized by C' filters with every filter $\mathbf{k} \in \mathcal{R}^{K \times K \times C}$.

$$\mathbf{y}_{c'} = \sum_{c=1}^{C} \mathbf{k}_{c'c} * \mathbf{x}_c, \ \forall c' \tag{1}$$

ReLU Layer: The ReLU layer is used to increase the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convoluional layer.

$$\mathbf{y} = \max(\mathbf{0}, \mathbf{x}) \tag{2}$$

Max-Pooling Layer: The max-pooling layer is used to reduce the dimensionality of the output and variance in deformable objects to ensure that the same result will be obtained even when image features have small translations. The max-pooling operation is applied for every pixel (i, j) around its small neighborhood $\mathcal{N}$.

$$y_{i,j,c} = \max_{u,v \in \mathcal{N}} x_{i+u,j+v,c}, \ \forall i, j, c \tag{3}$$

3.2. Re-interpreting ReLU and Max-Pooling

The ReLU and max-pooling layers can be re-interpreted as components for feature selection in neural nets. During feedforward computation, ReLU and max-pooling layers select those neuron signals that are either *confident enough or locally maximal* to facilitate the invariance [22], spatial assignments [32], and proceed the message passing to higher levels.

We introduce a set of binary activation variables $\mathbf{z} \in \{0, 1\}$ to replace the $\max()$ operations in these layers. For the ReLU layer, $\mathbf{z}$ is the same size as the input $\mathbf{x}$ and Equation 2 can be rewritten as $\mathbf{y} = \mathbf{z} \circ \mathbf{x}$, where $\circ$ is the element wise product (Hadamard product). For the max-pooling layer, $\mathbf{z}$ are similar as convolutional filters except that they are location variant. Equation 3 can be rewritten as $\mathbf{y} = \mathbf{z} * \mathbf{x}$, where $*$ is the convolution operator.

For both layers, the $\max()$ functions are replaced with linear operations between the inputs and binary hidden variables, as binary activation variables. The binary activation variables perform feature selection. However, the values of $\mathbf{z}$ are completely determined by the bottom-up input

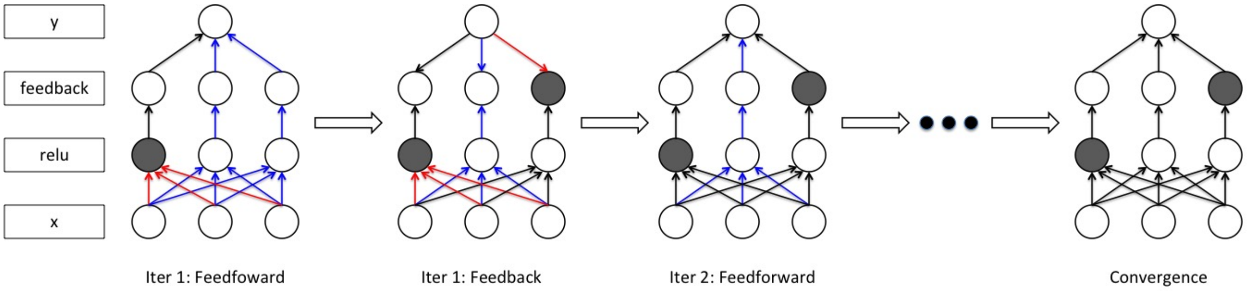


Figure 3. Illustration of our feedback model and its inference process. At the first iteration, the model performs as a feedforward neural net. After then, the neurons in the feedback hidden layers update their activation status to maximize the confidence output of the target top neuron. This process continues until convergence. (We show only one layer here, but the feedback layers can be tacked in the deep ConvNet.) Better viewed in color.

$\mathbf{x}$, meaning that the feature selections are purely based on bottom-up signals and won't be changed with any top-down information.

During the feedforward process, neural-net features need to keep an overall description of the image content and make image representation as general as possible, due to the lack of top-down semantic information. To achieve this target, middle-level neurons try to turn enough ReLUs on to avoid information loss; while the high-level fully connected layers are responsible for providing discriminability and descriptive ability. This works well when there is only one salient object in the image and no prior information is given. However, when the image contains multiple objects and complex scenes, the same feature may be too general to work equally well for all objects. So it will fail in the detection task.

3.3. Introducing the Feedback Layer

Since the model opens all gates and allow maximal information getting through to ensure the generalization, to increase the discriminability within feature level, it is feasible to turn off those gates that provide irrelevant information when targeting at particular semantic labels. This strategy is explained as selectivity in biased competition theory [6] and is critical to realize the top-down attention.

More technically, to increase the model flexibility to images and prior knowledges, we introduce an extra layer to the existing convolutional neural networks. We call it the feedback layer. The feedback layer contains another set of binary activation variables $\mathbf{z} \in \{0, 1\}$, similar to ReLU. However, the binary variable are activated according to the top-down message from outputs, instead of bottom inputs. The feedback layer is stacked upon each ReLU layer, and they compose a hybrid control unit to active neuron response in both bottom-up and top-down manners:

Bottom-Up Inherent the selectivity from *ReLU layers*, and the dominant features will be passed to upper layers;

Top-Down Controlled by *Feedback Layers*, which propagate the high-level semantics and global information back to image representation. This is achieved by activating only those gates related with target neurons.

Figure. 3 illustrates a simple architecture of our feedback model with only one ReLU layer and one feedback layer.

3.4. Updating Hidden Neurons in Feedback Loops

Given an image I and a neural network with learned parameters w, we optimize the target neuron output by jointly inference on binary neuron activations $\mathbf{z}$ over all the hidden feedback layers. In particular, if the target neuron is a class node in the top layer, we optimize the class score s_k by reajusting the feedback neurons at every layer l, channel c and pixel (i, j).

$$\begin{aligned} \max_{z} \quad & s_k(I, z) - \lambda ||z|| \\ s.t. \quad & z^l_{i,j,c} \in \{0, 1\}, \; \forall \, l, i, j, c \end{aligned} \tag{4}$$

This leads to an integer programming problem, which is NP-hard given the current deep net architecture. An approximated solution could be derived by applying a linear relaxation:

$$\begin{aligned} \max_{z} \quad & s_c(I, z) - \lambda ||z|| \\ s.t. \quad & 0 \leq z^l_{i,j,c} \leq 1, \; \forall \, l, i, j, c \end{aligned} \tag{5}$$

We use the gradient ascent algorithm to update the hidden variables through all layers simultaneously.

$$z_{t+1} = z_t + \alpha \cdot (\frac{\partial s_c}{\partial z}|_{z_t} - \lambda) \tag{6}$$

The initialization of feedback layer status z is set to be the corresponding ReLU activation after the first feedforward pass and truncate z when the updated values are either larger than 1 or smaller than 0 during inference.

3.5. Implementation Details

As for implementation details, we set the feedback layer on top of every ReLU layer except those taking the fully connected layers as inputs. It is suspected that the fully connected layers learn more embedding spaces rather than particular parts compared to convolutional layers. We set learning rate of hidden activations to 0.1 and update the neurons of all the feedback layers simultaneously. Each iteration performs a feedforward step of the neural net and a backpropagation step to send back gradients. This process usually converges in 10 to 50 iterations. The final hidden neuron activations are binarized by threshold 0.5.

4. Experimental Results

To verify the feedback model, we conduct qualitative experiments on class neuron visualizations and quantitative evaluations on the weakly supervised object localization task. We use the three most popular pre-trained ConvNet models, AlexNet [15], VggNet [25] and GoogleNet [29] for experiments. All three models are pre-trained with ImageNet 2012 classification training dataset [4], obtained from Caffe [14] model zoo. AlexNet achieves $\sim$15% top 5 classification error on ImageNet 2012 testing dataset, while VggNet and GoogleNet obtains $\sim$7.5%. GoogleNet slightly outperforms VggNet, but the gap is small and can be ignored.

4.1. Image Specific Class Model Visualization

Given an image I, a class label k and the hidden neuron activation states $\mathbf{z}$, we approximate the neural net class score s_k with the first-order taylor expansion in the neighborhood of I:

$$s_k(I,\mathbf{z}) \approx \mathbf{T}_k(\mathbf{z})^T I + b \tag{7}$$

where $\mathbf{T}_k(\mathbf{z})$ is the derivative of s_k with respect to the image at the point of I and $\mathbf{z}$. $\mathbf{T}_k(\mathbf{z})$ can be viewed as the linear template applied on image I for measuring how likely the image belongs to class k. We can visualize $\mathbf{T}$ since it's the same size as the image I. We use this technique to visualize the our feedback model throughout the paper.

Specifically, for a VggNet which uses a stack of piecewise linear layers (*i.e.* Conv, ReLU and max-pooling) to compute the class scores, once the hidden states $\mathbf{z}$ are determined, the final score is a linear function on the image, equivalent to the inner product between the template and the image.

Comparison of Visualization Methods: We compare the image gradient (template $\mathbf{T}$) after the feedback process against original gradient and Deconvolutional Neural Net (Deconv) [34] on a set of complex images containing multiple objects from different classes. We show the qualitative results in Figure 4. All techniques use the same pre-trained GoogleNet and ground truth class labels are given as a prior. The visualization results before feedback is the same as original image gradients, where all the hidden neurons states are determined by the bottom-up computations, while after feedback the visualizations are similar to Deconv. However, our feedback model captures more salient regions for the specific class while surpress irrelevant object area much better than Deconv.

Comparison of ConvNet Models: We also qualitatively compare AlexNet, VggNet and Googlenet by visualizing their feedback templates in Figure 5. All models are given the ground truth class labels as a prior. From the visualization results, we find that VggNet and GoogleNet produce more accurate visual attention than AlexNet, suggesting that using smaller convolution filters and deeper architectures could further distinguish similar and nearby objects. We also observe that, although both VggNet and GoogleNet produce very smilar image classification accuracies, GoogleNet better captures the salient object areas than VggNet. We hypothesize that the two 4096 dimensional fully connected layers (*i.e.* fc6, fc7) in VggNet (which GoogleNet does not contain) could harm the spatial distinctiveness of the final image features,

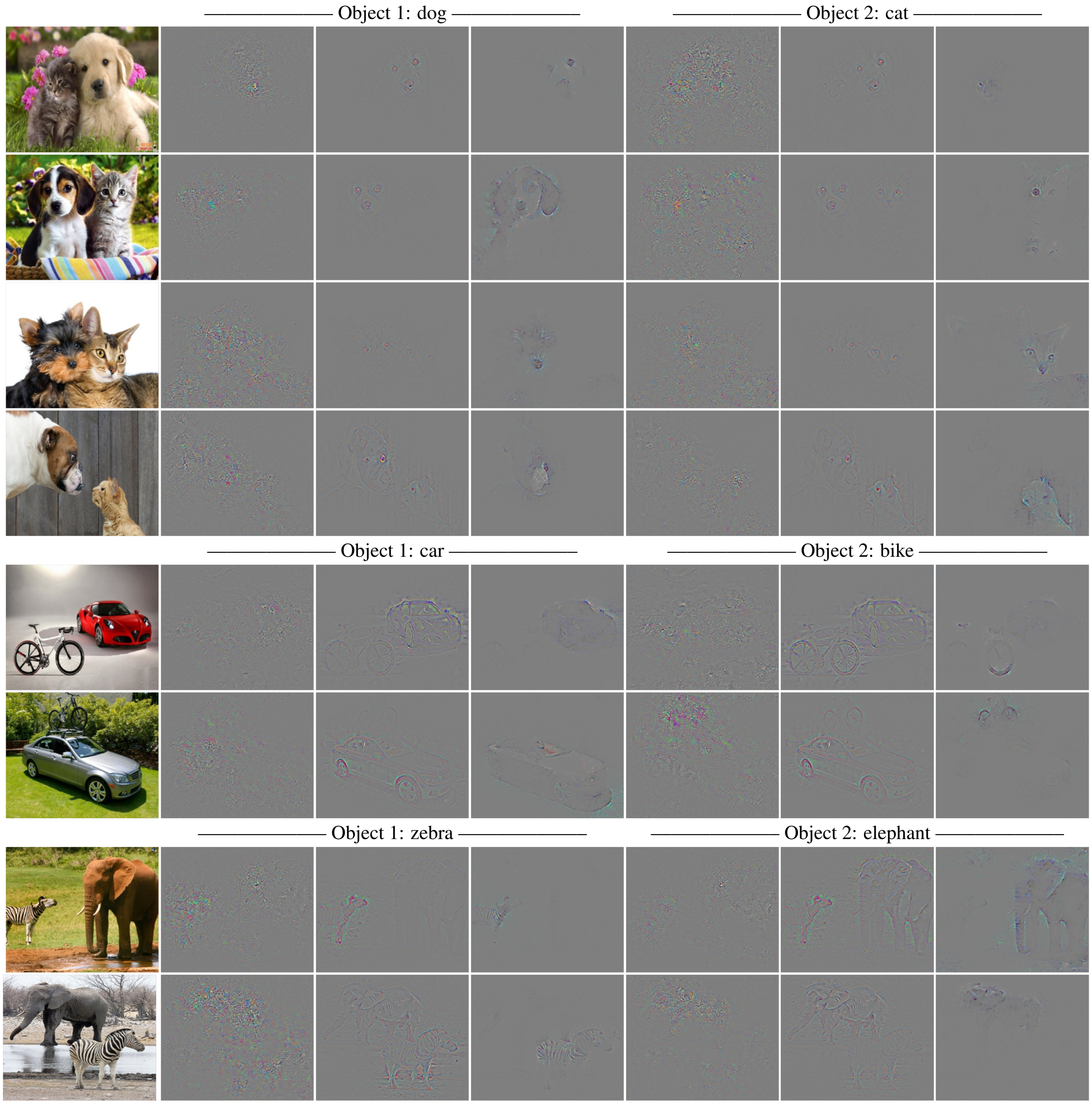
Feedback Visualization of Similar Classes: We also show a few intersting feedback visualizations for understanding the fine-grained classifcation models of GoogleNet. The GoogleNet is trained on ImageNet dataset with 1000 classes, among which there are $\sim$100 dog categories and $\sim$500 animal categories. We visualize the feedback templates of dog-cat images given paticular dog classes and animal classes in Figure 6. We observe that each class has its own special salient image features for distinction, for example, some classes (*i.e.* beagle, kit fox) look for local part features such as nose and ears, while others (*i.e.* terrier, tiger) focus on global attributes such as furry and tabby.

4.2. Weakly Supervised Object Localization

To quantitatively demonstrate the effectiveness of the feedback model. we experiment on the ImageNet 2014 localization task.

As pointed in [25], the magnitude of the elements in the model template $\mathbf{T}_k$ defines the class specific salience map on image I. Pixels with larger magnitudes indicate that they are more important to the class. We adopt the same saliency extraction strategy as [25] that a single class saliency value M_k for class k at pixel (i,j) is computed across all color channels: $M_k(i,j) = \max_{c\in rgb} |T_k(i,j,c)|$.

Although the three ConvNets are pre-trained for image classification, we could use the feedbacked salience map for weakly supervised object localization. Following [25], given an image and the corresponding class salience map, we compute the object segmentation mask using the Graph-



(a) Image (b) Gradient (c) Deconv (d) Feedback (e) Gradient (f) Deconv (g) Feedback

Figure 4. We demonstrate the effectiveness of feedback neural networks for class-specific feature extraction, by comparing the class model visualization results against original gradient [25] and Deconv [34] on selected images with multiple objects. All methods compute visualizations using a pre-trained GoogleNet trained on ImageNet 2012 classification dataset. Column (a) shows the input images (*i.e.* dog v.s. cat, car v.s. bike, and zebra v.s. elephant). Column (b) and (e) show the original image gradients given the provided class labels. Column (c) and (f) show the Deconv results. Column (d) and (g) show the image gradients after feedback. Comparing against original gradient and Deconv, the feedback visualization captures more accurate salient area of the target object. For example, in the 4th row, both original template and Deconv see the dog and cat, even provided with the target label. In the last row, when zebra is specified, Deconv finds it hard to supress the elephant area. Our feedback method supress the irrelevant object much better. Better viewed in color and zoom in.

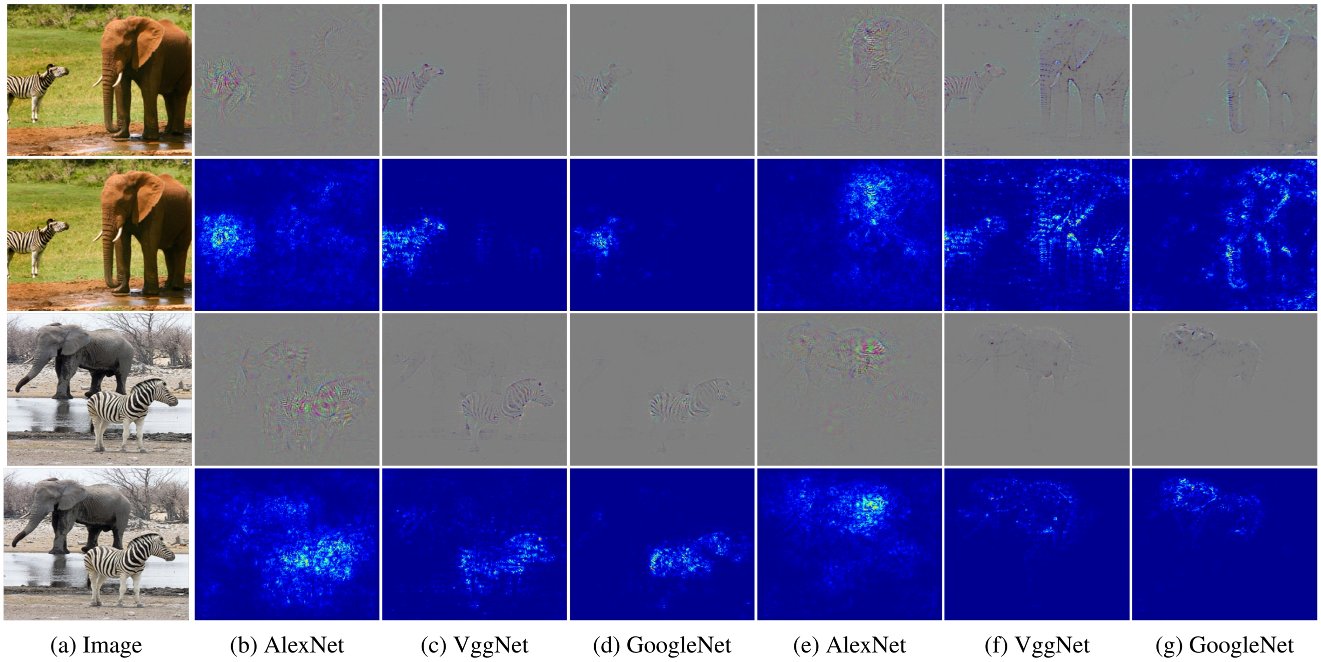


Figure 5. We visualize the feedback ability of three most popular pre-trained ConvNets: AlexNet, VggNet and GoogleNet, by visualizing the final image gradients and salience maps after feedback. We show the input images in column (a). We show the results of the three models feedbacked by ”zebra” in column (b), (c), (d) and by ”elephant” in column (e), (f), (g) repsectively. We find that VggNet performs quite better than AlexNet, especially in capturing salient object details, suggesting the benefit of usage of small convolutional filters and deeper architecture. Although both VggNet and GogoleNet produce similar classification accruacy, we find GoogleNet provides the better class specific feature separations. We suspect the two 4096 fully connected layers in VggNet (which GoogleNetdoes not have) could harm the spatial distinctiveness of image features.

Cut color segmentation [33]. During the inialization of graph cut, We set the pixels with the saliency higher than 95% quantile of the saliency distribution in the image as foreground and those with the saliency lower than 30% quantile as background. Once foreground and background segmentations are computed, the object segmentation mask is set to the largest connected component of the foreground pixels and the tighest bounding box is extracted as the localization result.

We test our localization results on ImageNet 2014 localization validation set, which contains ∼20000 images with each image associated with a label and a bounding box. The predicted bounding box is considered as correct if its overlap with the ground truth is over 50%. We resize every image to 224x224 as the models' required resolution and provide the ground-truth class labels for the localization prediction. No further preprocessing or multi-scale strategy is involved.

Comparison of Localization Methods: Table 1 shows the comparsion of our weakly supervised localization accuracy against the baseline method [25]. We use the same Googlenet and apply the same graph cut strategy. Our method obtains 38% localization error, significantly outperforming Oxford 44.6%, suggesting that in terms of capturing attention and localizing salient objects, our feedback net is better. Note that our weakly supervised localization error is even closer to a carefully trained supervised localization model (34.3%).

Comparsion of ConvNet Models: We also analyze the weakly supervised localization accuracy of the three ConvNets in Table 2, provided with the same testing protocol. Even provided with ground truth class, VggNet and GoogleNet significantly outperforms AlexNet suggesting that better feature representations are sharable between the two highly correlated visual tasks: recognition and localization. GoogleNet outperforms VggNet even further, which matches the observations in Figure 5.

5. Conclusion & Discussion

We propose a Feedback Convolutional Neural Network architecture in this paper, which achieves the selectivity of neuron activations by jointly reasoning outputs of class nodes and activations of hidden layer neurons during the feedback loop. The proposed Feedback CNN is capable of capturing high level semantic concepts and project down to image representation as salience maps. Benefit from the feedback mechanism implemented in our model, we uti-

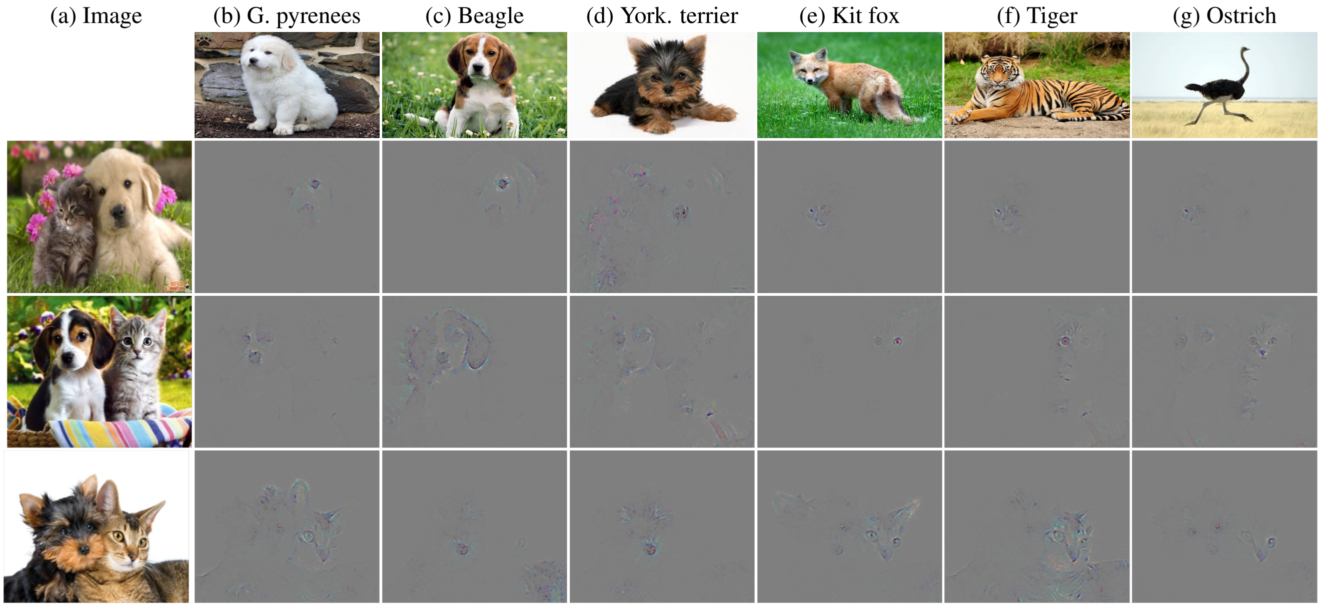


Figure 6. We show some interesting visualizations for the understanding of fine-grained classification by comparsing against the feedback gradients of ground truth labels and other classes. The top row shows the class labels and a representative image for each class for the ease of understanding. Column (a) shows the three examplar input images, their ground truth labels are great pyrenees, beagle and yorkshire terrier respectively. We can see that although (b), (c) and (d) are all dogs, their salient area for distinction are quite different. Noses are one of the most important feature for classifying dogs, but ears are specific feature for beagles, while fluffy is more importatnt to yorkshire terrier. When the top down is from (e) kit fox, features on the cat in the last row is more fox-specific: nose and ears. When top down is from (f) tiger, features on the same at is more tiger-specific: textures. And when it's (g) ostrich, nothing special come out.

Localization Errors Given Ground Truth Labels

Method	Localization Error (%)
Oxford [25]	44.6
Feedback	**38.8**
Oxford-Supervised [26]	34.3

Table 1. We compare our weakly supervised localization results on ImageNet 2014 validation set with the simplified testing protocol: the bounding box is predicted from a single central crop of images and the ground truth labels are provided. We show that our feedback method siginificantly outperforms the baseline method (44.6%) that uses the original image gradient to localize, and works even closer to a carefully trained supervised localization model (34.3%).

Localization Errors of Different Feedback ConvNets

Model	Localization Error (%)
AlexNet [15]	49.6
VggNet [26]	40.2
GooglNet [29]	**38.8**

Table 2. We analyze the attention ability by running our feedback mechanism on the three popular ConvNets. All models use the same testing protocol as Table 1. Even provided with ground truth class, VggNet and GoogleNet significant outperforms AlexNet suggesting they are learning better features. GoogleNet outperforms VggNet even further, which matches the observations in Figure 5.

lize the salience map to build a unified deep neural network for both recognition and object detection tasks, to answer questions of both "What" and "Where" simultaneously. Experimental results on ImageNet 2014 object localization Challenge show that our model could achieve competitive performance compared with state-of-the-arts, using only weakly supervised information.

The Feedback CNN has the potential to improve various computer vision and machine learning tasks, such as fine-grained recognition, multi-task learning, and object detection. Moreover, we are seeing the possibilities to implement semi-supervised CNN using the proposed feedback architecture as future work.

References

[1] D. M. Beck and S. Kastner. Top-down and bottom-up mechanisms in biasing competition in the human brain. *Vision research*, 49(10):1154–1165, 2009. 1, 2

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013. 2

[3] R. M. Cichy, D. Pantazis, and A. Oliva. Resolving human object recognition in space and time. *Nature Publishing Group*, 17(3):455–462, Jan. 2014. 1, 2

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 2, 5

[5] R. Desimone. Visual attention mediated by biased competition in extrastriate visual cortex. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 353(1373):1245–1255, 1998. 1

[6] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995. 1, 2, 4

[7] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2155–2162. IEEE, 2014. 2, 3

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. 2

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014. 2, 3

[10] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015. 2

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks, 2012. 1, 2, 5, 8

[16] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott. Deep Hierarchies in the Primate Visual Cortex: What Can We Learn for Computer Vision? 35(8):1847–1871. 1, 2

[17] Q. V. Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE, 2013. 3

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[19] T. S. Lee and D. Mumford. Hierarchical bayesian inference in the visual cortex. *JOSA A*, 20(7):1434–1448, 2003. 2, 3

[20] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 2

[21] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent Models of Visual Attention. *NIPS*, June 2014. 2, 3

[22] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999. 3

[23] N. C. Rust and J. J. DiCarlo. Selectivity and Tolerance (”Invariance”) Both Increase as Visual Information Propagates from Cortical Area V4 to IT. *Journal of Neuroscience*, 30(39):12978–12995, Sept. 2010. 2

[24] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455, 2009. 2

[25] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1, 3, 5, 6, 7, 8

[26] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv.org*, Sept. 2014. 1, 2, 3, 8

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 2

[28] M. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber. Deep Networks with Internal Selective Attention through Feedback Connections. *arXiv.org*, July 2014. 2

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. *arXiv.org*, Sept. 2014. 2, 5, 8

[30] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 3

[31] Q. Wang, J. Zhang, S. Song, and Z. Zhang. Attentional neural network: Feature selection using cognitive feedback. In *Advances in Neural Information Processing Systems*, pages 2033–2041, 2014. 3

[32] J. Weng, N. Ahuja, and T. S. Huang. Cresceptron: a self-organizing neural network which grows adaptively. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 1, pages 576–581. IEEE, 1992. 3

[33] B. Yuri and J. Marie-Pierre. Interactive graph cuts for optimal boundaryand region segmentation of objects in nd images. In *IEEEInternational Conference on Computer Vision. USA: IEEE*, volume 112, 2001. 7

[34] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014. 1, 3, 5, 6

[35] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *Computer Vision (ICCV . . .*, pages 2018–2025, 2011. 2