Learning perspective-free counting via dilated convolutions

Diptodip Deb  
College of Computing  
Georgia Institute of Technology  
Atlanta, Georgia, USA  
Email: diptodipdeb@gatech.edu

Jonathan Ventura  
Department of Computer Science  
University of Colorado  
Colorado Springs, Colorado, USA  
Email: jventura@uccs.edu

Abstract—We propose the use of dilated convolutions as a simpler approach to the perspective-free counting problem. Counting is a common problem in computer vision (e.g. cells in a microscope image or pedestrians in a crowd). Modern approaches to the counting problem involve the production of a density map via regression whose integral is equal to the number of objects in the image. This method of counting can also be used to locate objects in the image if the regressor used has enough accuracy and precision. However, objects in the image can occur at different scales (e.g. due to perspective effects) which can make it difficult for a neural network to learn the proper density map. A recent result for multiscale counting involves the use of a complicated pyramid of image patches. However, dilated convolutions have been shown to allow for the incorporation of multiscale information without such a complicated design in segmentation problems. We see that our dilated convolutional regression network obtains results comparable to and occasionally superior to the current state of the art.

Keywords—Computational and artificial intelligence, neural networks, machine vision.

I. INTRODUCTION

Many computer vision problems involve dense prediction [1]. Generally, these problems can involve discrete or continuous labeling of images. Counting objects in an image is a specific sub-problem of this sort. Specifically, the problem involves enumerating the number of objects in a given still image or video frame [2]. We have seen that good performance on counting tasks can be achieved without learning to detect and localize dense objects in images through the regression of a density map [2].

This means we can focus on the specific case in which each object has been labeled with a dot (one dot per object). In this case, the supervised learning agent simply learns a regression function to produce the density map such that the integral of the density map is equal to the number of objects in the image [2].

Indeed, we see that in many natural counting problems (such as those involving cells in a microscopy image, pedestrians in a crowd, or a traffic jam), individual detectors are not reliable [3]. This is due to a variety of challenges including overlap of objects, perspective shifts causing variance in shapes and sizes of objects, etc. [3].

A recent result in counting achieves state-of-the-art performance on counting objects that might be transformed by such perspective shifts [3]. This approach involves the regression of a density map as in [2], however their approach involves a complex convolutional neural network architecture that samples re-sized patches of different scales in order to incorporate multiscale information [3].

Indeed, convolutional neural networks have proven to provide state-of-the-art performance on a variety of dense prediction computer vision tasks [4], [5]. However, it is not clear whether the approach of sampling patches of varying scale is necessary for incorporating multiscale information [1].

Dilated convolutions, a simple modification to the structure of traditional, straightforward convolutional neural network designs, have proven to provide competitive performance in the dense, multiscale segmentation problem in which objects of different sizes in an image must be segmented [1]. We propose the use of such a network for the problem of regression of a density map for the purposes of counting rather than classification and segmentation. Such a network would have a much simpler architecture than that of [3], bypassing the need to sample multiscale patches of an image.

II. RELATED WORK

Counting using a supervised regressor to formulate a density map was first shown by [2]. In this paper, Lempitsky et al. show that the minimal annotation of a single dot blurred by a Gaussian kernel produces a sufficient density map to train a network to count. All of the counting methods that we examine as well as the method we use in our paper follow this method of producing a density map via regression. This is particularly advantageous because a sufficiently accurate regressor can also locate the objects in the image via this method. However,

Fig. 1. UCSD pedestrian traffic data (left) and simulated microscopy image of biological cells (right). These are example images that can be used in counting tasks.
The work of [1] proposes the use of dilated convolutions as a viable alternative to the more complicated architecture of the counting problem. These issues are addressed by the work of [3]. Rubio et al. show that a fully convolutional neural network can be used to produce a supervised regressor that produces density maps as in [2]. They further demonstrate a method dubbed HydraCNN which essentially combines multiple convolutional networks that take in differently scaled image patches in order to incorporate multiscale, global information from the image. The premise of this method is that a single regressor will fail to accurately represent the difference in values of the features of an image caused by perspective shifts (scaling effects) [3].

However, the architectures of both [3] and [6] are complicated due to requiring multiple image patches and, as discussed in [1], the experiments of [7], [8] and [9]–[11] leave it unclear as to whether rescaling patches of the image is truly necessary in order to solve dense prediction problems via convolutional neural networks. In [7], [8], upsampling is used to recover scale information from downscaled layers, which puts into question the necessity of downsampling scaled layers in the first place. Further, it is also unclear in [9]–[11] as to whether separate inputs of rescaled patches of the image are necessary. The work of [1] proposes the use of dilated convolutions as a simpler alternative that does not require sampling of rescaled image patches to provide global, scale-aware information to the network. It should be noted that other methods of counting exist, including training a network to recognize deep object features via only providing the counts of the objects of interest in an image [12] and using CNNs (convolutional neural networks) along with boosting in order to improve the results of regression for production of density maps [13]. In the same spirit, [14] combines deep and shallow convolutions within the same network, providing accurate counting of dense objects (e.g. the UCF50 crowd dataset).

In this paper, however, we aim to apply the dilated convolution method of [1], which has shown to be able to incorporate multiscale information without using multiple inputs or a complicated network architecture, to the counting problem.

III. METHOD

A. Dilated Convolutions

We propose the use of dilated convolutions as an attractive alternative to the more complicated architecture of the HydraCNN [3]. We largely keep the architecture shown in [1], making a simple modification in order to produce a regression network rather than a segmentation (classification) network which is described below. Dilated convolutions, as discussed in [1], allow for the exponential increase of the receptive field with a linear increase in the number of parameters with respect to each hidden layer.

In a traditional 2D convolution, we define a real valued function \( F: \mathbb{Z}^2 \rightarrow \mathbb{R} \), an input \( \Omega_r = [-r, r]^2 \in \mathbb{Z}^2 \), and a filter function \( k: \Omega_r \rightarrow \mathbb{R} \). In this case, a convolution operation as defined in [1] is given by

\[
(F * k)(p) = \sum_{s+t=p} F(s)k(t).
\]

A dilated convolution is essentially a generalization of the traditional 2D convolution that allows the operation to skip some inputs. This enables an increase in the size of the filter (i.e. the size of the receptive field) without losing resolution. Formally, we define from [1] the dilated convolution as

\[
(F *_l k)(p) = \sum_{s+t=p} F(s)k(t)
\]

where \( l \) is the index of the current layer of the convolution.

We mostly keep the architecture of the network in [1] in terms of the dilated convolution filters. Because we have a regression problem as opposed to a segmentation problem, we do not implement the front end to extract features, leaving this to the dilated convolutions themselves. Furthermore, we use a ReLU activation for all the layers in order to facilitate regression of floating point 32-bit pixel-values (which have a range of 0 to 1). We refer to this network as the dilated convolutional regression network, henceforth shortened as the DCR network.

B. Experiments

We evaluated the performance of dilated convolutions against the HydraCNN on a variety of common counting datasets: UCF50 crowd data, UCSD crowd data, and TRANCOS traffic data [3] and [15]. For each of these data sets, we used labels given by the corresponding density map for each image. An example of this is shown in Figure III-A. Currently, we have performed experiments on the four different splits of the UCSD data as defined in [3] and the split of the UCSD data as defined in [15] (which we call the Shanghai split). We also evaluated the performance of our network on the TRANCOS traffic dataset [16]. We have also experimented with testing on small patches from the UCF data (discussed in section IV-C).
We have so far observed that dilated convolutions produce density maps (and therefore counts) that are on par with or better than those of HydraCNN [3]. We measure density map regression loss via L1 loss. We compare accuracy of the counts via mean absolute error for the crowd datasets and the GAME metric in the TRANCOS dataset as explained in Section IV-A3. Beyond the comparison to HydraCNN, we will also compare to other recent convolutional counting methods, especially those of [15], [12], [13], and [14]. Further experiments in counting could involve testing the effect of increasing dilation size and
deepening the network (which effectively increases the size of the receptive field) on the size of images the network is able to take in.

IV. Results

We perform experiments on various data sets. For all datasets, we use patched input images and ground truth density maps produced by summing a Gaussian of a fixed size ($\sigma$) for each object. This size varies from dataset to dataset, but remains constant within a dataset. We do not take any perspective information into account for training our network. All experiments were performed using Keras with the Adam optimizer at its default learning rate [17].

A. Datasets

1) UCSD: The UCSD crowd counting dataset consists of frames of video of a sidewalk. There are relatively few people in view at any given time (approximately 25 on average). Furthermore, because the dataset comes from a video, there are many nearly identical images in the dataset. For this dataset, there have been two different ways to split the data into train and test sets. Therefore, we report results using both methods of splitting the data. The first method consists of four different splits: maximal, downscale, upscale, and minimal. Minimal is particularly challenging as the train set contains only 10 images. Moreover, upscale appears to be the easiest for the majority of methods [3]. The second method of splitting this data is much simpler, leaving 1200 images in the testing set and 800 images in the training set [15].

2) UCF: UCF is a particularly challenging dataset. The difficulty is due not only to the very low number of images in the dataset, but also to the fact that the images are all of varying scenes. Furthermore, perspective effects are particularly noticeable for particular images in this dataset. The average image has on the order of 1000 people in a crowd in this dataset.

3) TRANCOS: TRANCOS is a traffic counting dataset that comes with its own metric [16]. This metric is known as $GAME$, which stands for Grid Average Mean absolute Error. $GAME$ splits a given density map into $4^L$ grids, or subarrays, and obtains a mean absolute error within each grid separately. The value of $L$ is a parameter chosen by the user. These individual errors are summed to obtain the final error for a particular image. The intuition behind this metric is that it is desirable to penalize a density map whose overall count might match the ground truth, but whose shape does not match the ground truth [16]. More formally, we define

\[
GAME(L) = \frac{1}{N} \sum_{n=1}^{N} \left( \sum_{l=1}^{4^L} |e_{n}^{l} - t_{n}^{l}| \right)
\]

where $N$ refers to the number of images, $L$ is the level parameter for $GAME$, $e_{n}^{l}$ is the predicted or estimated count in region $l$ of image $n$ and $t_{n}^{l}$ is the ground truth count in region $l$ of image $n$ [16].

B. UCSD Crowd Counting

For this dataset, each object is annotated with a Gaussian of size $\sigma = 8$. The ground truth map is produced by summing these. There are two different ways to split the dataset. We have experimented on the split that gave [3] the best results as well as the split used in [15].

We note that training this method using the symmetric dilation method as introduced in [18] results in density maps that are better in terms of the shape produced, but worse in terms of the actual count values. This can be seen in Figure IV-B. At best, symmetric dilations are approximately equivalent to the standard dilated regression network and at worst symmetric dilations are unable to learn anything at all on the same training set that was used to train the standard dilated regression network. Hence, we have proceeded with the standard dilation regression network.

1) Upscale Split: We see that the “upscale” split as defined in [3] gives us very good results on counting for this dataset. For this experiment, we sampled 1600 random patches of size $119 \times 79$ pixels (width and height respectively) for the training set and split the test set images into $119 \times 79$ quadrants that could be easily reconstructed by simply piecing them together without overlap. Results appeared consistent over multiple trainings.

2) Shanghai Split: We see that the “Shanghai” split as defined in [15] gives us somewhat worse results for counting on this dataset. For this experiment, we again sampled 1600 random patches of size $119 \times 79$ pixels (width and height respectively) for the training set and split the test set images into $119 \times 79$ quadrants that could be easily reconstructed by simply piecing them together without overlap. Results appeared consistent over multiple trainings. While the performance of the network was not as good, i.e. the network does not achieve state of the art performance, we see that the results are comparable to the state of the art and the previous state of the art. This is compelling because the purpose of the dilated regression network is to show that perspective-free counting can be learned without creating image pyramids or combining multiple CNNs learning features at different scales.

C. UCF Crowd Counting

For this dataset, we initially did not fully test the images. Instead, for this dataset we also test on random image patches of the same size as the training patches. We take 1600 random patches of size $100 \times 100$ for training. We do the same for testing. Ground truth density maps are produced by annotating each object with a Gaussian of $\sigma = 15$. We see that because the UCF dataset has over 1000 people on average in each image, the shapes output by the network in the density map are not as well defined or separated as in the UCSD dataset. This can be seen in Figure 3c. While the average error when testing on patches seems to be quite low as indicated by Table II, when we test on 5 cross validation folds of the data as defined in [3], we find that the error increases to an average of approximately 1000, which is far higher than the state of the art. The low average error on the patches is misleading because the average error of the patches is summed over the number of patches for each image. However, modifications to the DCR network could yield significantly better results for these dense images.
Fig. 4. Left: input counting image. Middle: Generated ground truth density map. Right: Dilated convolutional network prediction of density map on test image. Note the grid pattern in Figure 3a. The symmetric dilation network does somewhat mitigate this grid pattern as shown here, but it ultimately performs worse for obtaining an actual count.

For example, taking an average of overlapping patches of the image during testing, i.e. densely scanning using a stride of 64 pixels, yielded a mean absolute error on the first fold of approximately 640.

D. TRANCOS Traffic Counting

Our network performs very well on the TRANCOS dataset. We see that although the shapes in the density map no longer match as closely to the ground truth density maps, the counts are significantly more accurate than other methods. For training this dataset, we take $80 \times 80$ patches which we can stitch back together into the full-sized $640 \times 480$ images. As seen in Table III, we achieve state of the art results as measured by the $GAME$ metric [16]. We trained the DCR network with density maps produced with a Gaussian of $\sigma = 15$ as specified in [3].

V. CONCLUSION

A. Summary

We have proposed the use of dilated convolutions as an alternative to the complicated HydraCNN [3] or Multicolumn CNN [15] for the vision task of counting objects in images. While we largely keep the structure of the dilated convolutions the same as in [1], we use ReLU activations for the purposes
of regression. We have performed experiments on two different splits of the UCSD crowd counting dataset, the UCF crowd counting dataset, as well as the TRANCOS dataset. We obtain comparable or better results in two of three of these datasets as compared to [3]. In fact, the DCR network, which never uses perspective information in our experiments, occasionally outperformed HydraCNN with perspective information. These results show that the DCR network performs surprisingly well and is also robust to scale effects (the sizes of the cells in the images were varied randomly). Further, the DCR network shows promising results not only on the relatively low density UCSD dataset, but also on the higher density TRANCOS dataset. However, it performs rather poorly on the extremely dense and varied UCF dataset.

B. Future Work

We would like to compare this procedure on other large crowd datasets, specifically those of [6] and [22] for the World-Expo crowd dataset as well as [15] for the Shanghaitech crowd dataset. Further, we would like to attempt other techniques for training the UCF dataset to possibly improve results on highly dense images.

In addition to an analysis of performance on counting, we also plan to examine the ability of these different approaches to locate the objects in the image. As mentioned previously, if the regressor is accurate and precise enough, the resulting density map can be used to locate the objects in the image (and we expect this to outperform more traditional feature/localization-based methods for dense images where features may be difficult to extract). We expect that in order to do this, we will have to regress each object to a single point rather than a region specified by a Gaussian. Perhaps this might be accomplished by thresholding the activations of the final layer. Moreover, we might examine the effect of increasing the depth of the DCR network along with the size of its dilations on the resulting regressed density maps.

Because the results of the dilated convolution network are promising, we might consider extending this work to the case where perspective effects occur in such a way that objects “behind” the current plane of focus should have the same density shape, but a lower density value. This situation arises when looking at smFISH confocal microscopy, which results in a 3D stack of images representing slices of a cell going down in vertical space. If the regressor is able to accurately label the densities in these images, we could potentially use the dilated convolution regression network to feed features to a recurrent neural network and obtain accurate counts of these densely packed 3D single molecules.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. 1359275 and 1659788. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Furthermore, we acknowledge Kyle Yee and Sridhama Prakhya for their helpful conversations and insights during the research process.

REFERENCES


