



**Faculty of Science**  
Srinakharinwirot University

เรียน AI แล้วไปสร้างเกม

- เรียน AI
- สร้างเกม



**ภาควิชาวิทยาการคอมพิวเตอร์**

คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

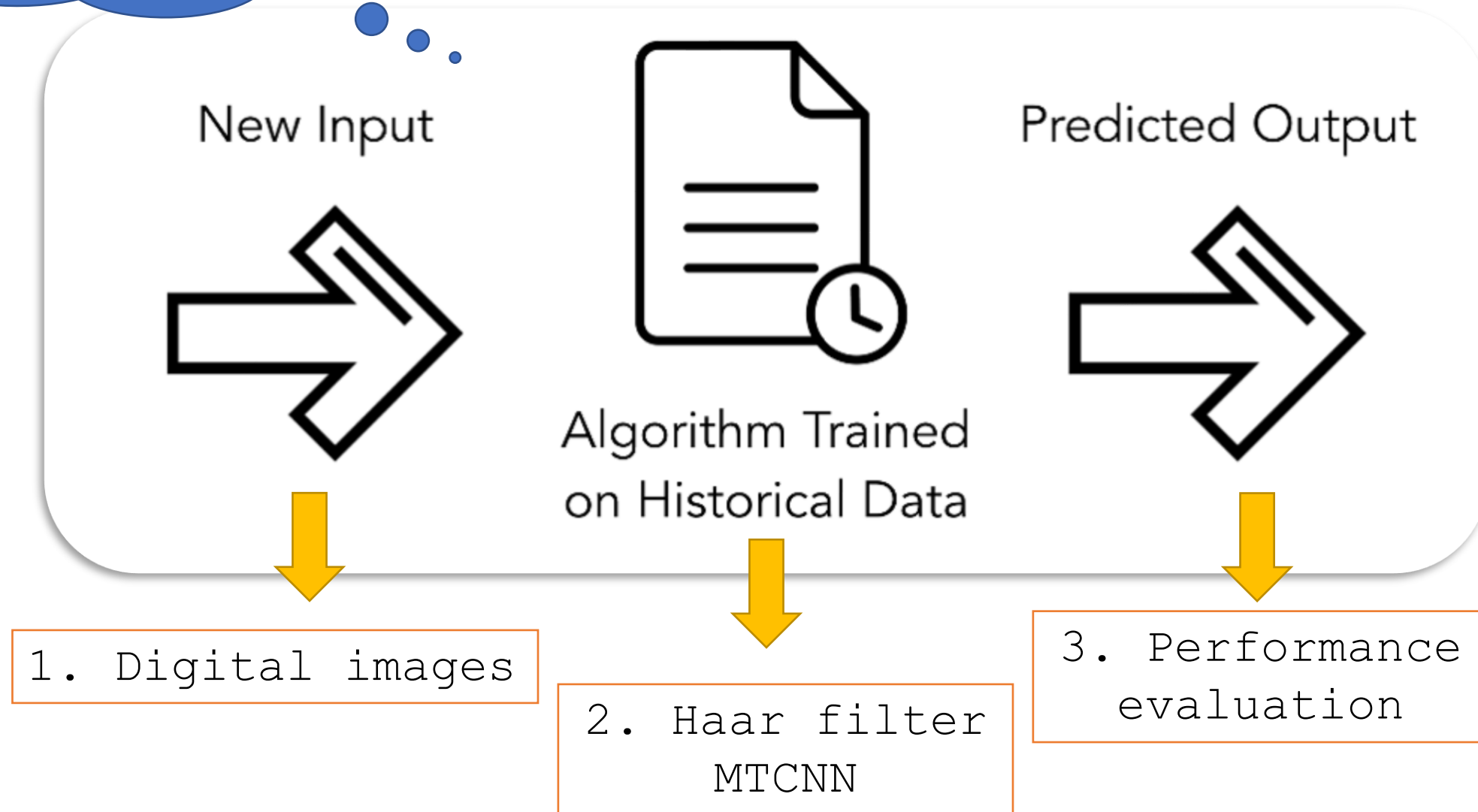
# เรียนรู้ AI จดจำใบหน้า Face detection

อ. นภา แซ่เป้, ผศ. ศศิวิมล สุขพัฒน์, อ. เรืองศักดิ์ ตระกูลพุทธิรักษ์

# How AI works



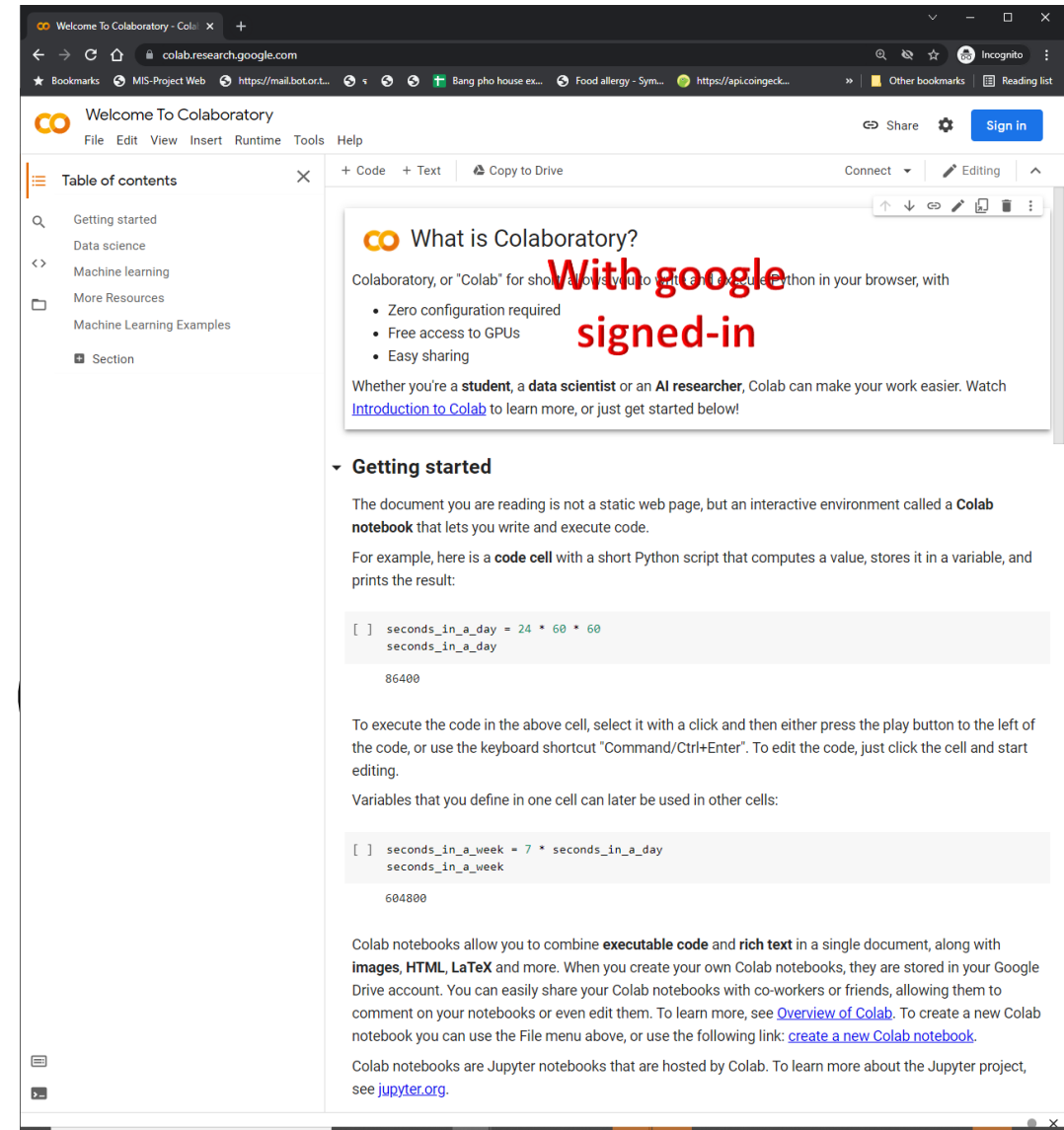
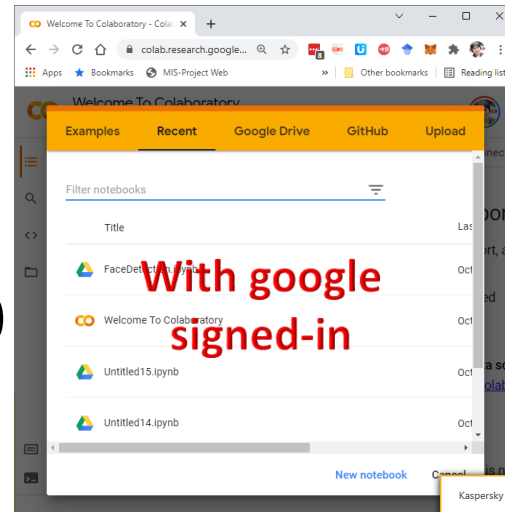
ภาควิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ



# Google-colab introduction

- <https://colab.research.google.com/>

Colab allows anybody to write and execute arbitrary python code through the browser

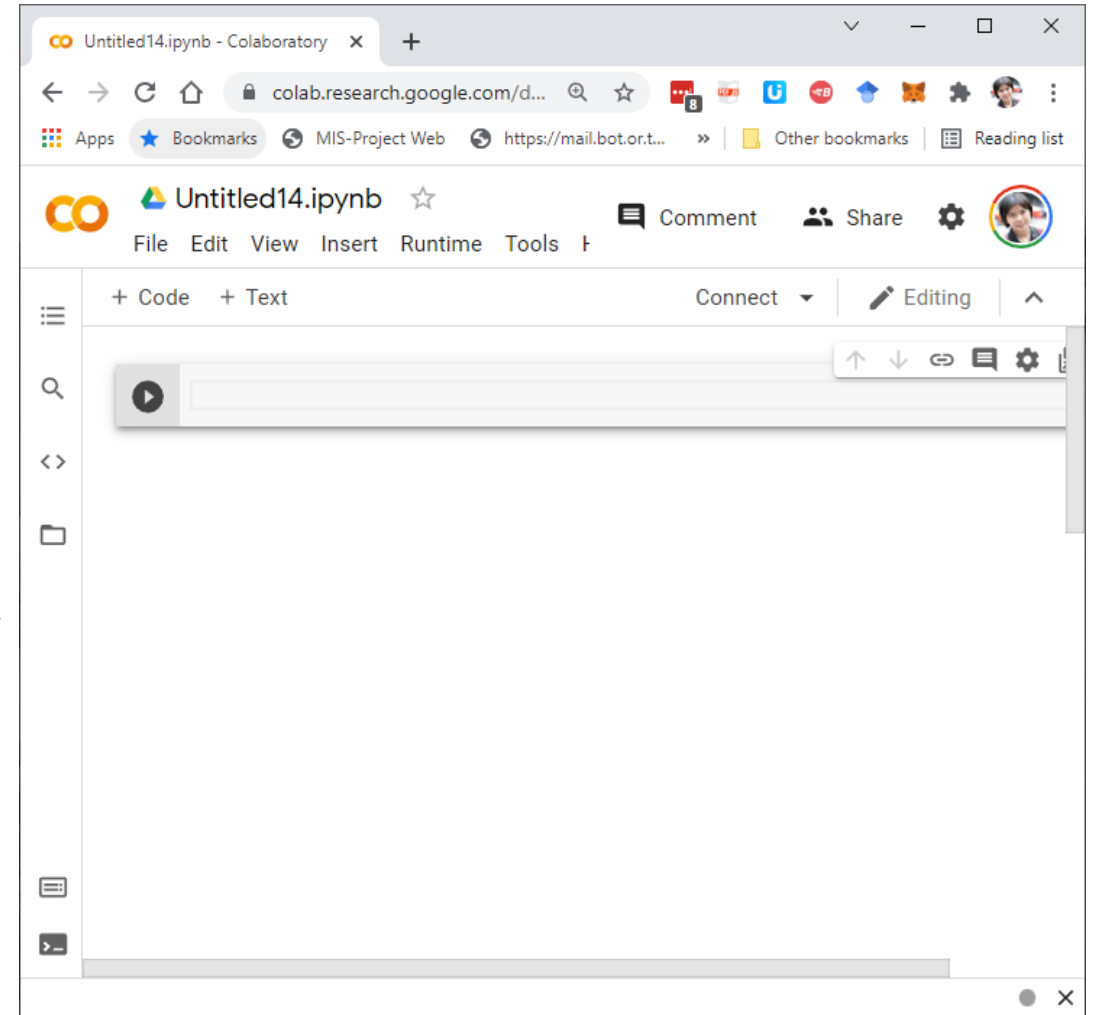
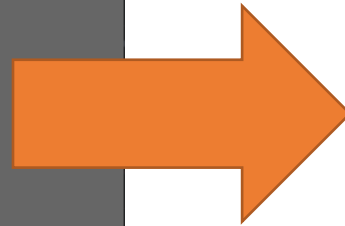
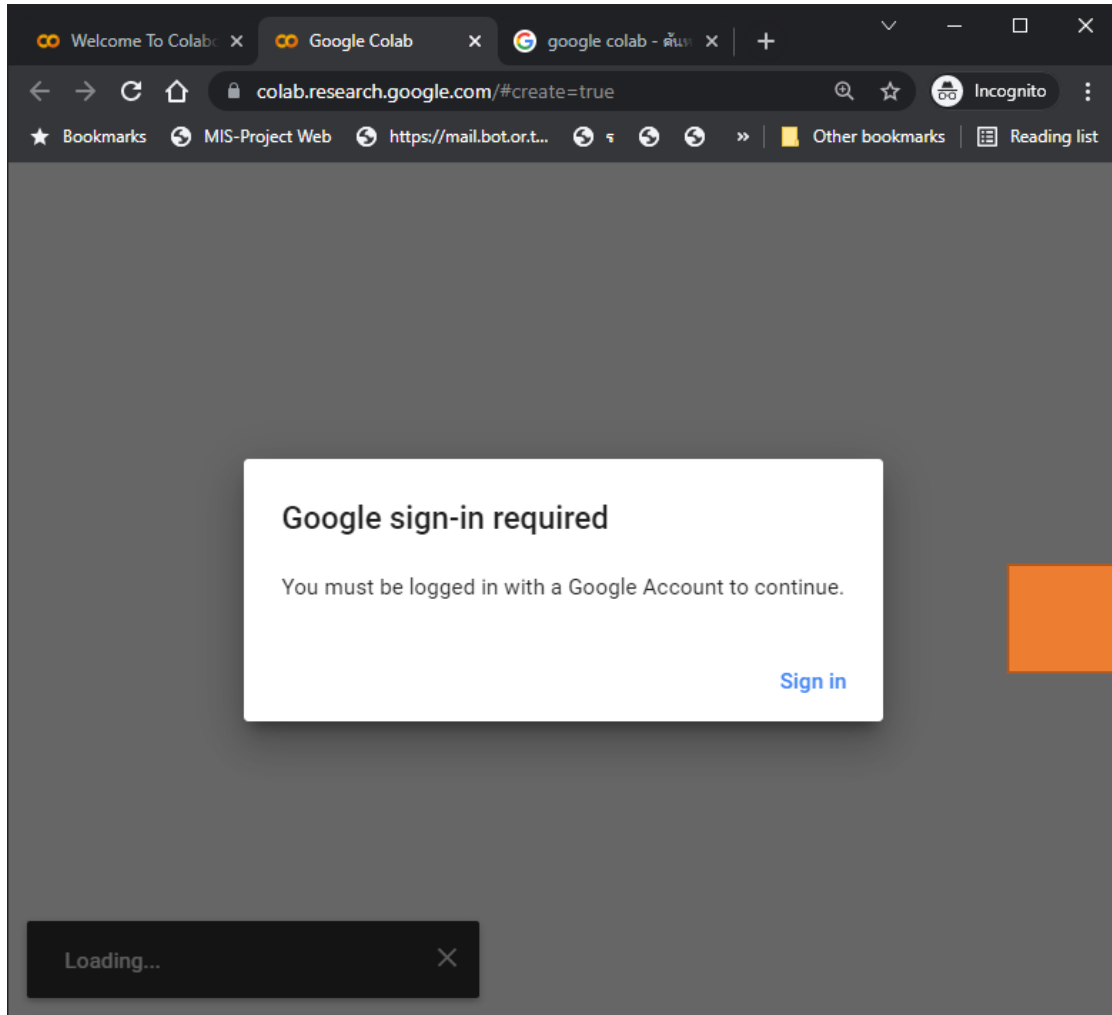




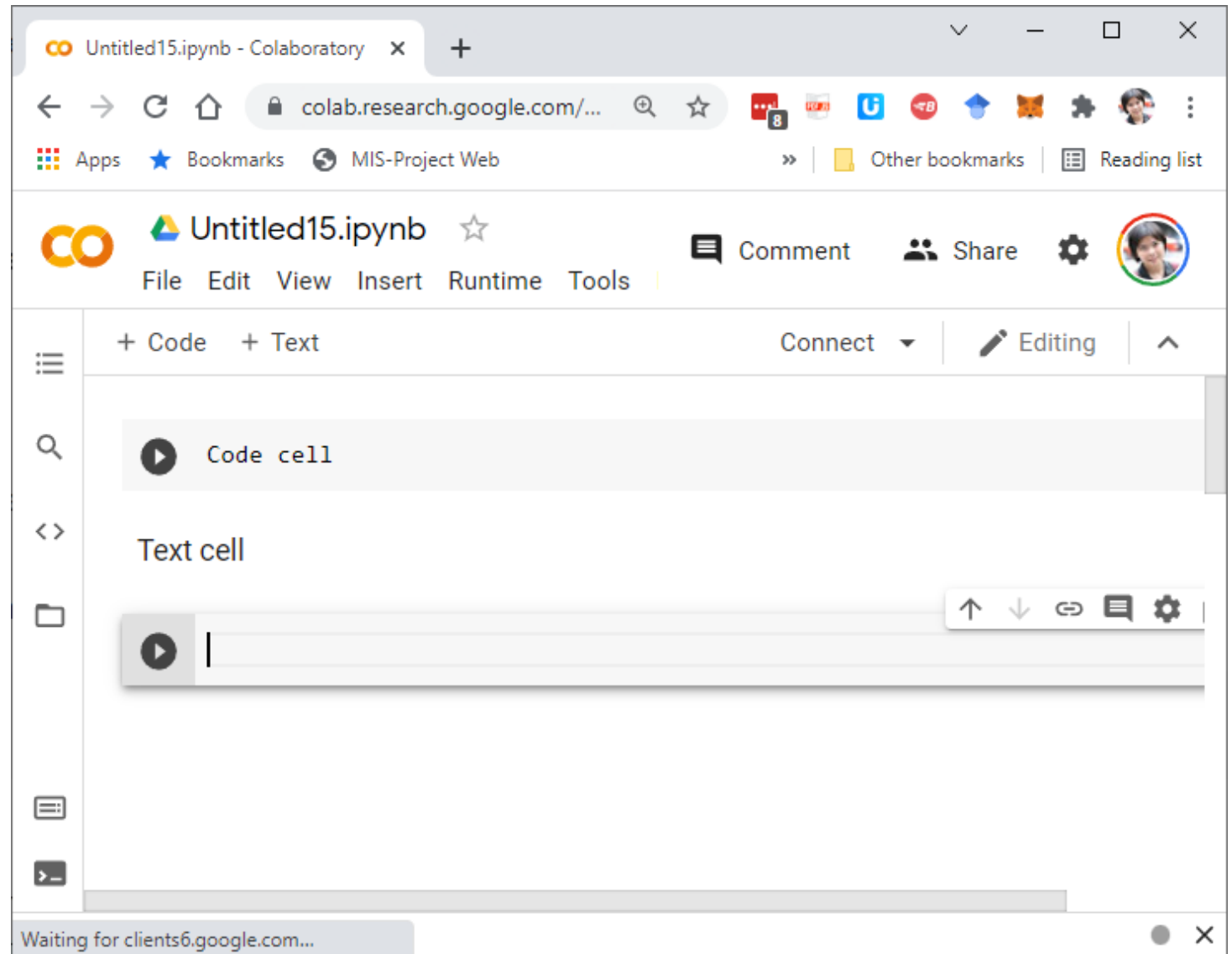
# New notebook

The image shows a web browser window displaying the Google Colaboratory (Colab) interface. The browser's address bar shows the URL `colab.research.google.com`. The Colab interface includes a top navigation bar with 'Welcome To Colaboratory', 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help' menus. A 'Sign in' button is visible in the top right. On the left, there is a 'Table of contents' sidebar with links to 'Getting started', 'Data science', 'Machine learning', 'More Resources', and 'Machine Learning Examples'. The main content area displays 'What is Colaboratory?' with a description and a list of features: 'Zero configuration required', 'Free access to GPUs', and 'Easy sharing'. A red rectangle highlights the 'File' menu, which is open, showing options: 'New notebook', 'Open notebook', 'Upload notebook', 'Rename notebook', 'Move to trash', 'Save a copy in Drive', 'Save a copy as a GitHub Gist', and 'Save a copy in GitHub'. The 'New notebook' option is circled in red.

To execute python code in google colab ,  
users are required to sign-in to google account first.



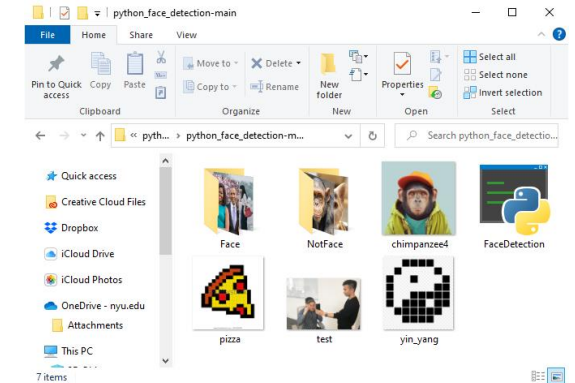
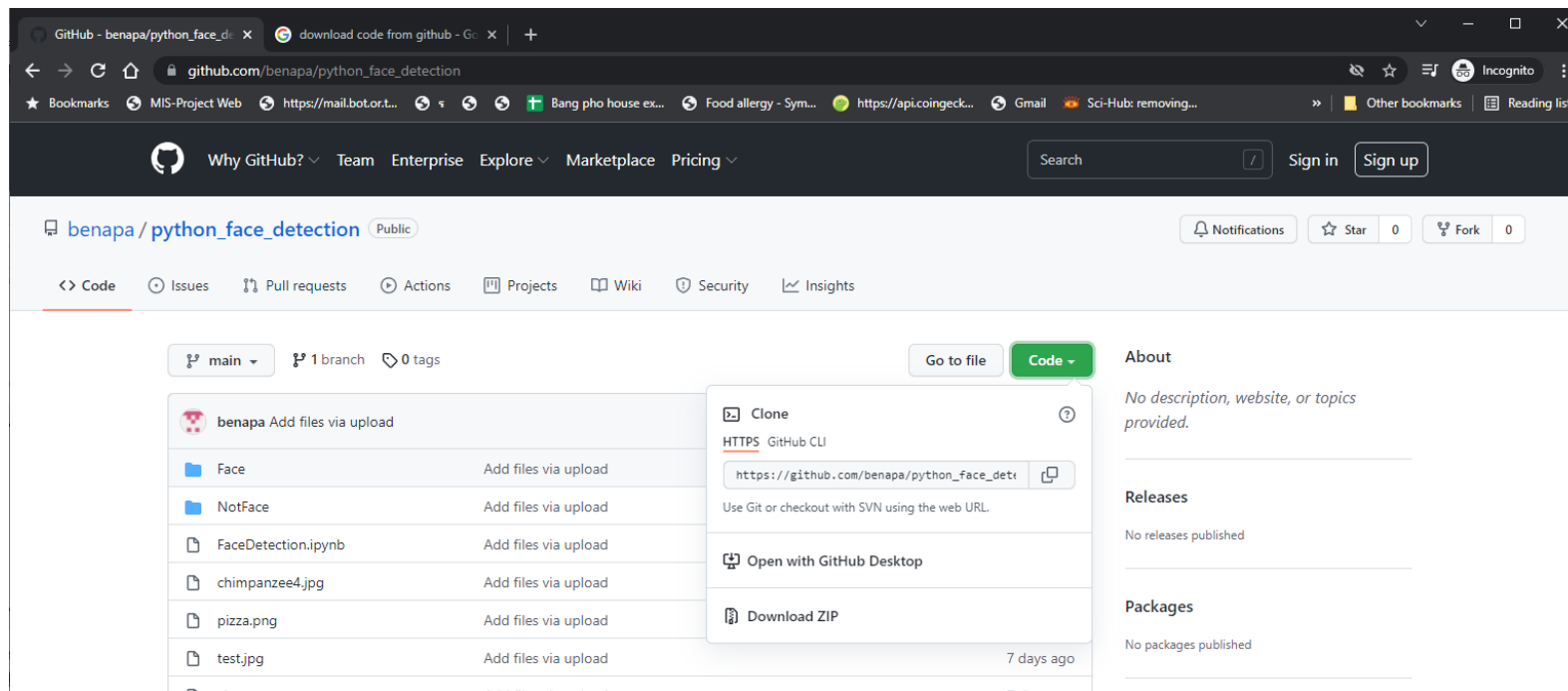
- Colab notebooks allow combination of
  - **executable code** and
  - **rich text**in a single document





# Get all files (\*.zip) from github

- Go to: [https://github.com/benapa/python\\_face\\_detection](https://github.com/benapa/python_face_detection)
- Unzip all files





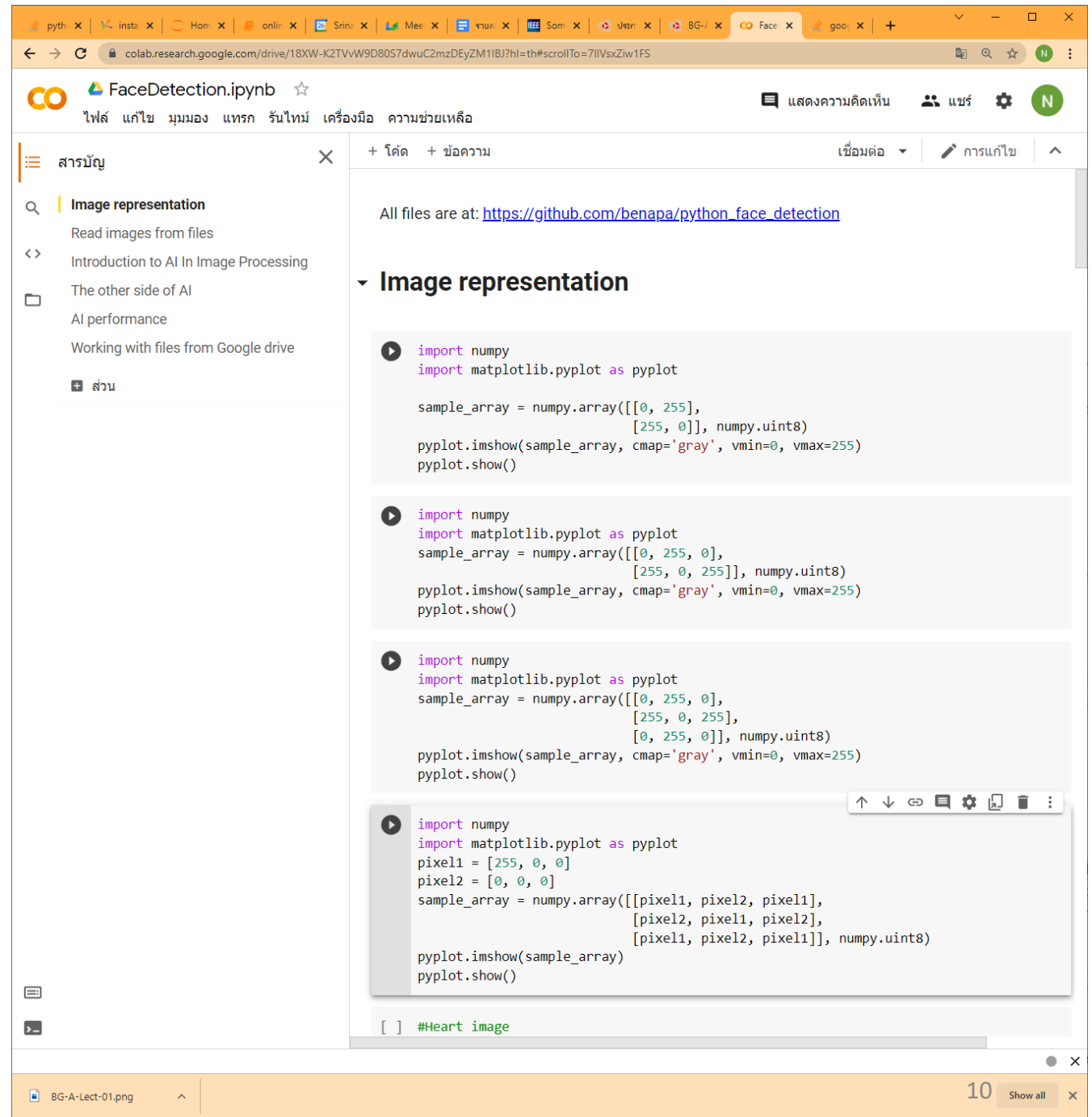


# Upload notebook to colab

The image shows a web browser window displaying the Google Colaboratory (Colab) interface. The browser's address bar shows the URL `colab.research.google.com`. The Colab interface includes a top navigation bar with 'Welcome To Colaboratory', 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help' menus. A 'Sign in' button is visible in the top right. On the left, there is a 'Table of contents' sidebar with links to 'Getting started', 'Data science', 'Machine learning', 'More Resources', and 'Machine Learning Examples'. The main content area displays a 'What is Colaboratory?' section, explaining that Colab allows writing and executing Python in the browser, with features like zero configuration, free GPU access, and easy sharing. Below this, the 'Getting started' section is partially visible. A 'File' menu is open, showing options: 'New notebook', 'Open notebook' (with a keyboard shortcut ⌘/Ctrl+O), 'Upload notebook' (highlighted), 'Rename notebook', and 'Move to trash'. The 'Upload notebook' option is the focus of the tutorial.

# 1. Upload

## “FaceDetection.ipynb”



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: `colab.research.google.com/drive/18XW-K2TVvW9D80S7dWuC2mzDEyZM1BJ?hl=th#scrollTo=7lIVsxZiw1FS`. The notebook title is "FaceDetection.ipynb".

On the left sidebar, under "Image representation", the following files are listed:

- Read images from files
- Introduction to AI In Image Processing
- The other side of AI
- AI performance
- Working with files from Google drive
- ส่วน

The main content area shows four code cells, each starting with `import numpy` and `import matplotlib.pyplot as pyplot`.

Cell 1:

```
sample_array = numpy.array([[0, 255],
                             [255, 0]], numpy.uint8)
pyplot.imshow(sample_array, cmap='gray', vmin=0, vmax=255)
pyplot.show()
```

Cell 2:

```
sample_array = numpy.array([[0, 255, 0],
                             [255, 0, 255]], numpy.uint8)
pyplot.imshow(sample_array, cmap='gray', vmin=0, vmax=255)
pyplot.show()
```

Cell 3:

```
sample_array = numpy.array([[0, 255, 0],
                             [255, 0, 255],
                             [0, 255, 0]], numpy.uint8)
pyplot.imshow(sample_array, cmap='gray', vmin=0, vmax=255)
pyplot.show()
```

Cell 4:

```
pixel1 = [255, 0, 0]
pixel2 = [0, 0, 0]
sample_array = numpy.array([[pixel1, pixel2, pixel1],
                             [pixel2, pixel1, pixel2],
                             [pixel1, pixel2, pixel1]], numpy.uint8)
pyplot.imshow(sample_array)
pyplot.show()
```

At the bottom of the notebook, there is a comment: `[ ] #Heart image`.

The bottom status bar shows a file named "BG-A-Lect-01.png" and a page number "10".

# Image representation

## 2. Run the first cell

colab.research.google.com/drive/18XW-K2TVvW9D80S7dwuC2mzDEyZM1IBJ?hl=th#scrollTo=SRjyoms1wvIC

FaceDetection.ipynb

ไฟล์ แก๊ไข มุมมอง แทรก รันใหม่ เครื่องมือ ความช่วยเหลือ

สารบัญ

- Image representation
  - Read images from files
  - Introduction to AI In Image Processing
  - The other side of AI
  - AI performance
  - Working with files from Google drive
- ส่วน

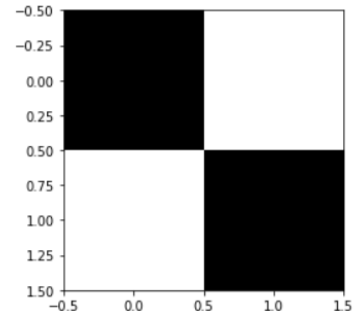
All files are at: [https://github.com/benapa/python\\_face\\_detection](https://github.com/benapa/python_face_detection)

### Image representation

```
import numpy
import matplotlib.pyplot as pyplot

sample_array = numpy.array([[0, 255],
                             [255, 0]], numpy.uint8)

pyplot.imshow(sample_array, cmap='gray', vmin=0, vmax=255)
pyplot.show()
```



```
[ ] import numpy
import matplotlib.pyplot as pyplot
sample_array = numpy.array([[0, 255, 0],
                             [255, 0, 255]], numpy.uint8)

pyplot.imshow(sample_array, cmap='gray', vmin=0, vmax=255)
pyplot.show()
```

```
[ ] import numpy
import matplotlib.pyplot as pyplot
sample_array = numpy.array([[0, 255, 0],
                             [255, 0, 255],
                             [0, 255, 0]], numpy.uint8)

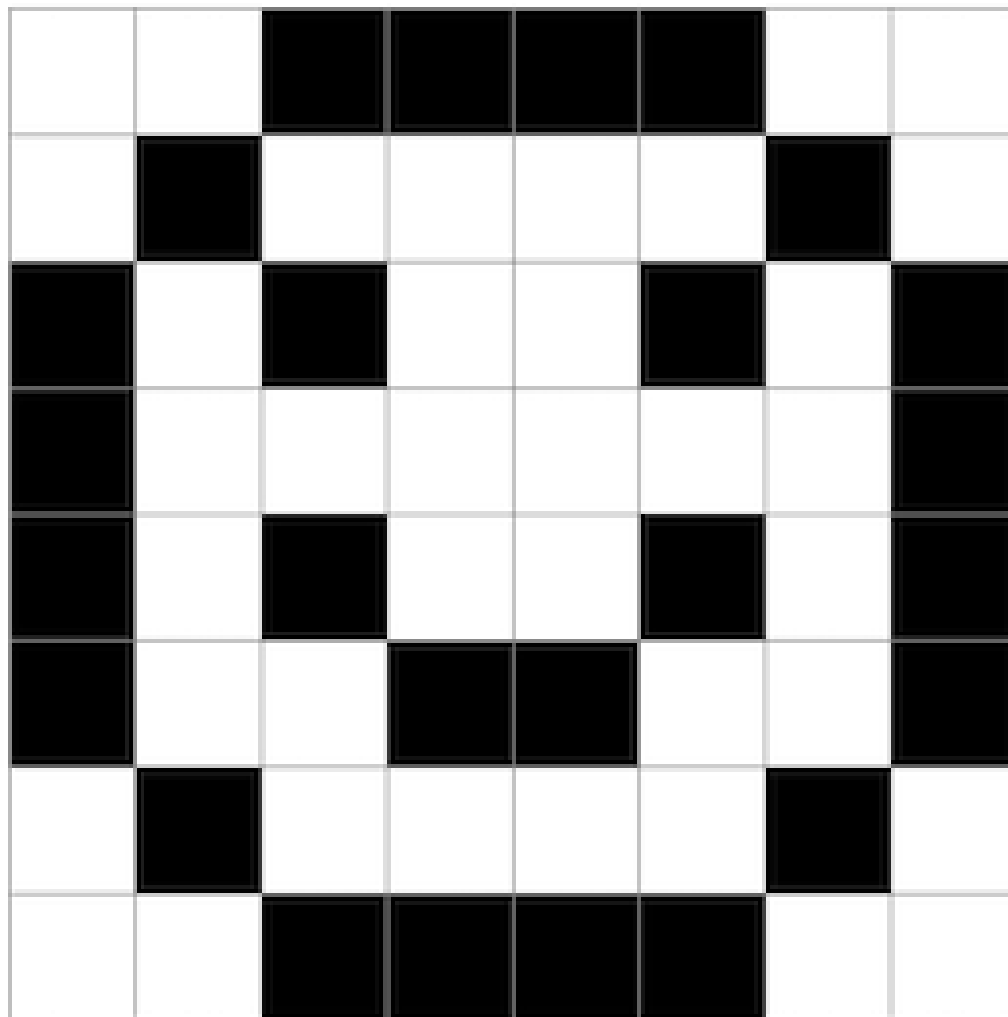
pyplot.imshow(sample_array, cmap='gray', vmin=0, vmax=255)
```

1 รันที่ เสร็จสมบูรณ์เมื่อ 16:21

11

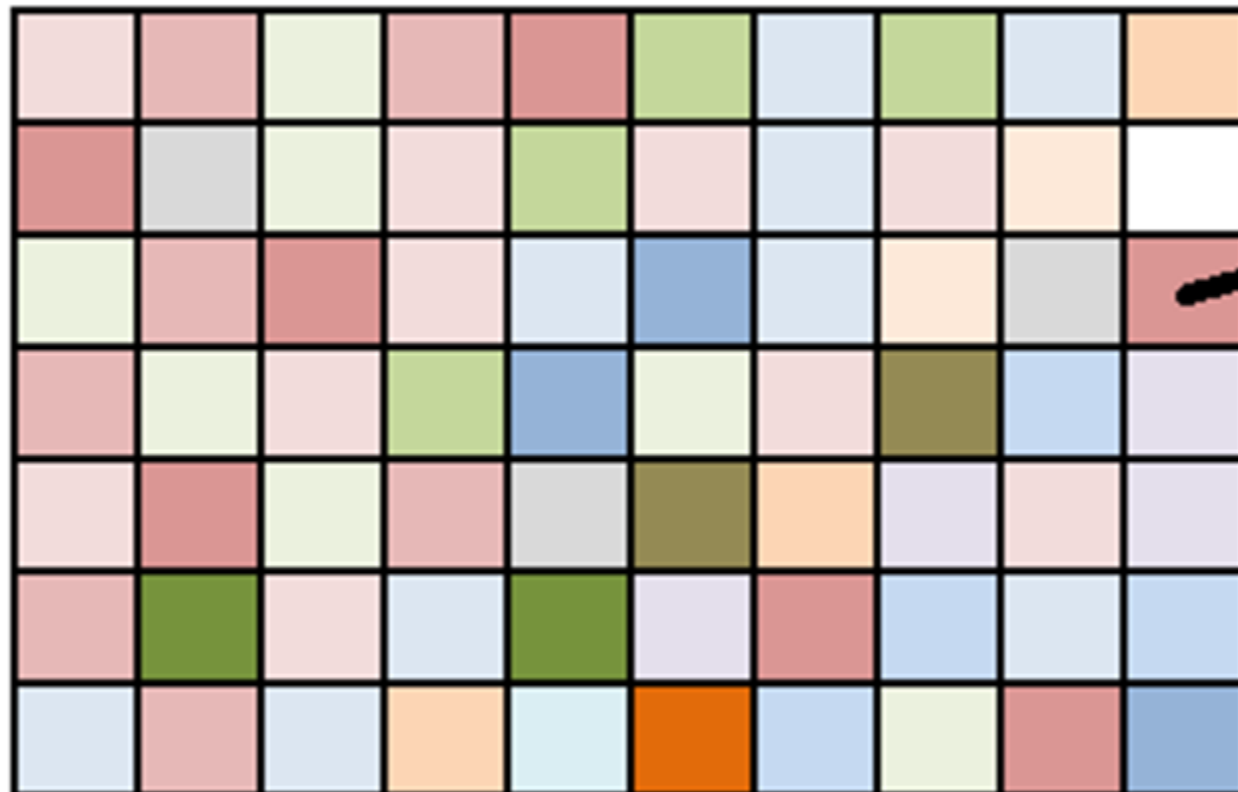
BG-A-Lect-01.png

Show all





# Color image



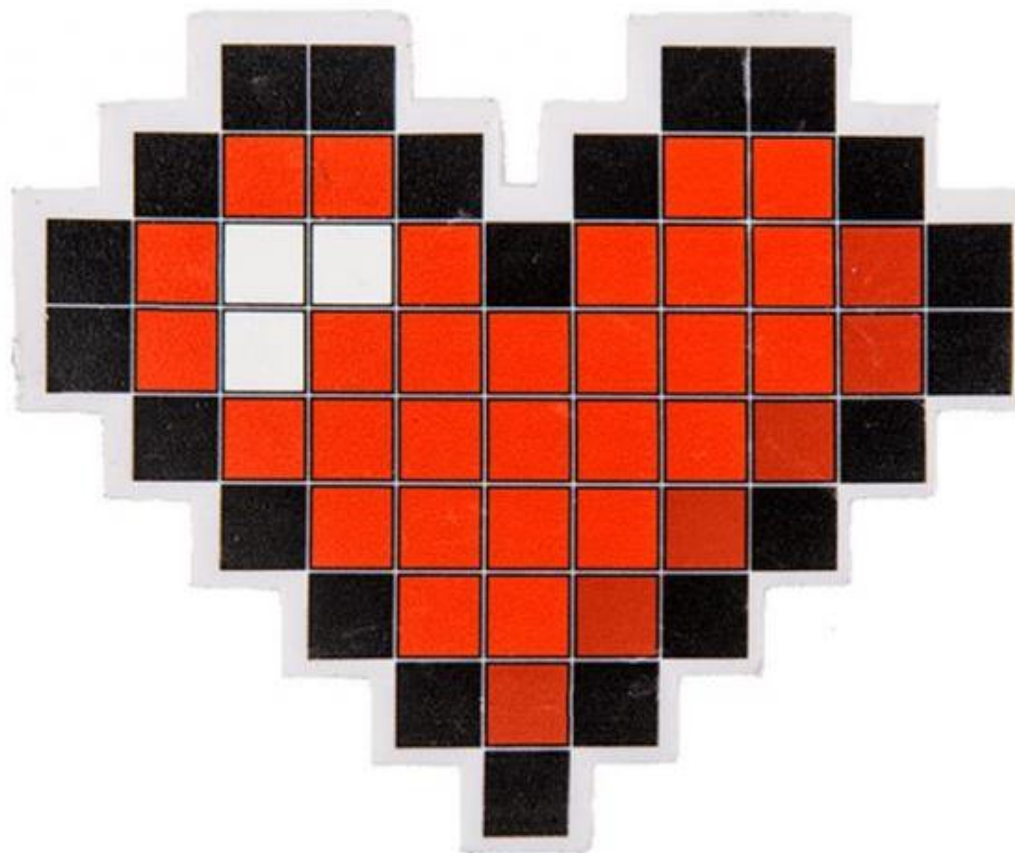
RGB (218, 150, 149)

## ANSI / ASCII RGB Color Chart

0, 0, 0					
170, 0, 0	128, 0, 0	194, 54, 33	187, 0, 0	205, 0, 0	255, 0, 0
0, 170, 0	0, 128, 0	37, 188, 36	0, 187, 0	0, 205, 0	0, 255, 0
170, 85, 0	128, 128, 0	173, 173, 39	187, 187, 0	205, 205, 0	255, 255, 0
0, 0, 170	0, 0, 128	73, 46, 225	0, 0, 187	0, 0, 238	0, 0, 255
170, 0, 170	128, 0, 128	211, 56, 211	187, 0, 187	205, 0, 205	255, 0, 255
0, 170, 170	0, 128, 128	51, 187, 200	0, 187, 187	0, 205, 205	0, 255, 255
170, 170, 170	192, 192, 192	203, 204, 205	187, 187, 187	229, 229, 229	
85, 85, 85	128, 128, 128	129, 131, 131	85, 85, 85	127, 127, 127	
255, 85, 85	255, 0, 0	252, 57, 31	255, 85, 85	255, 0, 0	
85, 255, 85	0, 255, 0	49, 231, 34	85, 255, 85	0, 255, 0	144, 238, 144
255, 255, 85	255, 255, 0	234, 236, 35	255, 255, 85	255, 255, 0	255, 255, 224
85, 85, 255	0, 0, 255	88, 51, 255	85, 85, 255	92, 92, 255	173, 216, 230
255, 85, 255	255, 0, 255	249, 53, 248	255, 85, 255	255, 0, 255	
85, 255, 255	0, 255, 255	20, 240, 240	85, 255, 255	0, 255, 255	224, 255, 255
					255, 255, 255



# Exercise#1 image construction





an image is a  
matrix of  
pixels





# Applications of image processing

- **Object classification:** การแยกประเภทวัตถุจากภาพ



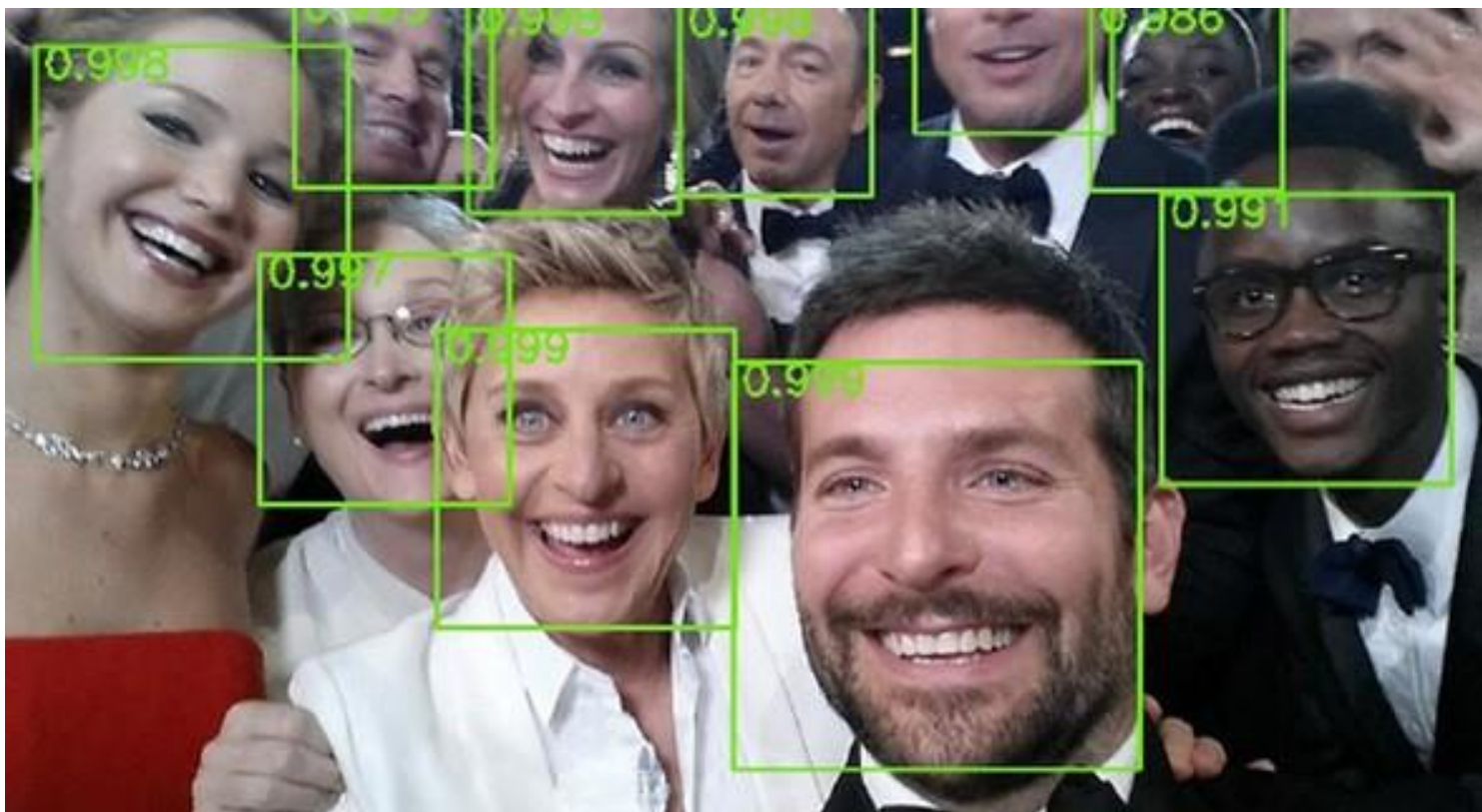
# Applications of image processing

- Object detection: การตรวจจับวัตถุเป้าหมายในภาพ



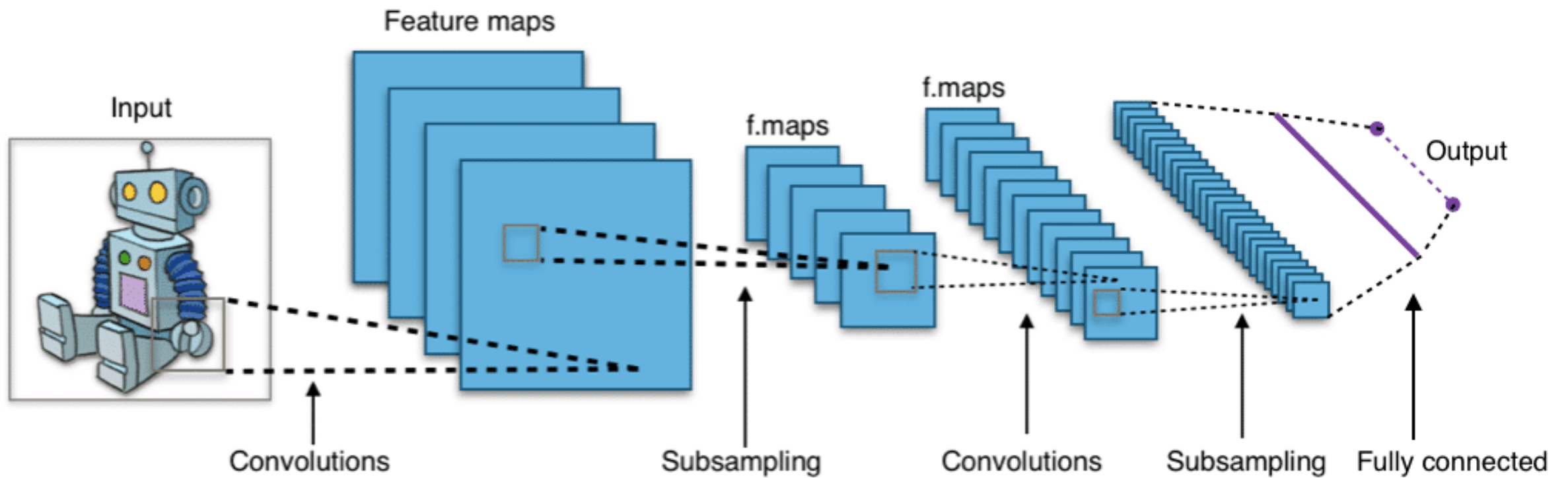
# Face detection

- Object detection: การตรวจจับวัตถุเป้าหมายในภาพ --> ใบหน้า



<https://www.technologyreview.com/2015/02/16/169357/the-face-detection-algorithm-set-to-revolutionize-image-search/>





<https://towardsdatascience.com/hands-on-machine-learning-example-real-time-object-detection-with-yolo-v2-ebdd8441c12a>



# Face detection library

- OpenCV Haar cascades (2001)
  - Developed in 2001
  - Based on edge detection features
- MTCNN
  - Developed in 2016
  - Based on deep learning model (Convolutional Neural Network)

FaceDe x | will sm x | object x | Diagn x | LNCS x | empty x | 2564 x | +

colab.research.google.com/drive/18XW-K2TVvW9D80S7dwuC2mzDEyZM1IBJ?hl=th#scrollTo=cdeQ3D1...

FaceDetection.ipynb ☆

ไฟล์ แก้ไข มุมมอง แทรก รันใหม่ เครื่องมือ ความช่วย

แสดงความคิดเห็น แชร์

RAM ดิสก์

ไฟล์

sample\_data

test.jpg

Read images from files

```
from matplotlib import pyplot

pixels = pyplot.imread('test.jpg')
pyplot.imshow(pixels)
pyplot.show()
```

[ ] pixels

Introduction to AI In Image Processing

ดิสก์ ใช้ได้อีก 61.54 GB

ไฟล์

sample\_data

test.jpg

ดิสก์ 61.54 GB

```
+ โค้ด + ข้อความ
```

```
implementation can be found at: https://pypi.org/project/mtcnn/.
```

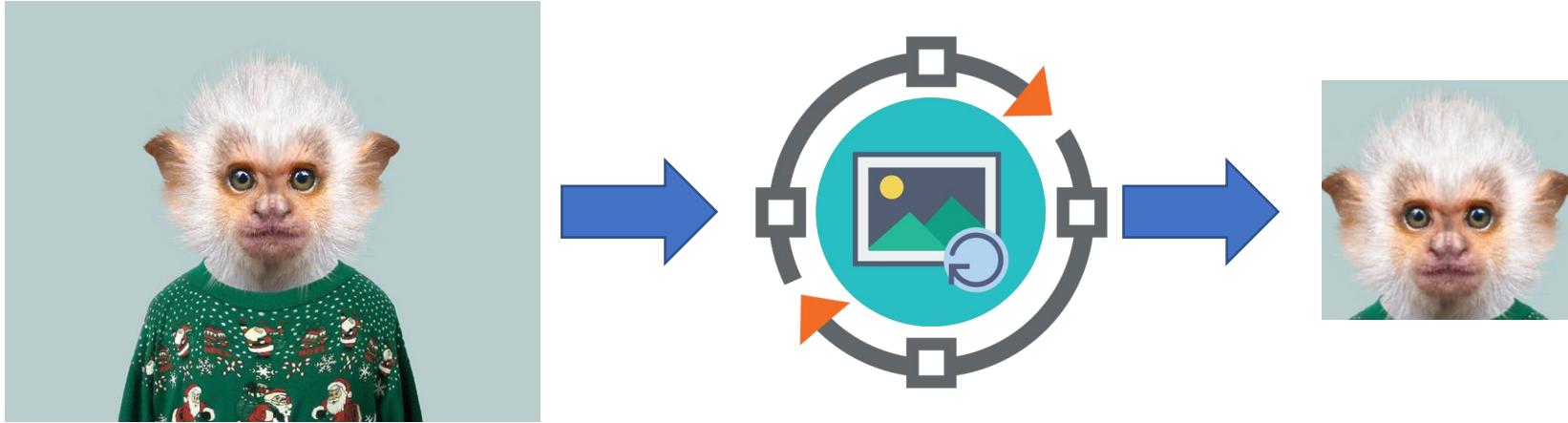
```
!pip install opencv-python
!pip install MTCNN
```

```
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-packages (4.1.2)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-python)
Collecting MTCNN
  Downloading mtcnn-0.1.1-py3-none-any.whl (2.3 MB)
    |████████████████████████████████████████| 2.3 MB 3.7 MB/s
Requirement already satisfied: opencv-python>=4.1.0 in /usr/local/lib/python3.7/dist-packages (from MTCNN)
Requirement already satisfied: keras>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from MTCNN)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from MTCNN)
Installing collected packages: MTCNN
Successfully installed MTCNN-0.1.1
```

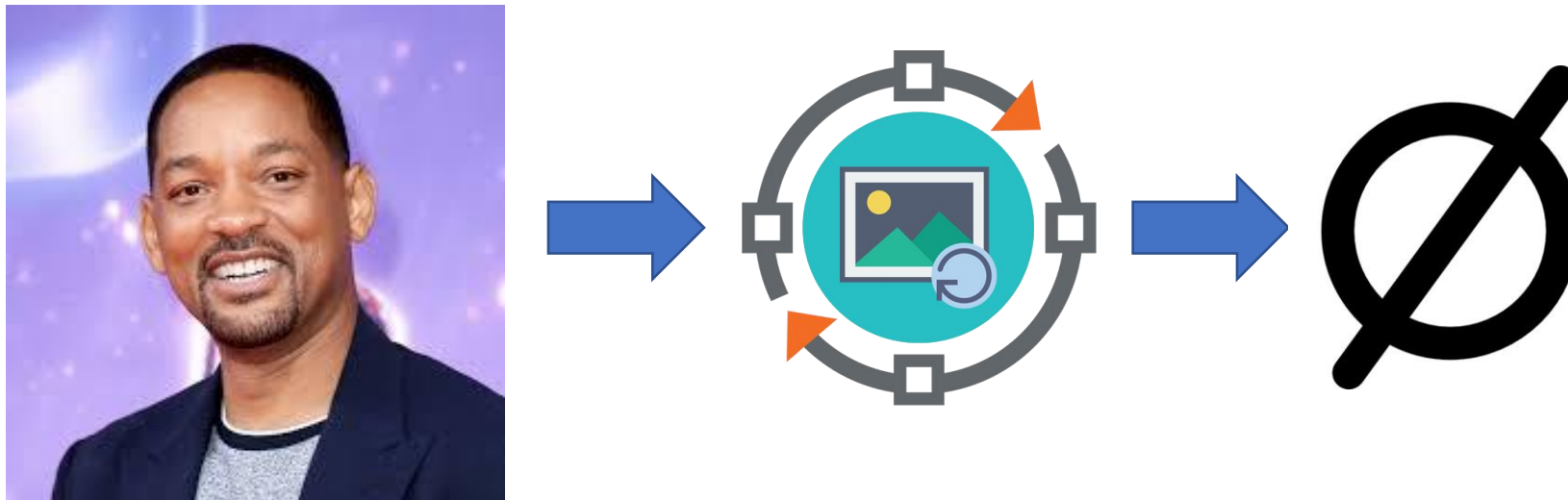
```
[ ] from matplotlib import pyplot
import cv2
```

ความผิดพลาดในการตรวจจับใบหน้า สามารถแบ่งได้เป็น 2 ชนิด คือ

- False detection --> detect non face object as faces



- False rejection - > fail to detect faces from images with faces





# Exercise#2 Face detection performance

- [https://github.com/benapa/python\\_face\\_detection](https://github.com/benapa/python_face_detection)