인공지능개론 기말 레포트

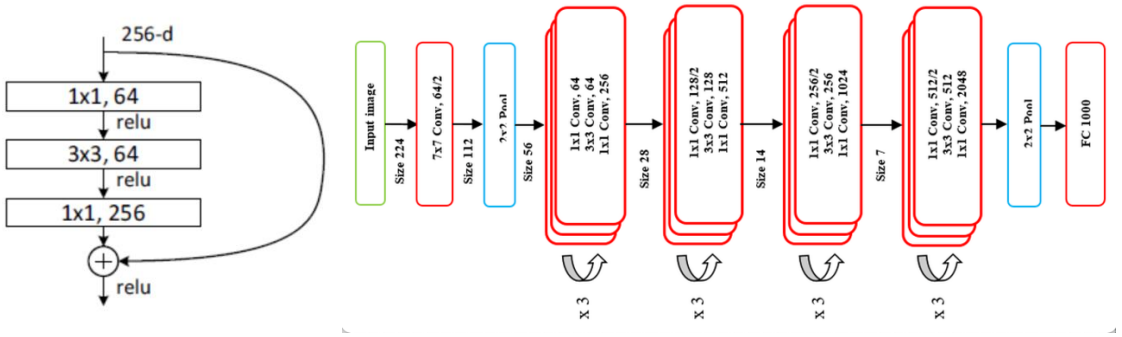
전기공학과
2017030919
문상인

1. ResNet-50 architecture

ResNet is introduced at “Deep Residual Learning for Image Recognition, CVPR 2016”. The motivation of ResNet is using residual connections to effectively train ‘deep’ networks (max 152 layers in this paper) and achieve remarkable improvements on several datasets.

Plain ResNet is inspired by VGG nets. The conv layers mostly have 3x3 filters and follow two simple design rules. (i) for the same output feature map size, the layers have the same number of filters, (ii) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer. Downsampling is performed directly by conv layers with stride of 2. Network ends with a global average pooling layer and 1000 way FC layer with softmax.

Basic architecture, BottleNeck, is shown below and the shortcut line means skip connection which prevents gradients from vanishing or exploding through the training. 1x1 conv layers are added to the start and end of network to reduce the number of connections while not degrading the model performance.

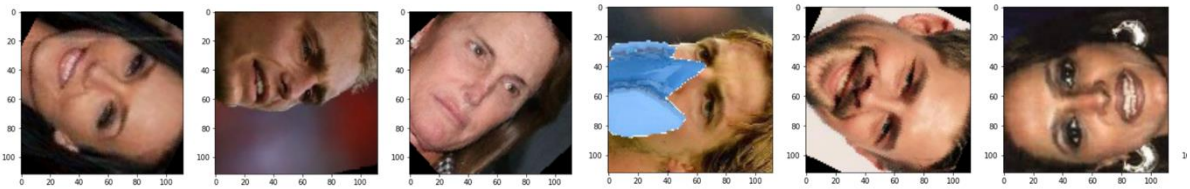


The figure below shows the first BottleNeck structrue of ResNet 50.

```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  (layer1): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
```

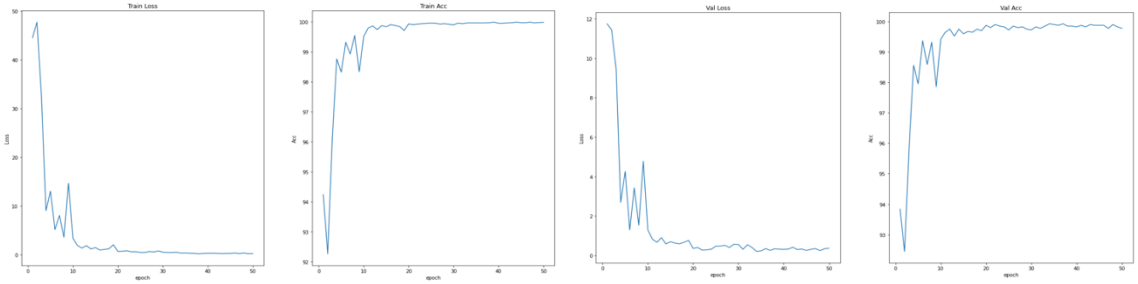
2. Augmented image samples

These images are randomly sampled at the first batch of trainloader. As you can see below, the pictures are randomly flipped, rotated and cropped.



3. Plot

These figures are showing the Loss and Accuracy during Training and Evaluation. Loss is calculated on all images of train loader and val loader. Thus, you can compare these two losses by multiplying 3.75 to validation loss. (Train image : 15k, Val image : 4k)



(From left, Train Loss, Train Acc, Val Loss, Val Acc)

4. Discussion

Datesets. In all of the following experiments, I assess the methods on a test set of 4k images, using as training set of 15k human faces. Custom image is consisted with 4 picture in each class (non-mask, mask). And the point is that 2 of them are strictly face-cropped and the others are not.

Architecture. I use ResNet-50 encoder, the output of which is average pooled, producing a feature $\phi \in R^{1000}$. By the fully-connected layer, it produces a number which can classifies the mask. I use SGD optimizer with a learning rate of 0.005 and batch size of 64. I train for 50 epochs, which takes an hour in Colab.

4.1. Ablation study

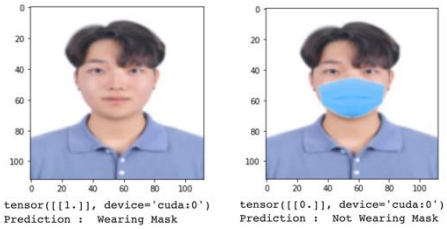
In this section, I present results from my ablation study, which investigates the effects of learning

rate and data augmentation during training. I compare four networks trained with: (i) no augmentation and lr with 0.01, (ii) only adjusted lr with 0.005, (iii) only data augmentation, (iv) both lr changing and augmentation. Evaluations are carried out with two types of datasets: val_loader and custom face images.

The Figure below illustrates that applying augmentation improves predictions of real images. Models with no augmentation shows worse report on not strictly face-cropped images. Rather, accuracy between two models don't show much improvements on validation accuracy.

Changing learning rate also don't show much improvements. But with lr=0.01, model shows over 99.99% at training data in epoch 7. Also, mini-batch loss becomes almost zero after that. Rather, lr with 0.005 doesn't show that issue and loss falls more stably.

Input	Method	Accuracy	Acc on custom 8 images
Validation dataset	Baseline	99.87%	50%
	Adjusting lr	99.87%	50%
	Data augmentation	99.90%	100%
	Augmentaton + lr	99.90%	100%



4.2 Limitation

If we use some object like paper, Model can't detect whether wearing mask or occlusion happens. Also, if the image is blurry or has colorful background, it seems to fail to detect.

5. Qualitative evaluation