Facial Landmarks Detection with Fake-it Dataset

github link https://github.com/Jenna-Che/Facial-Landmarks-Detection-with-Fake-it-Dataset

Chen Qiao, Jingwen Che, Haoran Ding

1. Introduction

Deep learning neural networks for facial landmarks detection require a sufficient amount of data for the purpose of training and testing. People often use real world photos to train the model. But collecting real world data and labeling them costs a lot. Can we use synthetic faces to train the model and get a same or even better result? In this project, we will design a deep learning neural network model for facial landmarks detection and train it on the CG dataset, then test it on the real world samples.

2. Dataset

The **training set** of our project is drawn from the Microsoft CG dataset (https://github.com/microsoft/FaceSynthetics) containing 100,000 images of synthetic faces at 512×512 pixel resolution. The labels of the training set are 2D landmark coordinates which are also provided alongside the images. The images of the **testing set** are chosen from Flickr-Faces-HQ (FFHQ) (https://github.com/NVlabs/ffhq-dataset). FFHQ was originally built as a benchmark for generative adversarial networks (GAN). It contains 70,000 high-quality images of human faces at 1024×1024 resolution.

3. Data preprocessing

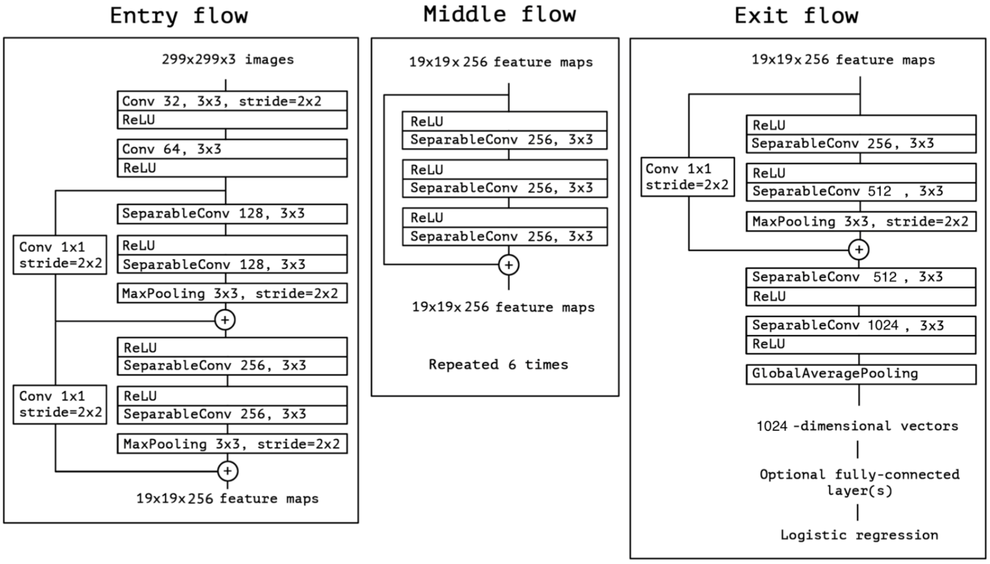
- Augmentations for faces:
  - Random Brightness
  - Random Contrast
  - Random Gamma
  - Random Saturation
  - Random Hue
  - Random Rotation
- Augmentations for landmarks:
  - Random Rotation

We tried various data augmentation techniques and found that the above set of augmentation techniques gave the most improvements to the result.

4. Models

- **Our Model** (Based on Xception)

  The following picture shows the structure of our model. It has three main flows. The entry, the middle, and the exit flow.



For the entry flow, the input first goes through some basic convolutions, and then it goes into a residual-like block. After that we repeat it, and then come to the middle flow. **For the middle flow** is much easier—just to do separable convolution several times and add it together with the original input of this flow. Then we repeat it 6 times. **For the exit flow**, we use a residual block which is similar to the block in entry flow, and then do separable convolution on it twice. For all the activation functions, we use **LeakyReLU** instead of ReLU. Finally we applied Global Average Pooling to flatten the output and use fully connected layers to do the final prediction.

- ### Other Network Architectures
  For the purpose of comparison, we also implemented some other popular network architectures.
  - **Xception** stands for Extreme version of Inception. The essence of the model is to assume that cross-channel correlations and spatial correlations can be mapped completely separately.
  - **ResNet-50** uses skip connections to jump over some layers. It helps in tackling the vanishing gradient problem using identity mapping.
  - **MobileNetV2** is a convolutional neural network that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers.

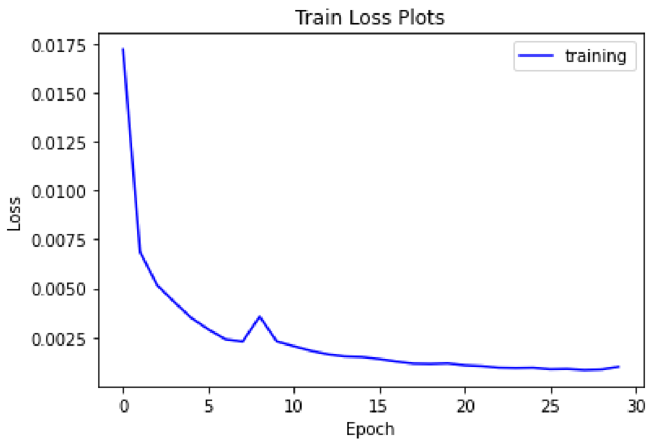
5. Minimum Viable Dataset Size

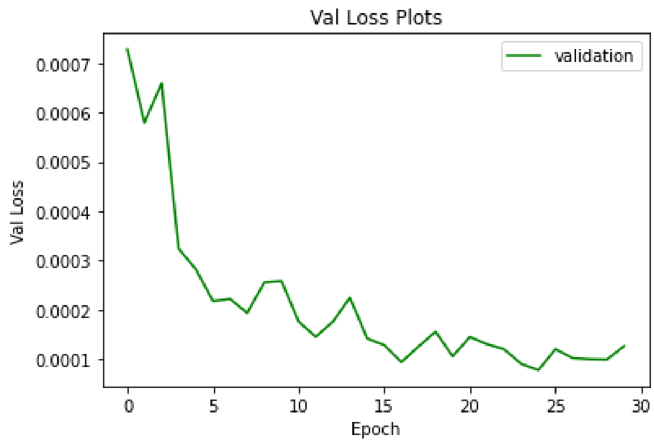
Val MSELoss	**Training set 2000**	**Training set 5000**	**Training set 10000**
Our Model	0.00070798	0.00010353	0.00003220
Xception	0.00091584	0.00010563	0.00003259
ResNet-50	0.00120577	0.00018337	0.00007236
MobileNetV2	0.00045519	0.00023415	0.00006439

When trying to determine the minimum viable training set, we found that a size of 5000 gave a reasonable training time with an acceptable loss.

6. Result

MSE Loss	Training loss	Validation Loss	Test Loss
Our Model data mixing	0.00094157	0.00007829	0.00189837
Our Model	0.00078196	0.00007003	0.00269198
Xception	0.00083827	0.00002770	0.00287737
ResNet-50	0.00187827	0.00046778	0.00385637
MobileNetV2	0.00116653	0.00011362	0.00336023





With our model and dataset size determined, we **optimized the hyperparameters** (learning rate, weight decay, epoch, etc.), then **implemented data mixing** during training. Compared with other network architectures with their best fit hyperparameters, our model is still the best—it gives the lowest training loss and test loss. Even when **tested on 30-fps videos**, our model gives excellent results (https://youtu.be/8jq60Haj4z4).

7. Reference

- Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, Jamie Shotton.: Fake it till you make it: face analysis in the wild using synthetic data alone. International Conference on Computer Vision 2021.https://openaccess.thecvf.com/content/ICCV2021/html/Wood_Fake_It_Till_You_Make_It_Face_Analysis_in_the_ICCV_2021_paper.html

- Chih-Fan Hsu, Chia-Ching Lin, Ting-Yang Hung, Chin-Laung Lei, Kuan-Ta Chen: Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization. arXiv:2005.08649. https://arxiv.org/abs/2005.08649

- Face Landmarks Detection https://github.com/braindotai/Facial-Landmarks-Detection-Pytorch