Project Final Report

Wentao Xu(wx225) Xingyue Dai(xd86) Ning Xu(nx43)
Cornell Tech
Image Recognition and Identification

Abstract

Our goal in this project is to compare different machine learning models and find the best fit to build an application to identify and verify objects(including human faces) from digital images and test them on a large dataset of images. We tackled two datasets in this project which are the 'facescrub' dataset and the 'CIFAR-10' dataset and built our models to train data with techniques including logistics regression, vgg, MobileNet V2, etc. With training more than 60k images, our application reached 99% training accuracy.

1 Introduction

Face recognition has been a hot application and research topic over the years. Recognizing faces with technical tools is important not only because it is useful to verify personal identity but also uncover criminals and deter crimes.As stated earlier in our milestone report, we successfully implemented logistic Regression together with One-vs-All(multi-class classification) and one-layer convolutional Neural Network to build and train models on a subset of the 'FaceScrub' dataset and got 99% training accuracy of the predictions on the training set.Since then we did more structured deep neural networks to improve the accuracy of predictions.We improved our model with new machine learning techniques (eg.VGG) and trained and tested it with much larger dataset(with 60k images) of not only face images but also images of objects in general in this project for broader use of our model. We also did some experiments with MobileNet.

Unlike before when we mainly used the 'FaceScrub' dataset which in total includes over 10k face images of 530 People as our input and training data, we recently switched gear to the 'CIFAR-10' dataset that contains 60k 32x32 color images in 10 different classes(airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks). Part of the reason is due to the unstable VPN(all of our group members are based in China currently). We were unable to download the complete dataset from the source after 5 hours' trying and even for those downloaded images, their qualities varied a lot. But more importantly, after the success of our model on the face recognition, we found building models on predicting larger categories more challenging and the 'CIFAR-10' dataset provided weigh more amount of image data than the 'FaceScrub' dataset could possibly do.

2 Background

To understand our work, basic knowledge about machine learning and deep learning techniques will be needed. During this section, we will briefly do an overview on our previous work as a background and describe some techniques in details in next section for you to better understand our work.

Previous work recap Our previous experiments on the subset of 'FaceScrub' dataset could be divided into two parts in general:image data preprocessing and modeling.

For the first part, we separated the dataset into three sub datasets: train set, validation set and test set and converted each image to gray-scale and resized it to 32x32.For the modeling part, we mainly implemented two techniques to build and train our model during the milestone phase. The first is One-vs-All with logistic regression model where we applied multi-class classification technique one-vs-all to break down a multi-classification problem into multiple binary classification problems. We created 6 binary classifier models such that each of them is a logistic regression model and then made predictions by using the model that was most confident. It turned out that our model has about 88% training accuracy of the predictions on the training set, about 76% accuracy of the predictions on the validation set and about 74% accuracy of the predictions on the test set.And the second is one layerd-convolutional neural network where we input the data with shape (500 x (32, 32, 1)) into the one-layered convolutional Neural Network and then set the number of training epoch to be 100. At the end, we got 99% accuracy on the training set and The test accuracy of the model reached 88%.

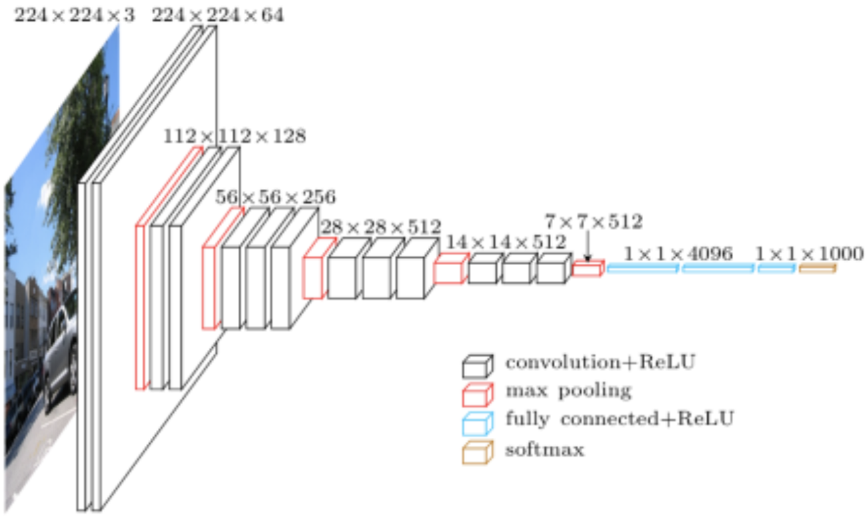
3 Methods

Techniques: Logistic Regression, One-vs-All(multi-class classification), One-layer Convolutional Neural Network, VGG, MobileNet V2.

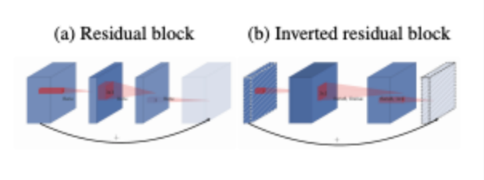
As described above, through our previous experiments, we used logistic regression and Convolutional neural network for data training and modeling and used one-vs-all techniques for categorization on the 'FaceScrub' dataset. We applied basic one-layered convolutional neural network to the training dataset, which had 64 filters with size 3x3, a max-pooling layer and a dense layer with size 100 and an output layer.

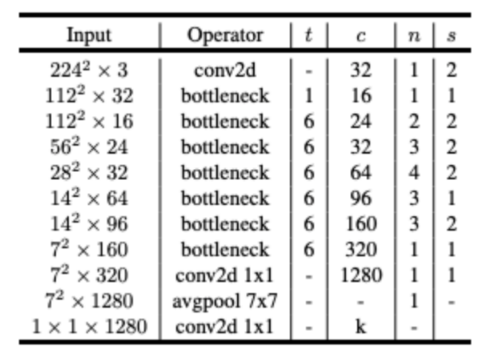
Since then, we mainly implemented two other powerful models: VGG and MobileNet V2 to improve our model on accuracy of image recognition and predictions in the large-scale image recognition setting.

VGG To keep the input image size consistent, VGG takes in 224x224 pixel RGB images. The convolution operation is carrying out from left to right, top down with 3x3 convolution filters. VGG has three fully-connected layers where the first two have 4096 channels each and the third has 1000 channels with each for a class as shown below



MobileNet V2 There are two types of blocks in the architecture of MobileNet V2, which includes a residual block with stride of 1 and a block with stride of 2 for downsizing.Unlike traditional residual models, MobileNet V2 is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers. The intermediate expansion layer is the depthwise convolution. Below are the details of this architecture





Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

4 Prior Work and Discussion

One layered Convolutional Neural Network The codes constructing one layered CNN in PyTorch

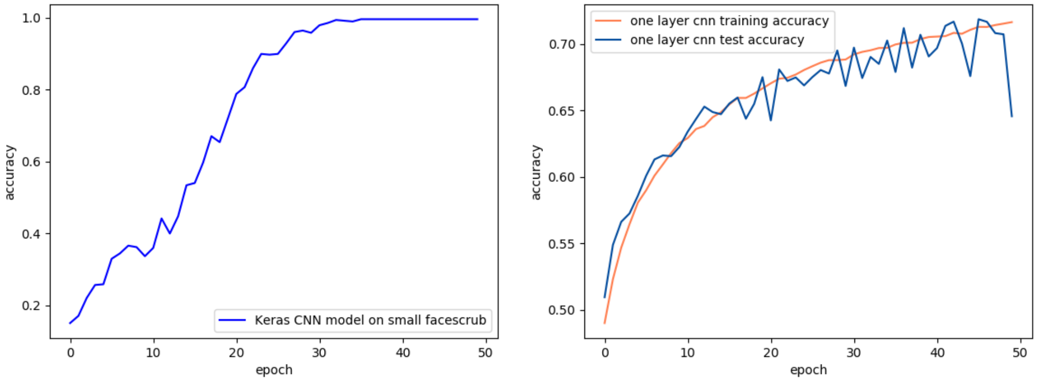
```
class cnn(nn.Module):
    def __init__(self):
        super(cnn, self).__init__()
        self.conv1 = nn.Conv2d(3, 64, kernel_size=3, padding=1)
        self.bn1 = nn.BatchNorm2d(64)
        self.classifier = nn.Linear(16384, 10)

    def forward(self, x):
        x = F.relu(self.bn1(self.conv1(x)))
        x = F.max_pool2d(x, kernel_size=2, stride=2)
        x = F.avg_pool2d(x, kernel_size=1, stride=1)
        x = x.view(x.size(0), -1)
        x = self.classifier(x)
        return x
```

During the milestone phase, our one layered convolutional neural network worked well on the samll dataset of the ’facescrub’ dataset. So, initially we decided to improve such one layered convolutional neural network and implement it on the CIFAR-10 dataset by using Pytorch instead of Keras which we used in the milestone phase.We so added a layer called BatchNorm for faster training to improve regularization and accuracy.Besides, we added max pooling layer and average pooling layer to extract informaton. In PyTorch, we used the linear layer as the dense layer to do the classification.

However, after so many improvements described above to our one layer convolutional neural network model,it turned out that the results on the CIFAR-10 dataset were still not ideal enough.

Below are two charts showing the accuracy we got from modeling using one layered CNN on facescrub and CIFAR-10 dataset respectively, where the left picture below is the training accuracy on small facescrub dataset with 50 training epoch, and the right picture is the training accuracy and test accuracy on cifar-10 dataset with 50 training epoch. As you can see,the accuracy on CIFAR-10 is around 70% which is weigh lower than 99% on the small facescrub dataset. It was due to such an undesirable performance of one layerd CNN on the ’CIFAR-10’ dataset, we decided to explore two other strong models: VGG and MobileNet which we will describe in details in the next section.



(a) CNN in Keras on small scale facescrub

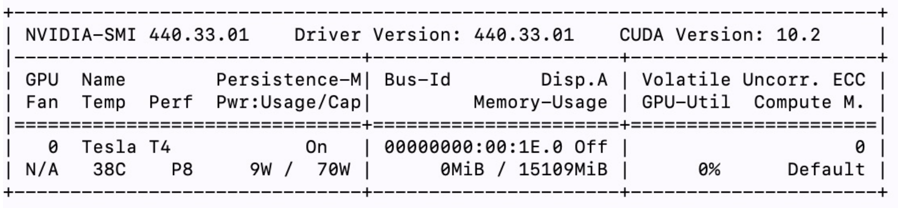
(b) CNN in PyTorch on CIFAR 10

Figure 1: The accuracy of one layer CNN on cifar-10 and small facescrub

5 Experiment Analysis

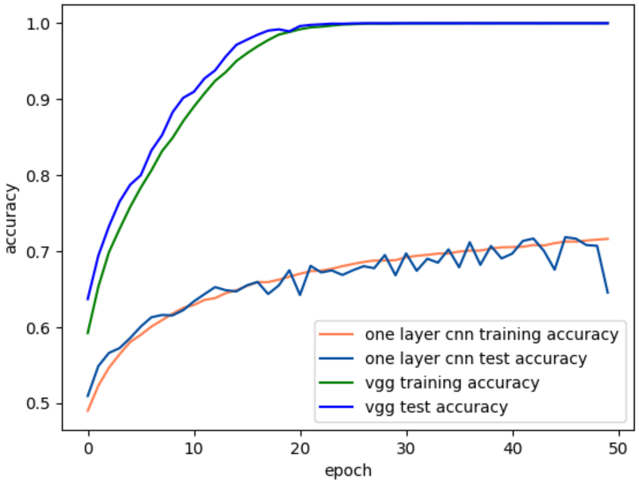
One layered CNN and VGG We did experiments on both one layered CNN and VGG on CIFAR−10 dataset. Since such a dataset is so large that it includes 60k images, we were unable to train it with our own personal laptops. So, we decided to rent AWS EC2 cloud computing service.

Below is the GPU configuration of our EC2:



```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.33.01    Driver Version: 440.33.01    CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla T4            On   | 00000000:00:1E.0 Off |                    0 |
| N/A   38C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

Let's first compare our results on the training accuracy of modified one layered CNN with those on the VGG, as shown in the chart below.



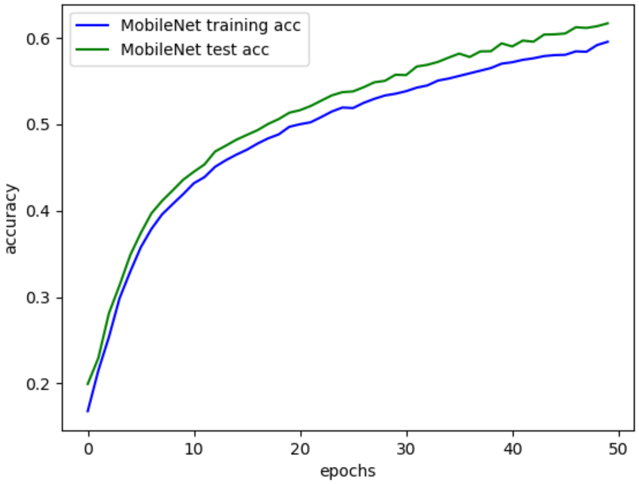
Here we run both VGG and one layered CNN with 50 epoch. Within 50 epoch, the one layered CNN finally reaches an accuracy with 0.7, which is unsatisfactory compared to the performance on small scale facescrub dataset.

Also, we found that the test accuracy line of one layered CNN in the graph fluctuated a lot, one possible explanation was that only one CNN layer cannot extract large information efficiently, i.e, we need deeper model.

The VGG training accuracy line and test accuracy line show that the VGG model converges fast, and the lines don't fluctuate, which means that the stability and the ability to extract information of VGG are strong. This is an ideal model as it returned a 99% training and test accuracy on the 'CIFAR-10' dataset.

The MobileNet

The training accuracy and test accuracy generated from MobileNet didn't reach our expectation:



One possible cause is that MobileNet is a special model used for mobile and embedded vision applicants, and it is a light weight deep neural network different from the previous two models.

Also, we didn't choose properly two hyper parameters in the MobileNet. But due to lack of budget, we didn't continue using EC2 service to do experiments with Mobilenet further.

6 Conclusion

Proper choices on models can save lots of model tuning and data processing time. In this project, we tried different models from one layered CNN, VGG to MobileNet. We noticed that different models with different structures have very different performances. It really depends on which dataset we work on to choose the proper model that yields the best outcome. Just like after so many experiments, we found VGG to be the best fit to train large scale dataset of images, the CIFAR-10 dataset in this case.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references.

[1] Wei, J. (2019, July 4). VGG Neural Networks: The Next Step After AlexNet. Medium. https://towardsdatascience.com/vgg-neural-networks-the-next-step-after-alexnet-3f91fa9ffe2c.

[2] https://zhuanlan.zhihu.com/p/31551004.

[3] Pytorch Team. MOBILENET V2. PyTorch. https://pytorch.org/hub/pytorch_vision_mobilenet_v2/.

[4] Tsang, S.-H. (2019, May 19). Review: MobileNetV2 — Light Weight Model (Image Classification). towards data science. https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c.

[5] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

[6] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition