Innovative Exploration and Method Integration Verification of Class Agnostic Counting Model

Aobotao Dai

Department of Computer Science
Fudan University
20307130234@fudan.edu.cn

Junwen Duan

Department of Computer Science
Fudan University
20307130235@fudan.edu.cn

Boxiu Zhou

Department of Nuclear Science and Technology
Fudan University
19300200036@fudan.edu.cn

Abstract

In response to the problem of class agnostic counting in images, this article aims to review existing counting methods and attempt to propose an algorithm to achieve object counting. The algorithm is based on density map, using depth convolution neural network to extract features from the original image, and uses transformer to further capture the global information and supervised learning technology to train the neural network to generate more accurate density map. Next, the density map can be used to estimate the number of objects in a specified category. The main contributions of this algorithm are twofold: firstly, it can effectively count objects; Secondly, it has high generalization performance and can be applied to various datasets and scenarios. The experimental results show that the improved algorithm is slightly better than the baseline model. After further exploration, it is expected to be used in practical scenarios

1 Introduction

Object counting in images is a fundamental research problem in computer vision, with numerous real-world applications. Given an image, the goal is to count the number of instances of a particular object or objects of interest. For instance, counting the number of cars on a highway, the number of shoppers in a store, the number of trees in a forest, and the number of cells in a biology experiment are all examples of object counting.

It has seen significant progress with the introduction of deep learning techniques in objects counting. Convolutional neural networks (CNNs) have shown great promise in this regard, as they can learn to

extract powerful features from images and use these features to make accurate counting predictions. However, object counting remains a challenging problem, especially when the objects of interest are densely packed or partially occluded.

In addition, the trained images also contain interference information such as noise and fog, making it challenging to count objects in these interference information. Most importantly, this project requires counting the selected types of objects in the image, which means that other types of entities cannot be counted and also become interference items.

The algorithm we proposed is a supervised learning method using the density map's deep neural network. We use a specific set of images as the training set, which contains information such as the sample boxes to be counted and the points of all objects to be counted. The purpose of neural network training is to effectively generate more accurate density maps for images and given sample boxes through feature extraction, and perform more accurate counting operations.

A density map is an image with a low density background of black and a high density point of bright color at the center of a given object. Given the counting of objects, it can actually be simplified to counting high-density bright spots. How to further extract features from the given sample boxes to help neural networks train more effectively, reduce noise and interference from other objects, and fit more accurate density maps has become a challenging aspect of this topic.

We train based on VGG16 and transformer deep neural network models. At the same time, the model's ability to absorb information is increased through operations such as image preprocessing and data enhancement in the early stage.

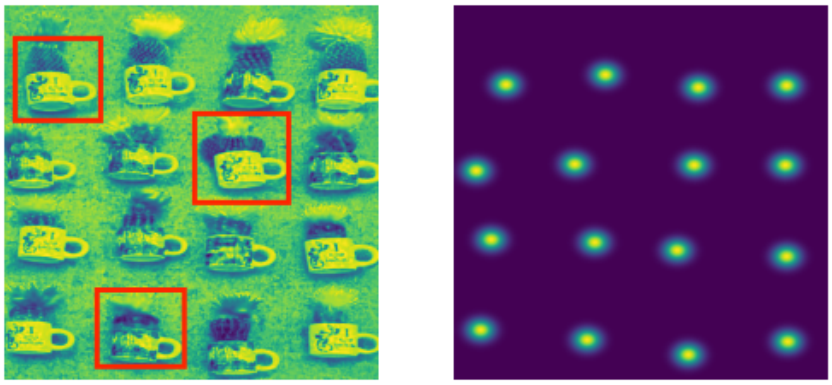

Figure 1: We use images with sample boxes and density map to count the numbers of specific objects

2 Related Work

The following are the relevant models developed by predecessors.

2.1 Traditional counting methods

The traditional image processing counting method can achieve object counting. The general method is to grayize and thresholding the target image into a binary image, and display the specific object as a color different from the background. However, when there are many types of objects and there are differences in the colors of the objects, it is difficult to ensure accuracy using simple color processing methods. In addition, there are methods such as edge detection and image segmentation, which have poor results when there are high overlapping image objects and many interfering objects.

In addition to image processing methods, deep neural networks were used for training in counting class agnostic objects based on detection counting and regression counting[6,7].We can also use sample boxes or make use of template matching[8,9].

Detection based counting methods are widely used in specific object recognition, such as crowd counting and cell counting. Because we can obtain a large number of training sets through this specific category to train object detectors. However, in the context of class agnostic objects, if this method is still followed, it is required to train a single detector for different objects, which means extracting a large number of samples in each image. This method is very difficult to implement when there are few labeled samples.

Based on regression counting, there is no need to train the feature extractor of the target object, and the detection problem is transformed into learning the mapping relationship between the image and the number of objects, simply using the number as the training label; Or the mapping relationship between image features and density maps, using the target density map as the training label. These methods are still difficult to eliminate the influence of interfering substances and lack supervision of samples.

2.2 Deep learning with fusion feature extraction function

Subsequently, a network specifically designed to solve the problem of unknown counting of image classes emerged, focusing on the content of sample boxes and the image itself, with a focus on feature extraction of the content of sample boxes.

The Generic Matching Network (GMN) proposed by Erika Lu, Weidi Xie, and Andrew Zisserman[1] consists of three modules: embedding, matching, and adaptation. Embedding is responsible for dual stream input and output, extracting features from the original and sample images respectively; Matching is responsible for converting the sample box content into vectors that match the size of the original image features for further processing; Adapting adds residual adapter modules to the embedding module, which freezes all parameters of the pre trained GMN during the adaptation phase. And train only the adapters and batch normalization layers. Finally, process the output density map.

The method proposed in this paper has the ability to extract exemplar images features, especially with good accuracy in counting objects such as cars, crowds, and cells. However, on the more complex dataset FSC147, the mean absolute error (MAE) of the density map still reaches 29.66, indicating that GMN [1] still needs improvement in the fitting of the density map. Therefore, we attempt to use GMN's technique [1] of extracting exemplar images features and appropriately improve the network and other training methods to better fit density maps on the FSC dataset.

The FamNet network[3] proposed used the Siamese network (a commonly used network in the field of facial recognition) for feature similarity measurement. Similarity has a significant impact on feature matching between local sample images and overall images. It can be said that the precision of similarity modeling can determine the final image generation effect. The later proposed BMNet (Bilinear Matching Network)[3] focuses on similarity modeling and proposes a universal framework for joint learning representation and similarity measurement. The network consists of a feature extractor, a learnable similarity measurement module, and a counter. The second one is the core part of the design, which can fuse image features and cross features and make further cross, and integrate similarity measurement into the loss function. Our work will further explore the design method of loss function.

3 Method

Our proposed model is based on the baseline model, combined with the previously proposed feature fusion method, and explores whether different preprocessing methods for input images can improve the entire network. At the same time, we also utilized bounding boxes and explored 1-shot to 3-shot (using different numbers of sample boxes for feature fusion). In addition, we also try to improve the design of loss function (involving the measurement of similarity).

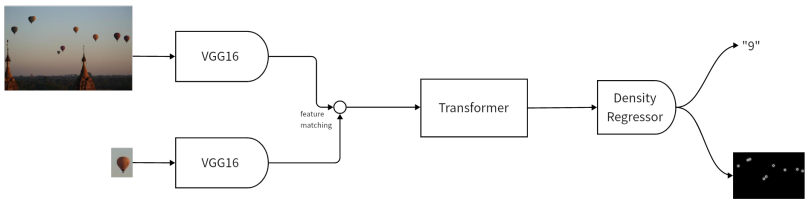

Figure 2: Our model's structure

3.1 Image preprocessing

With the emergence of deep learning, feature engineering is becoming increasingly unimportant. However, in this problem, we are faced with images, density maps, and supervised learning with density maps. If some special processing is applied to specific images, such as sharpening, noise reduction, etc., it will have a good effect on the training of the entire image. Therefore, we attempt to process the original image in different ways, with the aim of better exposing the features of the image to be counted and exploring what processing methods can help deep neural networks extract features more effectively.

Gaussian blur: Gaussian blur can eliminate high-frequency noise in the image during image feature enhancement,reduce small details in images such as edges and textures. At the same time, by reducing the number of data points in the image, Gaussian blur can be used to accelerate the processing speed of subsequent image processing steps. However, in object counting tasks, the elimination of edges may make it difficult to find the object to be counted, and there may also be negative effects.[4]

Edge operator: different from eliminating Gaussian blur of edges, edge operator is used to detect edges in images. Common edge operators include Sobel, Prewitt, Roberts, etc.[5]

These operators can detect edges by calculating the gradient of image grayscale values in different directions.

In object counting, edges can indeed play an important role. However, edge operators can easily make the points inside the object to be counted black, which may result in limitations during the matching process with the density map.

We try to conduct image processing such as edge weakening or enhancement before image feature extraction, which can further increase the exploration of density map supervised learning counting.

3.2 Feature extraction

We processed the original images and corresponding sample boxes (both of which may have been processed) using VGG16 deep neural networks (supervised by corresponding density maps, allowing the network to fit the corresponding relationship between the target and the density map).

Without the addition of sample boxes and relying solely on the supervision of the original image and density map, the model may be good at capturing the overall density, but it is difficult to train its ability to capture specific objects. The existence of sample boxes is actually a form of supervisory information, and we need to incorporate such features into our network design.

Ultimately, the features formed by the original image are vectors with a certain length and width; The feature formed by the sample box is a vector with a length and width of 1 * 1. Through the overall neural network framework and the supervised learning of this process with a given label (density map), we can conduct feedback training for two different neural networks. In the end, the two VGG neural networks extracted different features, one is the feature of the density map generated by the original image, and the other is the feature of the sample box image.

Figure 3: Original, Gaussian filtered, Laplace processed and Prewitt processed

3.3 Feature fusion

The original image and sample box will form features of different sizes. In order to fuse these features at an appropriate size, the length and width 1 * 1 vector extracted from the sample box is amplified.

The amplification method of the sample box is to directly copy and overlay equal 1 * 1 vector values to the size of the extracted features from the original image.

The fused features possess both the attributes of the original image and the sample box. This operation can integrate the attributes of the sample box (whether using 1-shot or 3-shot), allowing the neural network to choose independently

Then the fused features will be input into a transformer network for further processing and feature extraction. Transformer is a neural network architecture based on attention mechanism, where each input tag can interact with other tags and calculate their importance weights in the context of the encoder and decoder. In fact, we use VGG16 to extract and fuse the features of the original image and sample box for encoding and decoding operations.

3.4 Density regression

Use a set of convolutional neural networks to perform the final processing on the transformed network. This is a set of convolutional neural networks that, under the supervision of density labels, obtain the final predicted density map, serving as an output layer.

Because of the output of the final results, the density regression needs to be designed with a loss function.

In the density regression, we attempted to increase the signal-to-noise ratio. Mainly utilizing the similarity between the output density map and the target density map. Within the similarity point area between the two (i.e. the points covered by the target density map), more adjustments are needed to output the density map; Within the non similarity region (i.e. the black background of the target density), a smaller weight adjustment is given.

We have defined two concepts: positive sample and negative sample. Extracting the overlap between the predicted density map and the target density map is the focus of feature fitting, that is, positive samples, which are the parts of the predicted density map that are included in the density points

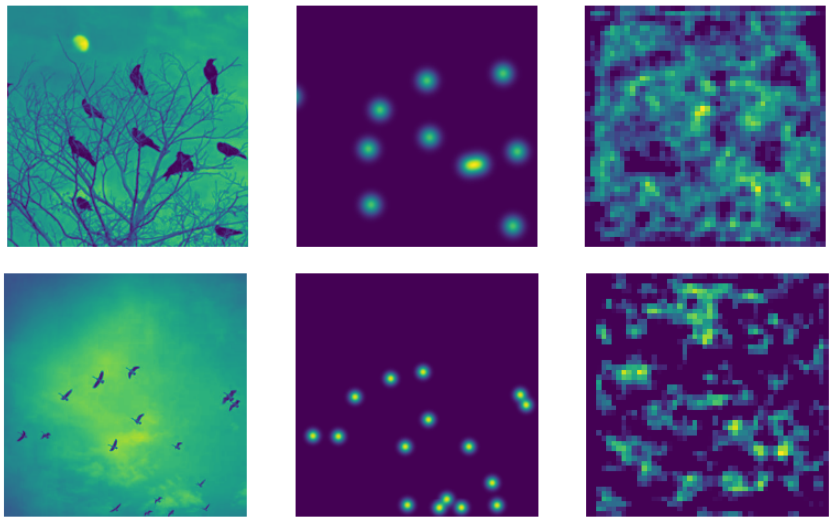

Figure 4: Original prediction of density map(without any training)

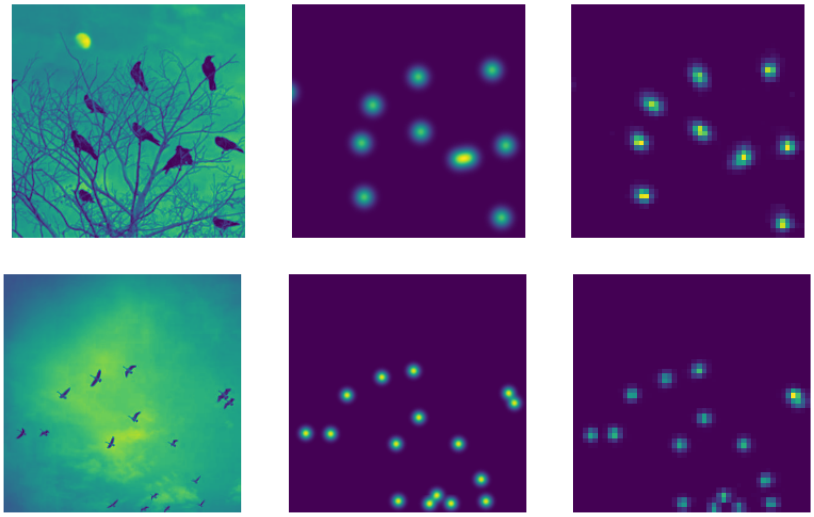

Figure 5: Predictions after several epoches of training

of the target density map (simplified as logic and operators in the following formula). We attempt to increase the weight on top of them; The non overlapping part of the two, namely the negative sample, which is the part of the predicted density map that is not included by the density points of the target density map (simplified as the logical difference operator in the following formula), has little relationship with the target, and we assign smaller weights to them(positive,negative or original samples).

$$y\_pos_i = y_i \& \hat{y_i}$$

$$y\_neg_i = y_i - \hat{y_i}$$

$$loss\_ori = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

$$loss\_pos = \frac{1}{n}\sum_{i=1}^{n}(y\_pos_i - \hat{y_i})^2$$

$$loss\_neg = \frac{1}{n}\sum_{i=1}^{n}(y\_neg_i - \hat{y_i})^2$$

$$loss = a * loss\_ori + b * loss\_pos + c * loss\_neg \quad (a, b, c \quad are \quad weights)$$

After several epoches' training,our model has shown a great improvement of prediction on density maps.

4 Experiments

4.1 Comparison of different methods

We try different network improvement methods, as well as different image processing methods and feature fusion methods, and increase the signal-to-noise ratio in the loss function. Under 100 epochs, observe the changes of some indicators on the validation set, such as loss, MSE (mean square error), MAE (mean absolute error), etc.The smaller these values, the more accurate the model fit.

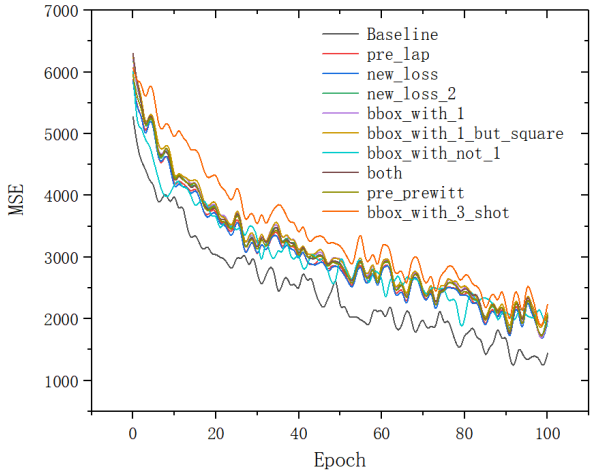

Figure 6: MSE(mean square error) of different methods

- baseline: no improved model
- pre_Lap: use Laplace operator to sharpen image edges during preprocessing
- pre_Prewitt: use Prewitt operator for image edge sharpening during preprocessing
- new_loss: use positive and original samples to change the loss function of baseline
- new_loss2: use positive and negative samples to change the loss function of baseline
- bbox_with_1: Directly transform bbox into a 1 * 1 vector through the network and fuse it with the original image features
- bbox_with_1_but_ square: Before introducing the VGG neural network, expand the bbox to a square, then resize it to a size of 96 * 96, and then change it to a size of 1 * 1 through the network
- bbox_ not_ 1: Change the VGG16 network output size of bbox to 6 * 6. However, due to size mismatch, it cannot pass val and test.
- both: Combination of 'bbox_not_1' and 'new_loss'

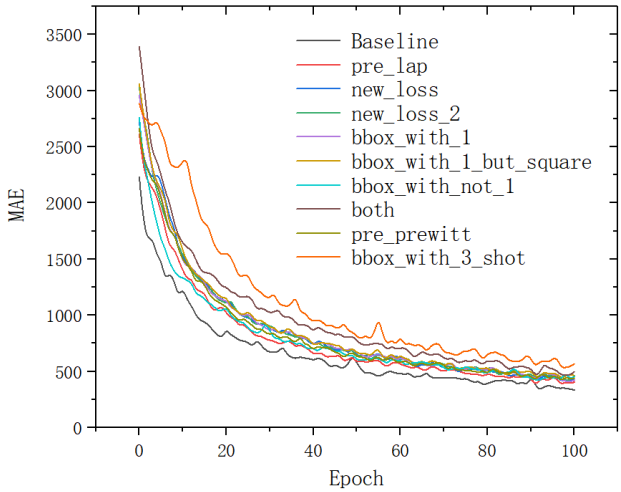

Figure 7: MAE(mean absolute error) of different methods

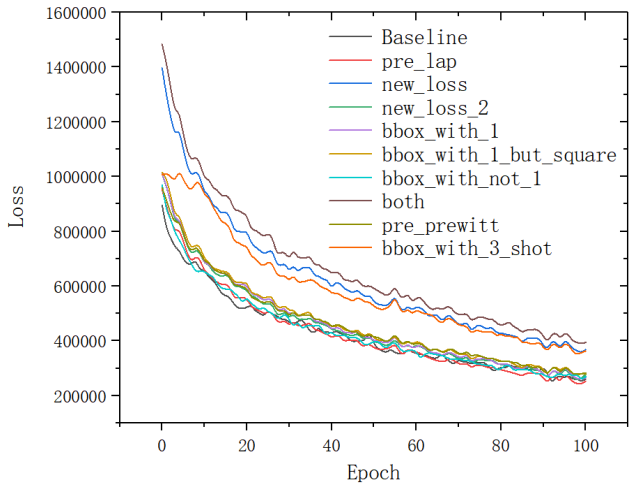

Figure 8: Losses of different methods)

Except for 'both', everything else maintains a single variable difference from baseline.

In fact, we found that in most cases of image preprocessing (especially edge extraction), our experimental data was poor and instead showed a decrease in fitting accuracy. It is speculated that the neural network lost internal information while sharpening the edges during feature extraction of images.

We separately counted the MSE and MAE of the validation set and the test set on different methods. Due to our limited computing power and time, we only conducted 100 epochs of training.

The results of designing different training methods are shown in the table below:

When improving the network structure, such as adding feature extraction for sample boxes, we found a slight improvement in the accuracy of the model compared to baseline. We have tried more methods to extract sample boxes and found that the optimal one is still the extraction of a single sample box. Perhaps the method we extracted is incorrect.

When improving the loss function for operation, our idea is to improve the signal-to-noise ratio, and we want to make more effective use of the similarity and overlap between the predicted density map

method	Val MAE	Val MSE	Test MAE	Test MSE
pre_Prewitt	22.45	135.11	65.50	160.91
bbox_ not_ 1(3-shot)	26.66	126.42	66.03	160.97
pre_Lap	24.75	133.31	66.11	161.00
new_loss2	20.57	124.14	66.04	160.87
new_loss	22.20	132.73	65.00	160.73
bbox_with_1_but_ square	22.19	124.29	66.04	160.87
both	23.32	126.15	66.04	160.87
bbox_with_1	20.93	122.11	65.00	160.73
Baseline	22.06	133.10	65.00	160.73

Table 1: Error comparison of different methods in the model

and the target density map, and adjust the loss function by enlarging the correlation difference and reducing the uncorrelated difference. But in reality, the effect is still poor, and the loss value actually increases. It is speculated that due to the weight modification of the density map, the corresponding features of our image have changed, which cannot reflect the difference in the density map well.

4.2 Comparison with other networks

Extract the best results from them and compare them with other networks.All on the FSC147 dataset.[3]

model	Val MAE	Val MSE	Test MAE	Test MSE
ourmodel	22.45	135.11	65.50	160.91
GMN	29.66	89.81	26.52	124.57
FamNet	24.32	70.94	22.56	101.94
FamNet+	23.75	69.07	22.08	99.54
BMNet	19.06	67.95	16.71	103.31
BMNet+	15.74	58.53	14.62	91.83

Table 2: Comparison of errors between different models

Limited by the computational power, training time, and the need to optimize the method, our model does not reach the accuracy of previous models. However, through exploration, we can eliminate many inefficient methods and facilitate further innovation.

5 Conclusions and Discussions

This paper mainly talks about how to do some pretreatment, network improvement, loss function change and other practices based on the network structure of baseline and the basic method of counting density map, and tries to help the neural network better fit by fusing the characteristics of the sample box.

By reducing errors such as loss, MSE, and MAE, our final count of objects will also approximate the true count. By using deep neural networks, the difficult image counting problem is cleverly transformed into a target fitting problem that deep learning excels at.

The advantage of our method lies in the use of previous image processing techniques, attempting noise reduction and edge extraction operations, and using methods such as feature engineering and data augmentation to help the network extract features; Simultaneously incorporating image trained sample boxes (bbox), attempting different methods for feature extraction of sample boxes, and incorporating them into the original image features; At the same time, we also try to improve the loss function, improve the signal-to-noise ratio of density map features, and increase the training effect.

Our method needs improvement, which is to enhance the feature mining during image training. We need to find the correct method that does not damage the original training features of the image

(otherwise the results will be counterproductive). We also need to design appropriate preprocessing methods, design excellent network structures, use better formulas to handle image similarity and improve signal-to-noise ratio methods, And there is still room for exploration to achieve a higher level of fitting accuracy compared to baseline.

References

[1] Lu, Erika, Weidi Xie, and Andrew Zisserman. "Class-agnostic counting." Computer VisionACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part III 14. Springer International Publishing, 2019.

[2] Ranjan, Viresh, et al. "Learning to count everything." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[3] Shi, Min, et al. "Represent, compare, and learn: A similarity-aware framework for class-agnostic counting." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[4] https://web.archive.org/web/20061109221710/http://www.cee.hw.ac.uk/hipr/html/gsmooth.html

[5] J. Canny, "A Computational Approach to Edge Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[6] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, Ling Shao; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12397-12405

[7] T. Stahl, S. L. Pintea and J. C. van Gemert, "Divide and Count: Generic Object Counting by Image Divisions," in IEEE Transactions on Image Processing, vol. 28, no. 2, pp. 1035-1044, Feb. 2019, doi: 10.1109/TIP.2018.2875353.

[8] Desai, Chaitanya, Deva Ramanan, and Charless C. Fowlkes. "Discriminative models for multi-class object layout." International journal of computer vision 95 (2011): 1-12.

[9] Tali Dekel, Shaul Oron, Michael Rubinstein, Shai Avidan, William T. Freeman; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2021-2029

A Appendix

Our code url:

https://github.com/aoweiyin/CHSNet.git

Our division of labor:

Aobotao Dai: Responsible for providing main ideas, network structure design, training method design, as well as main code writing and presentation

Junwen Duan: Collab environment configuration, proposing some ideas, writing some code, mainly responsible for paper layout and writing

Boxiu Zhou: (Joined the group a bit later) PPT production,reference paper collection, training data extraction and graph drawing