A CNN-based Machine Learning for Image Classification Application

FENG Xuezhi (2030026034) FENG Ziao (2030026036)

PENG Rongwei (2030026116) QIAN Jingbin (2030026117) ZHENG Yuguang (2030026217)

*Abstract*—In computer vision and machine learning, there is a mechanism called CNN (Convolution Neural Network). It is very useful for computer vision assignments, such as image classification, image detection, image segmentation, which can be used in many fields, such as medical treatment, business and face recognition and so on. This paper is going to research CNN in terms of image classification. Image classification is a method of image processing which refers to an image processing method that distinguishes different characters and types of images to different targets, and it uses CNN network to quantitative analyze images to divide every into one of the several categories, instead of human’s visual interpretation. However, there exist some shortcomings which slow processing speed and low classification accuracy need to be improved. This paper is going to introduce the image classification technique based on CNN. In the introduction part, there is a brief introduction of construction of CNN and image classification, and some models to implement image classification. In the literature review part, it will show the models – “Lenet”, “Alexnet”, “VGGNet”, “ResNet”, “EfficientNet” and in details. In the last part, we implement these models and using CIFAR-10 datasets to train these models to see their performance, including the experiment results and the analysis of these model. And then we develop an image recognition program, which users can upload a picture and choose different CNN models to see their prediction Finally, there is a clear conclusion of this paper.

*Index Terms*—Convolutional Neural Network, Image Classification, Machine Learning.

# INTRODUCTION

T

HIS part is going to introduce the idea of CNN and its general operations. Then, some famous models of CNN are going to be briefly introduced.

## The Birth of Convolutional Neural Network

There are three distinct drawbacks when using fully-connected neural networks to process large-sized images:

(1) Expanding the image into vectors will loss of spatial information;

(2) Too many parameters are inefficient and difficult to train;

(3) A lot of parameters can quickly lead to over-fitting of the network.

The convolution neural network can solve the three above problems very well.

## Concept of Convolutional Neural Network

CNN, which is a feed-forward neural network, has a convolution structure. Through the deep network, the operation of convolution can reduce the memory occupied. There are 3 main operations, local receptive field, weight sharing, and pooling layer. They can try to avoid the problem of over-fitting. What’s more, they reduce network parameters effectively.

## Structure of CNN

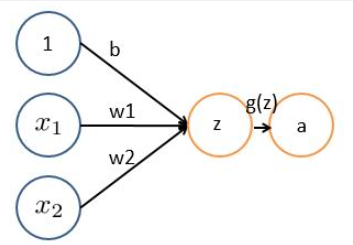
The most common form of a convolution network is to put some convolution layers together with the activation layers, followed by a pooling layer, and repeat this until the image is spatially reduced to a size small enough that it transitions to a fully-connected layer somewhere. The final fully-connection layer is the output. The most common structure of a convolution neural network is as follows:



Where ‘\*’ refers to the number of repetitions, ‘POOL?’ refers to an optional convergence layer. Among them, N≥0, usually N≤3, M≥0, K≤0, usually K＜3.

### *Neural Networks*

*Neural networks* consist of a number of neurons. Each neuron receives linear/nonlinear input. And the neural network will have different outputs with different weights and activation functions.



(Single neuron. X1 and x2 are input. W1 and w2 are weights. G(z) is an activation function. B is a bias.)

图示

描述已自动生成

(Neural network)

### *Activation Layer*

The activation layer maps the output of the convolution layer nonlinearly through the excitation equation. Sometimes we regard the convolution layer and the activation layer as the convolution layer.

In order to increase the nonlinearity of the neural network model, the activation function was introduced. It allows the neural network to approach each different nonlinear function at will. Therefore, the neural network can be used for a lot of different non-linear models. There are many non-linear activation functions such as ReLU, tanh, sigmoid, and so on.

图示

中度可信度描述已自动生成

sigmoid

图表, 折线图

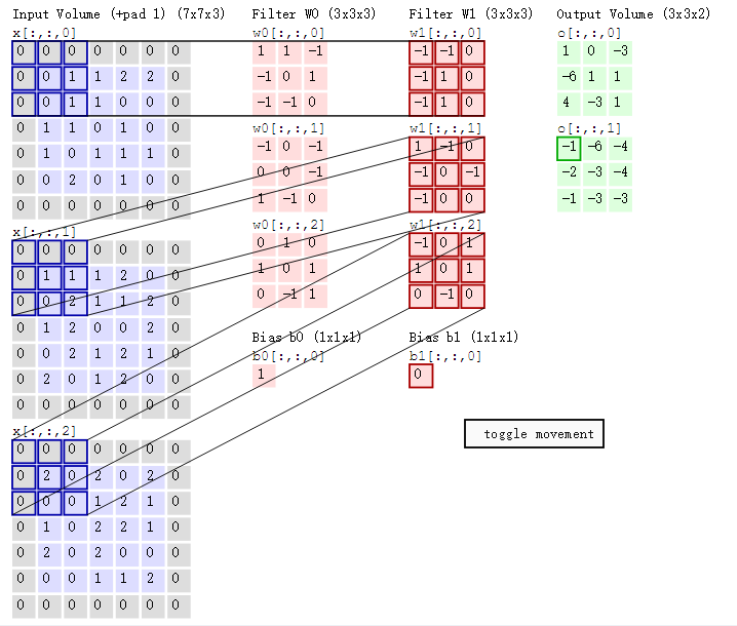
描述已自动生成

ReLU

### *Convolution Layer*

*What is convolution?*

Convolution is an operation that gets the inner product between the image (the data of different windows) and the filter matrix.



The calculation of convolution refers to sliding the window of the convolution kernel at a certain period, multiplying the numbers of the convolution kernel at each location by the corresponding elements of the input, then summing them up, and saving the result to the corresponding location of the output.

Parameters involved:

Depth: The number of neurons that determine the depth thickness of the output. It also represents the number of filters.

Step stride: Decide how many steps you can slide to the edge.

Padding: add a few rings of 0 (or other numbers) to the periphery edge to make it easy to slip from the initial position to the end position in steps, which in general is to divide the total length by steps.

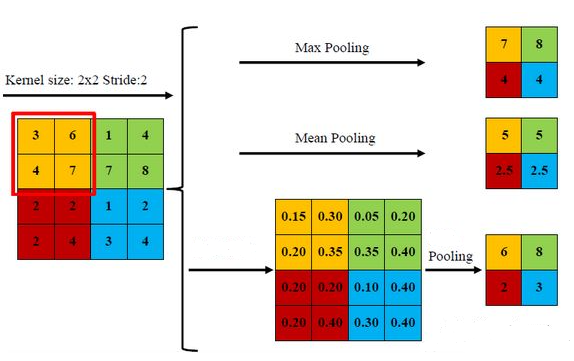
### *Pooling Layer*

Pooling, also known as ‘convergence’, is actually a down-sample process that reduces the size of high and long directions, reduces the size of the model, improves the speed of operation, and improves the robustness of the extracted features. In short, it is to extract the main features of a certain area, reduce the number of parameters, and prevent the model from overfitting.

Pooling layers usually occur after the convolution layer, and they alternate, and one convolution layer corresponds to one pooling layer.

Common pooling functions are:

Average Pooling/Mean Pooling, Max Pooling, Min Pooling, and Stochastic Pooling.



### *Fully-connected Layer*

The fully-connected layer is generally located at the end of the whole convolutional neural network and is responsible for converting the two-dimensional feature map of the convolution output into a vector of one dimension, thus realizing the end-to-end learning process. Each node of the fully-connection layer is connected with all nodes of the upper layer, so it is called the fully-connection layer. Due to its fully-connected characteristics, the parameters of the full connection layer are generally the largest

## Famous Model for CNN

### *LeNet*

LeNet is the first successful neural network. Its creator-LeCun is one of the winners of the 2018 Turing Prize. In the 1990s, due to the development of Support Vector Machine (SVM) and other algorithms, the development of deep learning has been greatly hindered, and the problem of gradient disappearance has not been well solved. But LeCun and others persevered and continued to work in the field. In 1998, LeCun proposed the LeNet-5 network to solve the problem of handwriting recognition. LeNet-5 is known as the "Hello Word" of convolutional neural networks, which is enough to see the importance of this paper.

### *AlexNet*

Alexnet is an excellent CNN network that won the first place in the 2012 ImageNet competition. Its authors are Alex Krizhevsky and Geoffrey E. Hinton et al. of the University of Toronto. Alexnet won the first place in the ImageNet ILSVRC2012 competition with a 15.3% error rate of the top-5 test, surpassing the 26.172% error rate of the second-place group using traditional computer vision. At the time, this performance shocked the entire computer vision community, setting off a research upsurge of deep convolutional neural networks in various fields. Although a large number of convolutional neural network structures that are faster and more accurate than Alexnet have appeared one after another. Alexnet, as the pioneer, still has a lot of places worth learning and reference. It has set the tone for subsequent CNN and even R-CNN and other networks. The features of this network are the use of ReLU function, two-GPU net, local response normalization (LRN), overlapping pooling and the unique design of the overall architecture of the entire network.

### *VGGNet*

In 2014, the computer vision group at Oxford University worked with researchers at Google DeepMind to develop a new deep convolution network called VGGNet and won the second prize at ILSVRC2014 competition. VGGNet can be seen as an enhance version of AlexNet, consisting of five convolutional layer, three fully connected layer, a softmax output layer. Although the number of layers is deeper, its structure is simple, and its convolution kernel is small than AlexNet. It designs six types of networks structures – A, A-LRN, B, C, D, E – to fit the different numbers of sub convolution layers. There are two types of VGG networks which are VGG16 (16 convolution layers) and VGG19 (19 convolution layers), but VGG19 does not improve so much. Nowadays, most of us use VGG16.

### *ResNet*

Resnet was proposed by Kaiming He and four other Chinese of Microsoft Research. The depth of the convolutional neural network reached an astonishing 152 layers and won the championship with a top-5 error rate of 3.57%. The depth of Resnet is much higher than that of VGGNet, but its reference number is lower than that of VGGNet and its effect is more prominent. One of the most innovative aspects of Resnet is the introduction of Residual Units

### *EfficientNet*

When scaling up ConvNets using method on scaling up depth, width and resolution, it is often difficult to find optimal accuracy and efficiency. EfficientNet demonstrated a new method to efficiently scale a network in three dimensions without manually tuning. Instead of previous ways using manual experiences that arbitrary scale these factors, this method configurate depth, width and image data resolution with a group of quantified compound coefficients. This is the first quantified method to show the relationship among all those three factors. Based on this method and neural architecture search (NAS) (Zoph & Le, 2017; Tan et al., 2019) a family of models named EfficientNets are successfully constructed.

# Literature Review

This part is going to analyze the five models in detail.

## LeNet

Although LeNet5 is a small network, it contains the basic modules of CNN (deep learning): Convolution layer, pooling layer, activation layer, and fully-connected layer.

LeNet-5 has 7 layers in total, and each layer contains trainable parameters, and they have many feature maps. Each feature map has many neurons. And they extract one feature of the input through a convolution filter.

图示

描述已自动生成

### Input layer： (It is not included in LeNet hierarchy)

### C1-Convolution layer:

Input: 32\*32

The size of kernel: 6\*6

The type of kernel: 6

The size of feature map: 28 (32-5+1)

The number of neurons: 28\*28\*6

Number of connections: (5\*5+1) \*6\*28\*28

### S2-Pooling layer

Input: 28\*28

The size of feature map: 14\*14

Sample area: 2\*2

The number of neurons: 1176

Sample type: 6

Number of connections: 5880

### C3-Convolution layer

Input: 14\*14

The size of kernel: 5\*5

The type of kernel: 16

The size of feature map: 10\*10 (14-5+1)

Number of connections: 10\*10\*(15+16)

### S4-Pooling layer

Input: 10\*10

The size of feature map: 5\*5

Sample area: 2\*2

The number of neurons: 6\*5\*5

Sample type: 16

Number of connections: 16

### C5-Convolution layer

Input: 5\*5

The size of kernel: 5\*5

The type of kernel: 120

The size of feature map: 1\*1 (5-5+1)

Number of connections: 120\*(16\*5\*5+1)

### F6- Fully-connected layer

Input: 120-dimensional vector

图示

描述已自动生成

### Output- Fully-connected layer

In the Output layer, there are 10 nodes in total, representing the numbers 0-9 respectively. ‘i’ is the result of network recognition which represents node ‘i’. The network connection mode of the Radial Basis Function is adopted. The calculation method of RBF output is:

yi​=j∑​(xj​−wij​)2

(Let x be the input of the upper layer.

And Let y be the output of RBF.)

The wij in formula—i is from 0 to 9, and j is from 0 to 83. The closer the output of RBF gets to 0, the closer it gets to ‘i’, and the closer it gets to the ASCII of ‘i’. It indicates the identification result is character ‘i’.

*Summary*

It is one of the earliest CNN, which promotes the development of deep learning. CNN is good at processing the structural information of images, in which LeNet is used to recognize handwritten characters.

## AlexNet

### Overview

After Alexnet won the first place in the ImageNet competition in 2012, its author Alex Krizhevsky and Geoffrey E. Hinton et al. published the article "*ImageNet Classification with Deep Convolutional Neural Networks*".

The ImageNet dataset used by the authors in the ILSVRC-2010 and ILSVRC-2012 competitions contains more than 15 million labeled high-resolution images, roughly in 22,000 categories. The article describes the techniques used by the authors and optimization measures to reduce overfitting. The entire Alexnet network consists of five convolutional layers and three fully-connected layers with a final 1000-way softmax.

### ReLU Nonlinearity

Generally, the activation function of neurons will choose the sigmoid or the tanh function. However, the author found that these nonlinear saturated functions are much slower than the nonlinear unsaturated functions in terms of gradient decay at training time. The non-saturating nonlinearity function used in Alexnet is f=max (0, x), which is Rectified Linear Units (ReLU). The results show that if the deep network is to be trained until the training error rate reaches 25%, ReLU only needs 5 epochs iterations, but the tanh unit requires 35 epochs iterations, and ReLU is 6 times faster than tanh.

图表, 折线图

描述已自动生成图表, 折线图

描述已自动生成

The author believes that using ReLU can accelerate speed of network training. At the same time, using this function, the complexity of the calculation will be reduced, all kinds of perturbations will have stronger robustness, and to a certain extent, the problem of gradient disappearance will be avoided. The advantage of ReLU is that it is simple and does not require exponential calculation. At the same time, ReLU is not prone to the problem of gradient divergence. When the Tanh and Logistic activation functions are at both ends, the derivative tends to approach zero, and the gradient is more approximately equal to 0 after multi-level multiplication. However, ReLU can easily change the distribution of data, so a common improvement is to add Batch Normalization after it.

### Training on Multiple GPUs

To increase the scale of network operation and improve the running speed, the author adopts two-GPU net. And it is stipulated that the GPU can only communicate at a specific layer. That is, each GPU is responsible for half of the computing processing.

At that time, due to technical constraints, it was very difficult to train large-scale networks. The author needs to spend a lot of time and engineering to complete the implementation of cross-GPU training. Due to the rapid development of GPU in last few years, the support of various hardware and software has become more and more complete, and this problem is not so important now.

### Local Response Normalization

图片包含 图形用户界面

描述已自动生成

The author performed Local Response Normalization (LRN) after activation and pooling, using adjacent data for normalization. This strategy contributed an accuracy rate of 1.2%, decreasing the error rate from 13% without normalization to 11% with normalization.

The authors use LRN, which significantly improves the generalization ability of the model and increases the recognition accuracy. Because this method mimics the lateral inhibition mechanism of the biological nervous system, and it amplified the local value with large responses.

### Overlapping Pooling

Pooling can reduce the image dimension exponentially and extract the images’ key information in the local area. In the pooling method, the author uses the pooling layer with a pooling unit interval of 2 and a size of 3×3 for overlapping pooling, which has a slight error rate compared to the non-overlapping pooling of size 2×2. decrease. Alexnet all use the maximum pooling method rather than average pooling since average pooling may cause blurring effect.

### Overall Architecture

图示, 工程绘图

描述已自动生成

Input a 224 × 224 × 3 image, go through 5 convolutional layers and 3 fully-connected layers with a final softmax, and the output is the predicted probability distribution of 1000 class labels. Two GPUs were used for training, and the network model was cut in half for training on two GPUs. As the depth of the network increases, the image size in the convolutional layers decreases while the depth keeps increasing.

### Reducing Overfitting

The authors first used data augmentation to increase the diversity of the data, and then used a new technique to reduce overfitting - dropout.

The author expands the sample training set by means of image translation and horizontal reflections, adjusting image gray level (using PCA principal component analysis to change the intensity of RGB channels), etc. After the data is expanded, overfitting can be reduced, and the generalization ability would also be improved.

The author uses dropout as a highly efficient model fusion technique: the output of the hidden layer is set to 0 with 50% probability, and these ignored neurons do not participate in feedforward or backpropagation. Each input goes through a neural network of a different structure, but they share weights. By using the dropout method, the complex dependencies between neurons can be reduced effectively, and it helps model learn more robust features, and work well with some other random subset of neurons.

### Result

* ILSVRC-2010: top-1 error rates is 37.5% and top-5 error rates is 17.0%
* ILSVRC-2012: top-5 error rates were 16.4%

表格

描述已自动生成表格

描述已自动生成

## VGGNet

**VGG’s structure is similar to AlexNet, but the depth of VGGNet is deeper. VGG is made up of five convolutional layer, three fully connected layer, a softmax output layer. There is a maxpool between each big layer. And the ReLU function is used as the activation function. In the essay, “*Very Deep Convolutional Networks for Large-scale Image Recognition*”, the authors design six network structure according to the different numbers of sublayers. Here is the figure of the VGG network表格

描述已自动生成**

**The six structures are similar, but the difference is the number of the layers in each convolution layer. From A to E, the depth is increasing from 11 layers to 19 layers. In this table, the parameter of convolution layer is expressed as “conv(receptive field size)-(number of channels)”. Among these six structures, D is the well-known structure, VGG16, and E is VVG19. Then, we will focus on the detail of VGG16.**

图示

描述已自动生成

**VGG16 is made up of 16 sublayers. It uses 224\*224\*3 image to input the layer. For the convolution process, the first one uses sixty-four 3\*3\*3 convolution kernels to convolute, the other uses 64 kernels with 3\*3\*64. For the pooling process, it uses 2\*2 pooling units to do the max pooling. The whole process for the first convolution layer is:** **convolution🡪ReLU🡪convolution🡪ReLU🡪Pooling.**

**For the second part of this processing, the first convolution uses 128 convolution kernels (3\*3\*64) to convolute, the second counterpart use 128 kernels with 3\*3\*128. And the pooling method is the same as above. The whole process is the same as part one, but just the convolution method changes.**

**For the third convolution layer, also, only the two convolution parts changes. The first one uses 256 convolution kernels (3\*3\*128), and the second one uses 256 (3\*3\*256) kernels. All the others are the same.**

**For the fourth convolution layer, the first convolution part changes to use 512 convolution kernels (3\*3\*256). And the other part is 512 convolution kernels (3\*3\*512).**

**For the fifth convolution layer, the two-convolution part both use 512 kernels (3\*3\*512). And the procedure is the same as above.**

**We can easily find that, the author used almost the same strategy to convolute the image. The universal process is convolution🡪ReLU🡪convolution🡪ReLU🡪Pooling, and the only difference is using different size of kernels. Then, VGG16 uses three fully connected layer to integrate the image.**

**In these three layers, the first two layers’ procedure is FC🡪ReLU🡪Dropout. ReLU is to activate the result of 4096 neurons. And the Dropout is for avoiding overfit. In the training period, these are three fully connected layers, however, in testing period, three convolution networks are instead of these three layers. The direct reason is that, by replacing these three fully connected layers to convolution layers, the network model can accept any size of images. If the height of image is higher than 224, then the image will be cut to fit the input requirement of fully connected layer. The advantage of this method is to reduce the influence of the position of features.**

**Finally, the image data will go through the softmax layer. This layer is using softmax function to calculate the result of 1000 neurons and output the probability of prediction.**

**So, how is the accuracy of VGGNet. The authors give us its performance in their paper.**

**The single-scale evaluation:**

表格

描述已自动生成

**The multi-scale evaluation:**

**表格

描述已自动生成**

**The multi-crop evaluation:**

**表格

描述已自动生成**

We can find that the testing accuracy can be more exact if the layer is deeper. And the effect of multi-scale training model is significantly improved that that of single scale training model. VGG19 does not improve so much than VGG16, so nowadays, most of the people use VGG16.

**In conclusion, VGG improves the accuracy of large-scale image classification and confirm the direction of improve the depth of CNN network. However, this network is not deep enough, and the parameters are too much.**

## ResNet

### Highway Network

Highway Network believes that increasing performance of network to a certain extent. Suppose that the network output y and input x of a certain layer can be represented by . This is the result of the nonlinear transformation. On this basis, retaining some original input x, which can be obtained

Without matrix multiplication and nonlinear transformation, it has some percentage direct transmit to the next layer from the previous layer.

### Model Structure

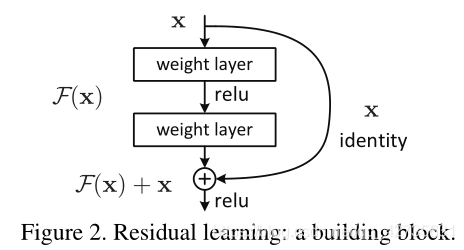
*Gradient vanishing problem*

To make the CNN deeper, some ideas to optimize the performance of the network have been proposed. The error rate that convolutional neural networks can achieve is also gradually decreasing. However, the front layer receive by the gradient propagates. And the repetition penalty may make the small gradient. Then the gradient problem arises.

### Identity Shortcut Connection

Resnet uses the shortcut connection to solve the problem of model degradation di deep network. As a result, it has an outstanding performance in the CNN image area. Meanwhile, Resnet has a short-circuit mechanism for every two or three layers compared with the ordinary network. And making the deep network play a role through residual learning.

Identity shortcut connection was cited by Resnet. The error will not increase when several congruent mapping layers are added for a network with relatively shallow saturation accuracy. When adding the congruent mapping layer, Resnet also allows to transferred directly to subsequent layers which raw input information, just like highway network.



### Residual error learning

On the left is a two-layer residual learning unit, and on the right is a three-layer bottleneck structure. It reduces or expands the feature map dimension by 1\*1conv. Therefore, the number of 3\*3 conv filters is independent of the input of the upper layer and the output of the lower layer. This method can save model’s accuracy, and reduces the network parameters. At the same time, it has a lower computation amount and saves the computation time.

图示

描述已自动生成

The introduction of residual learning units is equivalent to changing the learning objectives of Resnet. If you take the result of the convolution operation as the expected output H (x) of the network before the residual is added, your residual learning unit will not be learning H (x), but the difference H (x) -x between the output and the input.

### Resnet Network

The following figure shows how the Resnet network stacks the residual learning units. We can see that Resnet has many branches of the bypass that directly participate the input of the previous residual learning unit into the output. This allows the later residual learning unit to directly learn the residuals. For our ResNet18/34/50/101/152, the residual structures of the first layer of a series of residual structures corresponding to conv3\_x, conv4\_x, and conv5\_x in the table are all dashed residual structures. Because the first layer of the series of residual structures has the mission of adjusting the shape of the input feature matrix (reducing the height and width of the feature matrix to half of the original, and adjusting the depth channel to the channel required by the next layer of the residual structure). For ease of understanding, the network structure diagram of ResNet34 is given below, in which some information is simply labeled.

图表, 图示

描述已自动生成

### Network Configuration

In the paper that presented Resnet, the authors tried to extend the network to different depths. From the figure, we can see that these Resnet with different network configurations have similar infrastructure. Resnet introduced the residual learning unit structure. It successfully eliminated the problem of the accuracy decline of the test set because of the deepening of large number of layers in the previous convolutional neural network.

表格

描述已自动生成

## EfficientNet

### Model Scaling

For different resource constraints, there are many ways to scale a Convolutional Network. ResNet can be scaled up to Res Net-200 by adding more layers or scaled down to ResNet-18 (He et al., 2016) through one of scaling dimensions that is network depth(like Figure(c) below). Some networks increase the width of the network, that is, increase the number of kernels (increase the number of channel) to improve the performance of the network, as shown in Figure (b). Through previous works, it is obvious that three dimensions of network: resolution, depth(number of layers in a network) and network width(channel size of input matrix in each layer) are important to a network’s efficiency and accuracy. Here we focus on how these coefficients influence each other and how to manage these three dimensions.

图表, 箱线图

描述已自动生成

Based on previous observation, scaling up depth to the network results in getting richer, more complex features and is well applied in other use if needed (Zagoruyko & Ko-modakis, 2016). It should be noted that vanishing gradient problem will happen in very deep layers and the training will be difficult (Zagoruyko & Ko-modakis, 2016).

Wider networks can get features with more details and will be more convenient and better for later training (Zagoruyko & Ko-modakis, 2016). However, it is often difficult to learn the features in deep layers for networks with large width and small depth.

Increasing the resolution of the input image may potentially obtain more fine-grained feature combinations, but for very high input resolution, the gain of accuracy will decrease. Besides, large resolution images will cause heavy processing tasks.

The following figure shows the benchmark EfficientNetB-0 with increased width, depth, and resolution using this new scaling method. As can be seen from the below figures, Image top-1 accuracy tends to be saturated when it reaches 80%.

图形用户界面, 图表

描述已自动生成

Here’s another try, using different combinations of depth d and resolutuion r and constantly change the width of the network, the four curves as shown in the figure below are obtained. Through analysis, it can be found that under the same FLOPs, increasing d and r at the same time achieves the best result (Tan et al., 2019).

图形用户界面

描述已自动生成

### Compound Model Scaling

Here, EfficientNet use a new compound scaling method. This method uses a compound coefficient φ to uniformly scales those hyperparameters in a principled way, the specified formula is shown below:

文本, 信件

描述已自动生成

where s.t. stands for restrict condition and α,β,γ are constraints that can use a small grid search to determine these optimal hyperparameter values.

It should be noted that α, β, γ may not be same for different benchmark networks. It should also be noted that in the original paper (Tan et al., 2019), the authors also stated that searching for α, β, γ directly on larger model might get better results, but the cost of searching in the larger model is too high. Therefore, this article searches for the smaller EfficientNetB-0 models.

Use different φ can generate a family of EfficientNets name from EfficientNetB1 to EfficientNetB7

### EfficientNet Architecture

The network framework EfficientNet-B0 is shown in the following table, while B1-B7 modified the related three efficient network factors (number of channels, number of layers, and resolution) based on B0. The table below shows that the network has nine stages. The Stage 1 is a common convolution layer with a 3\*3 kernel and the stride is 2 (using BN and Swish as activation function). Stage2 to Stage8 are all using the MBConv structure. This structure is used in Mobile Net. Stage 2 repeat once and Stage3~8 used for 6 times, which means the first layer in MBConv will multiply 6 to the input matrix channel. Last layer has a Conv layer with kernel size1\*1 and an average pooling layer and a FC layer. This layer also use BN and Swish for activation function, which is used in MBConv Architecture.

表格

描述已自动生成

### MBConv Architecture

MBConv is similar to Inverted Residual Block in MobileNetV3, but different in activation function. Here Efficient Net adapted the Swish activation function, but MBConv used SE (Squeeze and Excitation) Module. Squeeze and Excitation Net or to say SE-Net can automatically obtain the importance of each channel in a wised choosing learning method. This model will also use this evaluation result to restrain some less important feature.

图表

描述已自动生成

EfficientNets did not use this activation function since the result of this method may miss choosing some features to be less important in deep layers and will influence the original three dimensions choosing.

Proposed AI-based ML System

As the development of the power of computation, scientists begin to optimize the structure of network and strike for higher accuracy, so there comes out many neural networks models which based on CNN. The knowledge of the field of image recognition is very large, so it is hard to tell in one chapter. Therefore, we just talk about some famous models based on CNN. CNN makes the task of image recognition easier, which has a special advantage that wherever one feature is in any position of this image it will not influence the recognition process. In 2012, the error rate of AlexNet whose weight is 60M is 16.4%. Two years later, the error rate of ResNet is only 3.5%. We can see that the pace of this development is very quick.

In the latter chapter, we implement different kinds of models (which we have talked above) and used the data from ImageNet to train our model. We also used cifar-10 to train our data. Then we evaluate and analyze the results of these models by comparing their accuracy, loss, validation accuracy and validation loss. At last, we build an image recognition program based on VGG and EfficientNet. User can upload an image file and choose which model to run, and you can compare the results of these two models.

图形用户界面, 应用程序

描述已自动生成

Then, we start to analyze the results of different CNN models (LeNet, ResNet, AlexNet, VGG16, EfficientNet).

Experimental Results and Analysis

## LeNet

Cifar-10 is an encapsulated data set, which includes 10 categories of things. Let's take a look at the effect of cifar10 classification based on Lenet-5.

The structure of Lenet5:

表格

描述已自动生成

In this experiment, we set batch size=128, epoch=50. Here is the visualization results:

图表

描述已自动生成

Since it is the earliest convolutional neural network, it has no very high accuracy rate, only about 60%. We can clearly see that the overfitting of the model is quite serious. It is normal to have such deficiencies.

电脑屏幕的照片

中度可信度描述已自动生成

What’s more, the accuracy of epochs has not been improved since it was increased from 50 to 180. This proves that simply increasing the number of epochs cannot improve the accuracy of the model.

Compared with VGG, the lenet5 with 40 epochs can achieve the effect of VGG with 10 epochs. But validation still is bad.

## B．AlexNet

We implement AlexNet model classification on CIFAR-10.

Here is the model structure of AlexNet. We construct this model using “TensorFlow Keras” and following the architecture mentioned in the reference paper [*ImageNet Classification with Deep Convolutional Neural Networks*](http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf) *.*

There are 8 layers with weight including 5 convolutional layers and 3 fully-connected layers.

表格

描述已自动生成

As you can see in the picture of the model structure. In this model, Relu is the activation function used in every convolutional layer. There are two local response normalization layers, one is behind the first convolutional layer, and another is behind the second convolutional layer. There are three MaxPooling layers in the model, and two MaxPooling layers are behind the first convolutional layer and the second convolutional layer following the local response normalization layer. The last MaxPooling layer is behind the fifth convolutional layer. In the end, we use dropout which are in the fully-connected layers to reduce the overfitting. Since the CIFAR-10 dataset has only 10 class, we add the last softmax layer with 10 outputs.

Here is the code of every layer in the model:

表格

中度可信度描述已自动生成

After fitting the model, we train the CIFAR-10 datasets with 10 epochs and 64 batch size. The following graph shows the accuracy and the loss of the training and validation.

图表, 折线图

描述已自动生成

From the diagram, we found that as the epochs getting bigger (the number of parameters increasing), the accuracy become higher and reaches 55%, while loss become smaller.

Since we want to compare the performance of different models, we unify the epochs=10. So, the maximal accuracy of Alex model here is only around 55%. But if we set the epochs to over 300, we can get close to 100%.

We also implement AlexNet by modify every layer in class.

文本

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

文本

描述已自动生成

## C．VGGNet

Here we construct VGG16 model by using Keras package. We define the layers of VGG16 and add them to the model. And here is the information of the model we constructed.

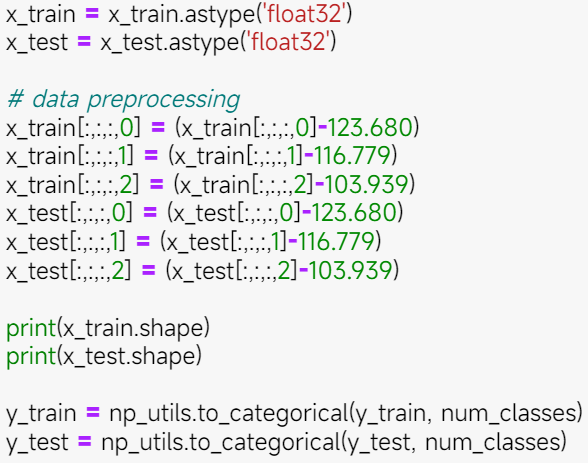
表格

描述已自动生成

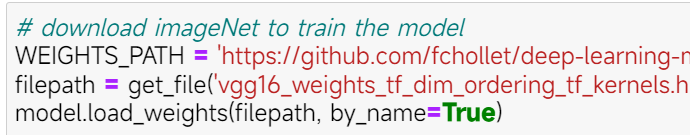
We can see that the total number of parameters is 33,638,218. We change the shape of output layer from 1000 to 10, because we need to fit the cifar-10 datasets.

After we constructed this model, we put the cifar-10 datasets into this model to do the training.

First, we do the data preprocessing to fit the input shape of this model.



And then we use the weight from imageNet to train this model.



We set the optimizer as SGD and set the learning rate as 0.01. Then we can use cifar-10 to train the model.

文本

描述已自动生成

Then we can plot this result.

图形用户界面

描述已自动生成

We can find that the more epochs the model are, the higher the training accuracy is, but the validation may decrease since it may overfit. After 10 epochs training, the accuracy of this model can be more than 80%. If we do more training, the accuracy may even higher than 90%.

We also design a VGG19 model, which has more layers and parameters.

图片包含 表格

描述已自动生成

And the training accuracy is better than VGG16, but the validation accuracy is lower than VGG16. It seems that the model which has more layers is easier to overfit.

图表, 折线图

描述已自动生成

## D．ResNet

In this section, we implement Resnet model classification on CIFAR-10.

Here we construct Resnet base on Keras Applications. Here for the experiment, we do classification on 10-Class dataset CIFAR-10 and show model Resnet parameter numbers and training accuracy and loss with validation accuracy and validation loss.

Since CIFAR-10 dataset has only 10 classes, we add a softmax layer as the last layer with 10 outputs.

图片包含 图表

描述已自动生成图片包含 图表

描述已自动生成

The first picture is Resnet 1, and the second model is Resnet 2. According to the above pictures, we can know that the second one’s accuracy trend is higher than the first one. And the loss trend is smaller than the first one. As a result, Resnet 2 is much better than Resnet 1.

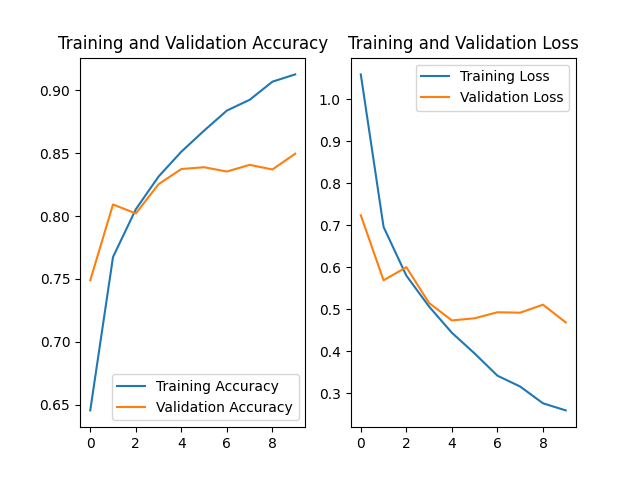
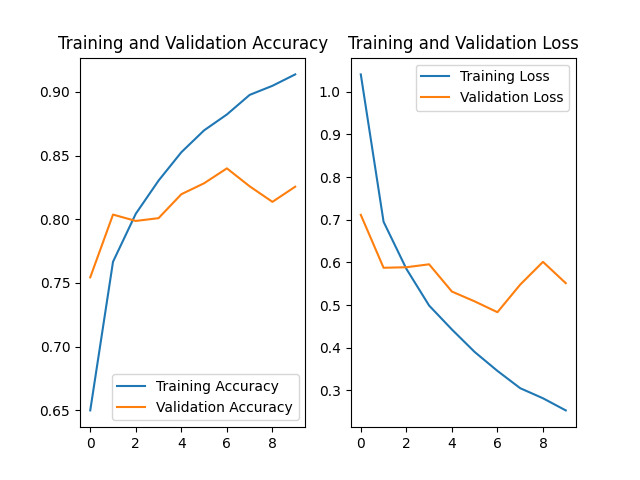
## E．EfficientNet

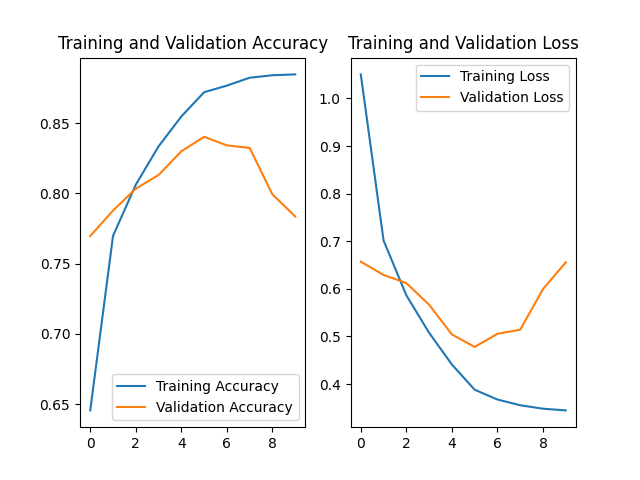
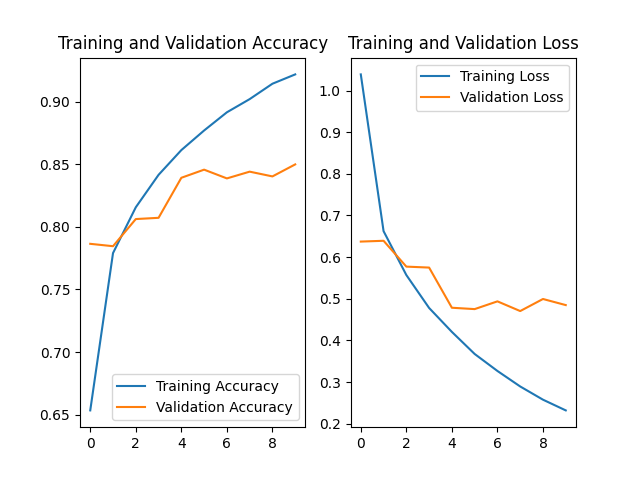
In this section, we implement Efficient Net model classification on CIFAR-10.

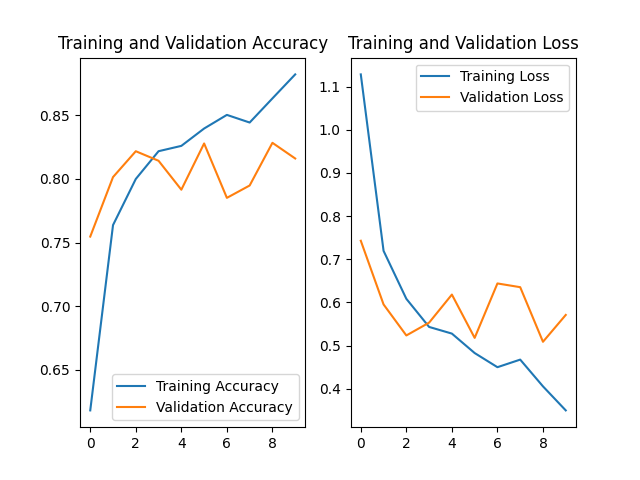
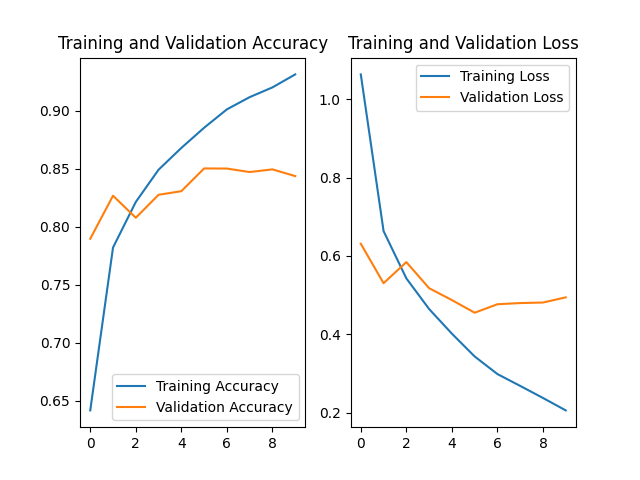
1.1 EfficientNetB0-B7 Classification on CIFAR-10

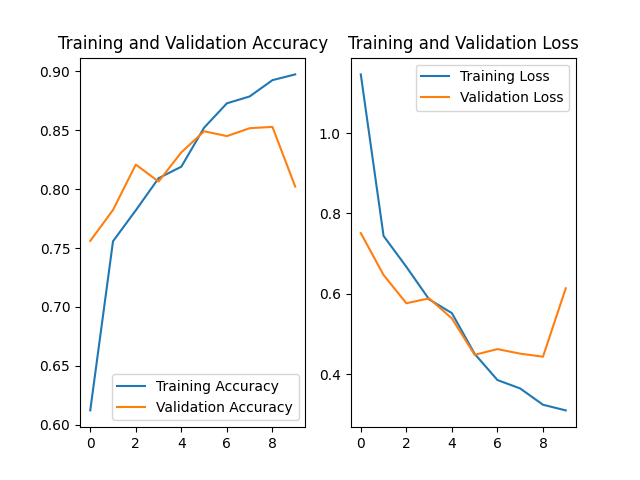
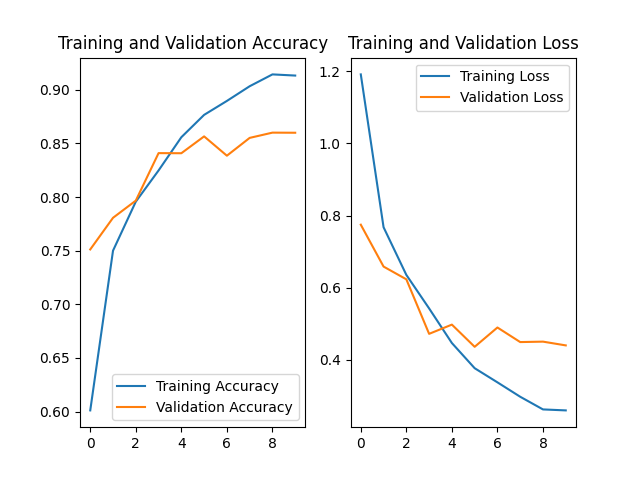
Here we construct Efficient Net model families base on Keras Applications. The used function reference paper is [EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks](https://arxiv.org/abs/1905.11946) (ICML 2019). Keras implement function returns this image classification model, with changeable weights and hyper-parameters. In this EfficientNet Keras Application model, input preprocessing is included as part of the model, which means the input of model already has the probability of preprocessing. Here for experiment, we use pre-trained weights on ImageNet and then do classification on 10-Class dataset CIFAR-10 and do show model EfficientNetB0-B7 parameter numbers and training accuracy and loss with validation accuracy and validation loss.

Since CIFAR-10 dataset has only 10 classes, we add a softmax layer as last layer with 10 outputs.









From these diagrams, we can see that as parameters become more and more, accuracy may not become better within same epochs.

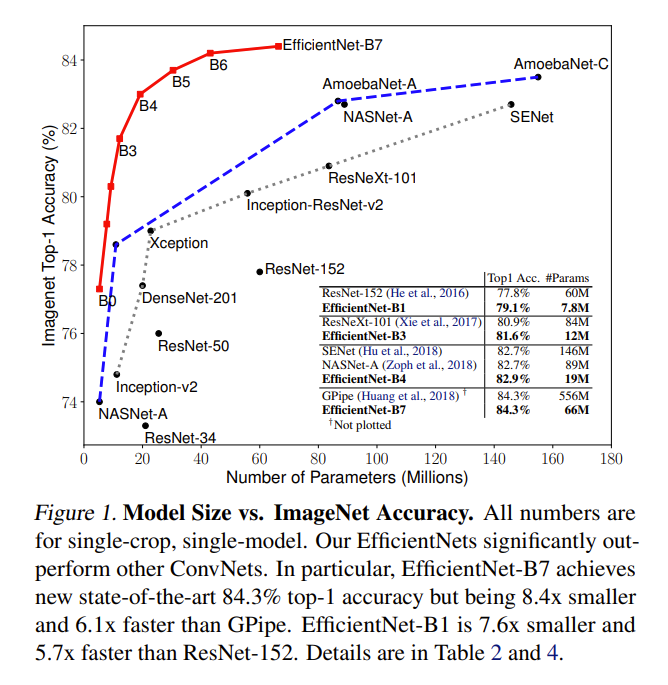
Table below shows each model training parameter numbers and total number:

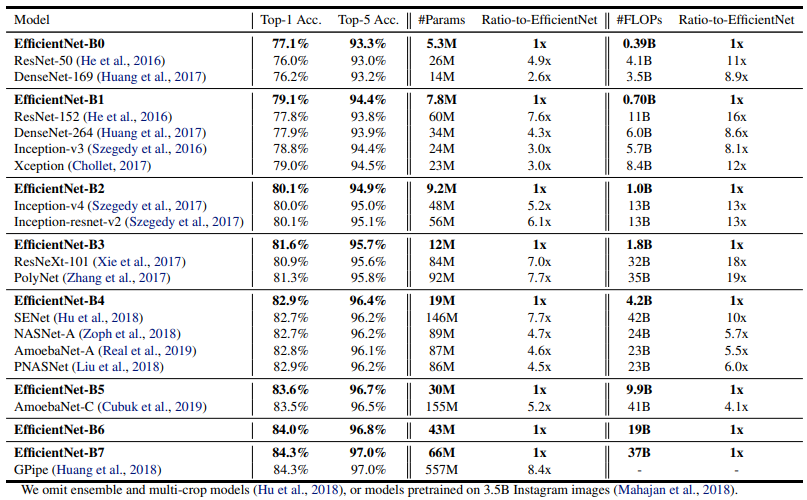
手机屏幕截图

描述已自动生成

From B0 to B7, the number of feature maps increased, which also cause the number of parameters increase. But compare to previous models, number of parameters in EfficientNet family is quite low.

Here, I give the table from original paper (Tan et al., 2019), showing that EfficientNet families use less parameters but provide better results.

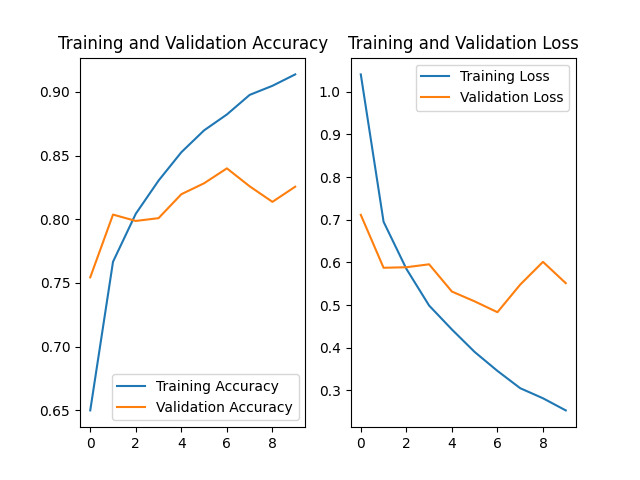




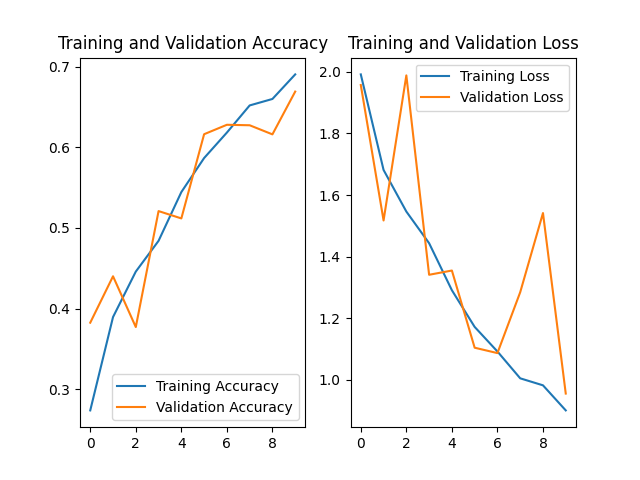
1.2 EfficientNetB0 with pretrained weights

Here we use EfficientNetB0 with pre-trained weights from ImageNet and without initial weight parameters to train on CIFAR-10 dataset.

1. ImageNet weights



(b) No pre-trained weights



As shown in (a) and (b), pre-trained weights can result in huge improvement in accuracy and loss. With 10 epochs, model train by random initialization can only reach an accuracy of nearly 70%, which is quite lower than pre-trained weights. Besides, loss variate a lot if no pre-trained weights given.

## Image Recognition Program

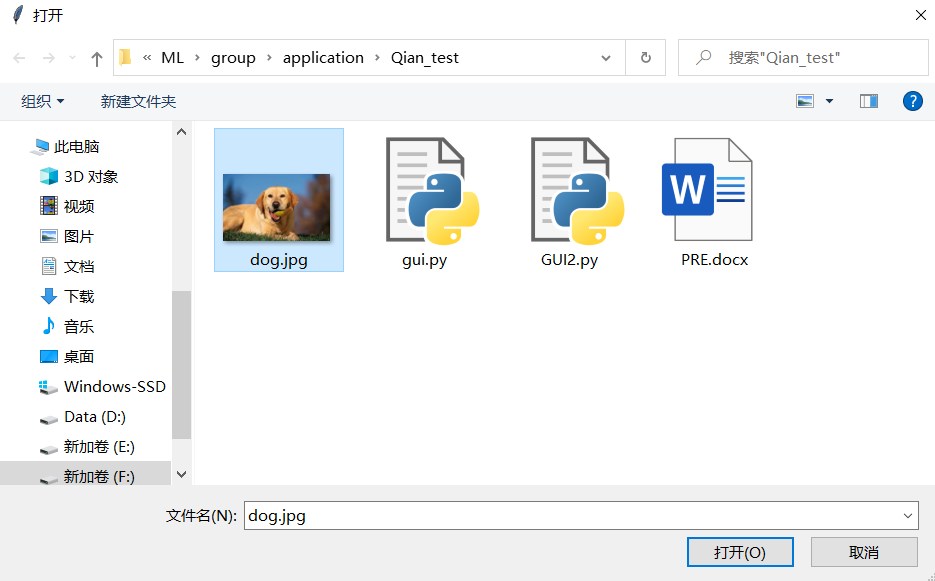
According to the network structure above, we selected VGG and EfficientNet to build an image recognition program, since their accuracy and efficiency are high. This is the interface we built.

图形用户界面, 应用程序

描述已自动生成

User can push the “Load Image” button to upload an image, and then they can select these two CNN models to predict what this image is.

As an example, we select a dog image.



For “VGG16 Prediction”, the image is recognized as golden retriever, which probability is 0.52.

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

For “EfficientNet Prediction”, the image is recognized as labrador retriever, which probability is 0.299.

图形用户界面, 应用程序

描述已自动生成

We can see that, the accuracy of EfficientNet is higher than that of VGG16. This phenomenon is the same as we analyzed above.

Summary

In this paper, we mainly discussed some basic structures of some traditional CNN models. We introduce some basic neural structures with some common layers. Beginning with the fundamental model (LeNet) to the advanced model that scientists developed (EfficientNet), we analyzed these models in detail, including the network structures, the performances and efficiency and so on. Then we do image classification with 10 epochs on CIFAR-10 dataset using different models and compare training loss and accuracy within similar type of models. For further application, we design a simple GUI application that use two models to classify an arbitrary image with classified types. For future transfer learning usage, these models can be well tuned to fit different dadtasets.

References

1. Boureau Y L, Bach F, LeCun Y, et al. (2010). Learning mid-level features for recognition[J].
2. Ciregan, D., Meier, U., &amp; Schmidhuber, J. (2012, June). Multi-column deep neural networks for image classification. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3642-3649). IEEE.
3. Frossard, D. (2016, June 17). VGG in TensorFlow: Model and Pre-trained Parameters for VGG16 in TensorFlow. Toronto. https://www.cs.toronto.edu/~frossard/post/vgg16/#introduction
4. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. CVPR, pp. 770–778
5. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861,
6. Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. CVPR
7. HU J., SHEN L, ALBANIA Set al. (2020). Squeeze-and-Excitation Networks [J].IEEE Transactions on Pattern Analysis and Machine Intelligence，42（8）：2011-2023.
8. Huang, Y., Cheng, Y., Chen, D., Lee, H., Ngiam, J., Le, Q. V., and Chen, Z. (2018). Gpipe: Efficient training of giant neural networks using pipeline parallelism. arXiv:1808.07233,
9. Hubel D. H., Wiesel T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex[J]. The Journal of Physiology, 160(1): 106-154.
10. Jiang, B. (2022). Image Classification Method Based on Improved ResNet Model. Modern Information Technology (12),83-85. doi: 10.19850/j.cnki.2096-4706.2022.012.021.
11. Kaiming He,Xiangyu Zhang,Shaoqing Ren & Jian Sun 0001.(2015).Deep Residual Learning for Image Recognition. CoRR.
12. Kaiming He,Xiangyu Zhang,Shaoqing Ren & Jian Sun 0001.(2016).Identity Mappings in Deep Residual Networks. CoRR.
13. Kim Bubryur,Natarajan Yuvaraj,Munisamy Shyamala Devi,Rajendran Aruna,Sri Preethaa K. R.,Lee DongEun & Wadhwa Gitanjali.(2022).Deep Learning Activation Layer-Based Wall Quality Recognition Using Conv2D ResNet Exponential Transfer Learning Model. Mathematics(23). doi:10.3390/MATH10234602.
14. Kornblith, S., Shlens, J., and Le, Q. V. (2019). Do better imagenet models transfer better? CVPR.
15. Krizhevsky A., Sutskever I., Hinton G. E. (2012). Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 1097-1105.
16. Krizhevsky, A., &amp; Sutskever, I., &amp; Hinton, G. (2017, June). ImageNet classification with deep convolutional neural networks. Communications of the ACM. 60(6), 84–90. <https://doi.org/10.1145/3065386>
17. Krizhevsky, A., &amp; Hinton, G. (2010). Convolutional deep belief networks on cifar-10. Unpublished manuscript, 40(7), 1-9. https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf
18. LeCun Y., Bottou L, Bengio Y, et al. (1998). Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 86(11): 2278-2324.
19. LeCun, Y., Kavukcuoglu, K., &amp; Farabet, C. (2010, May). Convolutional networks and applications in vision. In Proceedings of 2010 IEEE international symposium on circuits and systems (pp. 253-256). IEEE.
20. Lee, H., Grosse, R., Ranganath, R., &amp; Ng, A. Y. (2009, June). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th annual international conference on machine learning (pp. 609-616). https://doi.org/10.1145/1553374.1553453
21. Panda Manoj Kumar, Sharma Akhilesh, Bajpai Vatsalya, Subudhi Badri Narayan, Thangaraj Veerakumar & Jakhetiya Vinit.(2022).Encoder and decoder network with ResNet-50 and global average feature pooling for local change detection. Computer Vision and Image Understanding. doi:10.1016/J.CVIU.2022.103501.
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C.(2018). Mobilenetv2: Inverted residuals and linear bottlenecks. CVPR.
23. Simonyan K, Zisserman A. (2014). Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556.
24. Simard, P. Y., Steinkraus, D., &amp; Platt, J. C. (2003, August). Best practices for convolutional neural networks applied to visual document analysis. In Icdar (Vol. 3, No. 2003).
25. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... &amp; Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
26. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. (2019). MnasNet: Platform-aware neural architecture search for mobile. CVPR.
27. Tan, M., & Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv, abs/1905.11946*.
28. WANG Q. L., WU B. G., ZHU P. F. et al. (2020). ECA-Net： Efficient Channel Attention for Deep Convolutional Neural Networks [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition （CVPR）. Seattle： IEEE，11531-11539.
29. Werbos, P. (1974). Beyond regression:" new tools for prediction and analysis in the behavioral sciences. Ph. D. dissertation, Harvard University.
30. WOO S，PARK J，LEE J Y. et al. (2022,Mar) CBAM: Convolutional Block Attention Module [J/OL].arXiv:1807.06521[cs.CV].https://arxiv.org/abs/1807.06521.
31. Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. BMVC.
32. Zhou Feiyan, Jin Linpeng, Dong Jun (2017). A survey of convolutional neural networks [J]. Journal of Computer Science, 40(6): 1229-1251.
33. Zoph, B. and Le, Q. V. (2017). Neural architecture search with reinforcement learning. ICLR.