Light-Weight Facial Landmark Prediction Challenge

Caffeine Valid

資工二 洪國瀚
資工二 方泓傑
工海三 劉虹伶
生機三 陳勳蒲

OUTLINE

- Our Performance
- Data Augmentation
- Model
- Loss function
- Experiments
- Reference

Our Performance

- Architecture: ShuffleNet Ver.2 with rotate, color and wing loss + data unbalance
- model Size: 10.09 MB
- NME Score
    - **Total score: 2.094**

Data Augmentanion_transform

- random Rotate

- Color jitter

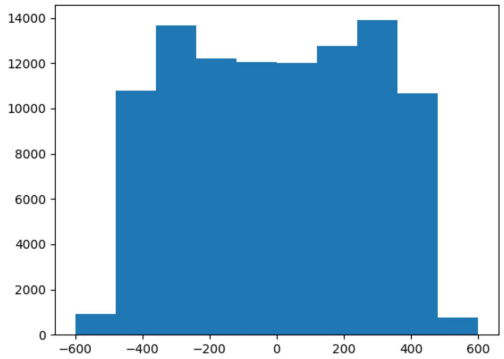
Data Augmentanion_Pose-based data balancing

Apply PCA (Principal Component Analysis) to the aligned training shapes and project the original shapes to the one dimensional space

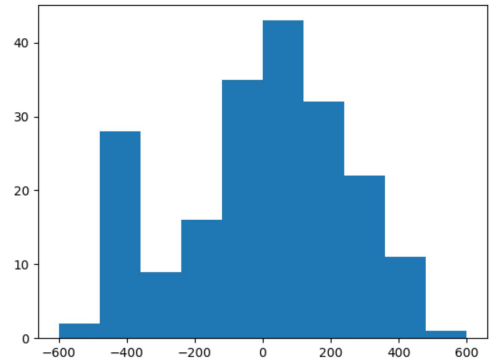
Data Augmentanion_Pose-based data balancing

Head angle’s histogram of train/validation datasets

- train dataset



- validation dataset



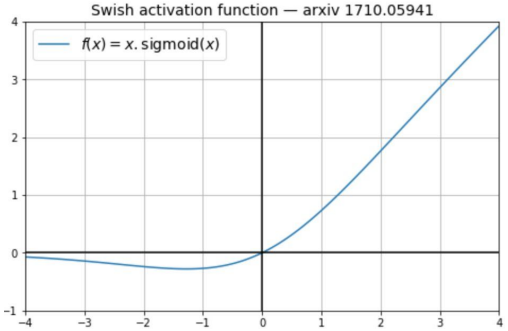
Model

- pointwise group convolution
- depthwise convolution
- Channel shuffle

ShufflenetV2[1][2]

Replace
activate fuction

Swish activation function[4]



- Spatial information

CoordCov[3]

- Better performance in paper's experiment
- More smooth than ReLu

Loss function

- MSE loss

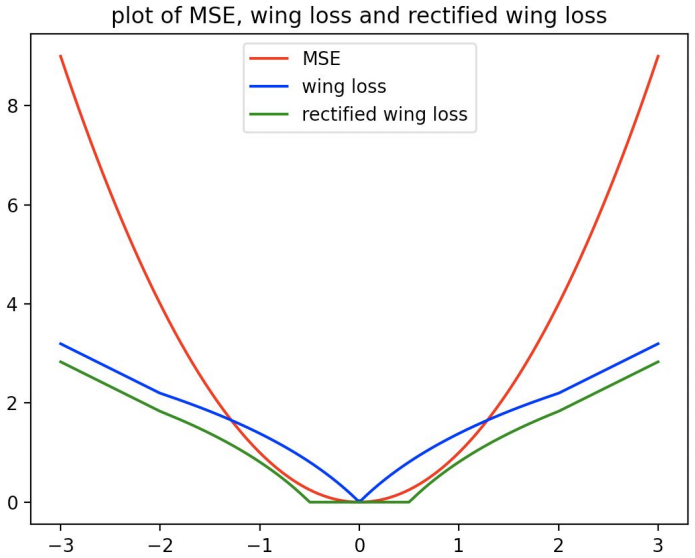
$$MSE(x) = x^2$$

- Wing loss

$$wing(x) = \begin{cases} \omega \ln(1 + |x|/\epsilon) \text{ , if } |x| < \omega \\ |x| - C \text{ , otherwise} \end{cases}$$

- Rectified wing loss

$$RWing(x) = \begin{cases} 0 \text{ , if } |x| < r \\ \omega \ln(1 + (|x| - r)/\epsilon) \text{ , if } r \leq |x| < \omega| \\ |x| - C \text{ , otherwise} \end{cases}$$



Experiment___data augmentation

- Color Jitter
- Random Rotate
- Random Posterize
- Random Solarize
- Random GaussianBlur

	pure	random Posterize	random Solarize	random GaussianBlur	color_jitter
score	4.661	4.81	4.85	5.015	4.59

Experiment___data augmentation

Pose-based data balancing

	origin	Pose-based data balancing	Pose-based data unbalancing
number of img	99756	142877	156510
score	4.661	4.746	4.488

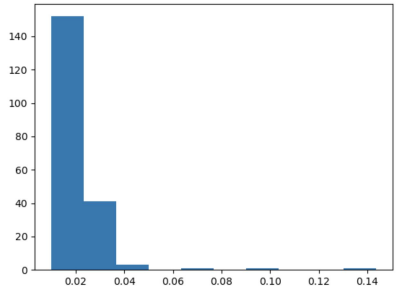
Experiment___Model

Add ORB keypoint channel		
	Original	New_version
NME	3.90	4.01

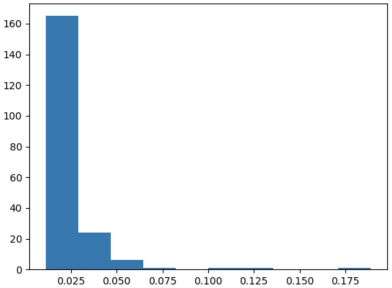
Replace inception block (googlenet)		
	Original	New_version
NME	3.90	3.98

Experiment___Loss Function

- Wing loss



- MSE loss



Loss function	MSE loss	wing loss	rectified wing loss
score	4.661	3.90	4.02

Experiment___hyperparameters

Shufflenet v2

model size (lr=0.0001)	0.5x (1.866MB)	1.5x (10.092MB)	2.0x (14.256MB)
score	4.39	4.59	4.31

Learning rate (model size=2.0x)	0.00005	**0.0001**	0.0005
score	4.594	**4.31**	4.92

Experiment___hyperparameters

Shufflenet_corr

model size (lr=0.0001)	0.5x (1.868MB)	**1.5x** (10.094MB)	2.0x (14.259MB)
score	4.14	**3.95**	4.384

Experiment___hyperparameters

Wing loss: $$wing(x) = \begin{cases} \omega \ln(1 + |x|/\epsilon) \text{ , if } |x| < \omega \\ |x| - C \text{ , otherwise} \end{cases}$$

epsilon \ omega	10	14	20
2	3.90	3.91	4.05
3	3.96	3.99	4.02

Reference

[1]Ma, Ningning, et al. "Shufflenet v2: Practical guidelines for efficient cnn architecture design." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

[2]Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[3]Liu, Rosanne, et al. "An intriguing failing of convolutional neural networks and the coordconv solution." *Advances in neural information processing systems* 31 (2018).

[4]Ramachandran, Prajit, Barret Zoph, and Quoc V. Le. "Searching for activation functions." *arXiv preprint arXiv:1710.05941* (2017).

[5]Xiang Wang, Kai Wang, Shiguo Lian,, et al.”*A Survey on Face Data Augmentation*”*arXiv preprint arXiv:*1904.11685.(2019)

[6]Feng, Z. H., Kittler, J., Awais, M., Huber, P., & Wu, X. J. (2018b).”Wing loss for robust facial landmark localisation with convolutional neural networks.” In IEEE conference on computer vision and pattern recognition (CVPR)

[7]Feng, Z. H., Kittler, J., Awais, M., Huber, P., & Wu, X. J. “Rectified Wing Loss for Efficient and Robust Facial Landmark Localisation with Convolutional Neural Networks.”

Reference

Github:

- CoordConv code source :https://github.com/mkocabas/CoordConv-pytorch
- Shuffenet code source :https://github.com/megvii-model/ShuffleNet-Series/tree/master/ShuffleNetV2
- Inception block code source :https://github.com/Lornatang/GoogLeNet-PyTorch/blob/master/googlenet_pytorch/model.py