1. Introduction
   1. Project description:

The intent of this project is to develop a mobile application to recognize the playing cards, the application should have the ability to take a photo of playing cards, then process the photo to segment the location of the cards and recognize these cards. To achieve this goal, the system trying to use deep learning technology.

Deep learning is a very powerful and popular approach in machine learning, in recent year, deep learning gets dramatic outcomes in many fields like image recognition, image segmentation, image synthesis, voice recognition, voice synthesis, language translate, memory, and automatic reasoning. For computer vision filed, convolutional neural network is widely used in deep learning, and in the last 4 years, a lot of convolutional neural networks has been researched, which get exciting outcome on some open data set like ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), all these network architectures get high accuracy on the challenge and push forward the progress of artificial intelligence and machine learning. After these years’ research, the application of deep learning in image recognition has been mature, get high performance no matter evaluates on accuracy, speed or memory consumption, and because the contribution of graphics processing unit (GPU), training complex model is possible and simplified model can be trained on a single machine.

Playing card recognition and segmentation is a task of computer vision, before wildly using deep learning to achieve image recognition, traditional computer vision method is using HOG or SIFT to extract the feature of image and for specific task design corresponding rules to find target in the image. The traditional computer vision method for playing card recognition is first using SIFT to extract features, and then try to find the four corner of each playing card to locate the card, then find top left corner and locate the label of the card, then using traditional machine learning method like support vector machine to classify the label.

In this paper will introduce a panoramic view of building a mobile app with deep learning technology to segment and recognize the playing cards, there are four sections of building this app, first section is talking about the system that used to collect data set, second part is discussing several convolutional neural network architectures used to recognize and segment playing cards, after that the server part for app will be discussed in the third part, the final part will detailed how the client part for app been developed. After discuss the implementation part the paper will going to evaluate how these convolutional neural networks worked on data and in real world. Before all these discuss, in related work will introduce some basic concept of knowledge like Artificial neural network and convolutional neural network.

1.2 Aims and Objectives:

1. App to recognize the playing card:

1. The app can take a photo of a card (or cards).
2. The app can send photo to server.
3. The app can receive and show the segmented and recognized image of the card.

2. Build a back-end to run the main classification program:

1. Using Python and web.py framework to handle HTTP request.
2. Using Python to segment and classify photo.
3. Write the image to disk and return the file path to client.

3. Build a back-end to collect data set:

1. Using PHP to handle the HTTP request.
2. Using PHP to invoke MATLAB to generate train/validation/test set.
3. Using Python script to create segment train/test set.

3. Build CNN system on Python to classify card:

1. recognition:

1. Build a prototype which can classify less than 5 different type cards.
2. Expand the prototype to 12 different type cards.

2. segmentation:

1. Build a prototype which can segment two non-coincide cards.
2. Build a prototype which can segment two coincide cards.

2. Related work

2.1 Traditional method:

There is detailed introducing the traditional method to find playing cards in [1], Because of the white card lies on the dark table, the deviation of the edge is intensive, So the algorithm using Canny detector to do edge detection, and then the algorithm find the trapezoids which corner’s inner angles close to 90 degrees in the edge map, the trapezoids are seen as the card and throw away the remaining. After find the trapezoids and corners, using corners to locate the card position and rotate the card to the vertical. Extracting the rank region is the next things to do, then remove noise, using Adaptive Threshold to binaries the image and isolate the label. Then the last thing to do is match the label with pre-defined pattern to classify the label. In [2] and [3] the pipeline is the same with [1], all these classifications are using traditional computer vision technology and creating many rules from prior knowledge to find the card.

2.2 Artificial neural network

Artificial neural network is a mathematical model that mimics human brain structure, this model has the ability to learn the regular pattern from data feed to it. Artificial neural network will be introduced by representation, evaluation and optimization three part.

2.2.1 Representation of artificial neural network.

In fact, the architecture of artificial neural network defines the maximum learning ability of the architecture, in other words, it defines how much knowledge the network can learn. A simple artificial neural network is shown in figure 1, every circle in the figure represents a normal neuron, blue circle represents the bias neuron, every arrow in the figure represents a weight connect two neural, and there are three layers in this network, first is input layer, second is hidden layer, the last layer is output layer, in common there is only one input layer and one output layer, every neuron connect with all normal neuron in the next layer, this connect way called fully connected, fully connected is the most basic, time-consuming, memory consuming but powerful connected way in artificial neural network. Forward calculation is simple, every neuron value except input layer’s neurons is computed by just add all weights pointed to that neuron together and then using an activation function, formula is shown in equation 1, using sigmoid function as activation function, the effect of activation function is allowing the network learn non-linear function. Sigmoid function is shown in equation 2, it can map the input into 0 to 1. That is representation of the artificial neural network, using this architecture with apposite weight the neural network can predict the class of input data (for classification) or some value of input data (for regression).

figure 1

figure 1

equation 1

equation 2

2.2.2 Evaluation of artificial neural network.

Similar to the process of human learning new knowledge, the neural network training process is to update the parameters of the network by telling the network the distance of its current output and the label. How to calculate the distance between the output and the label is a research topic, but there are two common ways to calculate the distance. The first one is seeing the network output as the point in the multi-dimensional space, and using Euclidean distance to calculate the distance. The cost function is shown in equation 3, where n represent the number of neurons in the output layer, and represent the i neuron in the output layer and target vector respectively. The next representation is seeing the network output and label as a probability distribution, using softmax function to map the output vector to satisfy the sum of entry equal to 1, softmax function is shown in equation 4, where n represent the number of neurons in the output layer, softmax function is the extension of sigmoid function. Then using cross entropy to calculate the distance between the two distributions, the formula is shown in equation 5, where is the i th label and is the i th network output neuron in the output layer.

equation 3

equation 4

equation 5

2.2.3 Optimization of artificial neural network.

Training network is similar with human study new knowledge, firstly have a try, then check if the result is correct or incorrect, if incorrect, find where have some problem and change that, continue to try again. This is the roughly steps of iteration learning, in mathematics, there is a common method called gradient descent, which give a cost function, try to find the parameters that minimum the value of the cost function, gradient descent calculates partial derivative for every weight in the network and update the weight by corresponding partial derivative in some way, the formulary is shown in equation 6, where k indicates the number of iteration, represents the corresponding partial derivative respect to weight, is called learning rate, which is controlling the speed and the accurate of training the network, large will speed up the training process but will lead to less accurate of convergence even cannot convergence, small will slow the speed of training process but can guarantee convergence in local minimum.

equation 6

2.2.4 General View of Artificial neural network

Artificial neural networks can theoretically approximate function by set suitable weights, but using fully connection need large memory space and computable capability, for example, there is a network with 3 layers, 100\*100 neurons in input layer, 1024 neurons in hidden layers, 10 neurons in output layers, the total weights of the network is (100\*100+1) \*1024+(1024+1) \*10 = 10251274, calculate these weight with a number of iterations will consume quite a lot time, there are more layers in more powerful neural network which will cost more time and computable capability, large networks even cannot be computed, because these shortcomings artificial neural networks are rarely used in machine learning tasks.

2.3 Progress of convolutional neural network

The difficult part of traditional computer vision is how to extract feature from images because it depends on human, traditional method of feature extract from image depend on texture, shape and color property of the image like Histogram of Oriented Gradient and Scale-invariant feature transform, these two methods are widely used to extract features, and these two methods are human beings over the years based on experience and mathematical derivation from the summary, they have good results but cannot contain all aspects of image information. But it changed in 2012, Alex using convolutional neural network to win the champion of ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 [4], which means a new framework of computer vision is better than traditional methods, and make deep learning be the super tool for Image recognition. After 2012, people keep working on convolutional neural network and obtain dramatic outcome. Microsoft win champion of ILSVRC 2015, it still uses an improved convolutional neural network and the accuracy higher than human level.

2.3.1 Convolution operation

Convolution is a mathematic calculation factor, it gives a kernel matrix, walk the kernel through the whole image with specific strides, for every step calculate formula equation 7, where x is the value of corresponding image region. After walk through the whole image there is a new image, a visualization process is shown in figure 2 [5] . The meaning of convolution operation is weighted summation of a region, this calculation method is used a lot in image processing. If there are multiple input layers, convolution operation using same kernel on all these layers and add them together, for each kernel there are only one output layer for that kernel.

equation 7

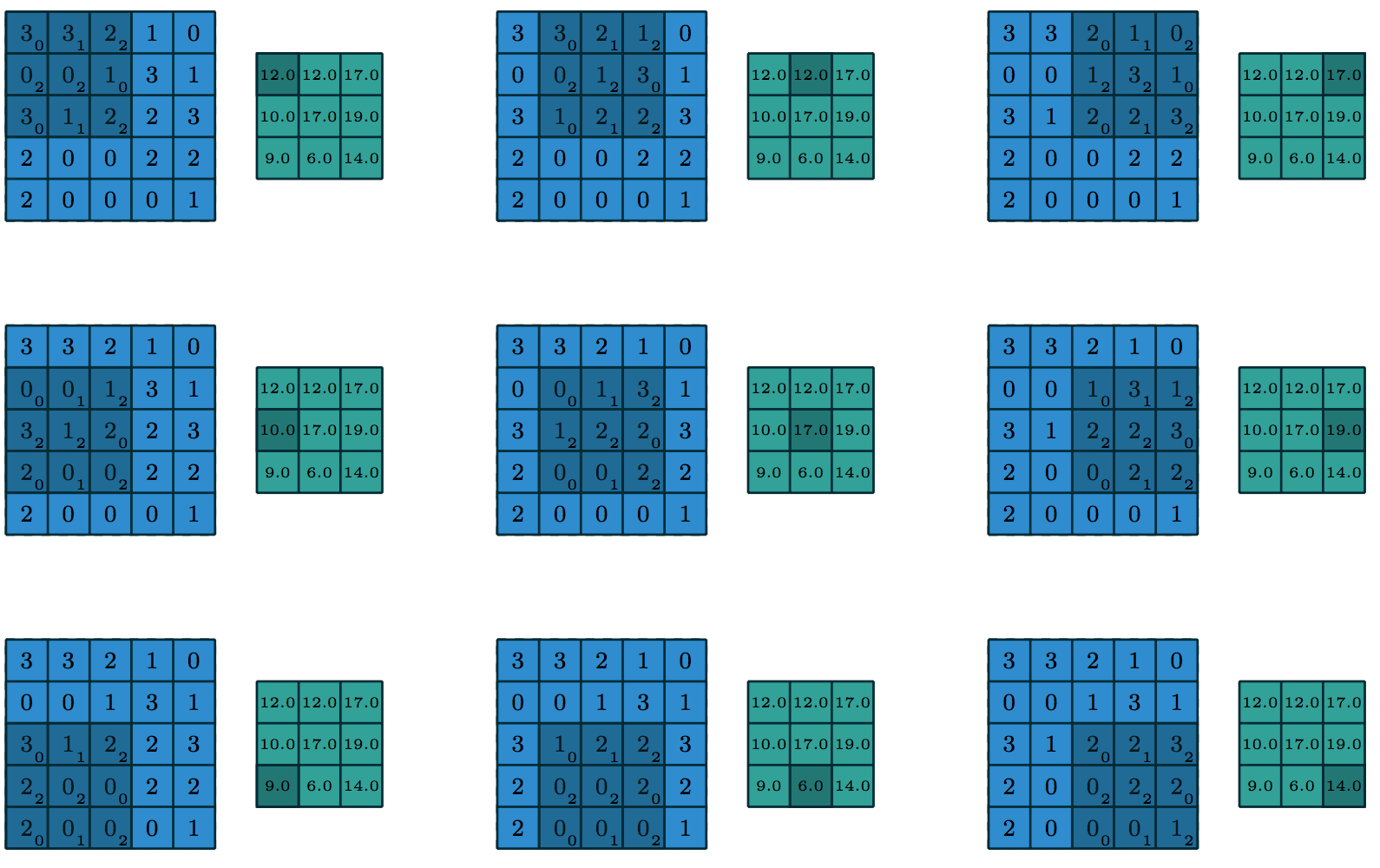
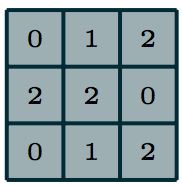


figure 2

gray matrix in left figure is the kernel matrix, blue matrix in right figure is the original image, green matrix is the new image

2.3.2 Pooling

The basic artificial neural network architecture requires a lot of weights connection, which makes training several layers’ network hard and painful, it also requires a great mass of data to train the network because there are so many variables need be trained. To avoid these disadvantage, convolutional neural network introduced a down sampling method called pooing, this process significantly reduces the needed weights of the feature maps by using some functions to summarize the information of a region, for example using the maximum value in the region to represent the whole region. Another function of pooling is making an assumption that the invariance property of images, a average pooling is shown in figure 3 [5].

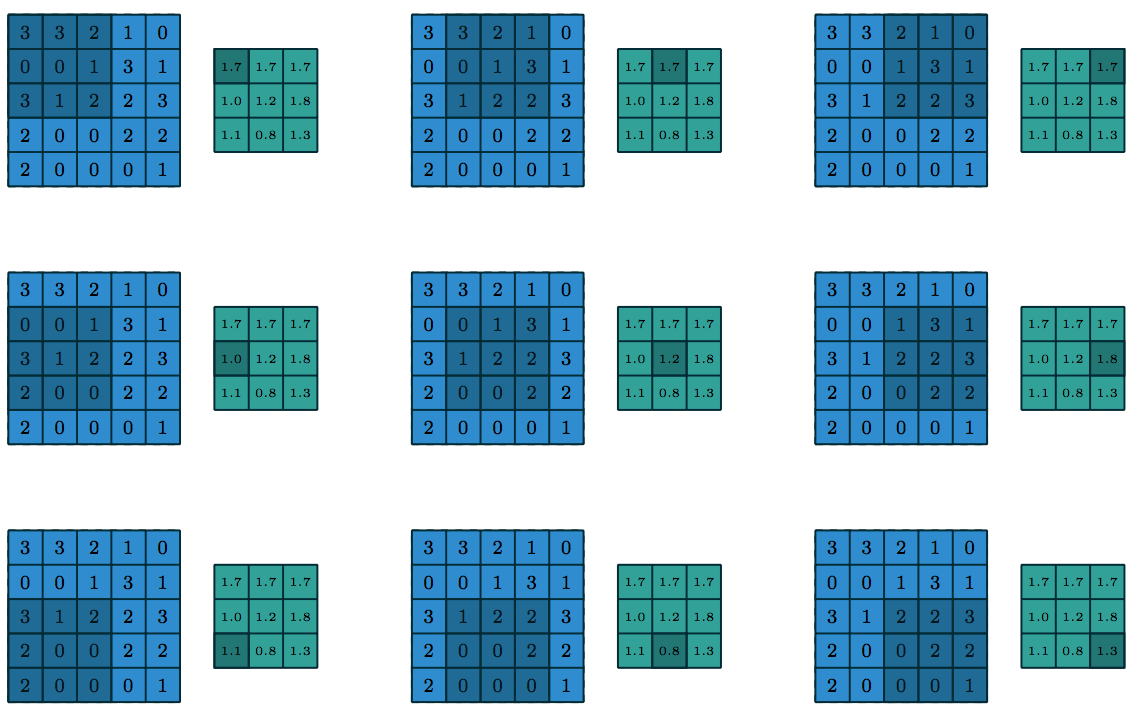


figure 3

2.3.3 Rectified linear unit (ReLU)

Activation function is an important part of deep neural network, the choice of activation function influences the speed and quality of training the deep neural network. In [4], give the definition of ReLU, the formula is shown in equation 8. In [4], the researchers test tanh activation function and ReLU on the CIFAR-10 dataset for a particular four-layer convolutional network. The require iterations for reach 25% training error rate on the two activation functions shows in figure 4[4], as the figure shows, the epochs need by ReLU is significantly less than tanh activation function. The reason of this situation is that tanh and sigmoid activation function are saturating nonlinearities, which means when the input of these function is very small or very large, the derivative at these points are very close to 0. This is demonstrated in figure 5, where the red line is the curve of sigmoid activation function and blue line is the curve of tanh activation function, it is clear that when the input large than 4 or smaller than -4, the derivative of both two functions is very close to 0, this property will lead to error very close to 0 during the optimization process using gradient descent, and this is the reason of why multiple layer neural networks using saturating nonlinearities activation function are slow and hard to training. ReLU is a non-saturating nonlinearities function, the yellow line in figure 5 is the curve of ReLU, when input is larger than 0, the derivative of ReLU is a constant 1, so the speed of using ReLU is significantly faster than using saturating nonlinearities, and this is the reason of why ReLU is widely used in deep learning.

equation 8

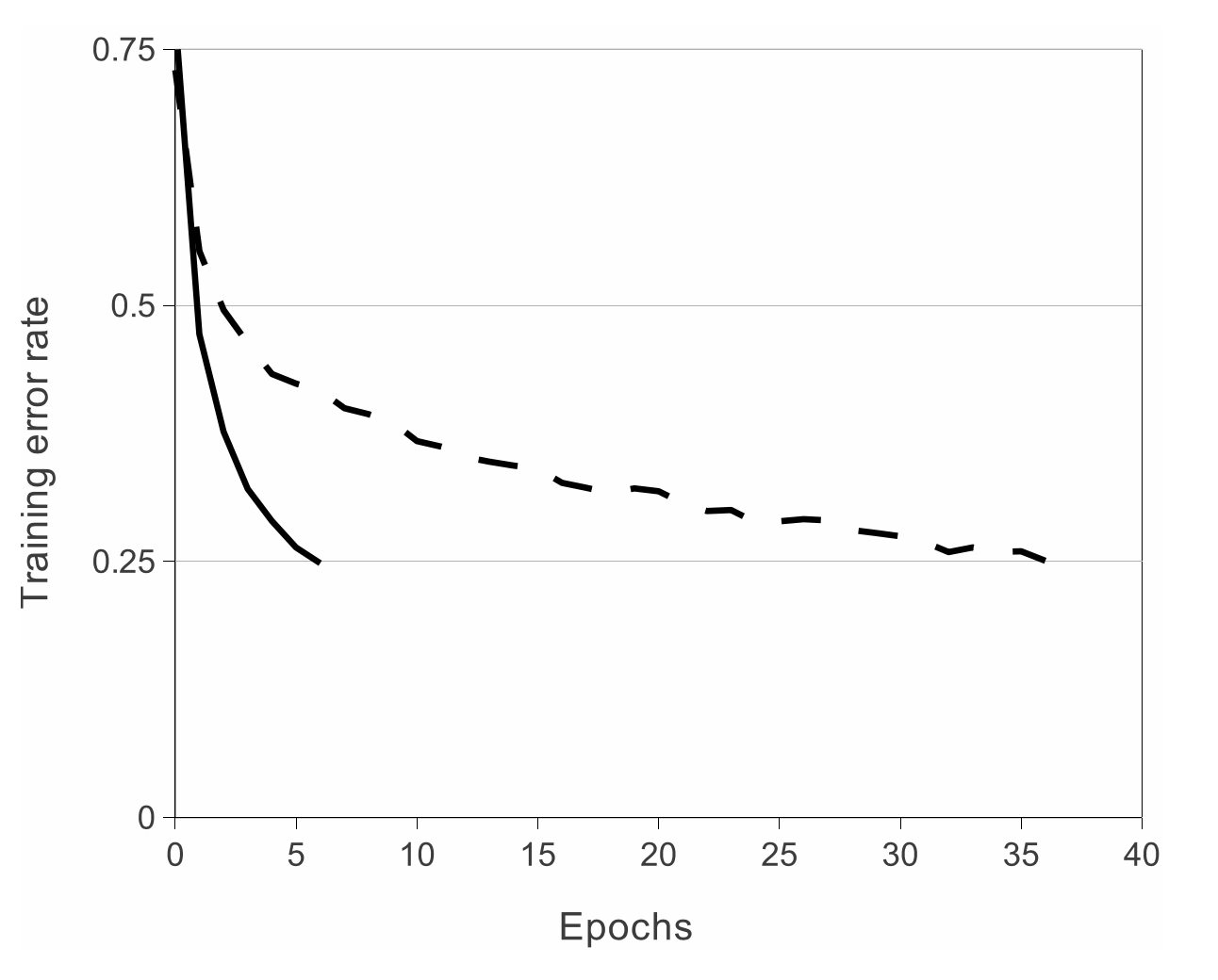


figure 4

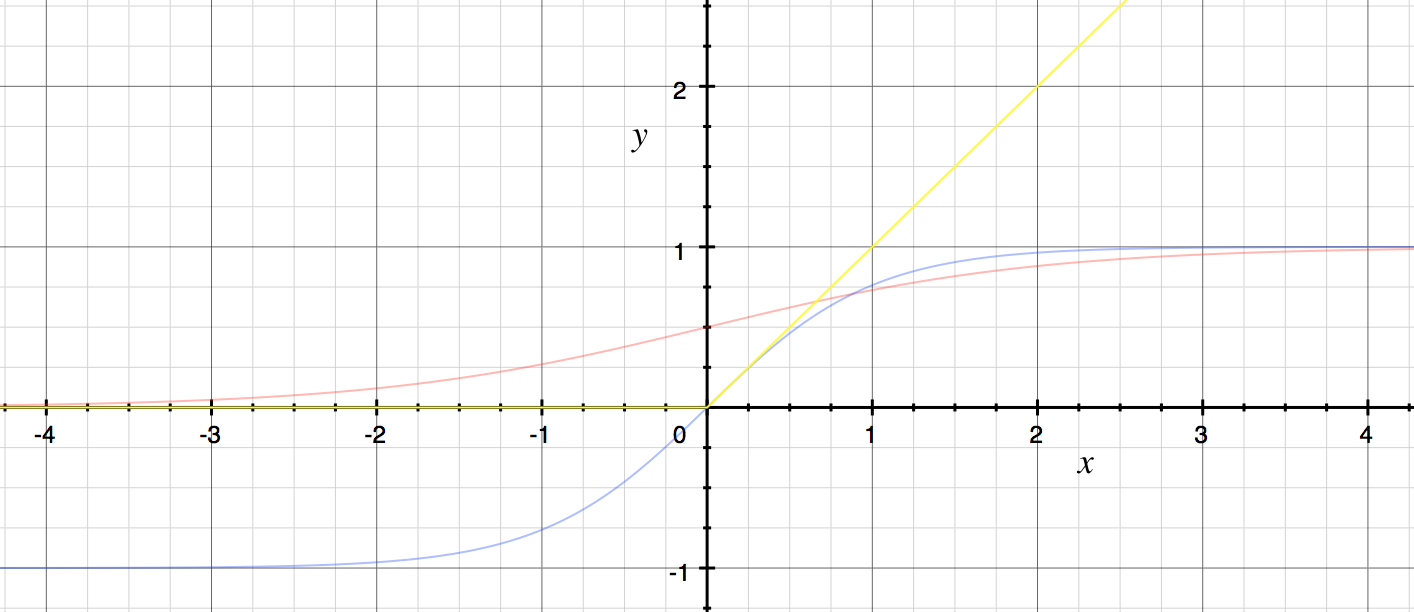


figure 5

2.3.4 Convolutional neural network

Convolutional neural network is a stack of learnable convolution layers which followed by a ReLU activation layer and connected to fully connect layers. A typically convolutional neural network framework is shown in figure 6, there are two convolutional layer in this framework, in the first convolutional layer conv1, the input layer has three feature layers(RGB), using convolution operation with 64 kernels on input layer then produce 64 new feature maps, after that, using max pooling produce a \* \* 64 feature maps. These are the basic framework of convolutional neural network. Training these learnable convolution kernels lead the convolutional layers obtain the ability to extract feature from images, then the fully connected layers obtain the ability to classify classes.



figure 6

3. Description of the work and Design

3.1 Project goal

The intent of this project is to develop a mobile application to recognize the playing cards, the application should have the ability to take a photo of playing cards, then process the photo to segment the location of the cards and recognize these cards. To achieve this goal, the system trying to use deep learning technology.

There are 2 sub problems need to be solved, the first problem is that the project need to develop a mobile application, which can take photo, process the photo and show the result. Second problem is need to obtain the trained deep neural network.

3.2 General idea of the approach

3.2.1 General idea for solving the mobile application sub problem.

To develop a mobile application, the first consideration is choosing which mobile application operating system to base on. There are two main candidates, iOS and Android. Because Android is an open source and free operating system, then Android has been chosen to develop the mobile application. The second consideration is architecture of the application, there is still two main candidates, the first candidate is put all calculation on the mobile, the second one is dividing the whole work into front-end and back-end, front-end is the application client, back-end is built on the server. The benefit of the first architecture is simple, because all the things are built on the mobile, do not need to design API and write server code, and the first architecture can work offline, easy to use, the problem of it is that the calculation is huge for mobile, it will take a really long time to compute the result and causing the phone to heat up. The advantage of second architecture is that it only uses the mobile to collect and show the information, the server to do calculation task, this design lead save time to do calculation and will not causing phone to heat up, the disadvantage of it is need internet and more complicate. Taking into account the advantages and disadvantages above, the second candidate architecture has been chosen.

3.2.2 General idea for obtain the trained deep neural network.

The basis of all machine learning algorithms is data, data decide what problem the machine learning algorithm can solve and directly influence the performance of the machine learning algorithm. Then the project need to obtain data for solve the problem, playing cards data are obtained in two methods, one is cut the video that recorded a single playing card into several images, another one is directly take playing card photos, first method can generate a lot of data by one video, so this method used mainly to generate recognition training data, the second method used to obtain validation, test and segmentation data. After get images data, there are more preprocess things need to do, for segmentation task, need to use the original photo to label mask and use original photo and mask to generate more images which some contain playing cards, some not contain.

After get data set, another important part is build the model, because train a model requires powerful computing ability, the system used to train the model need GPU resource to speed up the process of training. As introduced in section 2.2.3, training an artificial neural network need to calculate the derivative of each weight in the network, it’s hard to code a complicate neural network manually, then the system need have the ability to calculate derivative automatically. After trained the model, need to store the model into disk and can be load by other programs.

3.3 Functional specification

In this section will list all specific functions the whole system has, which can be used as a test part in evaluation section.

1. Mobile application:
   1. The application is based on Android mobile operating system.
   2. The application should have a text input box to allow the user to input the server IP address.
   3. The application should have an ability to take a photo.
   4. The application should have a box to show the photo which has been taken.
   5. The application should have the ability to post photo to the server.
   6. The application should have the ability to receive the image responded from server.
   7. The application should have the ability to show the image received.
2. Server for mobile application:
   1. The server should be built to waiting for the request from the mobile application.
   2. The server should offer an API to the mobile application to allow it post a photo.
   3. The server should have the ability to receive the photo.
   4. The server should have the ability to load trained model.
      1. Load two different model for two different purposes, one is segmentation and the other is recognition.
   5. The server should have the ability to find each playing card location in the photo.
   6. The server should have the ability to segment the playing cards in photo.
   7. The server should have the ability to recognize the playing cards in the photo.
   8. The server should have the ability to add text and mask on the photo.
   9. The server should have the ability to send the processed card back to the mobile application.
3. System for data management:
   1. The system should have the ability to convert a data into a specified data type.
   2. The system should have the ability to cut the video into several pictures according to the data set settings.
      1. Cut the video into train/validation/test set. Cut the video by time, i.e. The distribution ratio of the three data sets is 7:1:2, video before the of the time be cut into training set, to be cut into validation set, to be cut into test set.
      2. Cut the whole video into train set.
      3. Cut the whole video into validation set.
      4. Cut the whole video into test set.
   3. The system should have the ability to store these data set.
   4. The system should have the ability to cut a whole image into several small region and indicate if there is card in that region and generate a mask for that small region.
   5. The system should randomly fetch the data from the specified dataset.
   6. The system should find the left top corner of the card in the photo.
4. System for training deep neural network:
   1. The system should have the ability to build neural network architecture.
   2. The system should have the ability to use GPU resource.
   3. The system should have the ability to calculate derivative automatically.
   4. The system should have the ability to train the model.
   5. The system should have the ability to show the result of the training process.
   6. The system should have the ability to test the performance of model.
   7. The system should have the ability to store the trained model.

Reference:

1. P. Martins, L. Reis, L. Teófilo: Poker vision: playing cards and chips identification based on image processing. IbPRIA'11 Proceedings of the 5th Iberian conference on Pattern recognition and image analysis, pp. 436–443, Springer-Verlag Berlin, Heidelberg, 2011.
2. Zheng C, Green R (December 2007) Playing Card Recognition Using Rotational In-  variant Template Matching. University of Canterbury, Christchurch, New Zealand)
3. K. Zutis and J. Hoey, ‘‘Who’s Counting? Real-Time Blackjack Monitoring for Card Counting Detection,’’ in Computer Vision Systems: 7th International Conference on Computer Vision Systems, ICVS 2009, Li`ege, Belgium, October 13-15, p. 354, 2009.
4. A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, pages 1106–1114, 2012.
5. “ArXiv.org Stat ArXiv:1603.07285.” [1603.07285] A Guide to Convolution Arithmetic for Deep Learning. N.p., n.d. Web. 04 Dec. 2016.