

Faculty of computer and artificial intelligence.

CS396_Selected CS2 (2021-2022).

Team: #30

Name	ID
Ali Ehab Ali Yousef	201900483
Nehad Ahmed Ali NourEldein	201900902
Nehal Ashraf Elsayed Elsayed	201900903
Mohamed Ashraf Abdelaziz Ibrahim	201900635
Mohamed Hassan Ali Amen Hendy	201900654
Omar Abdel Nasser Tawfik Adam	201900517
Hesham Mahmoud Ibrahim	201900942
Nourhan Mouhamed Radwan	201900916

2-Paper details:

Received October 26, 2017, accepted December 31, 2017, date of publication January 25, 2018, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2796722

Image Classification Based on the Boost Convolutional Neural Network

SHIN-JYE LEE (1), TONGLIN CHEN (2), LUN YU (2), AND CHIN-HUI LAI (3)

(1) Institute of Technology Management, National Chiao Tung University, Hsinchu 30010, Taiwan

(2) National Pilot School of Software, Yunnan University, Kunming 650091, China

(3) Department of Information Management, Chung Yuan Christian University, Chungli 302023, Taiwan

Corresponding author: Chin-Hui Lai (chlai@cycu.edu.tw)

This work was supported in part by the Key Laboratory of Software Engineering of Yunnan Province under Grant 2017SE202, in part by

the Applied Basic Research Foundation of Yunnan Province under Grant 2016FB104, in part by the Yunnan Provincial Innovation Team

under Grant SJ2016-1, in part by the Natural Science Foundation of China under Grant 61402397, Grant 61663046, Grant 61379032, and

Grant 61662085, and in part by the Ministry of Science and Technology of Taiwan under Grant MOST 106-2410-H-033-013.

TABLE 1. Hardware and software of computer.

Item	Content
<i>Processor</i>	Intel(R) core(TM) i7-6700HQ CPU
<i>GPU</i>	NVIDIA GeForce GTX 960M
<i>Memory</i>	8G
<i>Operating System</i>	Windows 10
<i>Tensorflow</i>	TensorFlow 1.0
<i>Python</i>	Python 3.5
<i>Cuda</i>	cuda 8.0

TABLE 2. Specification of CNN configuration.

Input: 500000*3072		
<i>Hidden1 Layer</i>	conv	Size 5*5; quantity: 64; method: same
	ReLU	Max(0,x)
	Max Pooling	Size: 3*3; stride:2
	Batch Norm	alpha=0.001 / 9.0, beta=0.75
<i>Hidden2 Layer</i>	conv	Size: 5*5; quantity: 64; method: same
	ReLU	Max(0,x)
	Max Pooling	Size: 3*3, stride:2
	Batch Norm	alpha=0.001 / 9.0, beta=0.75
<i>Hidden3 Layer</i>	Full connect	Weight size: [1228, 384]
	ReLU	Max(0,x)
	Dropout	Probability of activation: 0.5
<i>Hidden4 Layer</i>	Full connect	Size of weight: [384, 192]
<i>Output Layer</i>	Softmax	Size of weight: [192, 10]

TABLE 3. Configuration of adaboost.

Input	Use the feature extraction data of the convolution network: [50000,192]
<i>Softmax1</i>	Size of weight: [192, 10]
<i>Softmax2</i>	Size of weight: [192, 10]
<i>Softmax3</i>	Size of weight: [192, 10]
<i>Softmax4</i>	Size of weight: [192, 10]
<i>Softmax5</i>	Size of weight: [192, 10]
<i>Softmax6</i>	Size of weight: [192, 10]
<i>Softmax7</i>	Size of weight: [192, 10]
Output	Results of weight voting of the categories

TABLE 4. Experimental comparison of CIFAR-10 testing datasets.

Classifier	Accuracy of Testing (%)
<i>Softmax</i>	35.5
<i>AdaBoost+ Softmax</i>	52.3
<i>CNN+Softmax</i>	85.3
<i>CNN+AdaBoost (this study)</i>	88.4

Project Description Document:

General Information on the selected dataset:

Rock-Paper-Scissors

https://drive.google.com/drive/folders/1ERpc8o3Z1o8srtvMkmrQKGf-5_1ZdiJH?usp=sharing

Total number of samples: 2892 sample.

the dimension of images: (227, 227, 1).

number of classes: (3).

their labels:

1-paper

2-scissors

3-rock

CIFAR-10

<https://www.cs.toronto.edu/~kriz/cifar.html>

Total number of samples: 60000 sample.

the dimension of images: (32, 32, 3).

number of classes: (10).

their labels:

(1-airplane, 2-automobile ,3-bird, 4-cat ,5-deer ,6-dog, 7-frog ,8-horse, 9-ship, 10-truck)

Implementation details:

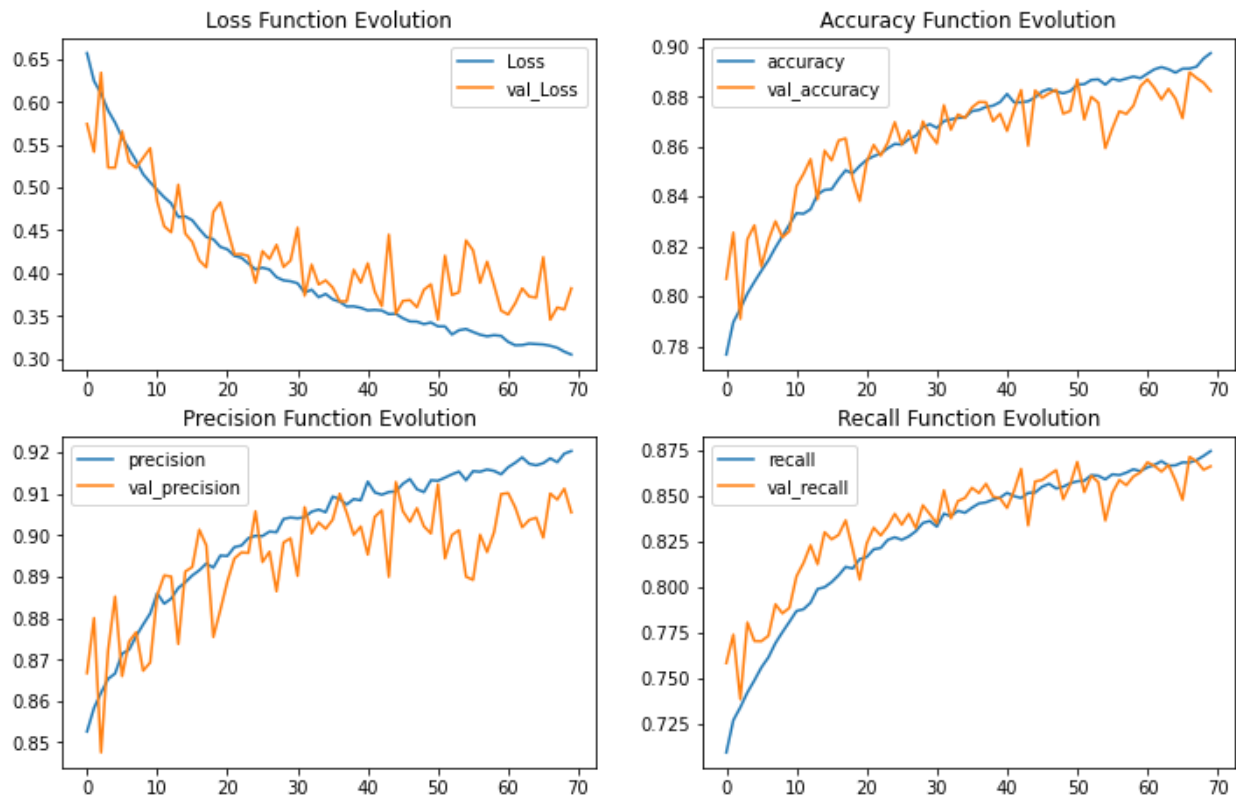
CIFAR-10

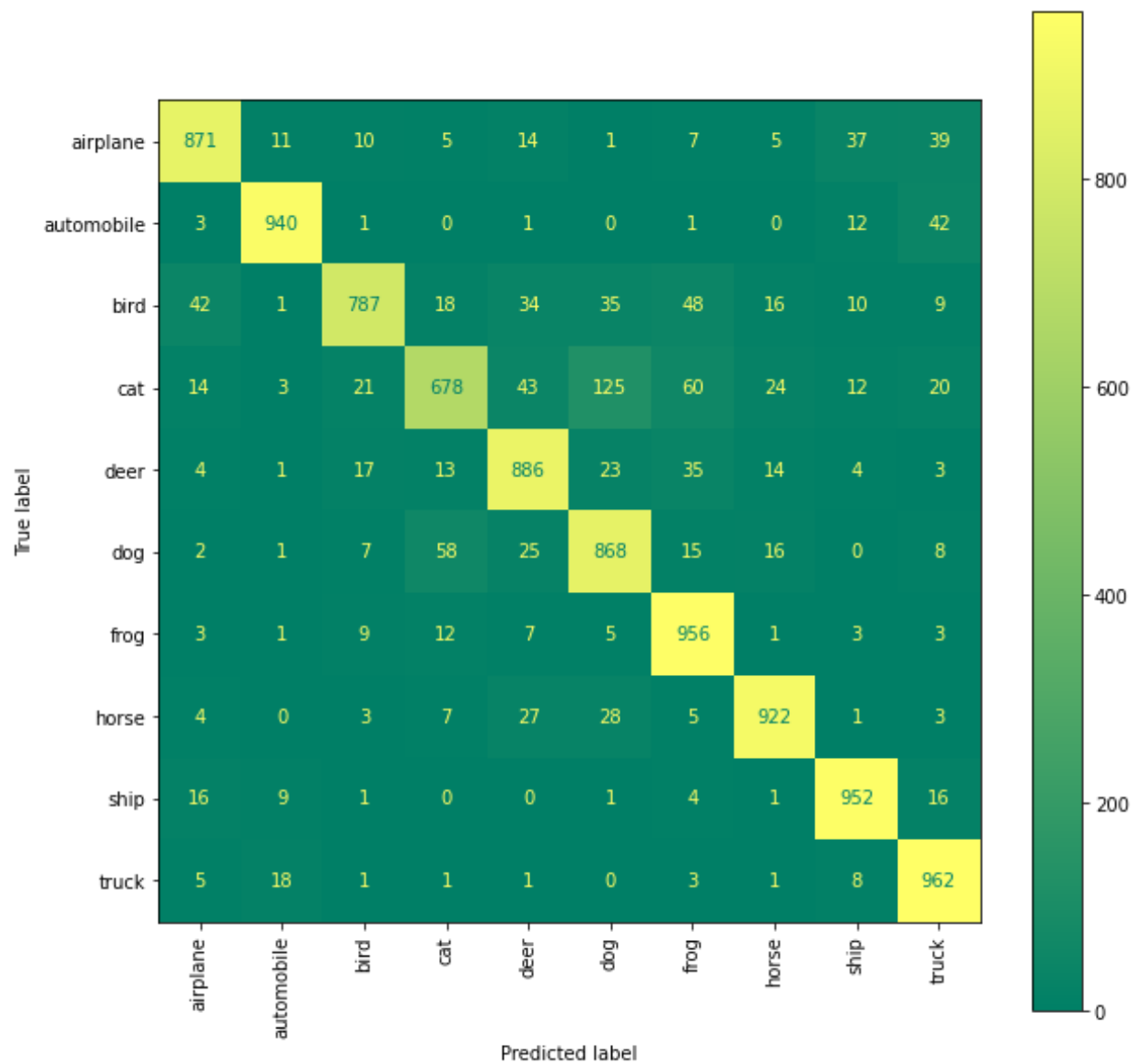
Training (83.3%=50000 image), validation (16.67%=10000) and testing (16.67%=10000).

Rock-Paper-Scissors

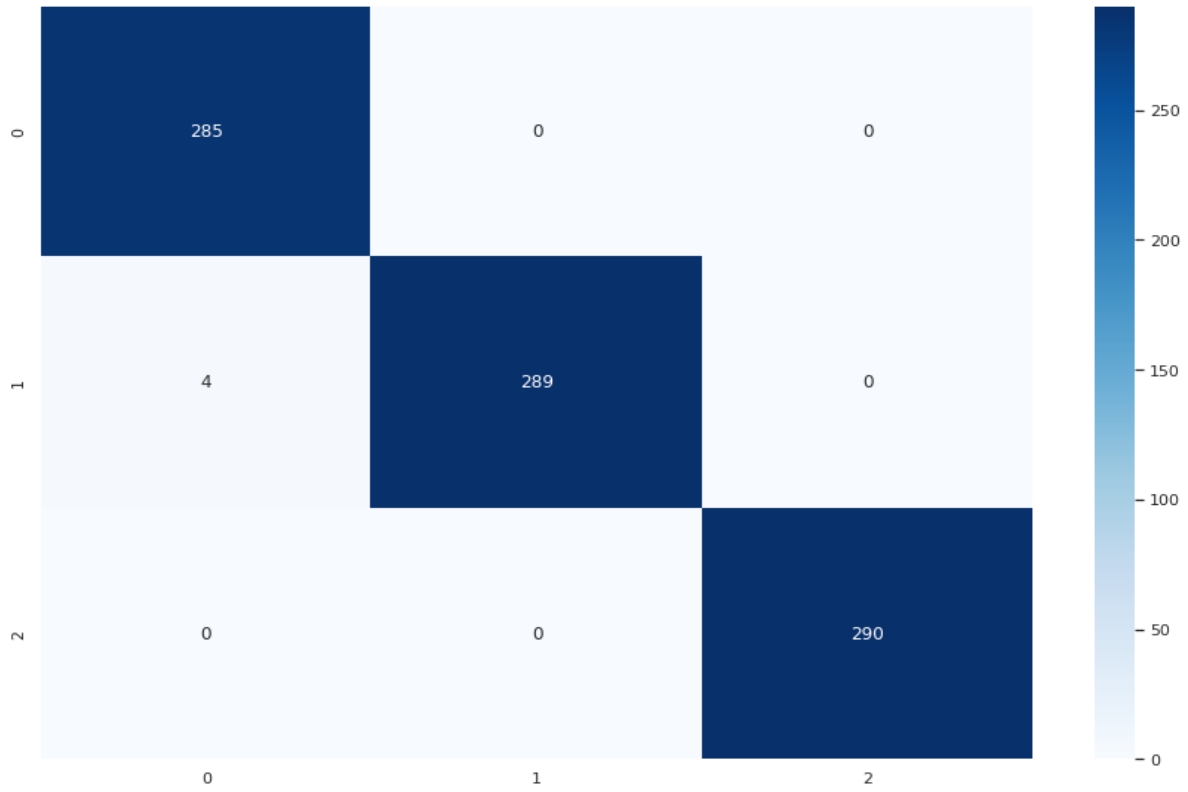
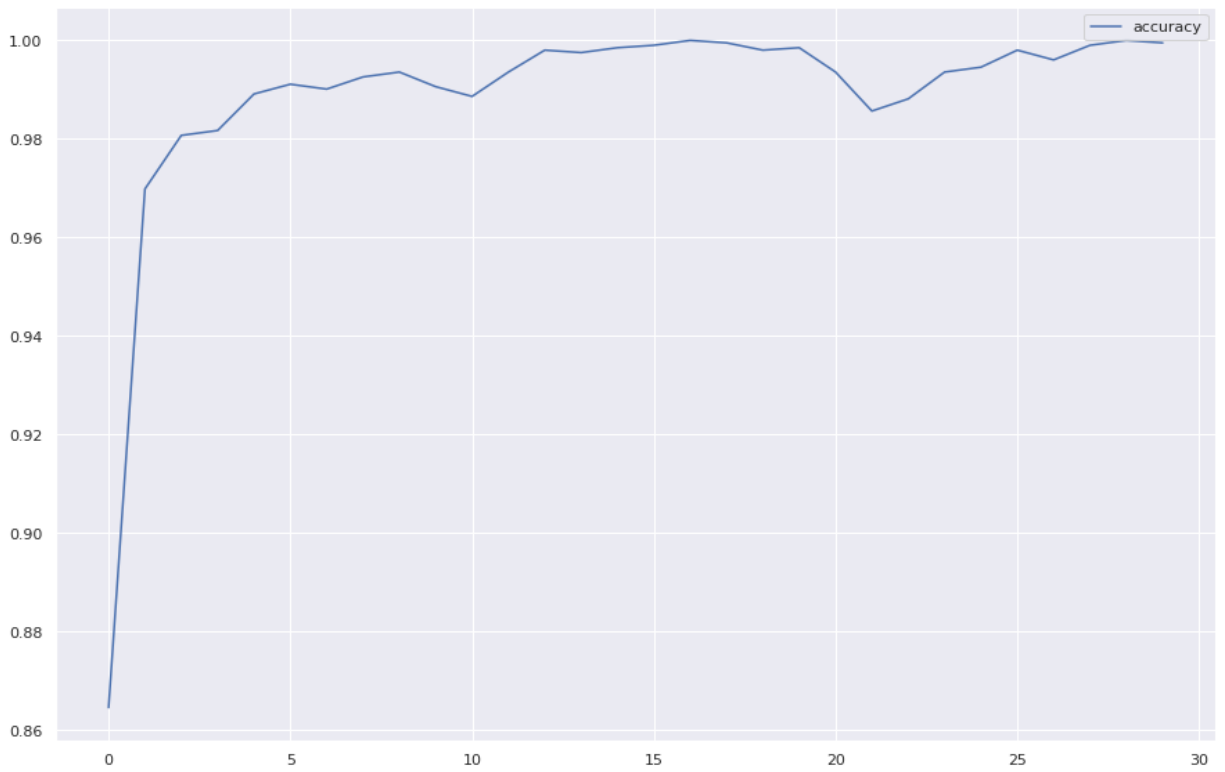
Training (87.13%=2520 image), validation (0) and testing (12.86%=372).

CNN on cifar-10





CNN on Rock-Paper-Scissors



Results details:

CNN on cifar-10

```
evaluation = model.evaluate(X_test, y_cat_test)
print(f'Test Accuracy : {evaluation[1] * 100:.2f}%')

313/313 [=====] - 2s 5ms/step - loss: 0.3822 - accuracy: 0.8822 - precision: 0.9055 - recall: 0.8663
Test Accuracy : 88.22%
```

CNN on Rock-Paper-Scissors

```
# Train
loss, acc = model.evaluate(x_train, y_train)
print('Train')
print(f'loss : {loss}')
print(f'acc : {acc*100}')

64/64 [=====] - 2s 30ms/step - loss: 2.9230e-07 - accuracy: 1.0000
Train
loss : 2.922998589838244e-07
acc : 100.0

# Test
loss, acc = model.evaluate(xtest, np_utils.to_categorical(ytest))
print('Test')
print(f'loss : {loss}')
print(f'acc : {acc*100}')

28/28 [=====] - 1s 29ms/step - loss: 0.0487 - accuracy: 0.9954
Test
loss : 0.04873378947377205
acc : 99.53917264938354
```