Implementation Details of the New Crowd Counting Model with Multi-Scale Attention and Deep NCL

Goural

November 2024

1 Implementation Details

1.1 Dataset Details

This model was evaluated on three widely used datasets for crowd counting: ShanghaiTech, UCF-CC-50, and WorldExpo'10.

- **ShanghaiTech:** Contains two parts (A and B), with each part having separate training and testing sets. Each dataset includes:
  - *images*: JPG image files
  - *ground-truth*: MATLAB files containing head coordinates
  - *ground-truth-h5*: Density maps
- **UCF-CC-50:** Contains 50 images and corresponding density maps without a pre-defined train-test split. 5-fold cross-validation is used to evaluate model performance.
- **WorldExpo'10:** Requires a university license and is not freely available.

1.2 Pre-processing

All the images are converted in 224*224 size using the data transform function to be able to be fed to the VGG-16 network. The customcollate function is then used to match the sizes of density maps and images.Data augmentation is applied, including random cropping and color jittering. The model uses images to generate density maps and trains using the ground truth density maps

2 Model Architecture

The architecture of our model is based on the original D-ConvNet-1 design with modifications to enhance feature extraction and attention mechanisms:

- **Feature Extractor:** VGG-16 layers up to conv4_3, followed by dilated convolutions for better receptive fields.
- **Attention Mechanism:** Multi-scale attention with adaptive weighting is applied at multiple scales to enhance focus on relevant features across density levels.
- **Regressors:** Group convolutional layers, each with a 1×1 convolution to output density maps.
- **Dropout:** A dropout rate of 0.3 is used to reduce overfitting.
- **Weight Initialization:** Xavier initialization is applied to the regressor layers.

3 Training Procedure

The model is trained with the following hyperparameters and settings:

- **Optimizer:** Stochastic Gradient Descent (SGD) with a momentum of 0.9.
- **Learning Rate:** 1e-5 for feature extraction layers and 1e-3 for regressor layers.
- **Loss Function:** A combination of Mean Squared Error (MSE) and a Negative Correlation Loss (NCL) penalty is used to minimize redundancy across regressors.
- **Scheduler:** A StepLR scheduler with a step size of 5 epochs and a decay factor of 0.5 is applied.
- **Epochs:** The model is trained for 20 epochs, with early stopping applied if no improvement is observed after 5 epochs.

4 Evaluation Metrics

Final output is taken as average of all the regressors, which is then compared with given density map to calculate MAE and RMSE values.
The model was trained using Kaggle GPU-P100.

5 Results and Analysis

The implementation given along with the paper was done using 'caffe framework' which is very old and is now not looked at by developers. I tried to set up the caffe to be able to run their code for days but couldn't succeed. I then tried to implement the model myself using the details given in the paper. The results of the paper varied from that of the previous model because some details like how is random cropping of the images done, upsampling details, what is the pool size decorrelated regressors, lambda parameter used in the Negative correlation loss function, etc. are not mentioned in the paper. There are a lot of such minute details required for accurate model working.
Improvement using Multi-scale Attention mechanism has been done on the previous model made.
The following are the best results I could achieve after tuning the aforementioned parameters:

- UCF-CC-50 dataset

  - Claimed Results in Paper MAE=288.4, RMSE=404.7

  - Previous Model Results MAE= 365.77, RMSE=399.58

  - New Model Results MAE= 337.62, RMSE=372.14

- ShanghaiTech-PartA

  - Claimed Results in Paper MAE=73.5, RMSE=112.3

  - Previous Model Results MAE= 181.62, RMSE=200.56

  -New Model Results MAE= 161.09, RMSE=183.08

- ShanghaiTech-PartB

  - Claimed Results in Paper MAE=18.7, RMSE=26.0

  -Previous Model Results MAE= 26.01, RMSE=28.32

  - New Model Results MAE= 20.24, RMSE= 22.73

The performance of the new model on UCF-CC-50 and ShanghaiTech (Parts A and B) is close to the original results reported in the paper, though with some differences due to minor implementation variations. Challenges included:

- Lack of specific details in the original paper, such as cropping methods, upsampling strategies, and exact lambda parameters in NCL.
- Limitations of model reimplementation in PyTorch as opposed to the original Caffe framework.

Despite these challenges, our model showed improved generalization on UCF-CC-50 with 5-fold cross-validation, providing comparable performance to the state-of-the-art on ShanghaiTech datasets.

UCF-CC-50 doesn't have separate testing and training datasets unlike the ShanghaiTech dataset, 5-fold cross-validation has been used to evaluate model performance in the case of the UCF-CC-50 dataset. The rest of the parts of both codes are exactly the same. The two separate files for the PartA and PartB datasets in the shanghaiTech dataset only differ in the trainloader and testloader lines of the code.