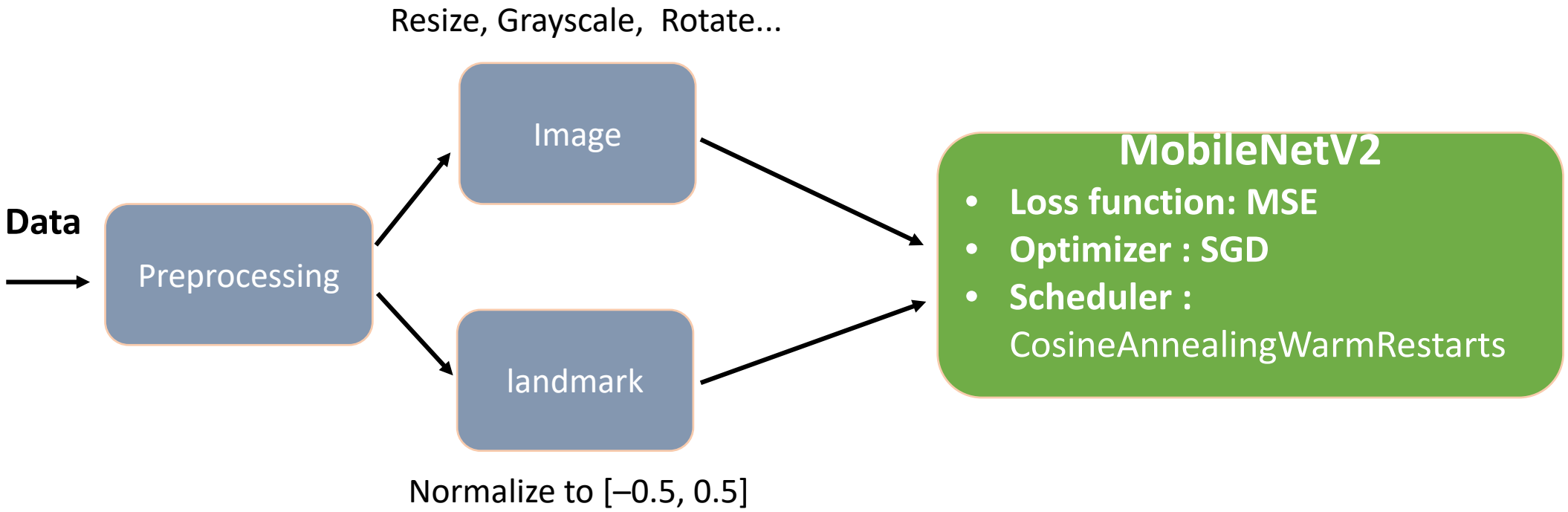
Computer Vision
Light-Weight Facial Landmark Prediction Challenge

Team AAA

蕭郁澄 r09942168, 楊欣睿r10942079, 郭承賢 r09943098

Global picture and results

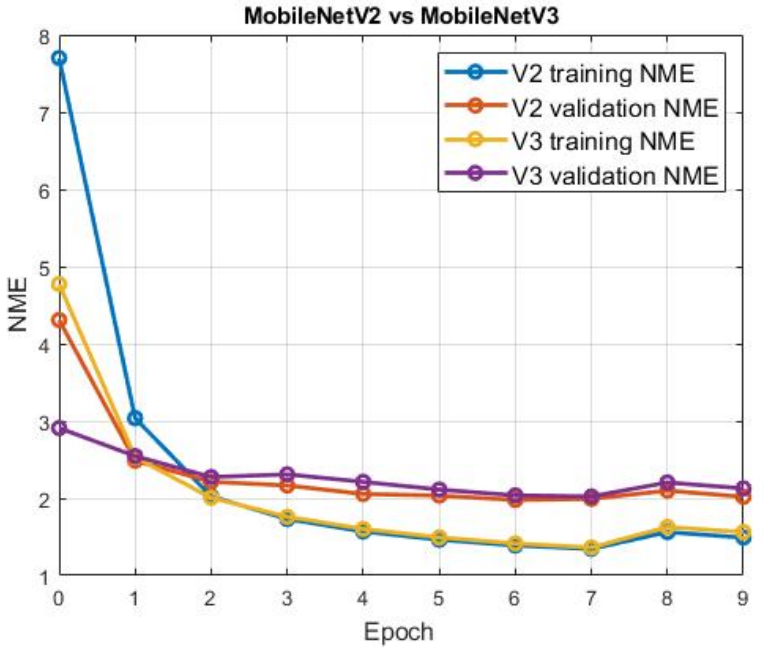


Leaderboard Rank	Validation NME	Public testing NME	Private testing NME
4	1.87095	2.03235	2.024754

Model selection

- It is well known that the MobileNet family perfoms well in the computer vision. Although [1] shows the MobileNetV3 outperforms the MobileNetV2 on the image classification taks.
- After the exhaustive experiments, we found the **MobileNetV3 is easier to overfit** the training dataset and the **MobileNetV2 is more suitable** for our task.
- Due to the model size constraint(15MB), we choose the **1.25 multiplier MobileNetV2 (14.25 MB)** as our training model.

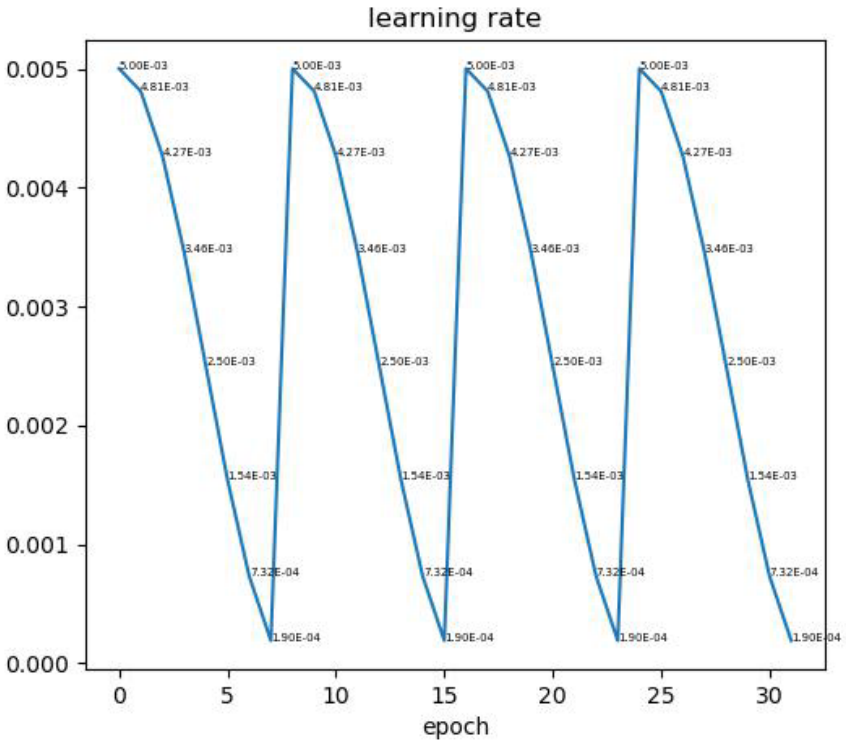
Network	Top-1	MAdds	Params	P-1	P-2	P-3
V3-Large 1.0	**75.2**	**219**	5.4M	**51**	**61**	**44**
V3-Large 0.75	73.3	155	4.0M	39	46	40
MnasNet-A1	75.2	315	3.9M	71	86	61
Proxyless[5]	74.6	320	4.0M	72	84	60
V2 1.0	72.0	300	3.4M	64	76	56
V3-Small 1.0	**67.4**	**56**	2.5M	**15.8**	**19.4**	**14.4**
V3-Small 0.75	65.4	44	2.0M	12.8	15.6	11.7
Mnas-small [43]	64.9	65.1	1.9M	20.3	24.2	17.2
V2 0.35	60.8	59.2	1.6M	16.6	19.6	13.9

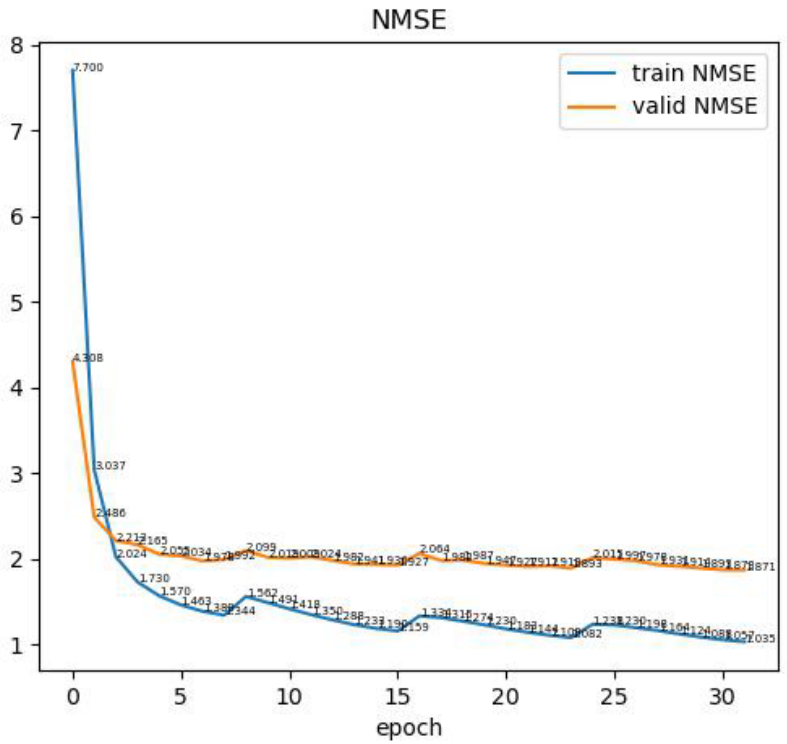


[1] Howard, Andrew, et al. "Searching for mobilenetv3." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

Scheduler

- optim.lr_scheduler.CosineAnnealingWarmRestarts(optimizer, T_0 = 8, T_mult = 1)





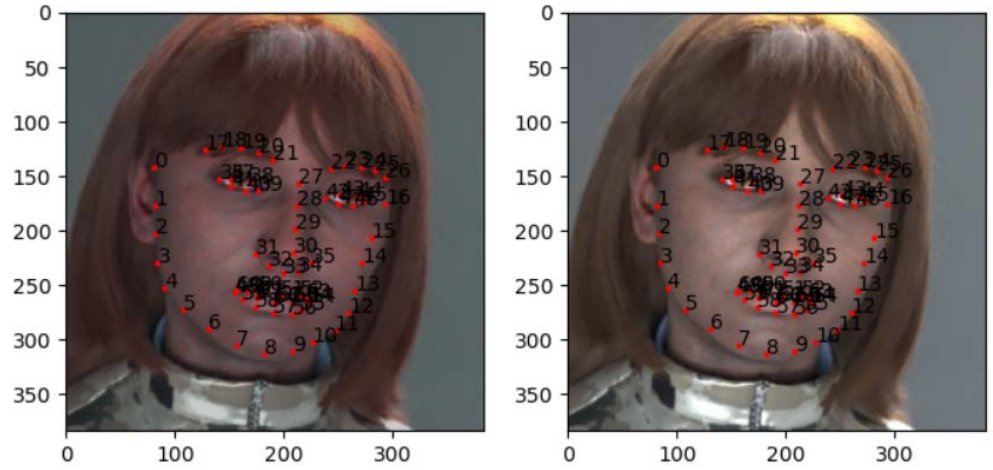
Preprocessing

- We adopt the following preprocessing for landmarks and images, respectively.
  - Landmarks:
    - Normalized to [-0.5, 0.5]

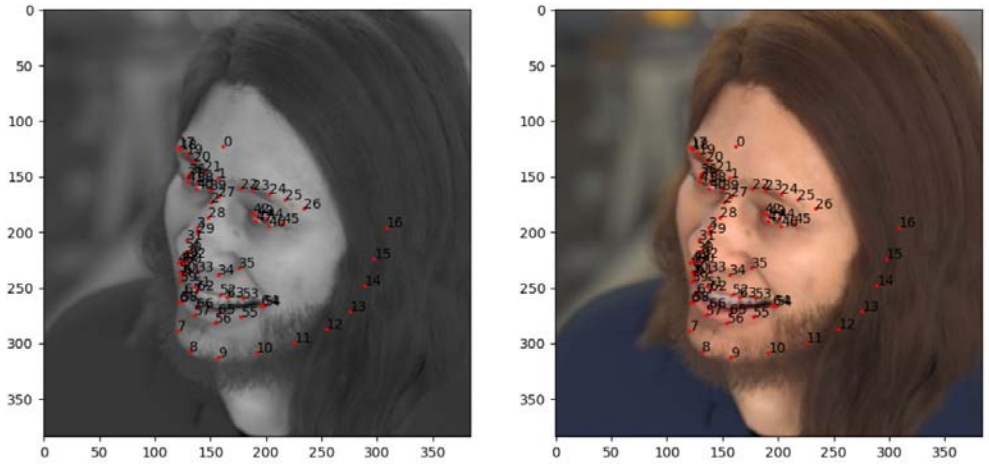
  - Image:
    - RandColorJitter, RandGrayscale, Rrandom_horizontalfip
    - RandomRotation, Random_scale, Random_erasemargin, RandomBlur, RandomErasing

Visualize the preprocessing

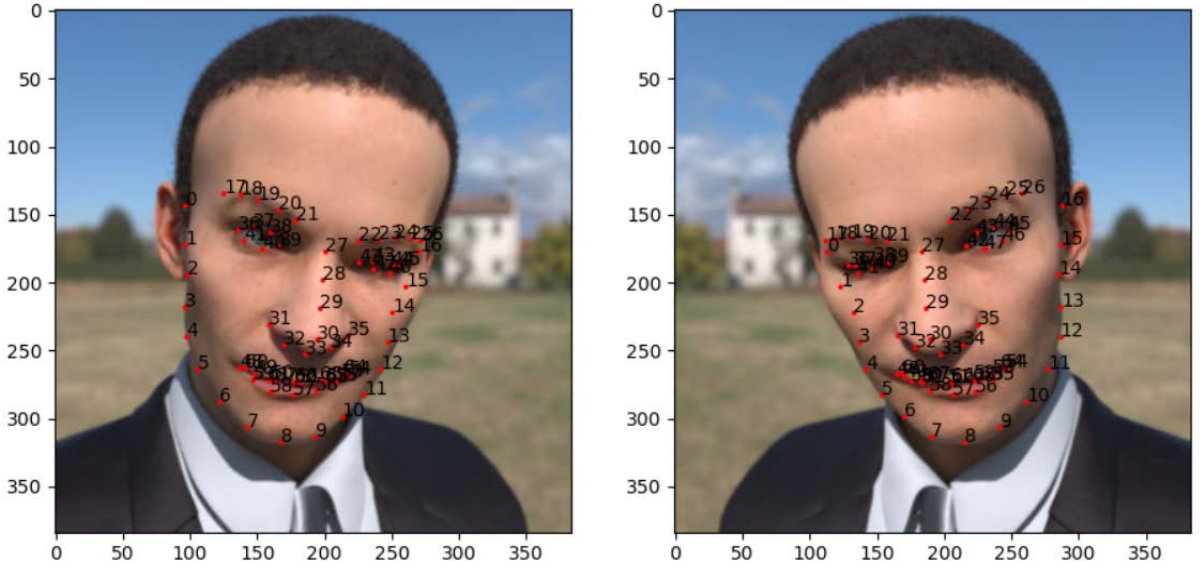
- **ColorJitter**



- **RandomGrayscale**



- **Random_horizontalfip**



Difference of dataset

- Train on synthetic data, test on real image

Tilted head

Black margin

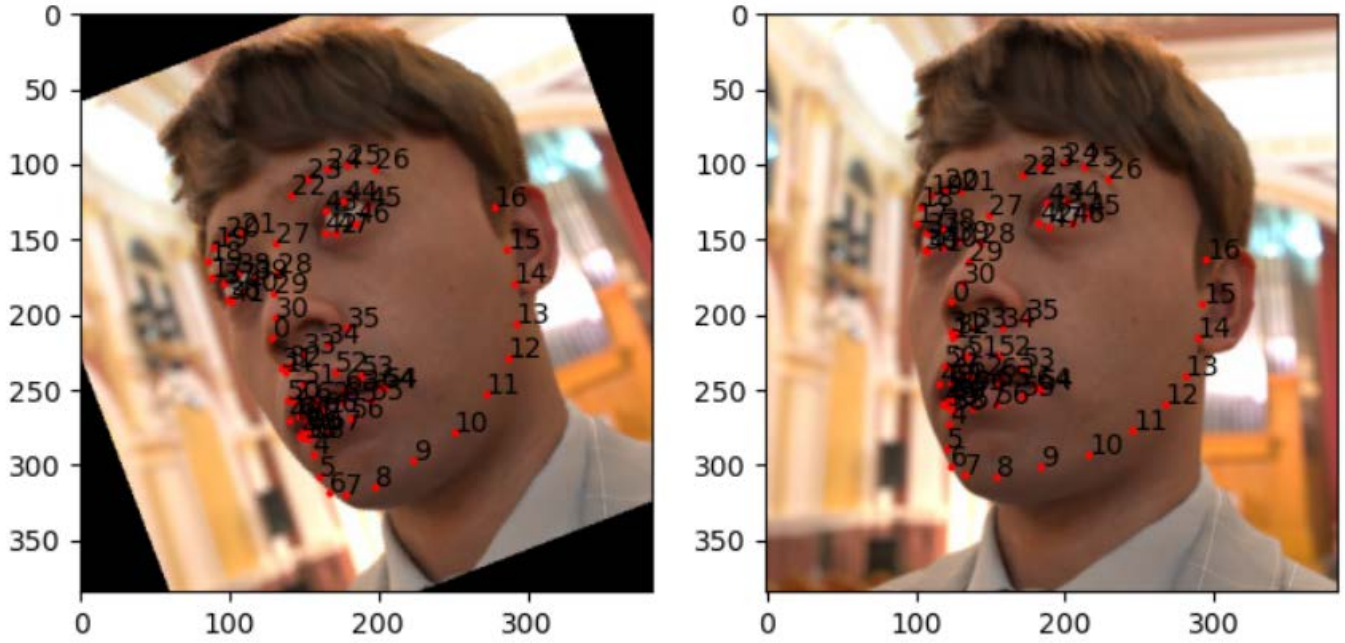
Low resolution

Blocked by something

Difference of dataset

Tilted head

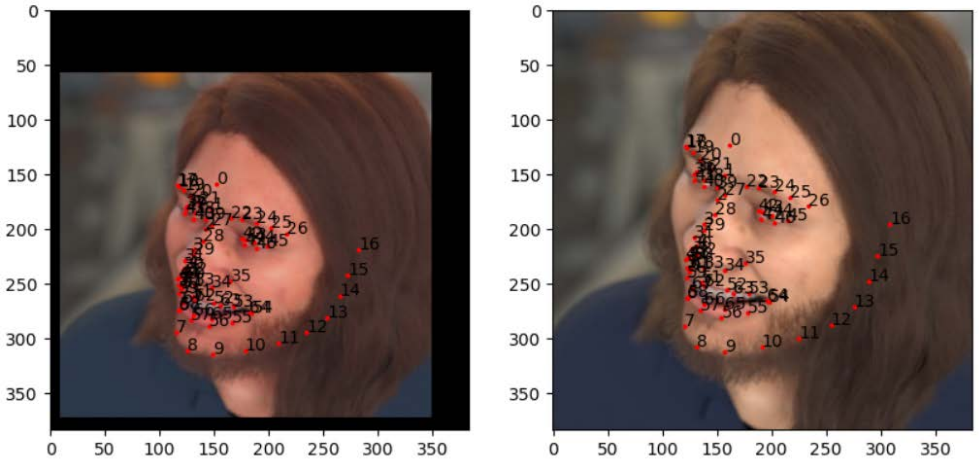
- **Random_rotate**



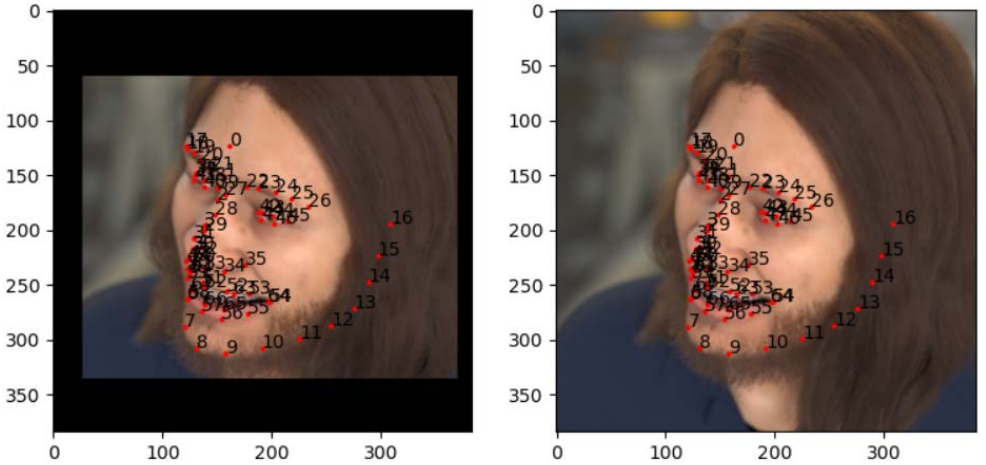
Difference of dataset

Black margin

- **Random_scale**



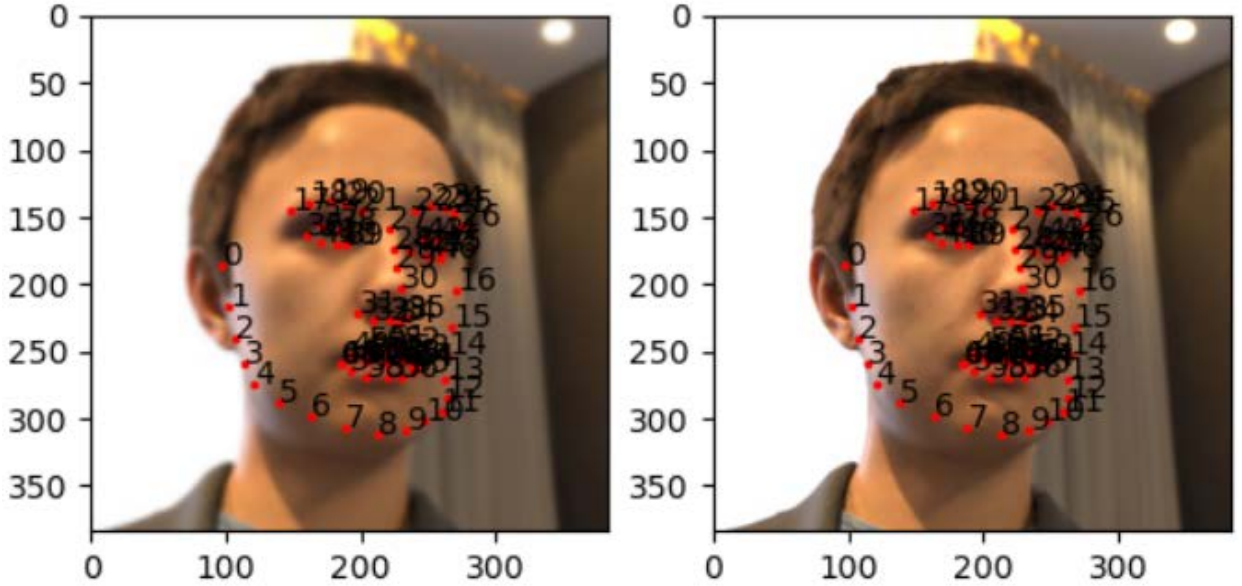
- **Random_erasemargin**



Difference of dataset

Low resolution

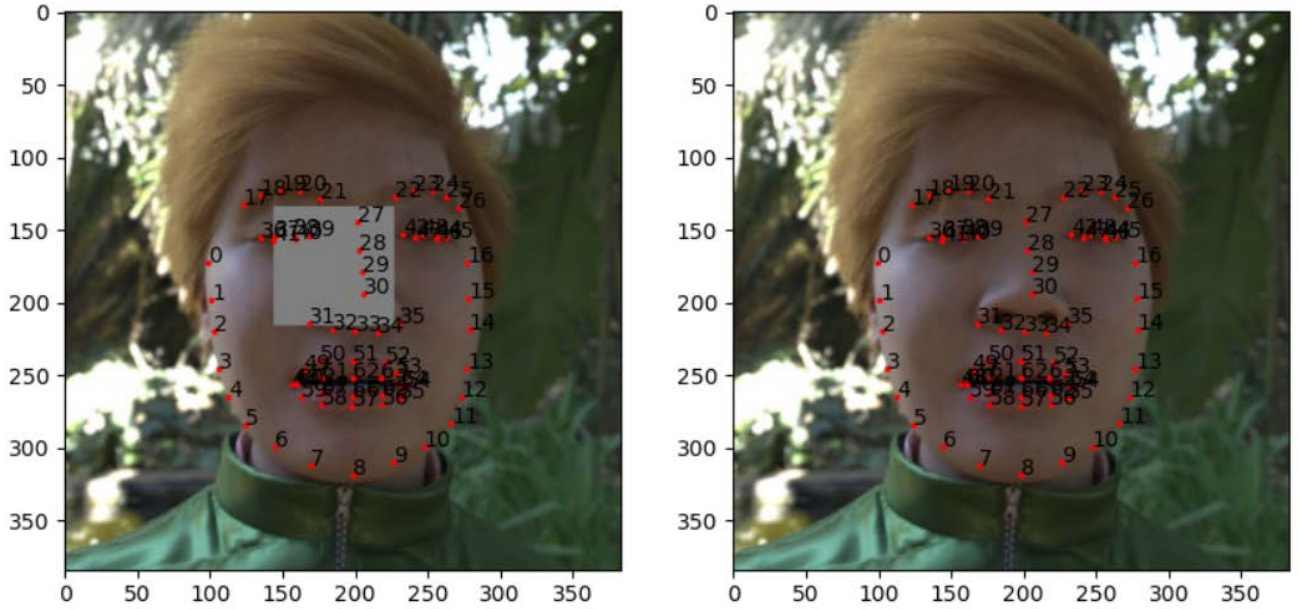
- **Random_blur**



Difference of dataset

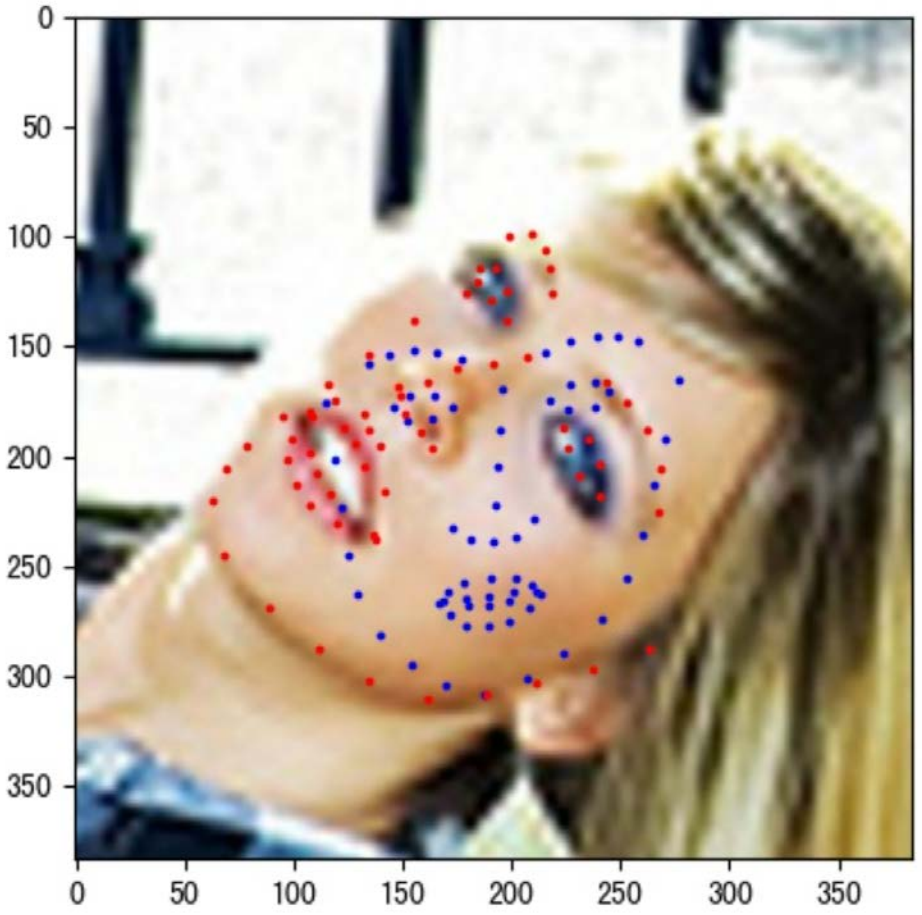
Blocked by something

- **RandomErasing**

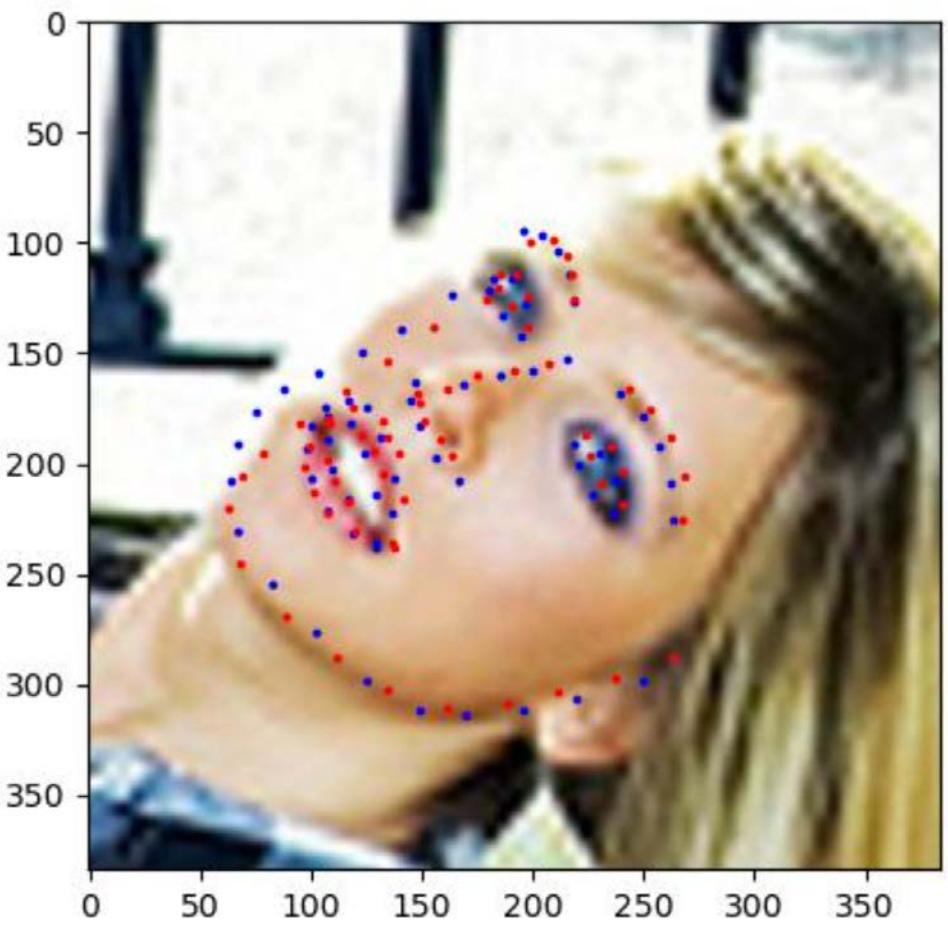


Data augmentation (red : ground truth, blue: predicted landmarks)

- Without data augmentation

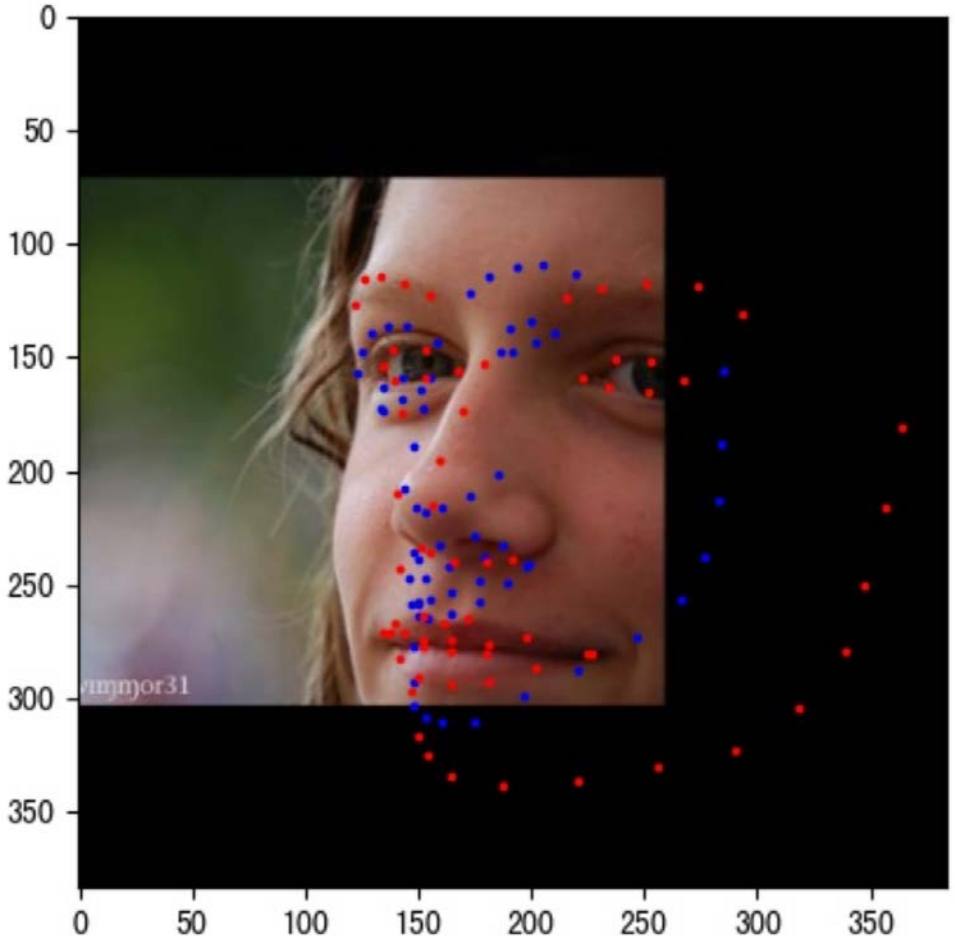


- With data augmentation

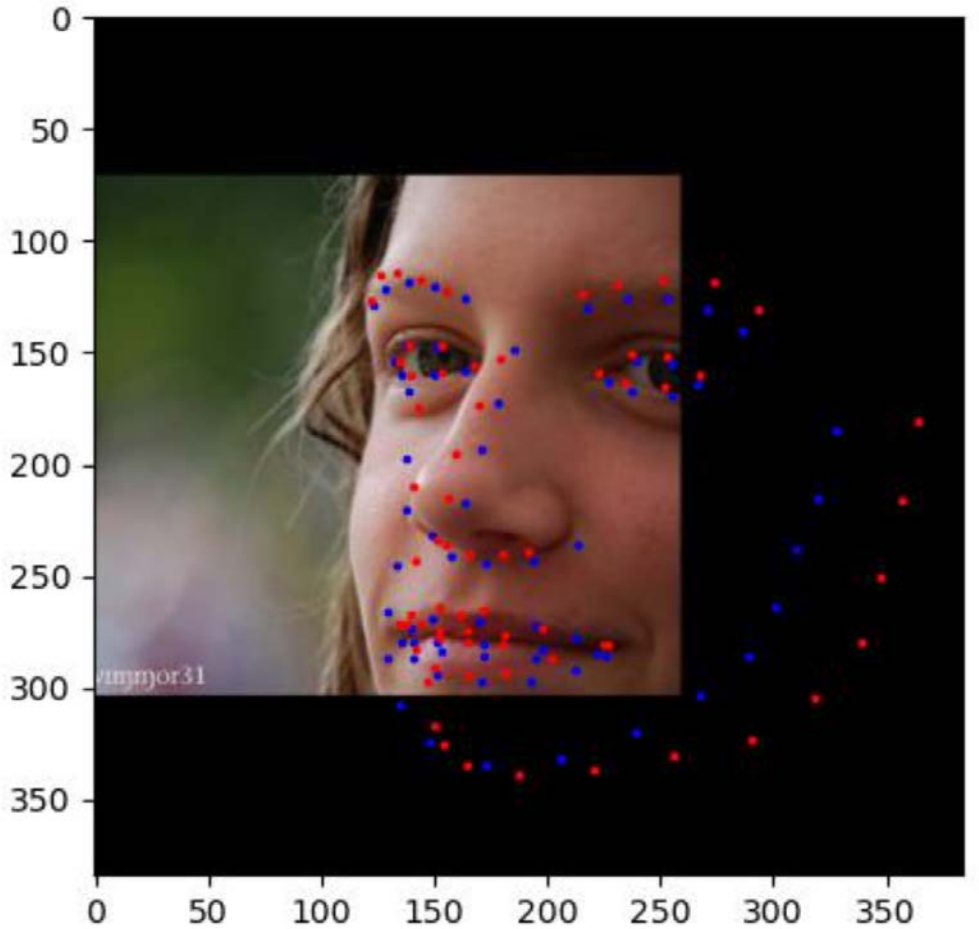


Data augmentation (red : ground truth, blue: predicted landmarks)

- Without data augmentation



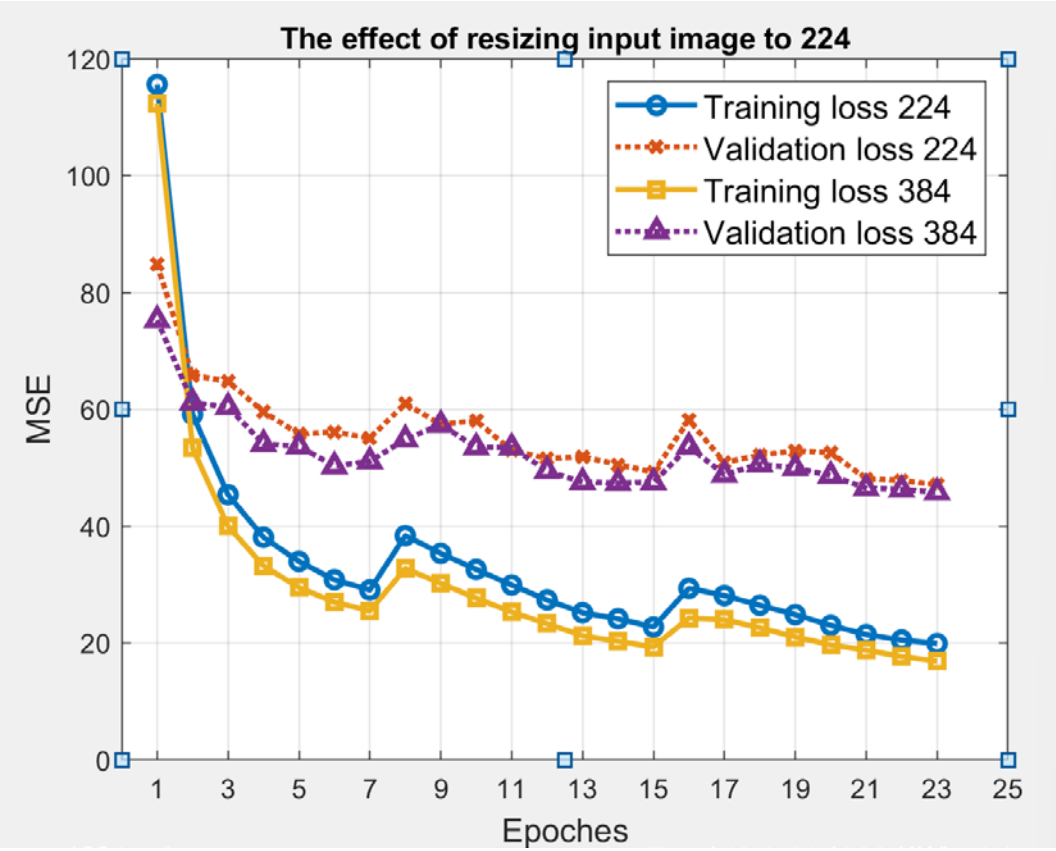
- With data augmentation



Experiment : resize 384x384 image to 224x224

- Pros : reduce the training time by half and it's convenient to tune hyperparameters.
- Cons : loss of precision.

Except the image size, other parameters are same.



Thank you

Appendix: Fine-tune Hyper prarameter

```
hyper = {
    'batch_size': 16,
    'img_resize': 384,
    'lr':0.005,
    'num_epoch': 32,
    'cos_T_0':8,
    'cos_T_multi':1,
}
```

```
p_dict = {
    'ColorJitter_b':0.3,
    'ColorJitter_c':0.3,
    'ColorJitter_s':0.3,
    'ColorJitter_h':0.1,
    'RandomGrayscale':0.4,
    'random_scale_p':0,
    'random_scale_scale':0.8,
    'random_rotate_p':0.4,
    'random_rotate_degree':45,
    'random_blur_p': 0.2,
    'random_blur_min': 128,
    'random_horizontalfip_p':0.5,
    'random_erasemargin_p':0.4,
    'random_erasemargin_ratio':0.7,
    'RandomErasing_p':0.1,
    'RandomErasing_scale_min':0.02,
    'RandomErasing_scale_max':0.1,
}
```