

Crowd counting on fixed camera images

Pierpaolo D’Odorico

pierpaolo.dodorico@studenti.unipd.it

Massimiliano Conte

massimiliano.conte@studenti.unipd.it

Abstract

In this work we compared different computer vision techniques in order to estimate the number of people in a frame. The counting is performed on images captured from a fixed camera placed in a shopping mall. Some applications of this kind of counting on a static view are security and safety tasks, estimating the number of visitors on a mall for a/b testing purpose, planning spaces and services or verify compliance with covid-19 social distancing.

1. Introduction

The crowd counting problem received a lot of attention in recent years, due to its direct connection with crowd control and public safety. For this reason many techniques were recently proposed. Our idea is to compare two main techniques in this fixed camera setting, one that is fast to implement and the other one more challenging, in order to verify if it is worth spending time for a more sophisticated solution. The first approach is to directly estimate the number of people performing regression with a deep convolutional neural network, such as the VGG16 network [1]. We chose this very deep network since it is easy to handle for our purposes, unlike more complex architectures that have, for example, skip layer connections, and also because it is the base for the second technique. The second approach performs an undirect estimate of the number of people. First it is estimated the density of people in the image, then starting from the obtained density map the count is inferred. This second approach represents the base idea for the state of the art methods in crowd counting, where images could have a completely different number of people on different environments and perspective. After implementing the two approaches on our problem, we found that a simple regression based on neural networks could perform as good as density based approach, probably exploiting the fixed background and because density estimation methods are well suited for dense datasets.

2. Related work

2.1. VGG net

We based our work on information contained in different papers about computer vision tasks and crowd counting. The first one is related to the base network of both approaches, the VGG16 net [1]. In this paper the authors investigated the effect of a deeper (with respect to previous architectures) convolutional neural network on the classification accuracy in the large-scale image recognition setting, specifically on the *imageNet* dataset [2]. In this convolutional neural networks they used very small (3x3) convolution filters, which have shown a significant improvement on the prior state of the art configurations. They also pushed the depth to 16-19 weight layers (Fig.1). Those convolutional filters learned during *imageNet* classification task can be useful in our application, giving a meaningful feature extraction for finding people in images.

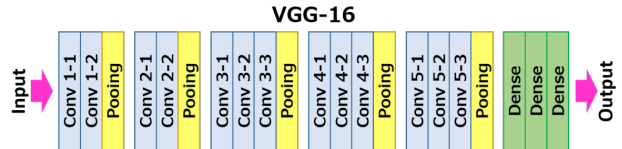


Figure 1: VGG16 deep CNN architecture.

2.2. Density based approach

The crowd counting based on density map estimation is a well known approach. One solution that is robust with respect to a variety of image properties is the *Multi-Column Convolutional Neural Network (MCCNN)* [3]. The authors designed a system capable of detecting heads of different sizes, both in dense or sparse situations. Their work is first based on the generation of ground-truth density maps via geometry-adaptive kernels, for handling both dense and sparse images. Their architecture is designed in such a way that is able to detect heads of different sizes, in particular they built a neural network with three branches, one for head size, small, medium and large heads. For the training they pre trained separately each branch, and then the full network. The output of the network was designed

and trained to be the estimation of the ground truth density maps. In order to train such a network, since are required ground truth density maps, a dataset with head annotations is a requisite, and for this reason they also introduced *Shanghai-tech*, a new large scale crowd counting dataset.

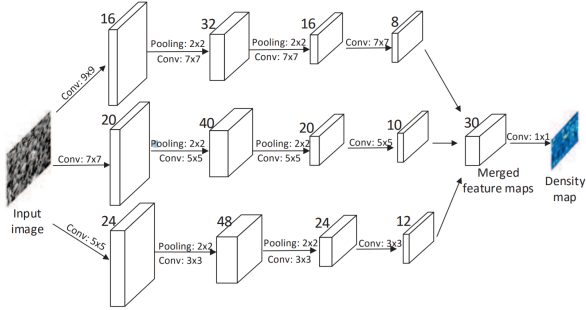


Figure 2: MCCNN architecture.

The models that we implemented are related to a more recent paper that simplifies the convolutional neural network of the previous work, by removing the multi-branch structure and going further with the deepness. They also experimented with the stride of convolutions, in an architecture they called *CSRNet* [4]. Designing a model which is automatically able to distinguish heads of different sizes achieved better results with respect to *MCCNN*. This architecture is basically the *VGG16* net (without the classifier), with other convolutional layers on top of it. For the training they fine tuned the network, starting from *ImageNet* learned weights for the *VGG16* part and random normal initialization for the others. Since those methods [3, 4] don't use dense layers, they can work with images of any size.

3. Datasets

3.1. Mall dataset

3.2. Shanghai tech dataset

This dataset was introduced in the *MCCNN* paper [3]. This is a large-scale crowd counting dataset which contains 1198 annotated images, with a total of 330,165 people with centers of their heads annotated. It consists of two parts: there are 482 images in Part A which are randomly crawled from the Internet, and 716 images in Part B which are taken from the busy streets of metropolitan areas in Shanghai. The crowd density varies significantly between the two subsets, in particular the part A is extremely dense, while part B is sparser, but still a dense dataset. Both Part A and Part B are divided into training and testing: 300 images of Part A are used for training and the remaining 182 images for testing, and 400 images of Part B are for training and 316 for testing. We used Part B because is similar to the *mall dataset*, since

input(unfixed-resolution color image)
front-end
(fine-tuned from VGG-16)
conv3-64-1
conv3-64-1
max-pooling
conv3-128-1
conv3-128-1
max-pooling
conv3-256-1
conv3-256-1
conv3-256-1
max-pooling
conv3-512-1
conv3-512-1
conv3-512-1
back-end
conv3-512-2
conv3-512-2
conv3-512-2
conv3-256-2
conv3-128-2
conv3-64-2
conv1-1-1

Figure 3: Best CSRnet (kernel size - # of filters - stride).

it is sparser.



(a) An image from Part A.



(b) An image from Part B.

4. Method

4.1. Regression based approach

4.2. Density based approach

4.2.1 Ground truth generation

Since we want the model to estimate the crowd density, we need ground truth density maps in order to perform supervised learning. A density map is a single channel image with positive values associated to each pixel. The generation of this image involve the head annotations: for each labeled image we have the coordinates of each head in that image. We can see each annotation as a sparse matrix, whose dimensions are the same as the image, with all zeros but one entry equals to one, corresponding to the pixel in the center of the head. As it is done in [3], we used geometry-adaptive kernels. The density map F is obtained via the following formula:

$$F(x) = \sum_{i=1}^N \delta(x - x_i) * G_{\sigma_i}(x) \text{ with } \sigma_i = \beta \bar{d}^i \quad (1)$$

Where:

- x represent a given pixel coordinates;
- N is the number of annotated heads;
- x_i is the i -th annotated head pixel coordinates;
- $\delta(\cdot)$ is the discrete version of the delta function: this function is zero on all possible points but in zero. In our case $\delta(0) = 1$ (in general we would have $\delta(0) = +\infty$ s.t. $\int_{Domain(x)} \delta(x) d(x) = 1$);
- $*$ is the convolution operator;
- G_σ is the 2-d gaussian filter;
- \bar{d}^i is the mean euclidean distance between x_i and k neighbors.

As suggested in [3], we used $\beta = 0.3$.



Figure 4: An example of ground truth density estimation.

4.2.2 CSRnet training

Our goal is to use this model on the *Mall dataset*, but we don't have annotation for those images. So we used *Shanghai-Tech Part B* dataset in order to pretrain the model. The first step was to build the architecture as described in [4], in particular the one that performed the best in their experiments (Figure 3). We froze all the *VGG16* layers, with *ImageNet* pretrained weights, and train the other layers using SGD as described in [4], using *Mean Squared Error* pixel by pixel as loss function for the reconstruction of the density map.

$$MSE = \frac{1}{m} \sum_{i=1}^m \sum_{x \in X^{(i)}} (\hat{F}^{(i)}(x) - F^{(i)}(x))^2 \quad (2)$$

Where $F^{(i)}$ is the i -th ground truth density map and $\hat{F}^{(i)}$ the estimation done by the model.

Then we fine tuned the whole neural network by train it again having unfrozen all the parameters, so we let the *VGG16* layers to adapt their filters in order to catch features that are more relevant for the crowd counting task. We

checked the performance of the network by making predictions and evaluating them using *Mean Absolut Error*.

$$\hat{y}^{(i)} = \sum_{x \in X^{(i)}} \hat{F}^{(i)}(x) \quad (3)$$

Basically the prediction is the sum over each pixel of the estimated density map.

$$MAE = \frac{1}{m} \sum_{i=1}^m |\hat{y}^{(i)} - y^{(i)}| \quad (4)$$

Since *VGG16* has 3 max pooling layers (Figure 1), we built the ground truth density maps such that its dimension would be 1/8 of the original images, in order to make the predictions match the ground truth values.

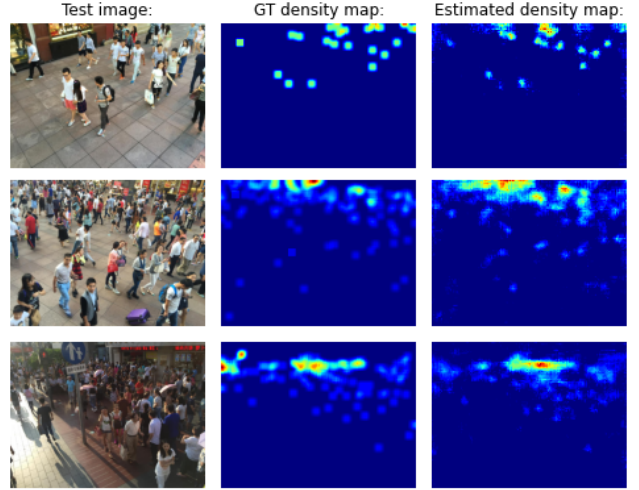


Figure 5: Test images, GT and CRSnet estimated density

As we can see from Figure 5, the estimation of the density maps provided by the trained *CSRnet* model seem reasonable.

4.2.3 More layers

Once we had the *CSRnet* trained on *Shanghai-Tech Part B* dataset we adapt this model to the *Mall dataset* in order to let it exploit the static background and the fixed perspective. We decided to use the fine tuned *CSRnet* as feature extractor and to build on top of it a regressor that estimates the number of people instead of integrating the density map. So we froze all the *CSRnet* weights and added some dense layers (we experimented different configurations that will be explained in the next section), and that we trained the regressor with *Adam*, using as loss function the *MAE* on the prediction of the number of people. We could add dense layers since images on *Mall dataset* have fixed size. Then we also tried to fine tune the whole network.

5. Experiments

In this section we describe the experiments we run in each stage, and what’s the best configuration that we found.

5.1. Regression based approach

5.2. Density based approach

5.2.1 Shanghai Tech Part B

During the different phases of the *CSRnet* model training we checked the performances on *Shanghai-Tech Part B* dataset and on the unseen *Mall dataset*, in order to check whatever the model can be used for the specific unseen dataset. We report the *MAE* (Equation (4)) computed on the test sets.

CRSnet training procedure	<i>Shanghai-Tech Part B</i> MAE	<i>Mall dataset</i> MAE
VGG16 pre trained layers frozen and SGD on the new layers	40.17	14.34
Full previous trained neural network fine tuned using SGD	28.35	12.63

Table 1: CSRnet results.

In order to evaluate the capabilities of the model on the *Mall dataset* we need to know how a dummy regressor performs. Since we are using *MAE*, the best dummy regressor with respect to this metric is the median. A dummy regressor that always predicts the median has a *MAE* of around 6. So the *CSRnet* is performing very badly, and it needs to adapt to the specific characteristic of the dataset.

5.2.2 Mall dataset

In this section we performed several experiments starting for *CSRnet*. The reconstructions seem good (Figure 5), but the results on this dataset are bad. We started experimenting different network architectures for the new layers, we tried to use techniques such as dropout, resizing the starting images and we tried both the previous learned weights as feature extractor and to fine tune the whole architecture.

From Table 2 we can see how fine tuning the whole net does not work in this case. Having 2 dense layers perform slightly better, but it takes longer to train. Dropout is useful but too much bring distortion to the estimator. We tried also SGD but we didn’t see significant differences.

6. Conclusion

The best model we found is bla, with a *MAE* of bla99. In such a fixed scenario does not help to use density maps approach, those methods are good on very dense images and performs similarly to simpler approaches on this setting, but they are harder to train, so it is not worth using them.

Architecture	Training procedure	<i>Mall dataset</i> MAE
1 dense layer	Adam on full network	5.35
1 dense layer	<i>CSRnet</i> frozen + Adam	2.48
2 dense layers	<i>CSRnet</i> frozen + Adam	2.50
224x224 images reshape + 1 dense layer	<i>CSRnet</i> frozen + Adam	2.6
224x224 images reshape + 2 dense layer	<i>CSRnet</i> frozen + Adam	2.56
1 dense layer + Dropout(0.1)	<i>CSRnet</i> frozen + Adam	2.44
2 dense layers + Dropout(0.1)	<i>CSRnet</i> frozen + Adam	2.43
1 dense layer + Dropout(0.4)	<i>CSRnet</i> frozen + Adam	2.51
2 dense layers + Dropout(0.4)	<i>CSRnet</i> frozen + Adam	2.65

Table 2: Results on *Mall dataset*.

1 dense layer: 1 unit - relu activation;

2 dense layers: 1024 units - relu, 1 unit - relu;

Dropout is placed before each dense layer.

References

- [1] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).
- [2] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [3] Yingying Zhang et al. “Single-image crowd counting via multi-column convolutional neural network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 589–597.
- [4] Yuhong Li, Xiaofan Zhang, and Deming Chen. “Csr-net: Dilated convolutional neural networks for understanding the highly congested scenes”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1091–1100.