Detecting cars in aerial photographs with a hierarchy of deconvolution nets

Satyaki Chakraborty Daniel Maturana Sebastian Scherer

CMU-RI-TR-16-60

November 2016

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

***Abstract*—Detecting cars in large aerial photographs can be quite a challenging task, given that cars in such datasets are often barely visible to the naked human eye. Traditional object detection algorithms fail to perform well when it comes to detecting cars under such circumstances. One would rather use context or exploit spatial relationship between different entities in the scene to narrow down the search space. We aim to do so by looking at different resolutions of the image to process context and focus on promising areas. This is done using a hierarchy of deconvolution networks with each level of the hierarchy trying to predict a heatmap of a certain resolution. We show that our architecture is able to model context implicitly and use it for finer prediction and faster search.**

***Keywords*—Object detection, Neural networks, Deconvolution nets**

I. Introduction

While there are several methods for detecting objects like cars on roads, prevalent object detection algorithms perform well when the target object to be detected occupies a significant portion of the image. The main reason why such algorithms fail to provide impressive results in our case is that in our dataset cars are often so small that they might just as well be treated as noise. Also given that, such aerial images captured from a MAV are usually huge, a few megapixels, a sliding window approach will take a significant amount of time processing the entire image. Even downsampling the image to a lower resolution is not a good idea, as lowering the resolution means potential loss of information(a small car might just vanish when the image is downsampled to a much lower resolution). Hence we don't want to naively downsample or use a sliding window based method. Instead we propose an intermediate strategy where we process the image at low resolution to choose areas to examine at higher resolution. We do this via a hierarchical network architecture. At every level of our architecture, we take help of context to locate areas of interest where we might find cars in an image and narrow down our search scope as we go up to the next level. This not only helps us to detect cars in a large photograph without downscaling it but also discard areas where there is absolutely no chance of finding cars, thereby resulting in faster prediction.

II. Related Work

A. Object detection methods

One of the very common ways of doing object detection and semantic segmentation in natural images is rcnn (regions with cnns) [1]. Essentially, this approach involves identifying region proposals in an image and passing it through a convolutional neural network. Features are hence extracted from the CNN and used for classification tasks, refining bounding boxes etc. One thing that is to be noticed here is that the region proposals used are bottom-up proposals based on approaches like selective search [2], category independent object proposals[3], multi-scale combinatorial grouping [4] etc. RCNN, however is a relatively slow process and not meant for detecting and identifying small objects. In recent years, there have been several modifications to the original RCNN approach. Fast RCNN [5], for example, surpasses the performance of the original RCNN architecture in terms of accuracy while being 213x faster during test time. While both these approaches rely on bottom-up region proposal detection approaches like selective search, the faster RCNN [6] shares full-image convolutional features with the object detection network, thus enabling nearly cost-free region proposals and real-time object detection.

YOLO [7] is another approach for fast object detection in natural images (which also have generalised well in case of artwork). The basic idea is to divide the image into SxS grid cells and for each grid cell predict a bounding box with a confidence score. Simultaneously, each grid cell is also categorized into what class it belongs to. Finally both pieces of information are merged to produce the final result, The main reason why YOLO performs so fast is because it treats the object detection problem as a regression problem by predicting bounding boxes around objects and hence it does not require a complex pipeline. Although YOLO is a faster method compared to RCNN it is relatively less accurate and hence is even worse for detecting small objects.

Figure 1 A sample image showing cars that are to be detected from the aerial vehicle. Some cars are too small to be identified distinctly by a human eye.

S. Chakraborty e-mail: (satyaki.cs15@gmail.com D. Maturana email:(dimatura@gmail.com) S. Scherer email:(basti@andrew.cmu.edu) This work was done when S. Chakraborty was a summer intern at the AirLab, RI, CMU under the mentorship of D. Maturana and S. Scherer.

B. Methods for capturing context

Exploiting context becomes useful for a variety of tasks from object detection to semantic segmentation. rCPN [14], for example, semantically segments an image using a recursive neural network that propagates context. [13] uses spatial context to learn to generate rich visual representations, which are then used for unsupervised visual discovery of objects in an image. [12] tries to model the relationship between an object and its surrounding context and in the process shows that context can be a rich source of information about an object's identity.

One approach that is of particular interest and can be viewed as a possible solution to the problem is the memex model [8] which takes an image as input and builds a graphical model of different segments or objects present in the image. Such a model is capable of capturing the spatial relationship between different objects present in an image. Once we have built the graphical model from an image, we can easily perform queries, search for objects etc. Although, this might give a solution to our problem, this method requires presegmentation of the image, which becomes quite expensive for large aerial photographs.

C. Deconvolution nets and semantic segmentation

Deconvolution networks [9] as introduced by Zeiler et al. helps capturing features beyond certain edge primitives. A deconvolution network consists of an encoder part which is similar to a convolutional network without the softmax and the MLP layers, and a decoder part which is basically a mirror image of the encoder part i.e. the encoder part consists of convolution and pooling layers while the decoder part comprises of deconvolution (transposed convolution) and unpooling layers. While Zeiler et al. had originally used deconvolution networks for building representations, deconvolution networks have also been used for semantic segmentation and scene labelling. For example, [15] also advocates using a stack of deconvolution layers for non-linear upsampling while performing semantic segmentation. Noh et al. [10] used deconvolution networks for identifying pixel level labels and segmentation masks.They have also shown that finer details of objects in a scene are revealed as features are forward propagated through the layers of the deconvolution network. This suppresses noisy activations and amplifies only those which are related to any target class, thus aiding better scene labelling. We take inspiration from these methods and use deconvolution nets for object detection while modifying our architecture in certain ways, which is discussed in the following section.

III. Our approach

One thing that is evident, is that in order to narrow down our search space for finding cars, we need to use context as a tool or exploit spatial relationship. This is because of the generic assumption that a car is more likely to be on a plain road than on a grassland. In stead of looking directly into a particular patch at a high resolution, we first look into the entire image at a low resolution to find out areas of interest. Once we have done that we now look at promising area in the image at a relatively higher resolution and repeat this process for several times, each time increasing the resolution of the patch we are looking into. With our hierarchy of neural networks we aim to do so by capturing context implicitly – a particular level of the hierarchy aims to predict regions of interests with a certain degree of accuracy. Lower levels have a large context as input but make very coarse predictions. Higher levels only have a portion of the high resolution photograph as input but predictions are finer. The complete architecture is described in the following section.

Layer	kernel size	stride	pad	Output dim.
Input	-	-	-	1x128x128
Conv-1	3x3	1	1	64x128x128
Pool-1	2x2	2	0	64x64x64
Conv-2	3x3	1	1	128x64x64
Pool-2	2x2	2	0	128x32x32
Conv-3	3x3	1	1	512x32x32
Pool-3	2x2	2	0	512x16x16
Conv-4	3x3	1	1	1024x16x16
Pool-4	2x2	2	0	1024x8x8
Unpool-1	2x2	2	0	1024x16x16
Deconv-1	3x3	1	1	512x16x16
Unpool-2	2x2	2	0	512x32x32
Deconv-2	3x3	1	1	128x32x32
Unpool-3	2x2	2	0	128x64x64
Deconv-3	3x3	1	1	64x64x64
Unpool-4	2x2	2	0	64x128x128
Deconv-4	3x3	1	1	1x128x128

TABLE I. The first half of the network consists of convolution (conv) and pool layers, followed by a number of deconvolution (deconv) and unpooling layers.

A. Architecture

At the heart of the proposed architecture lie three deconvolution networks. The architecture of the three deconvolution nets in the three different levels of the hierarchy are identical and described in table 1. Similar to Zeiler et al.'s, work we also make use of switch matrices (in our case to aid construction of heatmaps).

Our motivation for using deconvolution networks for predicting heatmaps arises from their use as a tool for semantic segmentation as discussed in the previous section. In our case, we use such networks to predict heatmaps that highlight regions containing cars or regions with possibility of finding cars. All three networks have the same exact architecture taking a 128x128 image as input and producing a heatmap of the same resolution. The lowest level network takes the entire image as input at 128x128 resolution and is trained to predict areas where there is a possibility of finding cars. How we have trained different levels of the architecture is discussed in the training methodology section. After the prediction has been made at the lowest level, we perform binarisation using a threshold of 0.5(emperically chosen). Next we divide the binary image into 4 sections equally (each of dimension 64x64), and isolate those sections where there are white patches (i.e. possibilities of finding cars). We remember the positions of these patches and use this information for selecting 128x128 patches from the image at 256x256 resolution where we should

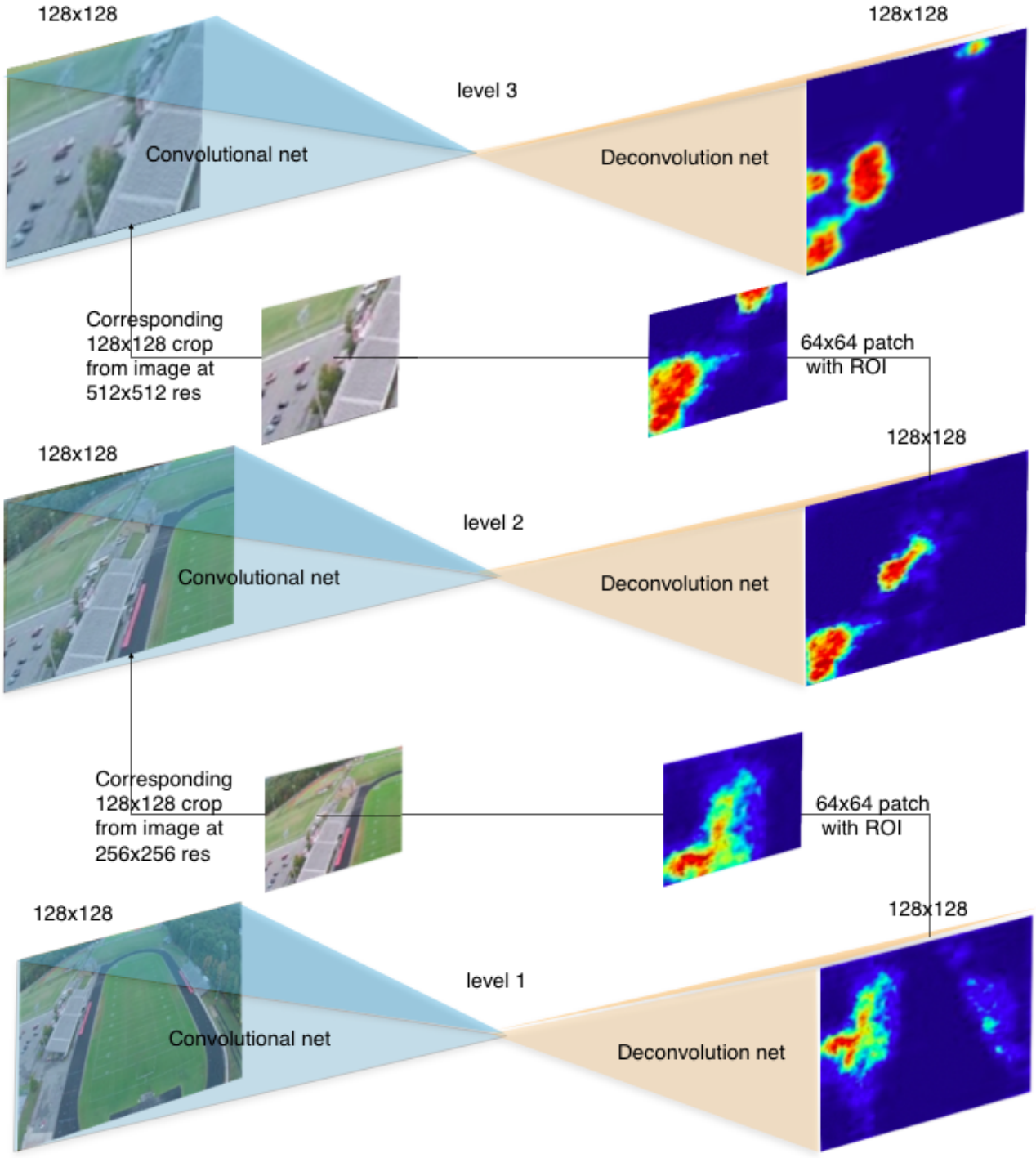

Figure 2 The overall architecture of the hierarchy of deconvolution networks. The hierarchy consists of 3 different levels. Each of the three levels essentially consists of a single deconvolution network, the exact architecture of which is described in Table 1. As we go up the hierarchy, the size of context provided as input decreases on one hand, but the prediction gets finer on the other.

look for cars at level 2 of the hierarchy. Now the entire process is repeated as we go from level 2 to level 3. Level 3 takes 128x128 patches from the image at 512x512 resolution (the highest resolution we have considered in our tasks) and tries to predict heatmaps of cars as accurately as possible. Finally we stitch all the level 3 predictions together to get the final heatmap.

B. Training Methodology

The dataset of aerial images used in this case consisted of roughly 300 images. Hence for better generalisation we pretrained the network on the MSCOCO [11] car dataset which consists of roughly 2000 labelled images and later on fine-tuned the parameters while training on our own dataset. How we have trained at different levels of the hierarchy is explained as follows.

As already discussed, the lowest level of the hierarchy aims to predict regions with high probability of finding cars, whereas the highest level tries to predict cars as accurately as possible. Hence, as we go up the hierarchy our prediction gets finer, although the context used for prediction decreases. The labels against which we train our network are different for the three different levels. For training at level 1, we downsample the label to 128x128 resolution and perform dilation and gaussian blurring on the same. The reason for dilation and blurring is that at this level we want to train the network in such a away that it is able to predict areas of interests and not the individual cars. For training at level 2, we downsample a label to 256x256 resolution, crop out 128x128 patches and then perform dilation and blurring on those. It is to be noted that as the level increases we reduce the parameters for dilation and blurring the labels. At level 3, we directly crop out 128x128 patches from the label at original 512x512 resolution and train the network without any dilation and blurring. Figure 3a and 3b show the labels for training at level 3 and level 1 respectively.

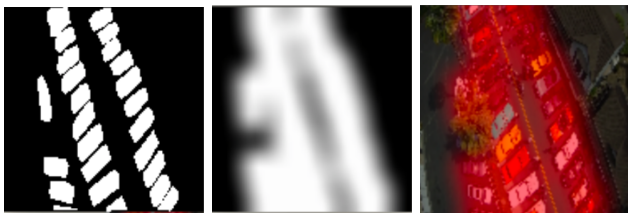

Figure 3a, 3b and 3c (from left to right). 3a is a sample label with which we train the network at level 3, the highest level. 3b is a sample label with which we train the network at level 1. 3c is a blended image which shows how simple dilation and blurring is able to capture part of the context surrounding the cars.

IV. Results and Analysis

We now show the performance of our approach on the test dataset. First we show the predictions made by the three different levels of the architecture in fig. 4. The results agree with our previous claim that as we go up the hierarchy our prediction gets finer.

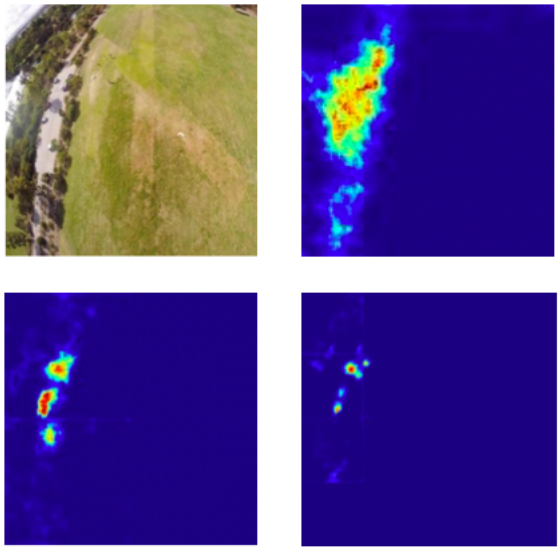

Figure 4: 4a(top left) is the actual input image, 4b(top right) is the prediction at level 1, 4c(bottom left) is the prediction at level 2, 4d(bottom right) is the prediction at level 3. Level 1 prediction is all about capturing the regions of interest like roads in this case, whereas level 3 tries to predict the location of the cars as accurately as possible.

Fig. 4 shows an example which highlights the importance of the dilation and blurring done on the labels while training the lower levels of the network. The prediction at level 1 tries to identify potential regions of interest. In this case the roads are predicted with somewhat higher probability compared to the grassy areas. This shows the ability of the lower levels of the hierarchy to identify potential regions of interest, as in areas where to look for our target object.

Fig. 4 also shows the significance of using a hierarchy of neural networks. As we go up the hierarchy, we only consider those patches or areas where the network thinks there is a chance of finding the target object(s). For eg., in the level 1 prediction the right half of the heatmap does not indicate any region of interest, so we only consider the left half of the input image at level 2. This not only decreases the search space efficiently, thereby reducing the overall time taken to locate the objects of interest, but also takes care of reducing the number of false positives as redundant patches are discarded as the level increases.

We show the precision recall curve for labelling cars at the pixel level in Fig. 5. The curve shows a low recall value in general. This is because of the presence of some false negatives at the level 3 (the highest level) prediction. The main reason for such is primarily the bottleneck of the neural network – as we go deep inside a neural network, by the time we reach the end of the convolution network, due to successive averaging out and pooling, most of the information about very tiny cars are essentially lost. Figure 6a shows a patch consisting of a congested scene fed to the level 3 network and figure 6b shows a feature map generated from the patch (figure 6a) at layer 4 of

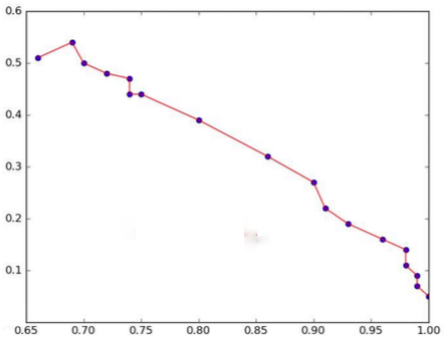

Figure 5 *Shows the precision recall curve. The X axis is the precision axis whereas the Y axis is the recall axis.*

the level 3 neural network. As is evident from the figure, most of the information of the small cars have essentially vanished. One thing that can be inferred from this is that we need to somehow preserve the information of such small objects as we go deep inside the neural network.

Figure 6: 6a(a sample input patch fed to the level 3 neural network), 6b(a sample feature map generated at layer 4 of the level 3 neural network from the input patch)

V. Conclusion and Future Work

Accurate detection of cars from aerial vehicles like MAVs can be extremely useful for traffic monitoring. We have shown why bottom-up approaches for object detection are not much helpful in this particular scenario and hence proposed a novel way of solving this problem from a top-down perspective by exploiting context. More importantly, our approach tries to solve a small piece of a bigger puzzle – by narrowing down our search space using context for detecting cars we aim to exploit spatial relationship to identify any particular object of interest. Since the architecture is able to capture context implicitly, hopefully, the same architecture can be trained to identify pedestrians, other aerial vehicles, any object of interest which might be too small to be detected from an aerial view.

However, experiments conducted on our small dataset have revealed the tendency of our architecture to detect false negatives, a possible explanation of which is given in section 4. One way to solve this issue would be to use connections across different layers in the same neural network. For eg., a layer can have the output of the previous layer as an input as well as a scaled version of the original input image patch. Fig. 7 shows an example of such an architecture. This way it might

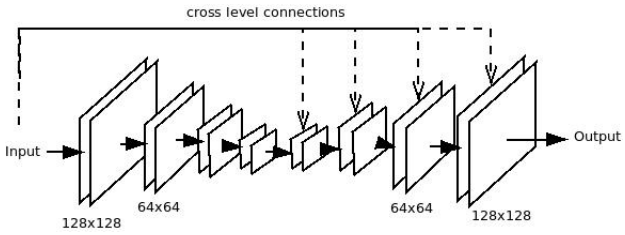

Figure 7 A possible solution to the stated issue by using cross layer connections from the input layer.

be possible to somehow counterbalance the information loss by preserving texture, shape, colour and other similar properties.

References

[1] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[2] Uijlings, Jasper RR, et al. "Selective search for object recognition." International journal of computer vision 104.2 (2013): 154-171.

[3] Endres, Ian, and Derek Hoiem. "Category independent object proposals." European Conference on Computer Vision. Springer Berlin Heidelberg, 2010.

[4] Arbelez, Pablo, et al. "Multiscale combinatorial grouping." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[5] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[6] Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

[7] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." arXiv preprint arXiv:1506.02640 (2015).

[8] Malisiewicz, Tomasz, and Alyosha Efros. "Beyond categories: The visual memex model for reasoning about object relationships." Advances in neural information processing systems. 2009.

[9] Zeiler, Matthew D., et al. "Deconvolutional networks." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010.

[10] Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[11] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European Conference on Computer Vision. Springer International Publishing, 2014.

[12] Torralba, Antonio. "Contextual priming for object detection." International journal of computer vision 53.2 (2003): 169-191.

[13] Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[14] Sharma, Abhishek, Oncel Tuzel, and Ming-Yu Liu. "Recursive context propagation network for semantic scene labeling." Advances in Neural Information Processing Systems. 2014.

[15] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.