1.Introduction

In this project, mmpose predicts 17 keypoints of cats. Traditional 2D pose estimation estimates 2D pose (x,y) coordinates for each joint from a RGB image. Post estimation is an import field in computer vision which is crucial for understanding object in images and videos. The goal of this experiment is to improve model AP for IoU=0.50:0.95 by using methods bellowing.

2.Methods

My best model achieve AP of 0.67.

The optimizing ways are hyper-parameter tuning, neural network architectures, data augmentation, Model pre-training.

2.1 Hyper-parameter tuning

The main method to tune hyper-parameter is to compare ablation experiment of different learning rate, batch size, warm up ratio and training epochs so as to find a balanced parameter.

2.2 Different neural network architectures.

The main method to tune network architectures is to compare ResNet, ResNest, HRNet and then try a couple of layers to some degree.

2.3.data augmentation

Proper data augmentation can enrich our dataset and help the model learn comprehensive information. There are abundant ways to tune like rotation factor, sacling factor. TopDownHalfBodyTransform is added to generate half_body picture to which has at least 8 point to ensure the train performance.

2.4. Model pre-training.

Mainly compare 3 NN model structure according to the NN structure: Resnet, Resnest, hrnet.

3.Experiment & analysis

3.1 Dataset analysis

1) Calculate the mean and std of train + validation dataset and compare with ImageNet data

	mean	std
ImageNet	[0.485, 0.456, 0.406]	[0.229, 0.224, 0.225]
Cat dataset	[0.489,0.446,0.402]	[0.235, 0.235, 0.237]

As is shown in the diagram, cat dataset in this experiment is in the same distribution with ImageNet.

2) Data set split ratio Split 1328 images into 1000 for training, 128 for validation and 200 for testing(Train : validation : testing = 8:1:2)

3.2 Ablation studies

3.2.1 Hyper-parameter tuning

1) Tune learning rate:On baseline model resnet50, set 3 ablation experiment.

Learning rate	AP
1e-5	0.153
1e-4	0.488
5e-4	0.228

Generally, a large learning rate allows the model to learn faster, at the cost of arriving on a sub-optimal final set of weights. A smaller learning rate may allow the model to learn a more optimal or even globally optimal set of weights but may take significantly longer to train. Based on the experiment result, learning rate is set as 1e-4.

2) batch size

On baseline model resnet50, set 3 ablation experiment.

Batch size	AP
8	0.172
10	0.167
16	0.155

Batch size controls the accuracy of the estimate of the error gradient when training neural networks. Batch, Stochastic, and Minibatch gradient descent are the three main flavors of the learning algorithm. There is a tension between batch size and the speed and stability of the learning process. Setting a big batch size can process the training speed and relieve overfitting to some extent.

3) training epochs

On baseline model resnet50, set 3 ablation experiment.

training epochs	AP
50	0.536
210	0.6

The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. As improving the epoch, the model can learn comprehensively as it can see the data more frequently. So the raining epochs is set as 210.

4) warm up

warmup_iters	warmup_ratio	step	AP(baseline)
10	0.001	[10, 15]	0.502
500	0.001	[170, 200]	0.506

Stepwise was used to reduce the learning rate at the 170th and 200th epochs respectively We used a warmup strategy with a very small initial learning rate.

5) Comparing image_size and heatmap_size

Image_size	heatmap_size	AP
resnest50 192*256	[48, 64]	0.527
Hrnet101 256*256	[64,64]	0.6
pose_resnet_152	[64,64]	0.637

As the input size increase, the model can extract more information while it slow down the feature extraction and is influenced by model structure as well while this parameter should be decided based on specific NN structure.

3.2.2 Different neural network architectures. Firstly, compare different network:

NN model	AP
resnet	0.153
resnest	0.488
HRnet	0.536

So HRnet is set as the main model. Next, we comparing network layers:

	AP
resnest50	0.527
resnest101	0.6

So network layer is set 101.

3.2.3 data augmentation: as rot_factor and scale_factor increase, AP increases slightly.

	rot_factor	scale_factor	AP
1	10	0.2	0.488
2	40	0.5	0.502

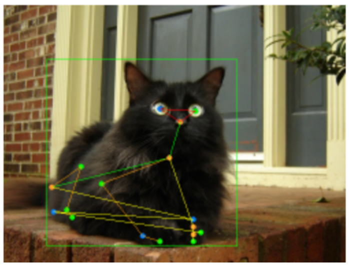
3.2.4 Model pre-training. Through comparing resnet, resnest, HRnet, the best configuration parameter is bellowing.

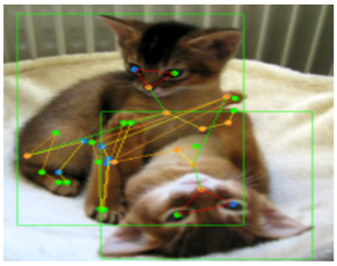
parameter	value	AP
NN model	HRnet	0.67
Pretrained model	Pose_hrnet_w32	
type	Adam	
lr	1e-4	
Warm_iters	500	
warmup_ratio	0.001	
step	[170, 200]	
total_epochs	210	
Input size	[256, 256]	
Heatmap size	[64, 64]	

4 Qualitative evaluations of challenging pictures

4.1 Good performance

The model can predict cats pose precisely even there are multiple object which can be mixed with another cat object

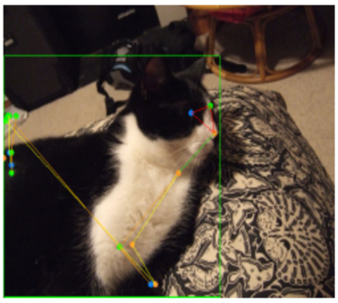


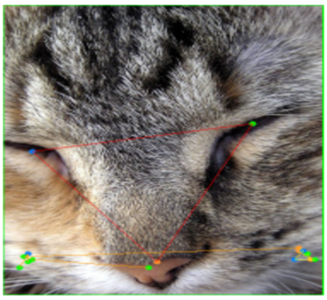


4.2 Failure case analysis

It is hard for the model to distinguish similar background and cat object.

When only part of the cat is shown, model prediction is also challenging while it give some kind of prediction.

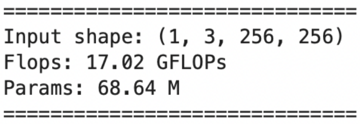




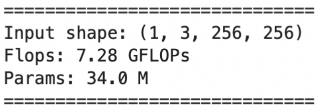
5 Model complexity & runtime analysis.

5.1 Model complexity

compute the FLOPs and params of our best model and baseline model. The best model decreases the parameters in half and increases flops doubly.



```
==============================
Input shape: (1, 3, 256, 256)
Flops: 17.02 GFLOPs
Params: 68.64 M
==============================
```



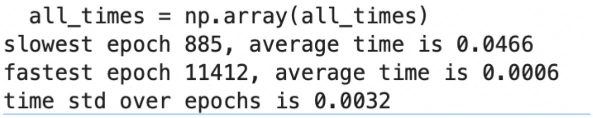
```
==============================
Input shape: (1, 3, 256, 256)
Flops: 7.28 GFLOPs
Params: 34.0 M
==============================
```

Best Model complexity

Baseline modell complexity

5.2 Runtime analysis

The runtime based on train log is shown in the bellowing picture. For different batch size and NN network setting, training time varies form each other.



```
  all_times = np.array(all_times)
slowest epoch 885, average time is 0.0466
fastest epoch 11412, average time is 0.0006
time std over epochs is 0.0032
```