**Comparison with others**

We have also used Alexnet and Googlenet for experiments. The experimental results are shown in Table 2. According to Table 2, Googlenet has the highest accuracy and Alexnet has the lowest accuracy. In terms of the number of parameters and the amount of calculation, our proposed network structure is significantly better than Alexnet and Googlenet. In general, the model we proposed is effective, which can achieve excellent classification performance, and

is more suitable for deployment on the mobile side.

rp1

Different sizes of memory cells were adopted to test the recognition accuracy, which

is shown in Fig. 1. LEBAI shows that the greater the size of memory cells, the higher

the recognition accuracy. The ultimate recognition accuracy achieved maximum

95.58 %.

The experiment selects randomly the 4000 learning item as the test set from the

data set, the other 16000 learning item as a training set. Table 1. show the

parameters.

Different sizes of memory cells were adopted to test the recognition accuracy, which

is shown in Fig. 1. LEBAI shows that the greater the size of memory cells, the higher

the recognition accuracy. The ultimate recognition accuracy achieved maximum

95.58 %.

Different sizes of training antigens were adopted to test the recognition accuracy,

which is shown in Fig.2 where sizes of memory cell are 14850, 10000, and 5200,

respectively. The experiments show that with the increase of training antigen, the

recognition accuracy is increased. However, when the size of memory cell is over

10000, the experiment results is stable to 95.87% because the convergence of the

memory cell

In order to prove that LEBAI improves the recognition accuracy, the comparison

of recognition accuracy is showed in Table.2, compared with FNNC, HSAC and etc

recognition algorithms. LEBAI shows that the recognition accuracy achieves the

highest recognition accuracy

rp2

Here in this paper handwritten character recognition using CNN is implemented and that method is evaluated based on Artificial Neural Network (ANN). CNN is a deep machine learning algorithm which doesn’t wants to extract the features manually. ANN is one of the existing system which also get implemented for comparing its performance with CNN. Both the CNN and ANN uses the same set of samples for training and testing.By running both the programs separately we get a result of accuracy out of 100 percentage. From that we can get a result of CNN is the best method than ANN. Because CNN have more accuracy and the features need not to be extracted. The algorithm itself do that procedure. **But in ANN the features extracted manually and have less accuracy compared to CNN.** For both ANN & CNN it uses training files of 45 samples per 62 folders and there are 10 samples for testing per folder. This mainly support English characters (both small and capital letters), digits and 26 special symbols.

The performance evaluation metrics used here are based on testing set as how many characters went wrong and correct out of total samples per folder. For example in this paper CNN uses 10 test samples per folder and there are 62 such folder. So 10 x 62= 620 is the no. of classification and based on that we are going to predict the percentage of accuracy out of 100. It also shows the counts of samples which went wrong and correct out of total samples plus the percentage out of 100.

Also included some 26 special characters to this in order to show the accuracy level. The special characters are: = > < - ? ” / \* \_ ! # $ % & ) ( ; , [ @ ] { } + . so there are 88 folders. That is 62 plus 26 special symbols. There is no standard dataset available for special symbols as result we need to create the dataset manually. So I created the training files consists of 5 samples per 26 folders and the test files consists of 2 samples per 26 samples. This is not actually enough for better accuracy. Because when we use more datasets the recognition accuracy is more.

By including special characters also the accuracy is predicted. So for that it uses 88 folders including 26 special symbols. This 26 special symbols have no any dataset is available and it is made manually by downloading the fonts of handwritten character. So for example in test file it includes 2 samples and in train files it includes 5 samples per 26 folders. So there are total of 2920 (i.e 62\*45+26\*5=2920) samples in 88 folder. From above two result it is very clear that the number of dataset is very much essential for good accuracy result. Here without using special symbols have much accuracy. If there are more number of datasets then the result is of good quality.

From the above Table: 1 it is very clear that English handwritten character recognition using Convolutional Neural Network (CNN) is having more accuracy than Artificial Neural Network (ANN).

rp3

To assess the performance of the proposed scheme, we train a

SHL-CNN model for both English and Chinese image character

recognition and conduct extensive experiments elaborated

as follows.

4.1. Databases

We choose the ICDAR 2003 character database for the English

recognition task, which a benchmark dataset used for the

robust reading task of the ICDAR 2003 competition. The objective

of the competition is to addressing text images that are

rather challenge to current commercial OCR packages. The

ICDAR 2003 character database is divided into three subsets:

a train subset (containing 6185 images), a test subset (containing

5430 images), and a sample subset (containing 854

images). These character images are of different sizes (e.g.,

5x12, 36x47, 206x223), fonts, colors, and present with varying

kinds of distortion. For simplicity, Yokobayashi[6] merge

the train and test set into a new train set, adopt the sample set

as the test set and select only number and alphabet for use.

Therefore, the new training set has 11492 samples including

6113 from original training set and 5379 from original test

set.

For Chinese image character recognition task, a news

video character database is collected from 38 different Chinese

TV news programs respectively as no public database

is available. Each of them is about 30-minute long and the

total time is about 50 hours. We have manually labeled text

appearing in these videos including its position and content,

eventually forming total 5036 text lines with 69340 characters.

These characters include superimposed text, scrolling

caption and a small quantity of scene text. These characters

are ranked according to their frequency, and based on the

ranking list, two character subsets, one formed by the top 500

categories and the other one formed by the next top 500 categories,

are obtained. The two subsets are named as TV-500-1

and TV-500-2, which contain 49480 and 9997 occurrences

of characters, respectively. Both subsets are divided into two

equal parts, one for training and the other for testing.

4.2. SHL-CNN V.S. Con-CNN

Experiments are conducted on the ICDAR-2003 and TV-500-

1 depicted before. The inputs to the conventional CNN (Con-

CNN) and SHL-CNN are both 48x48 RGB color image. For

ICDAR-2003 task, the output has 62 neurons including 10

numerals and 52 alphabets. For TV-500-1 task, 500 output

neurons are included. Since the sizes of local receptive fields

are sensitive for different tasks, three different models (listed

in Table 1 below) are used in the experiment in order to testify

the proposed scheme more comprehensively.

Table 2 compares the recognition error rate (RER) obtained

on the task specific test sets using the Con-CNN

(trained using only the data from that task) and the SHLCNN

(whose hidden layers are trained using data from both

two tasks). From the table we can observe that the SHL-CNN

outperforms the Con-CNN with a 16-30% relative RER reduction

(RR) across both the tasks for all models. Since the

model structure and training procedure are same between the

Con-CNN and SHL-CNN, we ascribe the gain of SHL-CNN

to cross-task knowledge. Additionally, since most of samples

come from superimposed text we can see that Chinese

character recognition task has a low RER.

4.3. SHLs for Unseen Characters

The SHLs extracted from the SHL-CNN can be considered as

an intelligent feature extraction module jointly trained with

data from multiple character recognition tasks. As such they

carry rich information to distinguish character classes in multiple

tasks and can be carried over to distinguish characters in

new tasks for image character recognition.

In this experiment, only the model behaving best performance

before (i.e.3x48x48-64C9-MP2-CN-64C9-CN-MP2-

64L7-32L7-500N) is adopted. The procedure of training a

model for new task TV-500-2 is simple. All SHLs from the

SHL-CNN are extracted and an untrained softmax layer with

500 nodes is added on top of them. During training process,

the weights of the SHLs keep intact and only the softmax

layer are updated using training data from training set of TV-

500-2. Using this method we achieve an RER of 3.6% on

the TV-500-2 test set. For comparison, the experiment of the

model randomly initialized weights without making use of

SHLs from SHL-CNN are also performed. The training process

is nothing different except that all weights are updated

using training data. We only achieve an RER of 6.3% on the

TV-500-2 test set.

According to the results, it indicates that the model leveraging

the SHLs extracted from SHL-CNN are more effective

than randomly initialized weights. An additional absolute

2.7% RER reduction has been observed by doing so.

4.4. Comparison with existing methods

In order to compare our system to the recent works in detail,

we test the seven subset of new test set of ICDAR-2003

OCR database mentioned before respectively. Fig. 3 shows

the results of our method compared with those of [6, 7, 1, 2].

The overall RER of our system reaches 9.03% compared to

14.04% of the best method [2] and it ranges from 32.14% for

seriously distorted images to 5.03% for clear images. Compared with the other methods, the performance of our system

behaves better especially in cases such as multi-color, uneven

light, little contrast and blurring. Since the large improvement

were made by cross-task training, we attribute it to cross-task

knowledge learned by proposed SHL-CNN, which may includes

more variations knowledge of character images.

rp4

In this section, before introducing mobile computing, let us

talk about the research status of the IoT and then briefly introduce

the principle of convolutional neural networks and their

application in OCR; finally, let us talk about the development

status of deep learning on mobile devices.

The Internet of Things is an emerging concept that is likely

to change our lives as much as the Internet has. In the Internet

of Things, IoT devices with smart features or objects can

communicate with each other. Each device will be given a

unique identifier that allows the device to access the Internet

and make decisions through calculations, thus creating a new

world of interconnected devices. These IoT applications are

sure to make a large difference in our lives. For example, in

the medical industry, HIT (Health Internet of Things) sensors

can be used to monitor important patient parameters (blood

pressure, heart rate, etc.), to record data to better respond to

emergencies, and to provide a better quality of life for people

with disabilities [10]. Another area of research based on the

Internet of Things is the IoUT (Underwater Internet of

Things), whose goal is to monitor a wide range of inaccessible

waters to explore and protect these areas [11]. The application

of interest in this article is OCR.

Optical character recognition is one of the litmus tests of

pattern recognition algorithms. In addition to pattern recognition,

optical character recognition involves some other fields

of knowledge, such as image processing, artificial intelligence

and linguistics [12–14]. Optical character recognition can

make computers handle real world texts directly. The research

results can be applied to a large number of practical problems,

such as mail splitting, check recognition, image retrieval, intelligent

robots and handwriting input. Commonly used

methods in character recognition can be divided into three

categories: template matching; feature extraction and classification;

and deep learning methods. In the template matching

method, a standard template set is first established for the

character to be recognized, and then the preprocessed images

of the character to be recognized are sequentially projected

onto the templates for matching. The best matching template

corresponds to the character being recognized. Feature extraction

and recognition are the most commonly used methods in

optical character recognition. The process consists of two

parts. **The first step uses an algorithm to extract the features**

**of the character image. The commonly used feature extraction**

**algorithms are: SIFT, SURF, and HOG**. The second step is to

use a classifier to classify the acquired features. Common

classifiers include the Bayesian classifier, support vector machine, K nearest neighbor algorithm, artificial neural network

algorithm, etc. [15, 16] Deep learning has become the

most commonly used method since its appearance. The most

commonly used deep learning model in the field of character

recognition is the convolutional neural networks.

A CNN is a feedforward neural network inspired by biological

processes. Unlike traditional network architectures, a

CNN contains very special convolution and down sampling

layers [17]. LeNet5 was born in 1994 and is one of the earliest

convolutional neural networks, which has promoted the development

of deep learning [18]. This network is mainly used

for handwritten character recognition. The features of the image

are distributed over the entire image. Convolution with

learnable parameters is an effective way to extract similar

features at multiple locations with a small number of parameters.

Therefore, the ability to save the parameters and the calculation

process is a key development. The two main features

of a CNN are local connections and weight sharing. The socalled

local connection means that the nodes of the convolution

layer are only connected with some nodes of the previous

layer and are only used to learn local features. This kind of

local connection greatly reduces the number of parameters,

speeds up the learning rate, and reduces the possibility of

overfitting to some extent. In addition, the convolutional neurons

are grouped into feature maps that share the same weight;

thus, the entire process is equivalent to convolution, and the

shared weight is the filter for each map.Weight sharing greatly

reduces the number of network parameters, thereby increasing

efficiency and preventing overfitting. The first few layers of

the convolutional network alternate between a convolutional

layer and a down sampling layer, followed by a number of

fully connected layers, each followed by an activation layer,

resulting in a classification result. By using gradient-descent

based methods and backpropagation to minimize the loss

function, a CNN is trained in a similar manner to other artificial

neural networks.

Since the 1990s, CNNs have gained continuous research

interest. During this time, Hinton and his colleagues conducted

extensive and meaningful fundamental research on deep

neural networks to improve algorithmic performance and optimize

architecture [19–22]. In 2012, **Alex Krizhevsky proposed**

**AlexNet, a deep convolutional neural network model,**

**which can be regarded as a deeper and wider version of LeNet**

**[23]. AlexNet includes several newer technical points. In the**

**CNN, functions such as ReLU, Dropout, and LRN have been**

**added. At the same time, GPUs have also been used to accelerate**

**the calculations, gaining the first place in the ILSVRC**

**2012 competition with significant advantages. This network**

**structure has also become a representative network structure**

**of current convolutional neural networks. Zeiler and Fergus**

**used random pools to tune the CNN**. Simonyan and

Zisserman proposed a very deep CNN to layers 16–19,

achieving the greatest accuracy (ILSVRC) for ImageNet’s massive visual recognition challenge [22]. The width of the

network also affects the ultimate effect of deep learning.

Google has proposed a deep convolutional neural network

structure called BInception^ that stacks 1 × 1, 3 × 3, 5 × 5 convolution

layers and 3 × 3 pooling layers. On the one hand, this

increases the network’s width; on the other hand, it also increases

the adaptability of the network to the scale of a particular

problem [24]. This network raises the level of technology

for classifying and identifying the ILSVRC14 data set. He

Kaiming et al. proposed a residual learning framework to reduce

network training, so that each layer can learn an incremental

transformation, which changes the internal network

parameters and characteristics of the transmission method

[25, 26]. The network is deeper than previously used networks,

and training speeds have increased significantly. In

March 2016, AlphaGo defeated Lee Sedol in five games.

This is the first time that the Go program has defeated professional

chess players [27]. It is an important milestone in artificial

intelligence research.

In pattern recognition tasks, traditional machine learning

tools often provide limited precision, and deep learning has

been shown to be one of the most promising methods to overcome

these limitations and achieve more powerful and reliable

reasoning [28]. However, although deep learning technology

has existed and has been successfully applied in many fields,

only a few DL-based mobile applications have been produced,

mainly due to the low-power computing resources provided

by mobile devices [29, 30]. There are two main models for

deploying DL applications on mobile devices: the clientserver

model and the client-only model. The former uses a

mobile device as a sensor (without data preprocessing), or a

smart sensor (with some data preprocessing), whose data are

sent to a server or a simple cloud that runs the DL engine and

sends the results back to the client. In this framework, maintaining

good connectivity is the most important requirement

for efficient operation of the entire system. The latter model

runs DL applications directly on the mobile device. This solution

can produce faster results without an operational network

connection but must cope with the lack of device resources

and therefore requires the right software and hardware

solutions. In terms of mobile services, Honghao Gao,Wanqiu

Huang and others have researched and innovated service patterns

and workflows [31, 32]. In the development of deep

learning, scientists have optimized model compression and

neural network structure, especially for the CNN. The

SqueezeNet [33] structure proposed in 2016 is not designed

to achieve higher precision, but to simplify network complexity,

reduce the size and calculation time of the model, and

achieve performance similar to that of some typical large networks

(such as Alexnet and VGG). In 2017, Google Inc. proposed

a small network structure, MobileNet for embedded

devices such as mobile phones, which can be applied to various

tasks such as target detection, face attribute analysis and scene recognition [34]. Similarly, there is an efficient network

ShuffleNet designed for mobile devices byVision Technology

[35].

Different input data and computing environments have different

structural requirements for neural networks. In this paper,

our work is to design a mobile computing framework and

a convolutional neural network structure based on data characteristics

and mobile computing requirements to achieve

Shui character recognition.

Experiment and result

4.1 Experimental preparation

We use our own Shui character dataset. First of all, we cut out

the images of the Shui characters from the classic scanned

documents. Second, these pictures are binarized and standardized.

Third, a clustering algorithm is implemented and these

image slices are used as input to the clustering algorithm.

Finally, add the class label to the clustering result. After these

steps, the image we get is a 52\*52 binary image. Since the

initial image slice is unlabeled and the convolutional neural

network is supervising the neural network, we must mark this

data. The data set obtained after preprocessing is shown in

Fig. 7.

We experimented in the **TensorFlow** environment using the

Python language.

4.2 Training

Among the 400 types of pictures that have been labeled, 3/4 of

the 50,000 data are selected as training data and 1/4 are used as

test data. To train on a convolutional neural network with 4

convolutional layers, we use the number of convolution kernels,

learning rate, and batch size as variables. The results

show that with the increase in the network scale, the results

of convolutional neural networks are getting better, but the training time is getting longer, and there may be problemswith

overfitting. When the batch size is large, it is easy to find the

optimal state of the model. Small batches will cause the recognition

rate to fluctuate. In other words, the model only fits to

a part of the data, but the batch size cannot be too large. When

it increases to a certain level, the model cannot be trained, and

the batch size is also limited by available memory.

When the batch changes, the learning rate has to be

changed. When the batch size is too large and the learning

rate is too low, the model cannot obtain sufficient training in

a fixed number of iterations, while a larger batch size is insensitive

to a high learning rate. In order to ensure training under

the same conditions, we set the momentum value to 0.9. As shown in Fig. 8, we tested the experimental accuracy of various

batch size and learning rate.

4.3 The result

In the training of convolutional neural networks, we mainly focus

on three parameters: the number of convolution kernels, the

learning rate, and the batch size. For the number of convolutional

cores, we can slowly increase it until it reaches its optimal state.

Of course,we cannot increase it all the time because it will lead to

overfitting problems. For batch size, we can use relatively large

numbers if resources allow, which helps us easily find the global gradient direction. After many trials, we chose a learning rate of

0.001 and a batch size of 128.

In Fig. 9, we can see that when the training reaches a certain

step size (400 epoch), the accuracy begins to oscillate, that is,

the over-fitting phenomenon begins to occur, and the training

should be stopped. Our final test accuracy is around 93.3%.

4.4 Comparison with others

We have also used Alexnet and Googlenet for experiments. The

experimental results are shown in Table 2.

According to Table 2, Googlenet has the highest accuracy

and Alexnet has the lowest accuracy. In terms of the number

of parameters and the amount of calculation, our proposed

network structure is significantly better than Alexnet and

Googlenet. In general, the model we proposed is effective,

which can achieve excellent classification performance, and

is more suitable for deployment on the mobile side.

rp5

1. V. EXPERIMENTAL RESULT

This is a simple method with precision and recall are over 90% and often with 95% on an average. However, this technique is not very effective for big size fonts as well as for some specific pictures for which corners are responding too much. Despite of these problems it is fast (and can be parallelized) and less complex as compared to other OCR tools and could be further improved in the future. Finally, this method seems also to be very efficient in extracting more complex layouts such as paragraphs, and lines.

According to research of \_\_\_rp1\_\_\_\_\_\_, simlar to our nepali printed , they had english alphabet recognition, Different sizes of memory cells were adopted to test the recognition accuracy. They concluded that greater the size of memory cells the higher the recognition accuracy. They achieved maximum accuracy of 95.58%.

According research of \_\_\_rp2\_\_\_, character recognition using CNN is implemented and that method is evaluated based on Artificial Neural Network (ANN). Used ANN.

According research of \_\_\_rp4\_\_\_,, Shui character dataset. First of all, we cut out

the images of the Shui characters from the classic scanned documents. Second, these pictures are binarized and standardized. algorithm is implemented and these image slices are used as input to the clustering algorithm. Finally, add the class label to the clustering result. After these steps, the image we get is a 52\*52 binary image. Since the

initial image slice is unlabeled and the convolutional neural network is supervising the neural network We experimented in the TensorFlow environment using the Python language. In the training of convolutional neural networks, we mainly focus

on three parameters: the number of convolution kernels, the

learning rate, and the batch size. For the number of convolutional

cores, we can slowly increase it until it reaches its optimal state.

Of course,we cannot increase it all the time because it will lead to

overfitting problems. For batch size, we can use relatively large

numbers if resources allow, which helps us easily find the global gradient direction. After many trials, we chose learning rate of0.001 and a batch size of 128. We have also used Alexnet and Googlenet for experiments. The experimental results are shown in Table 2. According to Table 2, Googlenet has the highest accuracy and Alexnet has the lowest accuracy. In terms of the number of parameters and the amount of calculation, our proposed network structure is significantly better than Alexnet and Googlenet. In general, the model we proposed is effective, which can achieve excellent classification performance, and is more suitable for deployment on the mobile side.