Crowd Counting Based on CSRNet and ResNet

Yashin Chen(yc3347) Wenqi Li(wl2620) Yun Liu(yl3763) Ziqi Zhang(zz2496)

Team O

Abstract

In this project, we based on the paper CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes [1]to provide a deep learning method that can understand highly congested scenes and do crowd counting estimation. We first replicated the paper and then replaced the VGG-16 with ResNet-18 and ResNet-34 since ResNet has outperformed VGG in recent ImageNet competition. Furthermore, we ran the different models on UCF dataset and then compared the results using different convolutional neural networks; however, we found that ResNet did not perform better than CSRNet. Next, instead of jumping from VGG to ResNet in one step, we built ResNet-like networks by adding 2 blocks and 3 blocks shortcut connections respectively to the VGG part. We then ran the ResNet-like networks on the UCF dataset again but found that when the VGG network becomes more ResNet alike, the result would become worse.

1 Introduction of CSRNet

CSRNet performs well on the crowd counting, especially for images with high density. Therefore, we began our project by first replicating this network.

1.1 Background

Crowd counting becomes more and more popular these days because it has a wide range of applications like video surveillance, traffic monitoring, public safety and other security services. It also plays an important part in doing vehicle counting and environmental survey. It has been proved that Convolutional Neural Network(CNN) works well on the crowd counting. So we decided to make use of CNN network. But instead of the normal one, we first worked on a network that combines VGG-16 with the dilated convolution, which is introduced by a paper called CSRNet: Dilated Convolutional Neural Networks, for Understanding the Highly Congested Scenes [1].

1.2 CSRNet Architecture

There are a lot of related traditional methods like detection-based approach, regression-based approach and density estimation-based approach used for crowd counting. However, hand-crafted image features often fail to provide robustness to challenges of occlusion and large-scale variations. With CNN, it shows the ability for learning nonlinear functions from crowd images to their corresponding density maps or corresponding counts.
CSRNet is built based on CNN [see Appendix a]. It uses the first 10 layers of VGG-16 as the front-end because of its strong transfer ability and its flexible architecture for easily concatenating the back-end for density map generation. The VGG-16 part is then attached to dilated convolutional layers as the back-end for extracting deeper information of saliency. Because there are 3 max-pooling layers in the front-end, the output size of this network is 1/8 of the original input size. As shown in the paper, dilated convolution has following advantages.

- Dilated convolutional neural network enlarges the receptive field without increasing the number of parameters or the amount of computation.
- The output from the dilated convolution neural network contains more detailed information.
- Using more convolutional layers with small kernels is more efficient than using fewer layers with larger kernels when targeting the same size of receptive field.

It uses bilinear interpolation with the factor of 8 to compare the output with ground truth.
Here are some training details mentioned in the paper:

- Ground Truth: head annotation with binary dot corresponding to each person.

- Density Map Generation: by blurring each head annotation using a Gaussian kernel (which is normalized to 1).
- Data Augmentation: crop 9 patches from each image at different locations with 1/4 size of the original image. Mirror the patches to double training set.
- Stochastic Gradient Descent(SGD): is applied with fixed learning rate at 1e-7 during training.

1.3 Data source

The paper introduces several data sources: UCF_CC_50 dataset, Shanghai Tech dataset, UCSD dataset, WorldExpo'10 dataset, TRANCOS dataset. And we focused mainly on the first two.

1.4 Paper replication

For the purpose of replicating the paper, we set all the environments on the Google Cloud Platform and used pytorch 0.4.0 and CUDA 9.2.
The original code is from GitHub [https://github.com/leeyeehoo/CSRNet-pytorch/tree/master]. In make_dataset.py file, we generated the ground truth density map using Gaussian filter with our own json files. Then we ran the train.py together with the model.py which constructed a CSRNet model to start our training process. During the training process, we saved the check points and best model. After training, we tested our performance using test dataset.
We ran on the folowing two datasets.

- Shanghai Tech dataset: The testing MAE is 68.16, which is close to 68.2 presented in the paper.
- UCF_CC_50 dataset: The testing MAE is 245.05, which is close to 266.1 presented in the paper.

2 Introduction of ResNet

Residual Net (ResNet), proposed in Deep Residual Learning form Image Recognition[2], is a recently popular convolutional neural network model. It is widely known by its performance since it has outperformed the VGG Nets in recent ImageNet Competition[see Appendix b].
ResNet plain baselines are mainly inspired by the philosophy of VGG Nets. To build the ResNet, one needs to base on the VGG Net's plain network and insert shortcut connections that turn the network into the counterpart ResNet version. Shortcut connections are skipping layers that add their inputs to the outputs of the stacked layers and thus form a block. When the input and output are of the same dimension, the input can be directly added to the output. On the other hand, when the dimensions increase, one needs to pad the input with extra zero entries and then add it to the output. With the design of the shortcut connections, ResNet has the following advantages.

- Deeper layers without degradation problem
  Empirical evidence shows that deeper neural networks are more difficult to train because deeper networks are exposed to degradation problem: accuracy gets saturated and then starts degrading significantly with deeper network depth.
  Compared with plain neural network architecture, residual networks are easier to be optimized, which eases the training of networks, and can gain accuracy from considerably increased depth. Thus, ResNet can have more stacked layers without suffering from degradation problem.
- SGD is still applicable
  Shortcut connections do not add extra parameter or computational complexity to the networks. Therefore, the entire network can still be trained end-to-end by SGD with backpropagation.

3 Experiments

Next we are going to show the experiments we conducted on ResNet-18 and ResNet-34.

3.1 ResNet-34

At first, we tried to substitute the VGG part in the original CSRNet by the ResNet-34. We removed the last average pooling layer and fully connected layer in ResNet, and connected the previous part by the dilated CNN directly. In this case, the last layer of the front end has the same depth with the first layer of the back end, which is the same to the situation in the CSRNet. We also changed two CNN layers with stride 2 to no stride to make the output size as 1/8 of the input size just like the CSRNet.
We tested the ResNet-34 on the UCF dataset and found that the testing MAE is around 500 [see Appendix Figure 3], which is far from the performance of the CSRNet. During the training process, we found that the validation MAE quickly jumped down to 400 level and started to oscillate around this level. We also found that the best model sometimes occurred at the first 100 epochs which implicated the existence of the over-fitting problem.
To overcome the above problems, we redefined the learning rate as a decaying step function afterwards. In addition, we omitted the best model produced by the first 100 epochs to assure the robustness of the model. To make it more similar to the CSRNet, we also tried the max pooling layer to complete the down-sampling process, instead of the using CNN layers with stride of two. However, all of the above modifications didn't improve the result significantly.
After discussion, we thought that there might be too many layers in ResNet-34 comparing to the CSRNet, which may lead to over-fitting problems. Hence, we tried Resnet-18 next.

3.2 ResNet-18

ResNet-18 shares a very similar architecture as ResNet-34 but is not as deep as the latter due to fewer blocks. Similarly, when constructing the network, we removed the first two strides in the original ResNet-18 network and only kept the stride of 2 in the last layer, thereby creating an output that is 1/8 of the original image. We then built ResNet-18 model and tested it on the UCF dataset without loading any pretrained weights. It turned out that the best model that we saved occurred at the very beginning of our training process. When tested on our testing sets, it performed worse than the last checkpoint.
Similar to ResNet-34, we also suspected that this best model happened to fit the validation set well. Therefore to avoid overfitting, we wanted to ignore the first few epochs when saving the best model. When we trained the unloaded weights ResNet-18 Network the second time, we did not start saving the best one until the program had run 70 epochs. The testing MAE we got this time is 366, which is still not as good as that of CSRNet. And the performance is not satisfactory on the Shanghai Tech dataset either [see Appendix Figure 5].
Next, we tried loading the pretrained weights of ResNet-18 up to the parts where we modified the model. We thought that perhaps the learned weights already contained information about recognizing image patterns such as edges and would thus direct the program to a better optimal point more quickly and accurately. However, the best model produced an MAE that is very close to the one we got without loading the weights. It seems that in this case, having the pretrained weights does not result in a significant improvement.

3.3 ResNet-like CSRNet

In the previous sections, we mentioned that what distinguish a ResNet network from plain networks is its shortcut connections. If we add some shortcut connections to the VGG part, then a CSRNet network would be mainly comprised with convulsion layers and shortcut connections and become a ResNet-like network. Therefore, we next tried to remove the last three convulsions in the CSRNet and replaced them with shortcut connections within 2 blocks and 3 blocks, in hopes of getting the best of VGG and ResNet. However, the results are not as good as we expected. The 2-block one achieved 376.3, whereas the 3-block one achieved 432.5 [see Appendix Figure 4]. Notice that both are worse than the original CSRNet result.

3.4 Some Observations

In the table below, we list our ResNet-18 experiment results along with the original CSRNet result. ResNet-like CSRNet has a performance between CSRNet and ResNet. It seems that, the further away we get from VGG, the worse the validation MAE of the best model becomes. In other words, the best model which ResNet-18 finds is not as good compared to CSRNet's best model. The testing MAE of the best models also confirms that CSRNet does a better job in finding the best model.
Meanwhile, we also display the results generated by the last checkpoint. In theory, the best model is supposed to be the best. Yet to our surprise, most of the time the last checkpoint generates a better MAE than the model we saved when tested on the testing dataset. We believe that this is

because we have too many layers in the ResNet networks and that the "best model" is overfitting and is found before the program has learned something.
On the other hand, the VGG16 + 2 Blocks model might have some potential when one takes a look at its last checkpoint performance. The last checkpoint has 290.6 as testing MAE, exhibiting a performance that is approaching that of the best CSRNet model. So far we've only tested this model on the UCF dataset. If given more time, we could also try it on other datasets such as the Shanghai Tech dataset.

	Best Model		Last checkpoint
Method	Val MAE	Testing MAE	Testing MAE
CSRNet	192	265.1	359.8
VGG16 + 2 Blocks	239	376.3	290.6
VGG16 + 3 Blocks	231	432.5	346.0
Resnet-18	Best after 70 epochs		
(weights unloaded)	297	366.1	356.5
Resnet-18	Best updated each time		
(weights unloaded)	325	608.4	363.4
Resnet-18 (pretrained)	277	376.6	372.1

4 Sanity Check

To check if our constructed networks are indeed learning, we've printed out the density maps generated by some of the different models and attached them in the Appendix section [see Appendix Figure 1 and Figure 2]. It seems that most of our models are functioning as we expected. But one will see that the best model of ResNet-18 without loading weights is perhaps saved too early.

5 Conclusion

We replicated the paper and replaced the VGG-16 with ResNet-18 and ResNet-34. Then, we ran the VGG and ResNet on UCF dataset and compared the results. Since ResNet has outperformed VGG in recent ImageNet competition, we thought that ResNet might have a better performance than VGG. However, we found that ResNet did not perform better than CSRNet. The best testing MAE on UCF dataset for ResNet-34 variations and ResNet-18 variations are 397.1 [see Appendix Figure 6] and 366.1 respectively . Next, instead of substituting VGG-16 with ResNet in one step, we built ResNet-like networks by adding 2 blocks and 3 blocks shortcut connections respectively to the VGG part. We then ran the ResNet-like networks on the UCF dataset again and found that VGG + 2 Blocks has testing MAE 376.3, whereas the VGG + 3 Blocks has 432.5. Both results are worse than the original CSRNet's result on testing dataset. As a result, we proposed when the VGG network becomes more ResNet alike, the result would become worse. Nonetheless, the VGG16 + 2 Blocks model might have some potential when one focuses on the last checkpoint performance. VGG16 + 2 Blocks model's last checkpoint has 290.6 as testing MAE, approaching that of the best CSRNet model. In further studies, we could make modifications and develop a better model based on our understanding on this one.

6 Future Plan

Here are the areas which we think could potentially improve our ResNet model results.

- Learning rate
  Learning rate is an important hyperparameter. Although we've already tried several kinds of choices including decaying learning rate, some other forms of learning rate adjustments can be tested.

- Optimizer
  We are using the SGD as the optimizer in our current model. More advanced optimizer like Adam optimizer are worth being tested.

- Over-fitting & dropout
  Sometimes we faced the over-fitting during the training process, which means the best model we got may not perform the best in the testing process. Dropout can be applied to overcome it.

- Input image size
  Since the picture is highly congested, it's possible that we might have lost some important information with a small input size. Higher quality of the input may improve the performance.

- VGG + 2 Blocks
  As mentioned previously, the lastcheck point of VGG + 2 Blocks model shows some potential. One could test it on some other datasets to confirm if it is indeed working well. If so, one might be able to conduct further analysis and start building a better model based on it.

- Inception v3
  Inception V3 is an improved model based on ResNet. In inception V3, each of the convolution's feature maps will be passed through the mixture of convolutions of the current layer. One could test this model and try to build new models from it then.

7 Appendix

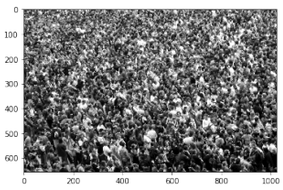
Configurations of CSRNet			
A	B	C	D
input(unfixed-resolution color image)			
front-end (fine-tuned from VGG-16)			
conv3-64-1 conv3-64-1			
max-pooling			
conv3-128-1 conv3-128-1			
max-pooling			
conv3-256-1 conv3-256-1 conv3-256-1			
max-pooling			
conv3-512-1 conv3-512-1 conv3-512-1			
back-end (four different configurations)			
conv3-512-1 conv3-512-1 conv3-512-1 conv3-256-1 conv3-128-1 conv3-64-1	conv3-512-2 conv3-512-2 conv3-512-2 conv3-256-2 conv3-128-2 conv3-64-2	conv3-512-2 conv3-512-2 conv3-512-2 conv3-256-4 conv3-128-4 conv3-64-4	conv3-512-4 conv3-512-4 conv3-512-4 conv3-256-4 conv3-128-4 conv3-64-4
conv1-1-1			

(a) CSRNet Architecture

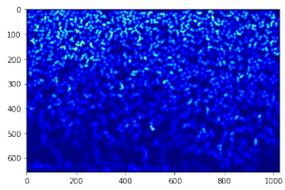
Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset.

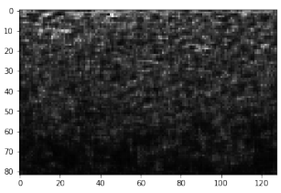
(b) Comparison of CNNs



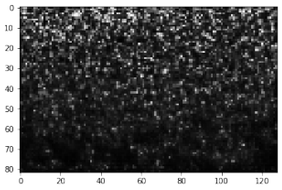
(c) Original Image



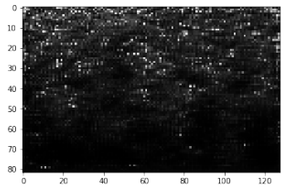
(d) Ground Truth, Count: 1358



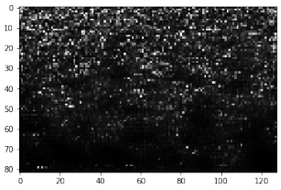
(e) CSRNet Best Model Model MAE: 265.1, Estimation: 1618.49



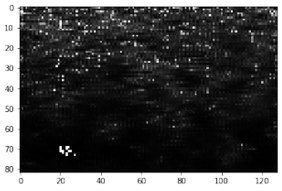
(f) CSRNet Last Checkpoint, Model MAE: 359.8, Estimation: 1704.84



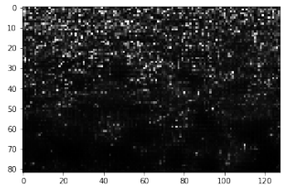
(g) VGG16 + 2 Blocks Best Model, Model MAE: 376.3 Estimation: 1135.15



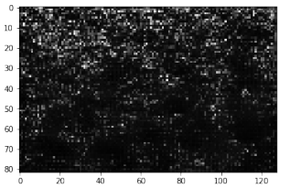
(h) VGG16 + 2 Blocks Last Model, Model MAE: 290.6 Estimation: 1435.61



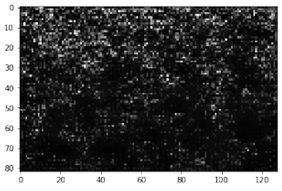
(i) VGG16 + 3 Blocks Best Model, Model MAE: 432.5 Estimation: 1073.49



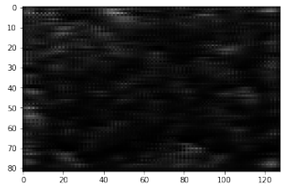
(j) VGG16 + 3 Blocks Last Model, Model MAE: 346.0 Estimation: 1251.41



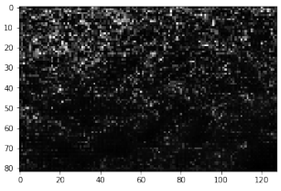
(k) ResNet 18 unloaded Best after 70, Model MAE: 366.1 Estimation: 1202.47



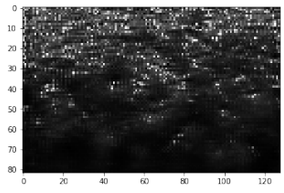
(l) ResNet 18 unloaded Last after 70, Model MAE: 356.5 Estimation: 1240.57



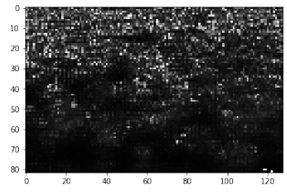
(m) ResNet 18 unloaded Best updated each time, Model MAE: 608.4 Estimation: 691.0



(n) ResNet 18 unloaded Last updated each time, Model MAE: 363.4 Estimation: 1136.04

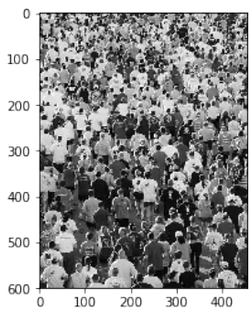


(o) ResNet 18 pretrained Best, Model MAE: 376.6 Estimation: 1828.21

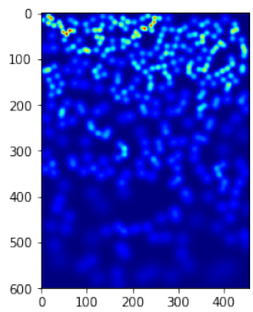


(p) ResNet 18 pretrained Last Checkpoint, Model MAE: 372.1 Estimation: 1656.68

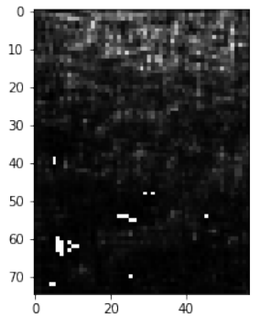
Figure 1: Sanity Check 1



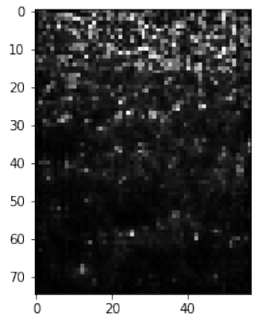
(a) Original Image



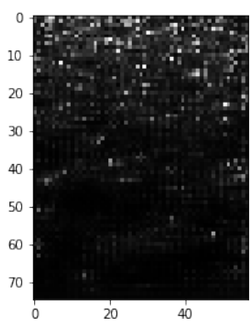
(b) Ground Truth Count: 326



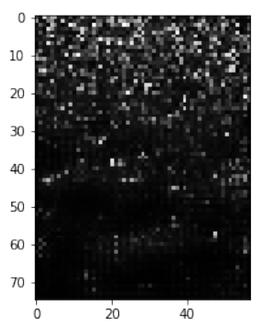
(c) CSRNet Best Model, Model MAE: 265.1, Estimation: 434.92



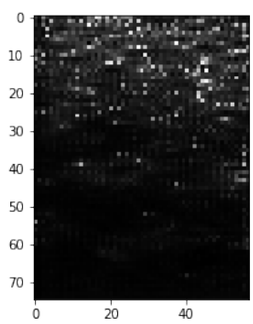
(d) CSRNet Last Checkpoint, Model MAE: 359.8 Estimation: 539.92



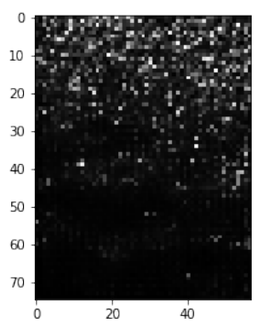
(e) VGG16 + 2 Blocks Best Model, Model MAE: 376.3 Estimation: 395.65



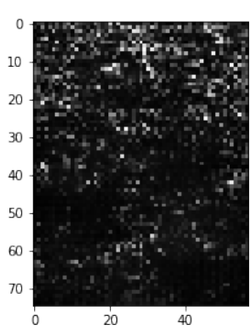
(f) VGG16 + 2 Blocks Last Model, Model MAE: 290.6 Estimation: 504.34



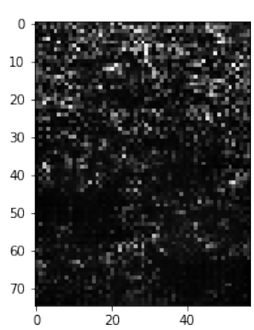
(g) VGG16 + 3 Blocks Best Model, Model MAE: 432.5 Estimation: 419.26



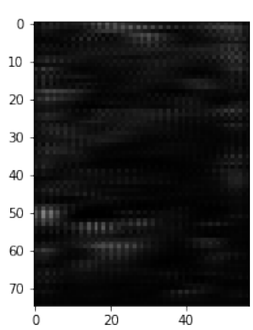
(h) VGG16 + 3 Blocks Last Model, Model MAE: 346.0 Estimation: 475.47



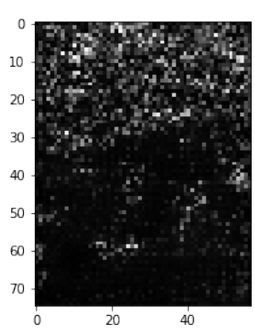
(i) ResNet 18 unloaded Best after 70, Model MAE: 366.1 Estimation: 507.76



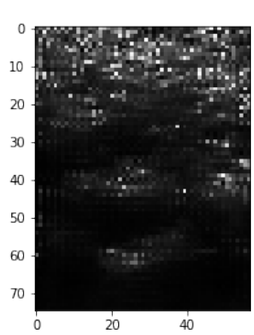
(j) ResNet 18 unloaded Last after 70, Model MAE: 356.5 Estimation: 525.89



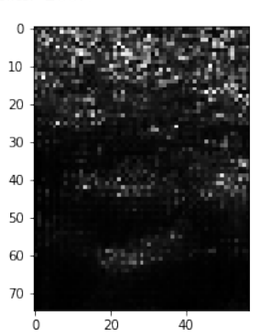
(k) ResNet 18 unloaded Best updated each time, Model MAE: 608.4 Estimation: 265.7



(l) ResNet 18 unloaded Last updated each time, Model MAE: 363.4 Estimation: 512.18



(m) ResNet 18 pretrained Best, Model MAE: 376.6 Estimation: 600.51



(n) ResNet 18 pretrained Last Checkpoint, Model MAE: 372.1 Estimation: 535.04

Figure 2: Sanity Check 2

```
0 63.6995849609375
1 697.43994140625
2 1622.110107421875
3 1838.4329833984375
4 2844.1058349609375
5 3125.0653076171875
6 3707.0031127929688
7 3728.0986938476562
8 4866.946472167969
9 5022.4456787109375
502.24456787109375
```

(a) ResNet34 Initial

```
0 95.9459228515625
1 723.9700317382812
2 1636.9653930664062
3 1780.1109924316406
4 2892.079620361328
5 3244.6258850097656
6 3634.767852783203
7 3701.867706298828
8 4942.599029541016
9 4946.6539306640625
494.66539306640624
```

(b) ResNet34 Pretrained

Figure 3: ResNet34 Testing MAE

```
0 222.394287109375
1 750.6480102539062
2 1296.8040161132812
3 1366.3680725097656
4 2032.5782775878906
5 2477.275665283203
6 2732.6412658691406
7 2922.728057861328
8 3748.8949279785156
9 3763.2702026367188
376.3270202636719
```

(a) CSRNet + 2 Blocks Best Model

```
0 78.06884765625
1 375.5052490234375
2 706.6683349609375
3 884.9125671386719
4 1283.7277526855469
5 1690.6277770996094
6 2186.302276611328
7 2298.0965881347656
8 2802.677032470703
9 2906.442413330078
290.6442413330078
```

(b) CSRNet + 2 Blocks Last Checkpoint

```
0 284.0516357421875
1 900.9177856445312
2 1525.9834594726562
3 1619.1526794433594
4 2560.2674255371094
5 3047.6879272460938
6 3280.6205444335938
7 3480.1746215820312
8 4290.409851074219
9 4325.158966064453
432.5158966064453
```

(c) CSRNet + 3 Blocks Best Model

```
0 106.1339111328125
1 596.8743286132812
2 1062.8158569335938
3 1212.2008666992188
4 1995.2085571289062
5 2447.9871826171875
6 2769.5250244140625
7 2804.3609619140625
8 3364.1253662109375
9 3459.8861694335938
345.9886169433594
```

(d) CSRNet + 3 Blocks Last Checkpoint

Figure 4: ResNet-like CSRNet Testing MAE

```
begin test
 * MAE 169.050
 * best MAE 125.226
zz2496@instance-1:~/Notebooks/CSRNet$
```

Figure 5: ResNet18 Shanhai Training Result

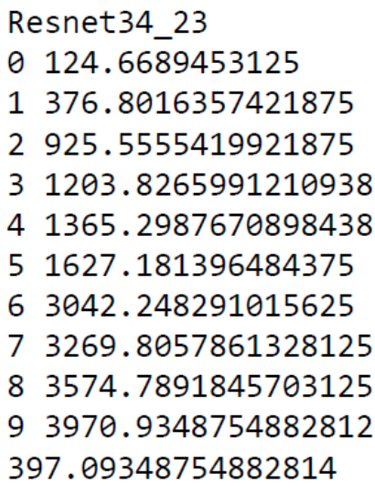


Figure 6: ResNet34 (removing 2 strides) Best Result on UCF

References

[1] CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes, *Yuhong Li, Xiaofan Zhang, Deming Chen.* University of Illinois at Urbana-Champaign, Beijing University of Posts and Telecommunications

[2] Deep Residual Learning for Image Recognition, *Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun.* Microsoft Research