A CLOSER LOOK: SMALL OBJECT DETECTION IN FASTER R-CNN

Christian Eggert, Stephan Brehm, Anton Winschel, Dan Zecha, Rainer Lienhart

Multimedia Computing and Computer Vision Lab
University of Augsburg

ABSTRACT

Faster R-CNN is a well-known approach for object detection which combines the generation of region proposals and their classification into a single pipeline. In this paper we apply Faster R-CNN to the task of company logo detection. Motivated by the weak performance of Faster R-CNN on small object instances, we perform a detailed examination of both the proposal and the classification stage, examining their behavior for a wide range of object sizes. Additionally, we look at the influence of feature map resolution on the performance of those stages. We introduce an improved scheme for generating anchor proposals and propose a modification to Faster R-CNN which leverages higher-resolution feature maps for small objects. We evaluate our approach on the Flicker data set improving the detection performance on small object instances.

Index Terms— Small objects, Faster R-CNN, RPM, Feature map resolution, Company logos

1. INTRODUCTION

Current object detection pipelines like Fast(er) R-CNN [1] [2] are built on deep neural networks whose convolutional layers extract increasingly abstract feature representations by applying previously learned convolutions followed by a non-linear activation function to the image. During this process, the intermediate feature maps are usually downsampled multiple times using max-pooling.

This downsampling has multiple advantages: (a) It reduces the computational complexity of applying the model, (b) helps to achieve a certain degree of translational invariance of the feature representation and (c) also increases the receptive field of neurons in the deeper layers. The flipside of these advantages is a feature map which has a significantly lower resolution than the original image. As a result of this reduced resolution it is difficult to associate features with a precise location in the original image.

Despite this potential drawback, this approach has been extremely successful in the areas of image classification and object detection. For most applications, pixel-accurate localization is not important.

In this paper we examine the suitability of feature representations from different levels of the feature hierarchy for the problem of company logo detection. Company logo detection is an application of object detection which attracts lots of commercial interest. On a superficial level, company logo detection is nothing but a special case of general object detection. However, company logos are rarely the objects which were intended to be captured when the picture was taken. Instead, they usually happen to get caught in the picture by accident. As a result, company logos tend to occupy a rather small image area.

Intersection over union (IoU) is the usual criterion by which the quality of the localization is assessed. By this measure, a detection which is off by a given amount of pixels has a greater influence on small object instances than large ones. Therefore, small object instances require a more precise localization than large instances in order to be classified as correct detections.

A simple way to resolve this problem would be to upsample the image and to repeat the detection but this simple approach is not very appealing since the effort for applying the convolutions grows quadratically with the side length of the image. This is especially true for company logo detection in which the object is typically small compared to the image, resulting in much unnecessary computation.

Our contributions are as follows:

1. We theoretically examine the problem of small objects at the proposal stage. We derive a relationship which describes the minimum object size which can reasonably be proposed and provide a heuristic for choosing appropriate anchor scales.

2. We perform detailed experiments which capture the behavior of both the proposal and the classification stage as a function of object size using features from different feature maps. Deeper layers are potentially able to deliver features of higher quality which means that individual activations are more specific to input stimuli than earlier layers. We show that in the case of small objects, features from earlier layers are able to deliver a performance which is on par with – and can even exceed – the performance of features from deeper layers.

3. We evaluate our observations on the well-known FlickrLogos dataset [3] in the form of an extension to the Faster R-CNN pipeline

Since FlickrLogos has been originally been conceived as a benchmark for image retrieval we have re-annotated the dataset for the task of object detection[1].

2. RELATED WORK

Low-resolution data has been previously studied by Wang et al. [4] in the context of image classification. They conclude that low-resolution classification problems do not benefit from deeper network architectures, more filters or larger filter sizes and also note substantial differences between the feature representation of large and small objects. However, [4] does not discuss its impact on object detection.

Bell et. al. [5] and [6] do consider object detection of small objects in the context of Fast-RCNN [1]. [6] explicitly consider the problem of company logo detection and notice a relationship between receptive field, object size and detection performance. [5] apply techniques like skip-pooling to create multi-scale feature representations. They also consider context features obtained by an recurrent network. However, both [5] and [6] only consider the classification stage of the pipeline. Also, they do not explicitly analyze the behavior of Fast R-CNN across multiple feature maps and scales.

3. SMALL OBJECTS IN FASTER R-CNN

Current object detection pipelines usually consist of two stages: The first step of current detection pipelines is usually to identify regions of interest (ROIs) from images. These ROIs serve as an attention model and propose potential object locations which are more closely examined in a second stage.

For our experiments we use a re-implementation of the Faster R-CNN [2] approach. Faster R-CNN extracts a feature representation of the image through a series of learned convolutions. This feature map forms the basis of both the object proposal stage and the classification stage. The first step is accomplished by a Region Proposal Network (RPN) which starts by generating a dense grid of anchor regions with specified size and aspect ratio over the input image.

For each anchor, the RPN – which is a fully convolutional network – predicts a score which is a measure of the probability of this anchor containing an object of interest. Furthermore, the RPN predicts two offsets and scale factors for each anchor which are part of a bounding box regression mechanism which refines the object's location. The refined anchors are sorted by score, subjected to a non-maximum suppression and the best scoring anchors are kept as object proposals which are fed into the second stage of the network.

[1]The updated annotations and evaluation script are made available here: `http://www.multimedia-computing.de/flickrlogos`

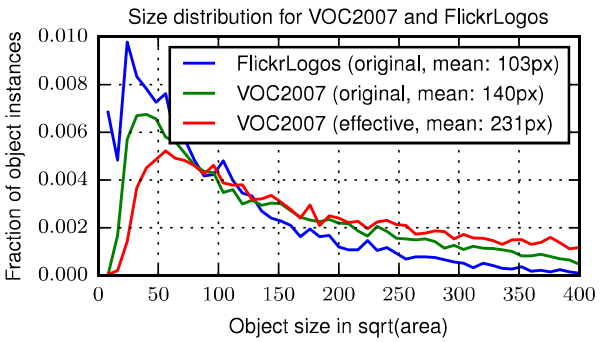

Fig. 1. Distribution of object instance sizes in VOC2007 and FlickrLogos. Because of dynamic rescaling of images in Faster R-CNN, the effective size distribution is shifted towards larger object instances.

At training time, anchors are divided into positive and negative examples, depending on the overlap they have with a groundtruth instance. Typically, an anchor is considered to be a positive example if it has an IoU greater than 0.5 with a groundtruth object.

Ren et. al [2] use anchors whose side length are powers of two, starting with 128 pixels. This choice of anchors delivers good results on datasets such as VOC2007 [8] where the objects are typically relatively large and fill a sizeable proportion of the total image area. Furthermore, [2] also dynamically rescale input images to enlarge the objects.

Upscaling of input images is typically not feasible for company logo detection. Figure 1 shows the size distribution of the FlickrLogos [3] dataset. The average object size is quite small compared with the average side length of the images (which is typically around 1000 pixels).

Figure 1 also makes it clear, that an anchor of side length of 128 is inadequate to cover the range of object sizes. In order to counter this problem one could simply add additional anchors using the same powers-of-two scheme used by [2]. However, we show that this scheme leads to difficulties – particularly for small objects – as it might fail to generate an anchor box with sufficient overlap.

To illustrate the problem we consider the situation in Figure 2a: We assume a quadratic groundtruth bounding box B_g of side length s_g and a quadratic anchor box B_a of side length s_a. Furthermore we will assume w.l.o.g. that $s_g \leq s_a$ and that both side lengths are related through a scaling factor $\alpha \geq 1$ by $s_a \geq \alpha s_g$. Under these conditions we can move B_g anywhere inside of B_a without changing the IoU.

In this case we can express the IoU as the ratio between the areas enclosed by these boxes:

$$t \leq IoU(B_g, B_a) = \frac{|B_g \cap B_a|}{|B_g \cup B_a|} = \frac{s_g^2}{s_a^2} = \frac{1}{\alpha^2} \qquad (1)$$

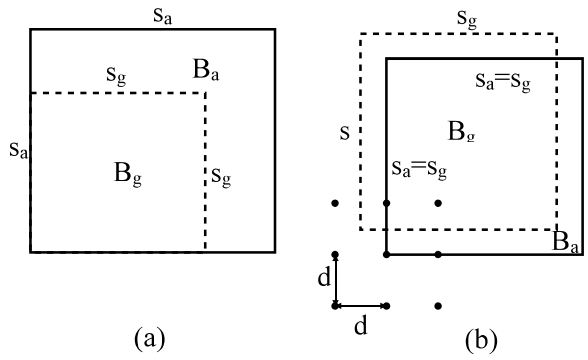

Fig. 2. (a) IoU can be expressed as the ratio of bounding box areas in the case of aligned bounding boxes of equal aspect ratio. (b) Worst case displacement of two bounding boxes of equal size when anchors are sampled with stride d

In order for an anchor box to be classified as positive example we require the IoU to exceed a certain threshold t. It follows that for $\alpha > \sqrt{t}^{-1}$ an anchor is unable to cover a groundtruth box with sufficient overlap to be classified as a positive example. The same relationship holds for non-quadratic anchors – provided the aspect ratio of groundtruth boxes and anchor boxes match.

Therefore, the side length of anchor boxes of neighboring scales s_{a_1} and s_{a_2} should be related by $s_{a_2} = \sqrt{t}^{-1} s_{a_1}$.

For the previous considerations we assume that there exists an anchor position at which the corner of an anchor is completely aligned with the groundtruth instance. In practice this is not true since the feature map of the network upon which the RPN is based usually has a much smaller resolution than the original image. A downsampling factor d^{-1} between the original image and the feature map effectively results in a grid of anchors with stride d.

To examine the influence of feature map resolution on the RPNs potential to identify small object instances we consider the situation in Figure 2b. We assume a quadratic groundtruth instance B_g and the existence of an anchor box B_a of identical scale and aspect ratio. In the worst case, both boxes are displaced against each other by a distance of $\frac{d}{2}$. The IoU between these boxes can be expressed by:

$$IoU(B_g, B_a) = \frac{(s_g - \frac{d}{2})^2}{(s_g - \frac{d}{2})^2 + 2(2\frac{d}{2}(s_g - \frac{d}{2}) + \frac{d^2}{4})} \quad (2)$$

Solving $t \leq IoU(B_g, B_a)$ for s_g while assuming $d > 0$ and $0 < t < 1$ and ignoring negative solutions for this quadratic expression, we obtain the following relationship for the minimum detectable object size:

$$\frac{d(t+1) + d\sqrt{2t(t+1)}}{2 - 2t} \leq s_g \quad (3)$$

For the VGG16 [9] architecture which is used as basis for Faster R-CNN $d = 16$. Assuming $t = 0.5$, this translates into a minimum detectable object size of $s_g \approx 44px$. This suggests that for the small end of our size distribution a feature map of higher resolution is needed. For the *conv4* feature map ($d = 8$) the minimum detectable object size is given by $s_g \approx 22px$. Since we do not expect to reliably classify objects smaller than 30px we use the next power of two as smallest anchor size.

Making use of our previous result we choose as our anchor set $\mathcal{A} = 32, 45, 64, 90, 128, 181, 256$ since we follow the recommendation of [2] and set $t = 0.5$.

3.1. Region Proposals of small objects

We want to evaluate the effectiveness of RPNs for different object sizes. The primary measure of an RPN's quality is the mean average best overlap (MABO). It measures the RPN's ability to generate at least one proposal region for each object with high overlap. If $\mathcal{C}$ represents the set of object classes, G_c the set of groundtruth objects of a particular class $c \in \mathcal{C}$ and $\mathcal{L}$ the set of object proposals, we can evaluate the performance of the RPN for a particular class c via its average best overlap $ABO(c)$ given by:

$$ABO(c) = \frac{1}{|G_c|} \sum_{g \in G_c} \max_{l \in L} IoU(g, l) \quad (4)$$

where $IoU(g, l)$ is the intersection over union between the groundtruth item g and the proposal l. The MABO is the mean over all ABO values for each object class.

In order to examine the influence of object size on the performance of the RPN, we create differently scaled synthetic variants of the FlickrLogos [3] dataset by applying the following algorithm to each image:

We start by selecting the point which has the maximum distance between two non-overlapping groundtruth bounding boxes. This point defines two axes along which the image is to be partitioned into four parts. We ensure that the axes of the split do not intersect with any other groundtruth items. If no such split can be found, the image is discarded. For each of the resulting partitions which contain more than one groundtruth item, the process is applied recursively. After applying this algorithm, each image contains only a single object instance which is then rescaled to match the desired target size.

Using this algorithm we create 11 differently scaled versions of the test set which we call $F_{test,x}$ where $x \in \{10 * i + 20 | i = 0 \dots 10\}$ represents the target object size, measured as square root of the object area. Additionally, we create a single training dataset F_{train} in which the objects are scaled in such a way that the square root of the object area is distributed evenly in the interval $[20\text{px}, 120\text{px}]$.

In order to observe the performance of the RPN for different layers we create three RPNs RPN_{conv3}, RPN_{conv4}

and RPN_{conv5} based on the VGG16 [9] architecture used by [2]. These networks use features from the *conv3*[2], *conv4* and *conv5* layer, respectively to predict object proposals. The features are passed through a normalization layer which normalizes the activations to have zero-mean and unit-variance. This is similar to batch normalization [10]. However we normalize the activations with respect to the training set and not with respect to the current batch as in [10]. We do this so that we can easily use an off-the-shelf Imagenet [11] pre-trained VGG16 network. Those pre-trained models usually have the property that the variance of activations decreases from layer to layer as the data progresses through the network. This property makes it hard to make certain changes to the network architecture. For example, adding additional branches of different depths will result in differently scaled activations in each branch which in turn leads to different effective learning rates in each branch. This normalization scheme circumvents this problem.

We place a standard RPN on top of this feature normalization which consists of a 3×3 convolution using the same number of channels than the preceeding layer. The output of this RPN is then used in two additional convolutional layers which predict anchor scores and regressors (see [2] for details). In the case of RPN_{conv3} we use the features from the *conv3* layer for predicting bounding boxes.

We fine-tune each of our RPNs on the F_{train} dataset for 40000 iterations with an initial learning rate of $\mu = 0.001$ on our set of anchors $\mathcal{A}$. The learning rate is decreased by a factor of $\gamma = 0.1$ after 30000 iterations. We then evaluate the trained RPNs on the different $F_{test,x}$ datasets while only considering the outputs for a single anchor at a time. As a result we are able to plot how effective the different feature maps are at predicting object proposals of a given size. Figure 3 shows the result of this experiment. Each point on the abscissa represents the result of an experiment with the corresponding $F_{test,x}$ dataset while the ordinate reports the performance for this experiment as MABO.

Figure 3 shows that for small objects the *conv5* feature map delivers results which are noticably inferior than the results generated by the *conv3* or *conv4* feature maps.

Another observation to be made is that earlier feature maps deliver a more localized response for every anchor than the *conv5* feature map. This manifests itself in a steeper performance drop as the object size moves away from the ideal anchor size. This is a consistent pattern over all examined object sizes: Even medium sized objects with a side length between 80px and 100px are better predicted by the *conv4* feature map. However, this in only true if the object size closely matches the anchor size. The *conv5* feature map is able to deliver a more stable performance over a larger range of object sizes.

[2] *conv3* refers to the output of the last layer of the *conv3* block which is *conv3_3* when using the naming convention of [9]

3.2. ROI Classification of small objects

After identifying ROIs, Faster RCNN predicts a score and bounding box regressants for each ROI and for every class. In the original approach, this stage re-uses the previously computed *conv5* feature map which was used to generate the object proposals. An ROI-Pooling [1] layer projects the ROI coordinates identified by the RPN onto the feature map using the downsampling factor of the network. The corresponding area of the feature map is converted into a fixed-dimensional representation with a pre-determined spatial resolution (usually 7×7). Each of these feature representations is then fed into several fully connected layers for classification and class-specific bounding box regression.

We perform an analysis of the performance of the classification stage by object size which is similar to our analysis of the RPN. Unlike RPNs, where each anchor by virtue of its size and the overlap criterion self-selects appropriate training examples, the classification stage does have this property. We therefore need to be careful about the size distribution in the training set.

For the scope of this paper we are interested in the maximum performance each feature map can deliver for a specific object size. In order to avoid any effects from size distribution we ideally want a separate training set for each test set $F_{test,x}$. To reduce the training effort, we combine multiple sizes into a single training set. For this purpose we generate four training sets $F_{train,a,b}$ where a represents the minimum object size and b the maximum object size as the square root of the object area. We choose $(a, b) \in \{(20px, 60px), (40px, 80px), (60px, 100px), (80px, 120px)\}$ to adequately cover the range of small objects in the FlickrLogos dataset (Figure 1).

Similar to our evaluation of the RPN, we generate three versions of the classification pipeline: CLS_{conv3}, CLS_{conv4} and CLS_{conv5}. CLS_{conv5} is identical in architecture to the default pipeline described in [1]. The other two networks are similar: They only differ in the feature map that they are based on, and the normalization layer described in chapter 3. During training, we only train the fully-connected layers and compare these results to a network where all layers are optimized (CLS_{conv5}(all)).

We train each of these networks on all of the training sets $F_{train,a,b}$ and evaluate their mean average precision (mAP) on all the test sets $F_{test,x}$ where $a \leq x \leq b$. Since the ranges of object sizes between the training sets overlap with each other, we obtain multiple mAP values for each object size x – represented by the test set $F_{test,x}$. We take the maximum mAP for each version of the classification pipeline. To eliminate the influence of bad proposals on the classification performance we assume a perfect RPN for our experiment and evaluate our networks using the groundtruth bounding boxes as object proposals.

Figure 4 shows the results of this experiment. Unsurpris-

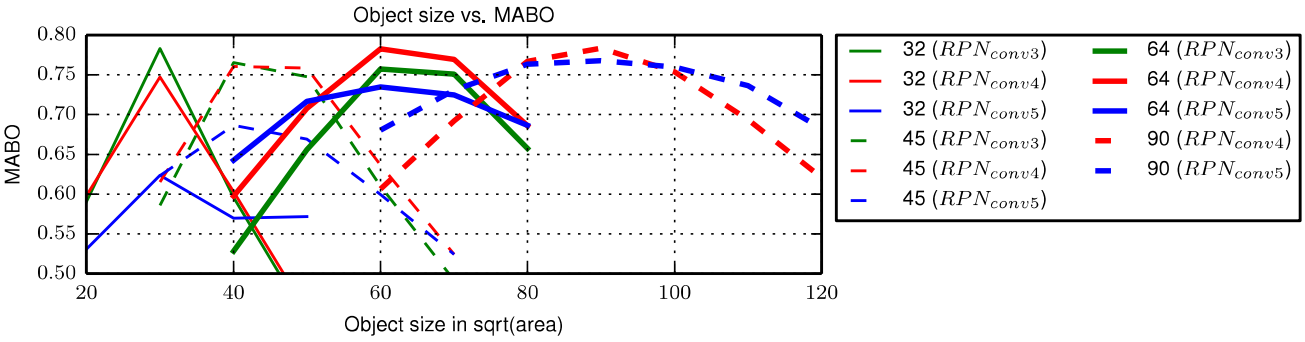

Fig. 3. RPN performance (MABO) for different anchor sizes as a function of object size. (green) performance of *conv3*, (red) performance of *conv4*, (blue) performance of *conv5*. Proposals for small objects can be generated more effectively by earlier layers while the performance of the *conv5* layer drops noticeably.

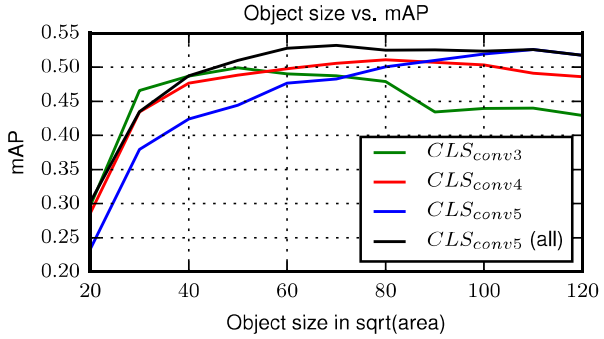

Fig. 4. Performance of the classification pipeline by size. The performance for *conv5* features drops noticeably for small object sizes. However, a full optimization (*conv5* (all)) is able to adapt to a wide range of scales.

ingly, the classification performance generally declines for small object instances. The performance of the CLS_{conv5} network declines more strongly than the performance of the CLS_{conv3} network. However, when all layers of the network are being optimized, the *conv5* feature delivers a good performance across all object sizes.

We therefore conclude that the classification stage in principle has similar difficulties to classify small objects given only low-resolution feature maps. However, the filters are able to adapt accordingly when given the option.

4. AN INTEGRATED DETECTION PIPELINE

We have shown that earlier feature maps can help to improve region proposals for small object instances. Furthermore, we have shown that the classification stage does not benefit from higher resolution feature maps if all layers of the network are being optimized. We want to exploit our observations in order to demonstrate their benefit on a real-world dataset. For this purpose, we propose a straightforward extension to the Faster R-CNN pipeline:

The results shown in Figure 3 show only a marginal performance benefit when using the *conv3* feature map compared to *conv4* features. For simplicity, we therefore only consider *conv4* and *conv5* features.

We have shown in Figure 4 that classification works best when using (fully optimized) *conv5* features. Figure 3 shows that anchors up to 64px are consistently predicted more accurately using the *conv4* feature map.

We therefore propose the following modified pipeline for Faster R-CNN: An additional branch is added to the original network architecture, beginning at the *conv4* feature map. This branch consists of a normalization layer (as described in section 3) and a separate RPN which is responsible for predicting a subset of anchors with scales $\mathcal{A}' = \{32, 45, 64\}$.

The main branch of the network remains unchanged. All other anchors are still predicted using the *conv5* feature map. There exists a single classification pipeline operating on *conv5* features. Like Faster R-CNN, the network can be fine-tuned end.

During test time, the proposals generated in each branch are subjected to their own non-maximum suppression. The proposals from both branches are then merged and undergo a joint non-maximum suppression. The number of all proposals is limited to $n = 2000$ which are fed into the classification stage.

We evaluate our approach by separately evaluating the RPN and classification performance on the original FlickrLogos dataset for several versions of the detection pipeline.

RPN (orig,default) refers to the RPN performance (in terms of MABO) for the original Faster R-CNN approach with the default anchor set. *RPN (orig,adj,old)* describes the original RPN with a set of anchors scales $\mathcal{A}_{ext} = \{32, 64, 128, 256\}$ which as been adjusted for better cover-

Configuration	RPN (MABO)	CLS (mAP)
orig,default	0.52	0.51
orig,adj,old	0.66	0.62
orig,adj,new	0.68	0.66
mres,adj,new	0.69	0.66

Table 1. Performance evaluation for both the proposal (RPN) and classifcation (CLS) stage on the FlickrLogos dataset.

age of the size distribution of the FlickrLogos datset but is following the traditional power-of-two approach of [2]. Similarly, *RPN (orig,adj,new)* describes the original RPN architecture which uses the anchor set $\mathcal{A}$ which employs our new strategy for scale selection as described in section 3. Finally, *RPN (mfeat,adj,new)* shows the performance of the new network architecture using multiple feature maps and our new set of anchors $\mathcal{A}$.

Likewise, *CLS (orig,default)* shows the classification performance (in terms of mAP) of the original pipeline using the default anchor set. *CLS (orig,adj,new)* refers to the original classification pipeline using our improved anchor set $\mathcal{A}$ and *CLS (mfeat,adj,new)* measures the performance of our new network architecture using the anchor set $\mathcal{A}$.

The results of this evaluation are shown in Table 4. The localization performance of the RPN can be improved by selecting anchors according to our scheme in chapter 3. If additionally, higher resolution feature maps are used, the quality of the proposals can be increased further. The above-mentioned techniques also improve the detection performance. However, the performance gains are less noticeable.

5. CONCLUSION

We have evaluated in detail the behavior of Faster R-CNN for small objects for both the proposal and the classification stage using artificial datasets. In our experiments we have observed that small objects pose a problem for the proposal stage in particular.

These difficulties are partially due to the inability of the RPN to accurately localize these objects because of the low resolution of the feature map. Also, we have shown that for small objects the choice of anchor scales is of great importance and have provided a criterion by which to choose anchor scales depending on the desired localization accuracy.

We have shown that the classification stage is able to adapt to small objects. Finally, we have validated our observation in the form of a simple extension to Faster R-CNN which is able to improve the overall detection performance on a real world dataset for company logo detection. In future work we would like to address how to improve the performance of the classification stage for small objects.

6. ACKNOWLEDGMENTS

This work was funded by GfK Verein. The authors would like to thank Carolin Kaiser, Holger Dietrich and Raimund Wildner for the great collaboration. Especially, we would like to express our gratitude for their help in re-annotating the FlickrLogos dataset.

7. REFERENCES

[1] R. Girshick, "Fast r-cnn," in *IEEE CVPR*, Dec 2015, pp. 1440–1448.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE PAMI*, 2016.

[3] S. Romberg, L. G. Pueyo, Lienhart R., and R. van Zwol, "Scalable logo recognition in real-world images," in *ACM ICMR*. 2011, pp. 25:1–25:8, ACM.

[4] Zhangyang Wang, Shiyu Chang, Yingzhen Yang, Ding Liu, and Thomas S. Huang, "Studying very low resolution recognition using deep networks," *CoRR*, vol. abs/1601.04153, 2016.

[5] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *IEEE CVPR*, June 2016, pp. 2874–2883.

[6] C. Eggert, A. Winschel, D. Zecha, and R. Lienhart, "Saliency-guided selective magnification for company logo detection," in *IEEE ICPR*, December 2016.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, June 2010.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML 2015*, 2015, pp. 448–465.

[10] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE CVPR*, June 2009, pp. 248–255.