Bayesian Crowd counting

Vaibhav Methi

October 22, 2024

1 Implementation and Training details

1.1 Network Details

The authors use a standard image classification network as their backbone, with the last pooling and the subsequent fully connected layers removed. They test two networks which are VGG-19 and AlexNet. The output of the backbone is upsampled to 1/8 of the input image size by bilinear interpolation, and then fed to a regression header, which consists of two 3×3 convolutional layers with 256 and 128 channels respectively, and a 1×1 convolutional layer, to get the density map. The regression header is initialized by the MSRA initializer and the backbone is pre-trained on ImageNet. The Adam optimizer with an initial learning rate 10^{-5} is used to update the parameters.

1.2 Training Details

Training data was augmented using random crop and horizontal flipping. Image resolutions in UCF-QNRF vary widely from 0.08 to 66 megapixels. However, a regular CNN can not deal with images with all kinds of scales due to its limited receptive field. Therefore, the shorter side of each image was limited within 2048 pixels in UCF-QNRF. At last 120 images were randomly sampled for validation set. Our model ran for total 999 epochs

2 Dataset Description

The main dataset used in this project was the **UCF-QNRF**, it is the the largest crowd counting dataset including 1535 images (both in .jpg and .mat fromat) crawled from Flickr with 1.25 million point annotations.I t has a wide range of counts, high image resolutions(which makes it a bit heavy to train on CPU), light conditions and viewpoints, all these make it a challenging dataset. The training set has 1,201 images which is further split for validation set and the remaining 334 images are used for testing.
The alternate datasets used by authors but not in this project are **ShanghaiTech** and **UCF_CC_50**

3 Results

The model results were evaluated with two eval metrics **MAE** and **MSE**

$$MAE = \frac{1}{K}\sum_{k=1}^{K}|N_k - C_k|$$

$$MSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(N_k - C_k)^2}$$

Datasets Methods	UCF-QNRF MAE	 MSE
CROWD-CNN [53]	-	-
MCNN [57]	277	426
CMTL [40]	252	514
SWITCH-CNN [3]	228	445
CP-CNN [41]	-	-
ACSCP [37]	-	-
D-CONVNET [38]	-	-
IG-CNN [2]	-	-
IC-CNN [32]	-	-
SANET [4]	-	-
CL-CNN [16]	132	191
BASELINE	106.8	183.7
Our BAYESIAN	92.9	163.0
Our BAYESIAN+	**88.7**	**154.8**

(a) Results mentioned in Official Implementation

```
('img_0318',) -89.4149169921875 1616 1705.4149169921875
('img_0319',) -71.6815185546875 922 993.6815185546875
('img_0320',) -12.32763671875 2414 2426.32763671875
('img_0321',) -181.76416015625 2471 2652.76416015625
('img_0322',) -229.5498046875 1613 1842.5498046875
('img_0323',) 48.122802734375 2033 1984.877197265625
('img_0324',) 70.039794921875 650 579.960205078125
('img_0325',) -457.73193359375 1659 2116.73193359375
('img_0326',) -2.1202545166015625 165 167.12025451660156
('img_0327',) -628.9195556640625 1408 2036.9195556640625
('img_0328',) 12.41717529296875 426 413.58282470703125
('img_0329',) 423.8050537109375 962 538.1949462890625
('img_0330',) -48.3897705078125 1668 1716.3897705078125
('img_0331',) -162.296875 2607 2769.296875
('img_0332',) 19.88543701171875 770 750.1145629882812
('img_0333',) -129.67413330078125 601 730.6741333007812
('img_0334',) -65.2188720703125 1508 1573.2188720703125
Final Test: mae 95.10819326617761, mse 169.57934000290606
```

(b) Resluts in my Implementation

As can be clearly seen , the results for both MAE and MSE are very close to the offical implementation

$$\text{Error in MAE} = \left(\frac{95.108 - 88.7}{88.7}\right) \times 100 = 7.22\%$$

$$\text{Error in MSE} = \left(\frac{169.57 - 154.8}{154.8}\right) \times 100 = 9.54\%$$

My training epochs results and final test results could be seen in output.log and test.log file respectively, in the github repository

Github repository link : Link