# Implementation Details

Goural

October 2024

## 1 Implementation Details:

The model has been made using the architecture mentioned in the paper which is as follows:

- Dataset Details —The paper uses three datasets for benchmarking namely ShanghaiTech, UCF-CC-50, and World-Expo'10. The first two are available at Kaggle in the following format:
  - images: the jpg image file
  - ground-truth: Matlab file contain annotated head (coordinate x, y)
  - ground-truth-h5: people density map

  The ShanghaiTech dataset has two parts A and B, both contain Training and Testing data separately.

  UCF-CC-50 contains 50 images and density maps and doesn't have training and testing datasets separately.

  WorldExpo'10 dataset is not available freely on the internet and requires a license from a university.

- 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. The model uses images to generate density maps and trains using the ground truth density maps.

- D-ConvNet-1 Model:
  - Feature Extractor: VGG16 layers up to `conv4_3` followed by dilated convolutions.
  - Regressors: Group convolutional layers, each with a final `1x1` convolution to output density maps.
  - Dropout: Added to prevent overfitting, with a dropout rate of 0.3.
  - Xavier initialisation is used for regressor layer.
- Training Procedure:
  - Optimizer: Stochastic Gradient Descent (SGD) with momentum of 0.9.
  - Learning Rate: 1e-5 for feature extraction layers and 1e-3 for regressor layers.
  - Loss Function: Combination of Euclidean Loss (MSE) and a correlation amongst the regressors as a penalty to minimize redundancy across regressors.
  - Scheduler: StepLR scheduler with a step size of 5 epochs and a decay factor of 0.5.
  - Epochs: 20 epochs with early stopping applied after 5 epochs without improvement.
- Evaluation:

  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.

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 vary from that of the 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.

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

  - Model Results MAE= 365.77, RMSE=399.58

- ShanghaiTech-PartA

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

  - Model Results MAE= 181.62, RMSE=200.56

- ShanghaiTech-PartB

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

  - Model Results MAE= 24.97, RMSE=32.25

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.