CS464 - Progress Report - Group 23

Berk Güler, Fuad Ahmad, Munib Emre Sevilgen,

Şeyma Aybüke Ertekin, Yağmur Özkök

Project Name: Crowd-Counting Using Convolutional Neural Networks

1. Introduction

Crowd counting is a machine learning technique used to estimate the number of people in a crowd in an image [1]. Crowd counting may have many advantages. It can be used to estimate and analyze attendance rate in the events. It is beneficial to provide an estimate about how much resource or how many staffs needed in the business. It can also be used for video surveillance or managing the traffic. In today's conditions, it is also very helpful to provide good health conditions by preventing people from getting together in large groups.

The machine learning methods for crowd counting can basically be divided into four [1]. Detection-based methods involve identifying people in the images. This method is used to detect faces and even if they work well for face detection, they are not a good solution for crowd counting for dense crowds, because people are not clearly visible and therefore, detectable [1]. Counting can also be done using regression models, which involve cropping patches from the images and then extracting low-level features like edge details and mapping the features and the count to apply the model [1]. Density estimation methods involve creating a density map and then learning the mapping between extracted features and density maps [1]. The fourth approach which is the technique we want to use is based on deep learning and convolutional neural networks. This technique offers higher accuracy rates compared to previous methods. Using CNNs, end-to-end regression is applied which takes the entire image as an input and predicts the count instead of making observations over the patches of the image [1].

2. Background Research

2.1. Convolutional Neural Network Techniques

There are many CNN based methods and models in the literature. These models can be divided into four [1]. Basic CNNs involve the initial deep learning approaches containing basic convolutional layers, kernels, and pooling layers. Scale-aware models involve multi-column or multi-resolution architectures. Context-aware models involve local and global contextual information incorporated into CNN. Multi-task frameworks involve tasks apart from crowd counting such as crowd-velocity estimation.

2.2. Crowd Counting Models

We have chosen three models to work with, which are CSRNet, MCNN, and Bayesian Model. From these models, CSRNet is a context-aware CNN model while MCNN is a scale-aware CNN model. We have also chosen a model, Bayesian, which is not a CNN

model in order to compare the performance of the CNN models with a model from a different approach.

2.2.1. CSRNet

The critical difference of this model from other CNN based crowd counting methods is to use dilated convolutional layers to increase the accuracy of the model. Dilated convolution can be defined as the equation 1.

$$y(m,n) = \sum_{i=1}^{M}\sum_{j=1}^{N} x(m + r \times i, n + r \times j)w(i,j) \quad (1)\ [2]$$

It is important to notice that the equality becomes the same as normal convolution when we set r = 1. Dilated convolution is needed, because the traditional pooling layer reduces spatial resolution while deconvolutional layers may cause an increase in the complexity [2]. Dilated convolution leads to alternating pooling and convolutional layers. We can try to train the model with different r values and measure how the accuracy changes when we increase r.

2.2.2. Multi-Column Convolutional Neural Network - MCNN

In order to create an adaptive model to people with different head sizes, filters of different sizes are used to model the density maps, because it is not possible to identify the scale of the heads with filters of the same size [3]. Figure 1 shows the general structure of the MCNN model.

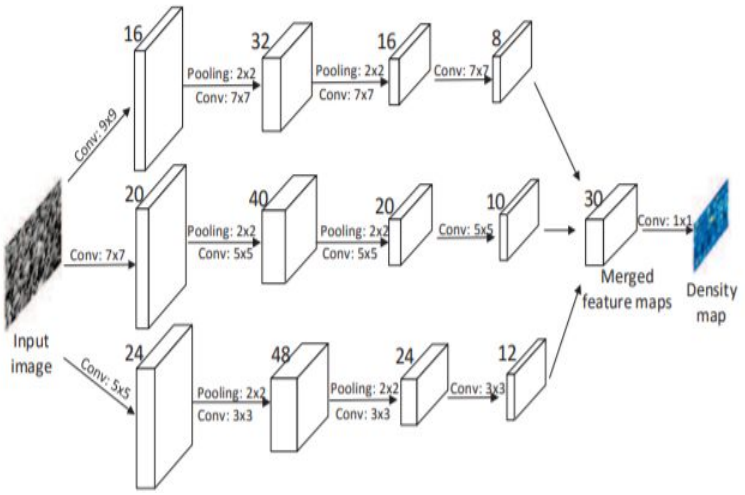

Figure 1: The Structure of MCNN [3]

2.2.3 Bayesian Model Adaptation for Crowd Counting

The aim of this method is to interpret Gaussian Processes with the advantages of Bayesian inference supported by the source model and adaptation dataset to collect observations [4]. Adaptation dataset is the limited part of the training set which is named as adaptation dataset. So, with the combination of Gaussian Processes and adaptation dataset, predictive distribution can be obtained. The predictive distribution shown below.

$$p(y_*^+|\mathbf{x}_*^+, X^+, \mathbf{y}^+, X, \mathbf{y}) = \int p(y_*^+|\mathbf{x}_*^+, \mathbf{w}) p(\mathbf{w}|X^+, \mathbf{y}^+, X, \mathbf{y}) d\mathbf{w}. \quad (2) \quad [4]$$

2.3. Evaluation of the Quality of Density Map

In order to evaluate the quality of the density maps created, Gaussian filters are used to create the ground truth density maps of the pictures, then, they are used to validate the model. The loss function shown by equation 3 is used to evaluate the quality of the density map, where F(Xi ; Θ) stands for the estimated density map.

$$L(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \|F(X_i; \Theta) - F_i\|_2^2, \quad (3) \quad [3]$$

2.4. Evaluation of the Accuracy of the Models

Equations 3 and 4 are used two evaluate the accuracy of the models, where , z_i is the actual number of people in the ith image, and $\hat{z}_i$ is the estimated number of people in the ith image.

$$MAE = \frac{1}{N} \sum_{1}^{N} |z_i - \hat{z}_i|, \quad MSE = \sqrt{\frac{1}{N} \sum_{1}^{N} (z_i - \hat{z}_i)^2} \quad (4, 5) \quad [3]$$

2.5. Fine Tuning

The models mentioned previously are designed so that they can be applied to different scenes. However, when we think about real life applications of crowd counting, it usually requires estimating the number of people in an area using a camera recording from the same place and the same angle. Therefore, we can use fine tuning in order to create scene-specific crowd counting applications with higher accuracies.

2.6. Data Augmentation

By taking the reflections of the images or cropping the images into patches from random places from the image, data augmentation can be achieved for crowd counting.

3. What is done

We have tried to understand and execute github codes for CSRNet and MCNN [5 , 6].

4. Tasks to do

For the first model, CSRNet, we can change the dilation factor, and observe the performance of the model for different dilation parameters.

For all models, we can try to fine-tune the parameters and observe the performance of the models when we use the images from the same scene. For this purpose, we need to select the images from the same scene from our large dataset.

Lastly, we can use data augmentation techniques in order to increase data size and observe the effect of the data size on the performance of the models.

5. Division of Work among Teammates

Background Research → Aybüke, Yağmur
Github Code Trial → Emre, Berk, Fuad

References

[1] Kurama, V. (2019, October 15). Dense and Sparse Crowd Counting Methods and Techniques: A Review. Retrieved November 16, 2020, from https://nanonets.com/blog/crowd-counting-review/

[2] Li, Y., Zhang, X., & Chen, D. (2018). CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. doi:10.1109/cvpr.2018.00120

[3] Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2016.70

[4] B. Liu and N. Vasconcelos, "Bayesian Model Adaptation for Crowd Counts," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[5] ZhengPeng7. (n.d.). ZhengPeng7/CSRNet-Keras. Retrieved November 16, 2020, from https://github.com/ZhengPeng7/CSRNet-Keras

[6] CommissarMa. (n.d.). CommissarMa/MCNN-pytorch. Retrieved November 16, 2020, from https://github.com/CommissarMa/MCNN-pytorch