Taiwan Deep Travel - Deep Learning-Based Recommendation APP for Taiwan Tourist Attractions

Speakers

Chi, Lee — Department of Psychology, SCU

Jane Han, Wu — Department of Economics, SCU

Bing Sheng, Gao — Department of Computer Science and Information Management, SCU

Advisors

Prof. Ching-Shoei, Chiang — Department of Computer Science and Information Management, SCU

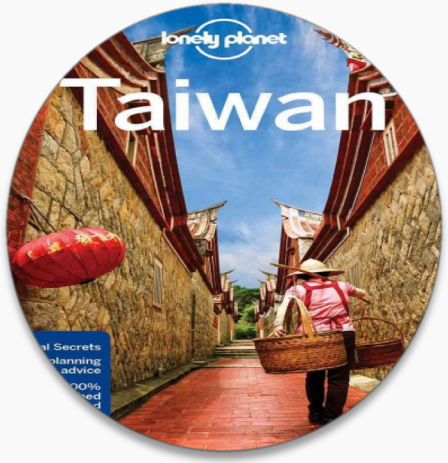
Prof. Yi-Ping, Chang — Department of Financial Engineering and Actuarial Mathematics, SCU

CONTENT

The Problem To Be Solved

PART ONE

Research Motivation



Taiwan is rich in beautiful natural and cultural scenes

Modern people live a busy life and have little time to search for travel information

The popularity of smartphones

Operating Mode — Import Image

Ocean
Islet

Island

TOP 1
Danger
rock island

TOP 2
Beinan Creek
Rafting

TOP 3
Waisanding
Sand Bar

TOP 4
Wangankou
Beach

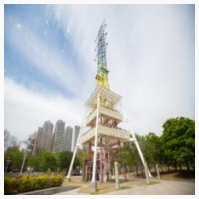
TOP 5
Guanshan
Xizhao

Operating Mode — Take Photos

Xiaonanmen

Building Façade Temple

TOP 1
Amitabha Temple



TOP 2
Tower of Light

TOP 3
Chiang Kai-shek Memorial Hall

TOP 4
Tree-footed church

TOP 5
Ci En Pagoda

Accurately meet the needs of users

Mountain

Mountain Path Forest

Sunset Canyon

Lawn

Focus on the recommendation of tourist attractions

Convenient user experience

Open Data

PART TWO

Model Dataset— Source

MIT CSAIL 「Place - 365 Standard」 dataset

- 365 scenes
- 256x256x3 pixels

Training Dataset — Each scene has 5,000 images

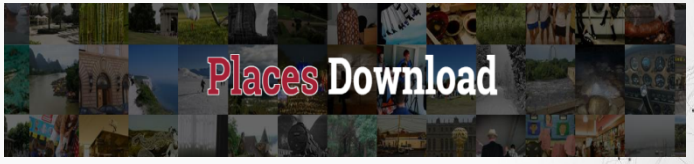
Validation Dataset — Each scene has 100 images

Model Dataset— Filtered

- 20 scenes
- 256x256x3 pixels

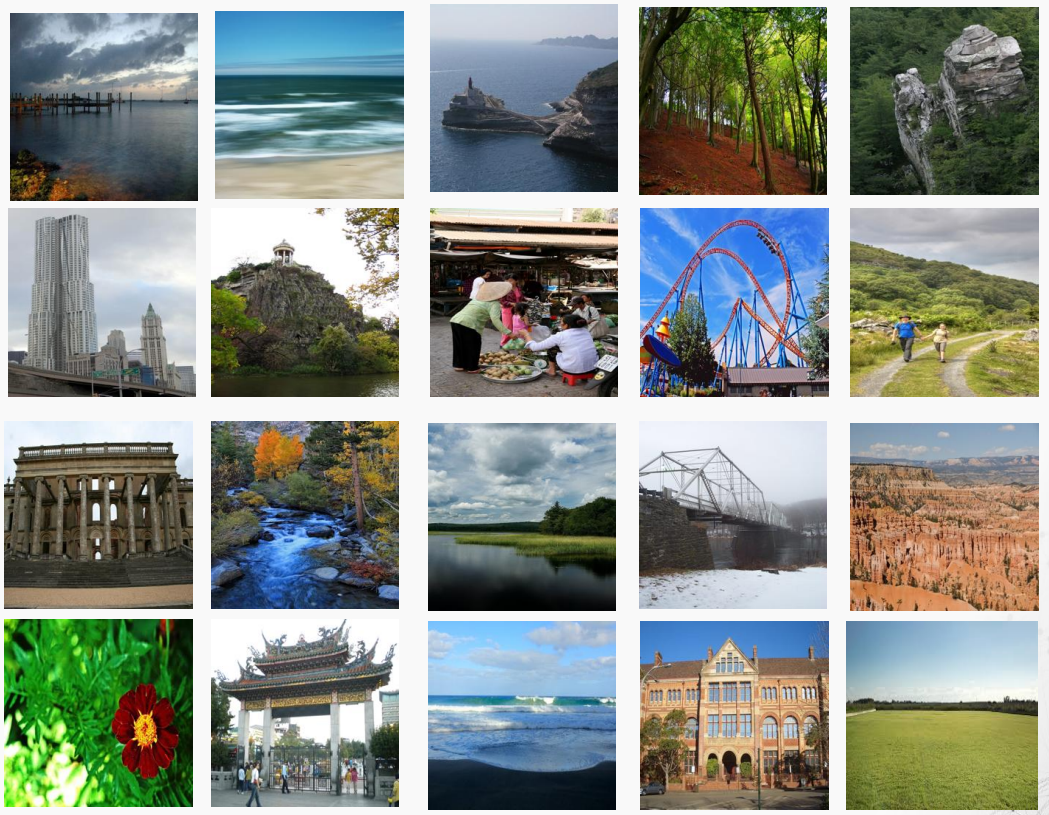
Training Dataset — Each scene has 5,000 images

Validation Dataset — Each scene has 100 images



Model Dataset— 20 Scenes

Amusement Park	Pier
Botanical Garden	Temple
Building Façade	Lawn
Mountain Path	Market
Skyscraper	Mountain
Canyon	Museum
Cliff	Lake
Creek	Beach
Forest	Bridge
Islet	Ocean



Taiwan recommended tourist attractions dataset— Source

Tourism Bureau, Ministry of Transportation and Communications Government Information Open Platform : "**Attractions - Sightseeing Information Dataset**" - covers 4,809 scenic spots

Taipei City Government Information Open Platform : "Taipei City, **Taipei Tourism Website - Attractions Data**" - covers 200 scenic spots

- Locations all over Taiwan
- Each scenic spot has an image and a text introduction about the scenic spot

臺北市政府 Taipei City Government

Taiwan THE HEART OF ASIA

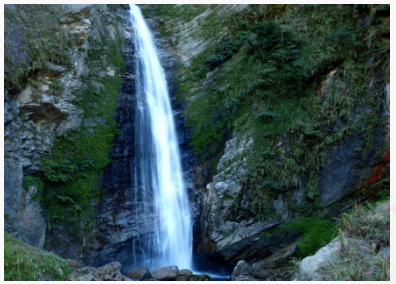
交通部觀光局觀光統計資料庫 Tourism Statistics Database of the Taiwan Tourism Bureau

Recommended tourist attractions dataset— Filtered

Recommended tourist attractions dataset

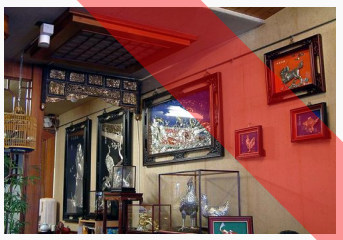
- Covers 895 scenic spots
- Locations all over Taiwan
- Each scenic spot has an image and a text introduction about the scenic spot



Recommended tourist attractions dataset— Filter Condition

Artwork



People

Tourist center

Experience in nature

Data usage

1. Input 20 different scenes to the model for training

- Each scene has 5,000 images

2. Input the attraction recommendation data set into the model

- Produces 895 1x20 vectors
- Each value in the vector represents the probability of each scene in the scenic spot predicted by the model

3. The user inputs an image of the attraction into the model

- Produces a 1x20 vectors

4. Recommend top five scenic spots with highest similarity to users

- Cosine similarity was compared 895 times between the vector generated by the image input by the user and the vector generated by the scenic spots

Technical Description

PART THREE

Deep Learning Tools

Keras

- Written in Python code
- Open source application running on TensorFlow
- Various neural network layers are built in, such as CNN and RNN
- Through the back end of the TensorFlow engine, can be run on the GPU
- Easier to understand and more modular than TensorFlow

Google Cloud Platform

Compute Engine

Create virtual machine instances

Remote control virtual machines

Install CUDA and TensorFlow, Keras libraries

Virtual Machine standard

- Two vCPU, 26 GB Memory
- Operating System: Windows Server 2016

  256GB SSD
- GPU standard：NVIDIA Tesla K80

  24GB Memory

Augment Original Validation Dataset

Keras Image Data Augmentation

Original Validation Data

- 20 scenes
- 100 images each scene

- 20 scenes
- 200 images each scene

Augmented Validation Data

- 20 scenes
- 100 images each scenes

Augmented Test Data

- 20 scenes
- 100 images each scenes

Modeling Procedure

1. Determine suitable Transfer Learning model
2. Determine Optimizer
3. Determine Batch Normalization (BN) & Activation Function
4. Determine to use dynamic Learning Rate or static Learning Rate
5. Adjust Hyperparameter (Avoid Overfitting)
6. Test model and evaluates the final results

Determine suitable Transfer Learning model

- Use Transfer Learning to train each model
- The result of training all layers 5 epochs

Model	Loss	Top-1 Acc	Top-3 Acc	Val loss	Top-1 Val_acc	Top-3 Val_acc	Size	Parameters	Depth	Time
VGG16 (2014 ILSVRC)	0.8538	0.7390	0.9250	0.9885	0.6760	0.9160	**266 MB**	23,238,356	27	47 ms/step
ResNet50 (2015 ILSVRC)	0.3632	0.8760	0.9800	1.0926	0.7060	0.9140	**417 MB**	36,435,476	180	40 ms/step
MobileNet (2017 Google)	0.9317	0.7000	0.9170	1.0584	0.6940	0.9030	110 MB	9,654,100	92	17 ms/step
DenseNet121 (2017 CVPR)	0.7228	0.7650	0.9350	0.9701	0.7030	0.9150	**154 MB**	13,462,740	431	**38 ms/step**
NASNetMobile (2018)	0.7656	0.7650	0.9320	0.8591	0.7320	0.9250	**128 MB**	10,895,656	774	**34 ms/step**
MobileNetV2 (2018 Google)	0.8226	0.7480	0.9240	0.9828	0.6780	0.9120	118 MB	10,288,852	160	20 ms/step

Determine Optimizer

- Compare Adam & Adafactor
- The result of training all layers 5 epochs
- The validation data is 2000 images of augmented validation data

Model	Loss	Top-1 Acc	Top-3 Acc	Val loss	Top-1 Val_acc	Top-3 Val_acc	Size	Parameters	Depth	Time
MobileNet with Adam (2015)	0.9624	0.6760	0.9030	1.5239	0.5565	0.8020	110 MB	9,654,100	92	22 ms/step
MobileNet with Adafactor (2018)	3.2579	0.0510	0.1300	3.4619	**0.0500**	0.1500	37 MB	9,654,100	92	21 ms/step

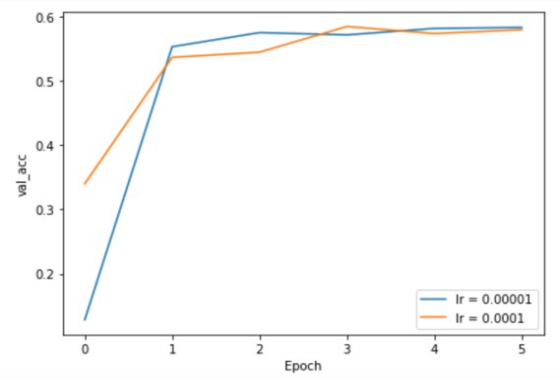
Determine Batch Normalization (BN) & Activation Function

- Compare ReLU with BN, ReLU without BN and SeLU with BN
- The result of training all layers 5 epochs
- The validation data is 2000 images of augmented validation data

Model	Loss	Top-1 Acc	Top-3 Acc	Val loss	Top-1 Val_acc	Top-3 Val_acc	Size	Parameters	Depth	Time
ReLU with BN	0.8410	0.7250	0.9230	1.3066	**0.5835**	0.8315	117.7 MB	9,855,316	95	21 ms/step
ReLU without BN	0.9624	0.6760	0.9030	1.5239	**0.5565**	0.8020	110 MB	9,654,100	92	22 ms/step
SeLU with BN (2015)	0.7213	0.7840	0.9390	1.3839	**0.5695**	0.8215	117.7 MB	9,855,316	94	20 ms/step

Determine to use dynamic Learning Rate or static Learning Rate

- Compare dynamic and static Learning Rate (LR) results
- The result of training all layers 5 epochs
- The validation data is 2000 images of augmented validation data



Model	Loss	Top-1 Acc	Top-3 Acc	Val loss	Top-1 Val_acc	Top-3 Val_acc	Size	Parameters	Depth	Time
MobileNet (LR = 0.00002)	0.8410	0.7250	0.9230	1.3066	0.5835	0.8315	117.7 MB	9,855,316	95	21 ms/step
MobileNet (LR = 0.0002)	0.7651	0.7460	0.9360	1.3139	**0.5800**	0.8400	112 MB	9,855,316	95	21 ms/step
MobileNet (LR = 0.002)	1.1672	0.6360	0.8590	1.7032	0.4620	0.7455	112 MB	9,855,316	95	22 ms/step
MobileNet (LR = 0.002 _ 0.00002)	1.0588	0.6670	0.8770	1.4206	0.5540	0.7965	112 MB	9,855,316	95	21 ms/step

Adjust Hyperparameter

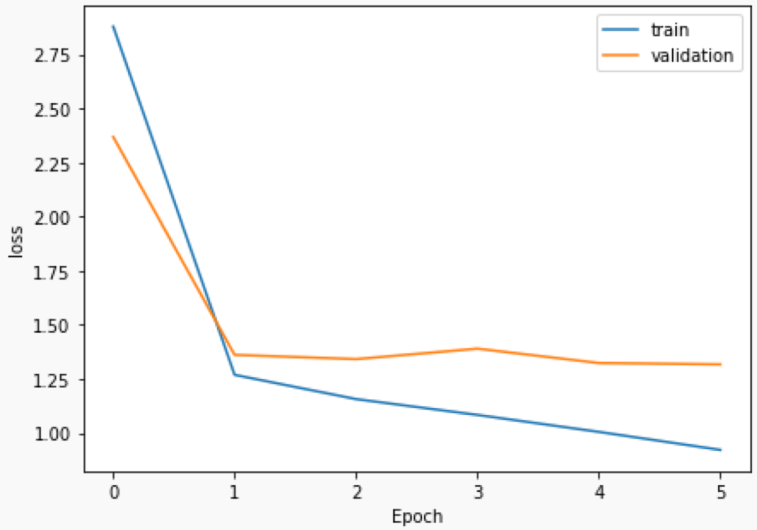
- The result of training all layers 5 epochs
- The validation data is 2000 images of augmented validation data

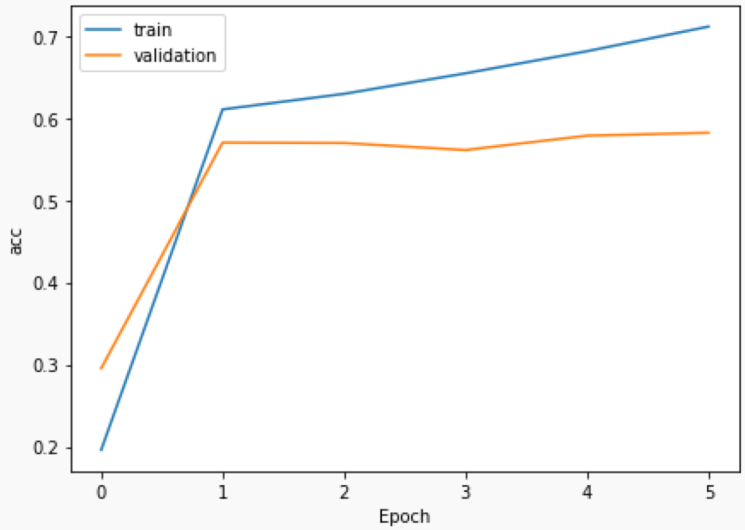
Depth	Time
94	21 ms/step
94	22 ms/step
94	21 ms/step
94	21 ms/step
94	21 ms/step
94	22 ms/step
94	21 ms/step
94	22 ms/step

kernel_regularizer activity_regularizer	Dropout	Dense	Loss	Top-1 Acc	Val loss	Top-1 Val_acc	Size	Parameters
0	0.5	128	0.9438	0.6880	1.2661	0.5840	112 MB	9,855,316
0	0.5	256	0.8881	0.6970	1.2515	0.5910	185 MB	16,281,044
0	0.2	128	0.8632	0.7090	1.2756	0.5700	112 MB	9,855,316
0	0.2	256	0.9959	0.6630	1.2755	0.5780	185 MB	16,281,044
0.01	0.5	128	4.4488	0.0500	4.4374	0.0510	117.7 MB	9,855,316
0.01	0.5	256	5.6078	0.1880	5.3630	0.2795	194.8 MB	16,281,044
0.01	0.2	128	3.6754	0.3710	3.6977	0.3865	117.7 MB	9,855,316
0.01	0.2	256	5.4007	0.4580	5.4992	0.4165	194.8 MB	16,281,044

Model Performance Evaluation

- The result of training all layers 5 epochs
- The validation data is 2000 images of augmented validation data





Model	Loss	Top-1 Acc	Top-3 Acc	Val_loss	Top-1 Val_acc	Top-3 Val_acc
5 epochs	0.9600	0.6900	0.9020	1.2687	**0.5835**	0.8515

Model Performance Evaluation

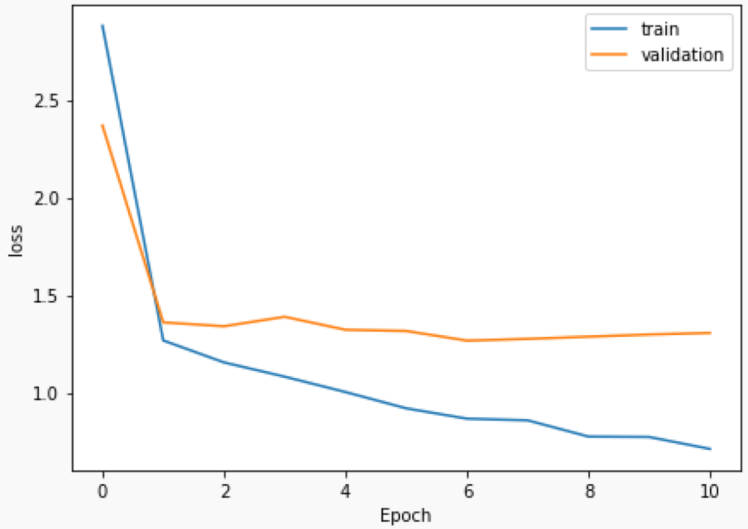
- The result of training all layers 5 epochs
- The test data is 2000 images of augmented validation data

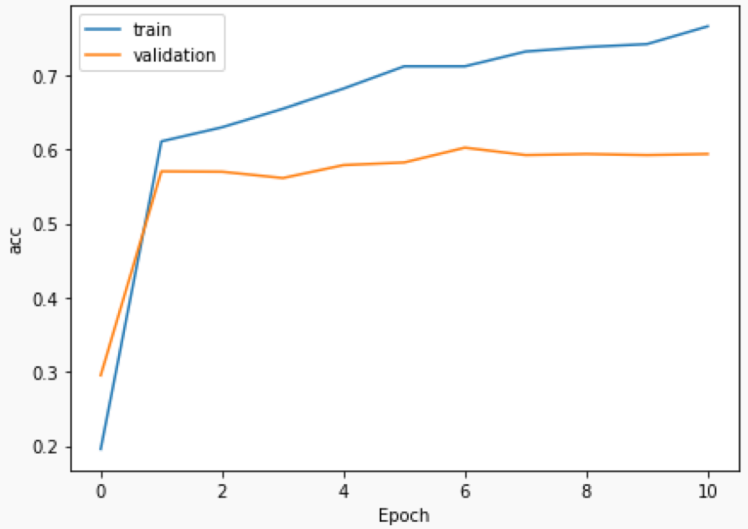
Test Data

Model	Loss	Top-1 Acc	Top-3 Acc
5 epochs	1.2499	**0.5970**	0.8510

Model Performance Evaluation

- The result of training all layers 10 epochs
- First 5 epochs' LR are 0.0002, last 5 epochs' LR are 0.00002
- The validation data is 2000 images of augmented validation data





Model	Loss	Top-1 Acc	Top-3 Acc	Val_loss	Top-1 Val_acc	Top-3 Val_acc
5_5 epochs	0.8040	0.7380	0.9360	1.2467	**0.6075**	0.8535

Model Performance Evaluation

- The result of training all layers 10 epochs
- The test data is 2000 images of augmented validation data

Test Data

Model	Loss	Top-1 Acc	Top-3 Acc
5_5 epochs	1.2206	0.6140	0.8465

Recommendation System

Cosine similarity after correction

$$\frac{\sum_{i=1}^{20}[(A_i - \boldsymbol{\mu_A})\times(B_i - \boldsymbol{\mu_B})]}{\sqrt{\sum_{i=1}^{20}(A_i - \boldsymbol{\mu_A})^2}\sqrt{\sum_{i=1}^{20}(B_i - \boldsymbol{\mu_B})^2}}$$

Cosine similarity

$$\frac{\sum_{i=1}^{20}(A_i\times B_i)}{\sqrt{\sum_{i=1}^{20}(A_i)^2}\sqrt{\sum_{i=1}^{20}(B_i)^2}}$$

Taipei 101

Cosine similarity:
0.9967668105098986

Xinbeitou Hot Spring Area

Cosine similarity:
-0.19923189416031017

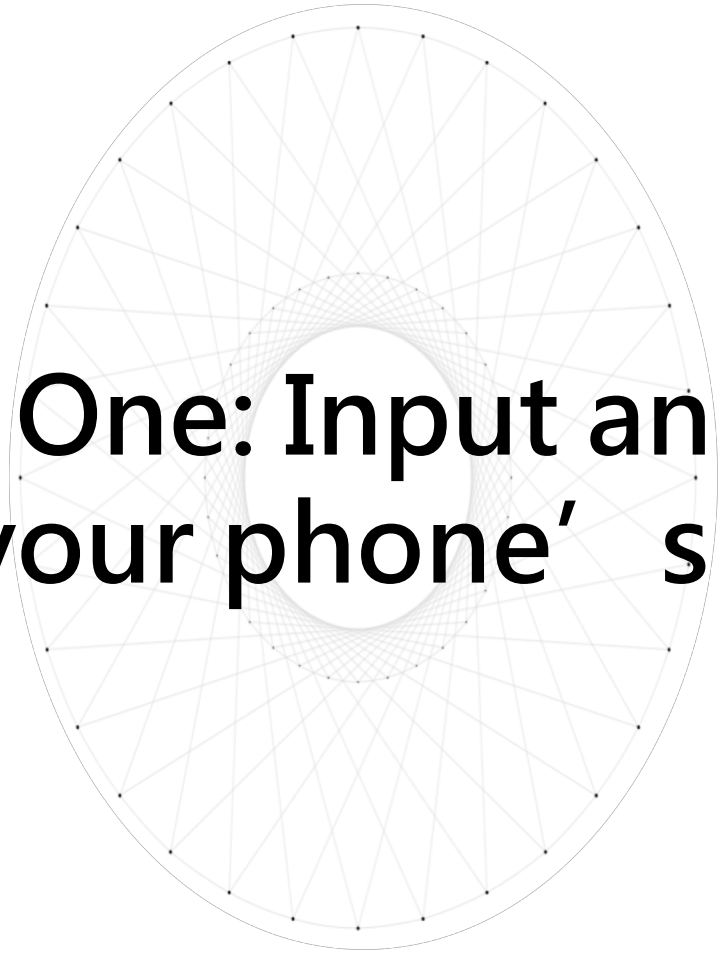
Integrated into mobile devices

Core ML 2.0

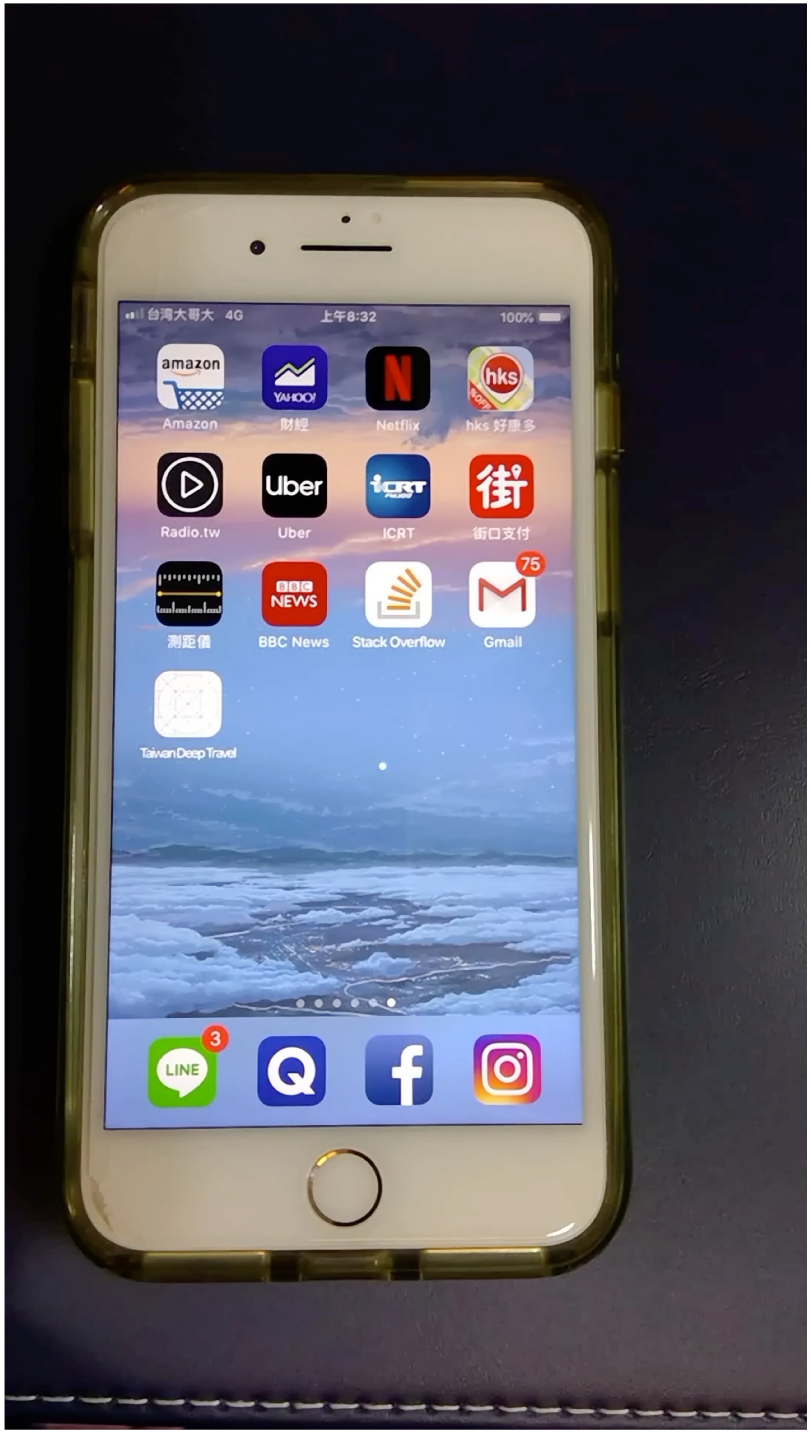
- Machine learning framework introduced by Apple at WWDC 2018
- Empowers the iOS app developers to integrate machine learning technology into their apps

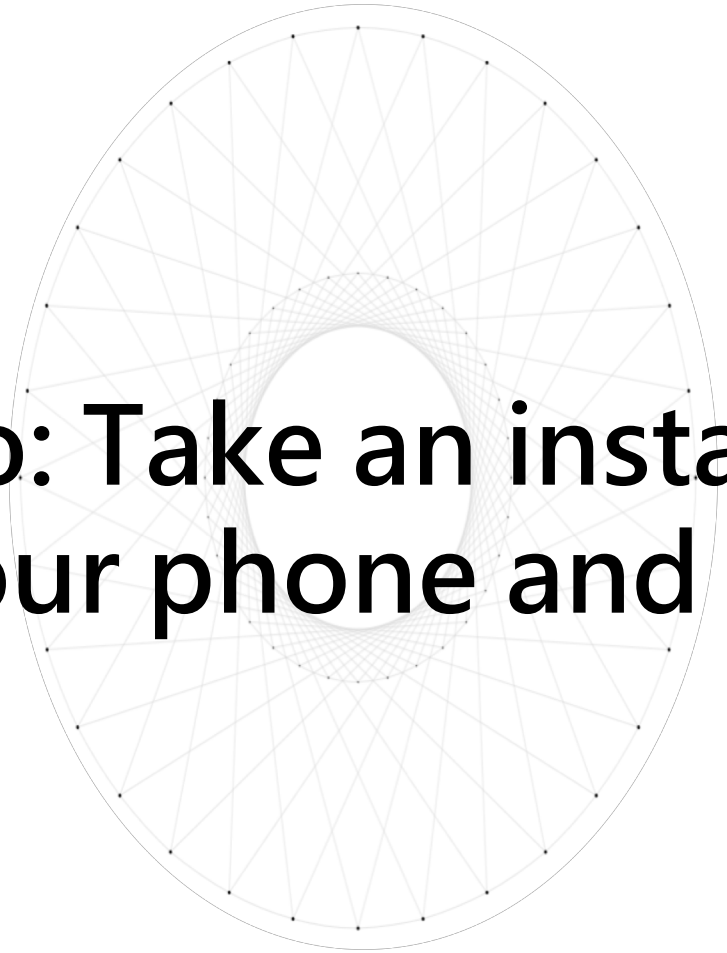
Demo

PART FOUR

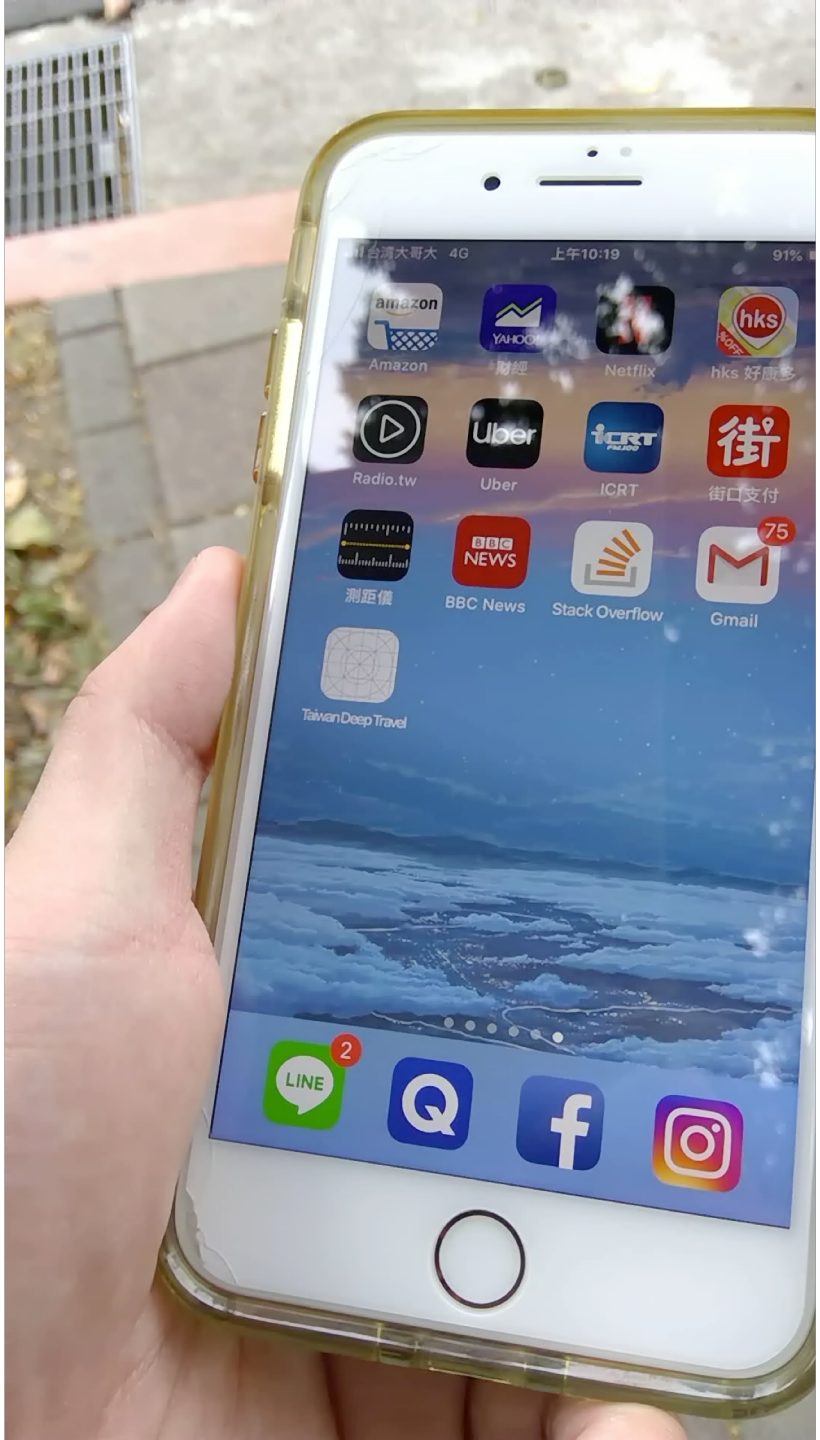


Video One: Input an image from your phone’s library


台湾大哥大 4G
上午8:32
100%
Amazon
財經
Netflix
hks 好康多
Radio.tw
Uber
ICRT
街口支付
測距儀
BBC News
Stack Overflow
Gmail
75
Taiwan Deep Travel
LINE
3



Video Two: Take an instant picture with your phone and input it


台湾大哥大 4G
上午10:19
91%
Amazon
財經
Netflix
hks 好康多
Radio.tw
Uber
ICRT
街口支付
測距儀
BBC News
Stack Overflow
75
Gmail
Taiwan Deep Travel
LINE
2

System Stability

Normal

TOP 1

TOP 2

TOP 3

TOP 4

TOP 5

Rotate 45 Degrees

TOP 1

TOP 2

TOP 3

TOP 4

TOP 5

System Stability

Normal

TOP 1

TOP 2

TOP 3

TOP 4

TOP 5

Extremely Bright

TOP 1

TOP 2

TOP 3

TOP 4

TOP 5

System Stability

Normal

TOP 1

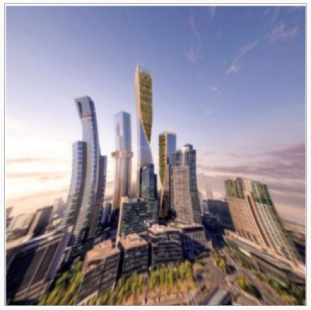
TOP 2

TOP 3

TOP 4

TOP 5

Twisted



TOP 1

TOP 2

TOP 3

TOP 4

TOP 5

Research Prospect

PART FIVE

1. Use more scenes to train the model to improve its ability to identify scenes and the accuracy of recommended scenic spots
2. Augment the training dataset for model training to improve the accuracy of validation and test data
3. Add more hyperparameter combinations for comparison
4. Use TensorBoard to visualize neural networks
5. Use more Open Data to augment recommended attractions
6. Extend the system to the Android mobile operating system
7. Collect users' comments and suggestions on the APP

References

PART SIX

- Wen-Lin Hung. 2016. Individual Recommender System based on Image Recognition using Deep Learning- A case study on Clothing Style, Master Thesis. Department of Engineering Science, College of Engineering, National Cheng Kung University
- Keras Documentation. Available at: keras.io/applications/
- Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 2015. Deep Residual Learning for Image Recognition
- Shao-Hua Sun. 2017. SELUs (scaled exponential linear units) - Visualized and Histogramed Comparisons among ReLU and Leaky ReLU. Available at: github.com/shaohua0116/Activation-Visualization-Histogram
- Justin.h. 2017.图片数据集太少？看我七十二变，Keras Image Data Augmentation 各参数详解. Available at: github.com/JustinhoCHN/keras-image-data-augmentation

- MIT CSAIL Com. 2018. Release of Places365-CNNs. Available at: github.com/CSAILVision/places365n

- Stackoverflow. 2018. Cosine Similarity between 2 Number Lists. Available at: stackoverflow.com/questions/18424228/cosine-similarity-between-2-number-lists

- Audrey Tam. 2018. Beginning Machine Learning with Keras & Core ML. Available at: www.raywenderlich.com/188-beginning-machine-learning-with-keras-core-ml

Taiwan Deep Travel - Deep Learning-Based Recommendation APP for Taiwan Tourist Attractions

Speakers

Chi, Lee — Department of Psychology, SCU

Jane Han, Wu — Department of Economics, SCU

Bing Sheng, Gao — Department of Computer Science and Information Management, SCU

Advisors

Prof. Ching-Shoei, Chiang — Department of Computer Science and Information Management, SCU

Prof. Yi-Ping, Chang — Department of Financial Engineering and Actuarial Mathematics, SCU