

Home

Galery

About

Shop



Present By

CodeBotix Learning Hub

# ARTIFICIAL INTELLIGENCE

## LANDING PAGE

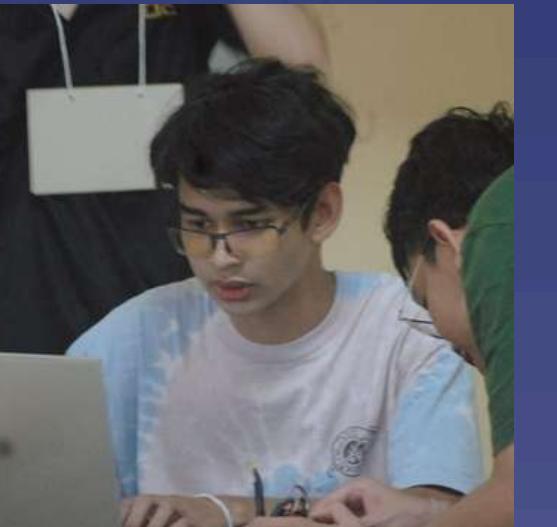
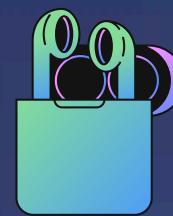
Loreum ipsum dolor sit amet, ius ei ancillae  
partiendo interpretaris, duo et reque dicta  
munere

FIND MORE

# Basic Image Processing And AI



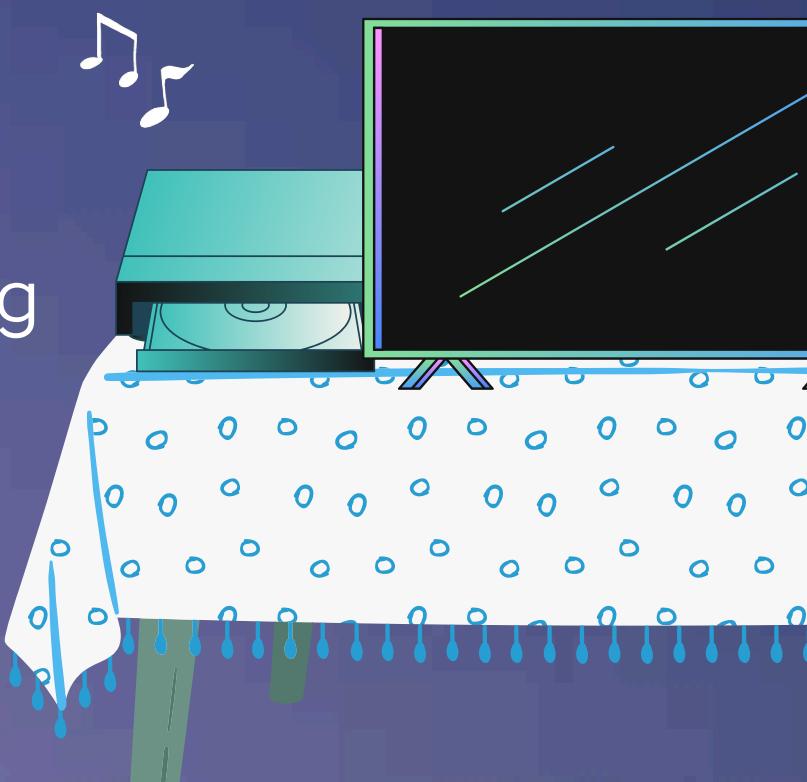
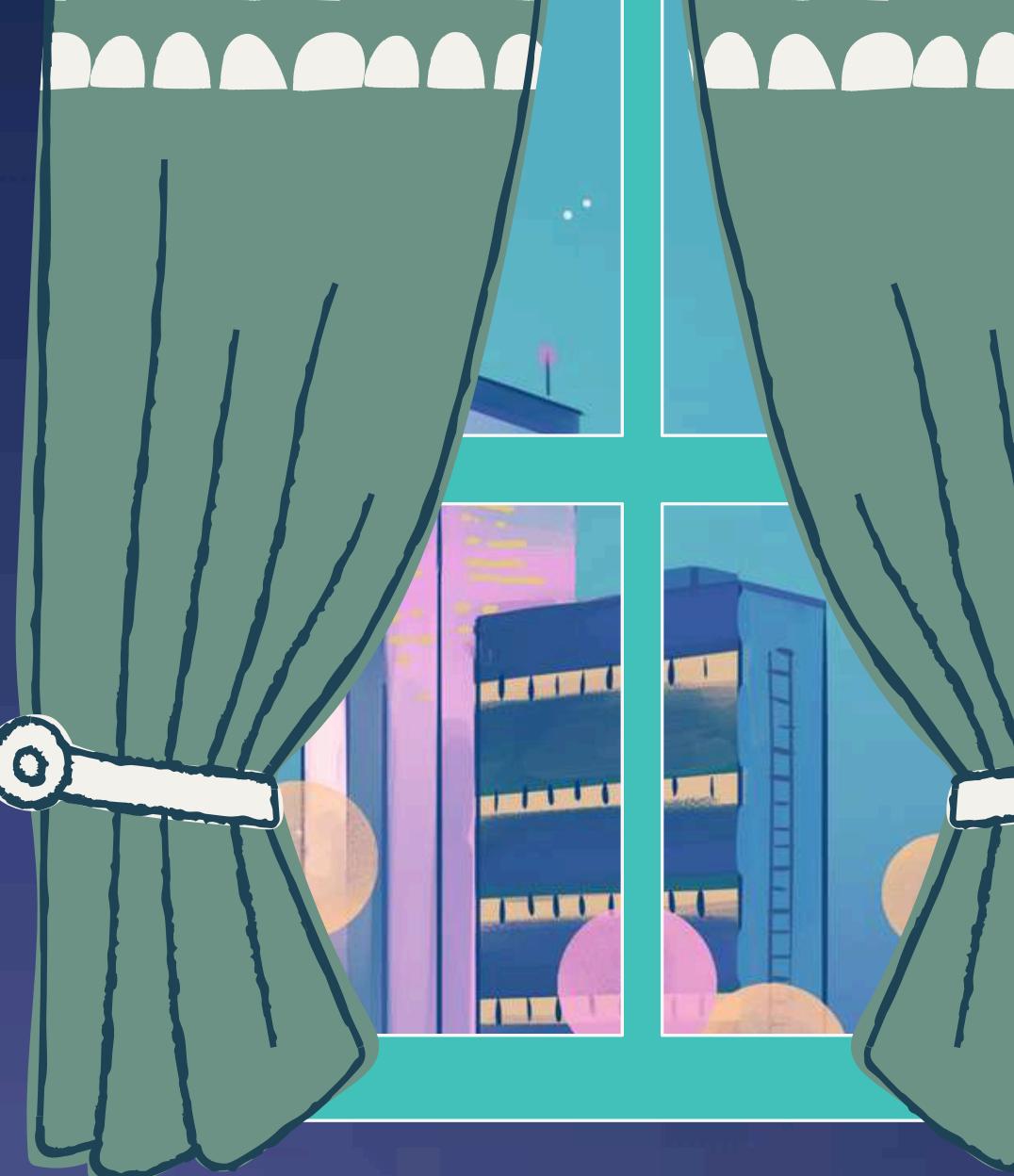
# Let's get to know each other first



P' BLUE

**Bachelor Computer Engineering**

King Mongkut's Institute Of Technology Ladkrabang  
(KMITL)



# Topic

which we could learning today?

1

OpenCV

2

Mediapipe

- - - - >

Mediapipe Holistic  
(guide line)

" Wisdom comes with age, so just live and learn."

By William Landry

# Open-cv

OpenCV  
(Open Source Computer Vision Library)

- is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.



<https://opencv.org/get-started/>



# Face detection

Q1

Install Python



open your command prompt

Q2

```
pip install opencv-python
```

```
pip install numpy
```

```
pip install requests
```

We'll use Visual Studio Code



- Create new file your project (.py file)

# Install Vscode

The screenshot shows the official Visual Studio Code website. At the top, there's a navigation bar with links for 'Visual Studio Code', 'Docs', 'Updates', 'Blog', 'API', 'Extensions', 'FAQ', 'Learn', 'Search Docs', and a prominent blue 'Download' button. A banner at the top right announces 'Version 1.30 is now available! Read about the new features and fix from January'. Below the banner, the main heading 'Code editing. Redefined.' is displayed in large white text, followed by the subtext 'Free. Built on open source. Runs everywhere.' On the left, a large call-to-action button says 'Download for Windows' with a dropdown menu showing options for 'macOS', 'Windows x64', and 'Linux x64'. On the right, a large image of the VS Code interface is shown, featuring a dark theme with various code snippets and extension icons.

open your install

3



Intellisense



Run and Debug



Built-in Git



Extensions

# Q3

## Coding

## Model (face , eye , etc.)

<https://github.com/opencv/opencv/tree/master/data>

```
import cv2
import numpy as np
import requests

xml_model_url = 'https://raw.githubusercontent.com/opencv/opencv/master/data/haarcascades/haarcascade_frontalface_default.xml'
xml_path = 'haarcascade_frontalface_default.xml'

def download_xml(url, path):
    response = requests.get(url)
    if response.status_code == 200:
        with open(path, 'wb') as file:
            file.write(response.content)
        print(f"Downloaded {path} successfully.")
    else:
        raise Exception(f"Failed to download {url}, status code: {response.status_code}")

# Download haarcascade XML file
download_xml(xml_model_url, xml_path)

cascade = cv2.CascadeClassifier(xml_path)
if cascade.empty():
    raise IOError('Unable to load the face cascade classifier xml file')
```

download model path

Initialize the model haarcascades



Link Video Basic AI



# Q3

## Coding

```
# Video Capture Start 'Camera' ---> Source 0
cap = cv2.VideoCapture(0) → Initialize the webcam
while True:
    ret, frame = cap.read()
    if not ret:
        break
```

Read a frame from the webcam

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
objects = cascade.detectMultiScale(
    gray,
    scaleFactor=1.3,
    minNeighbors=5,
    minSize=(30, 30)
)
```

Convert the frame to grayscale

```
for (x, y, w, h) in objects:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

Perform object detection

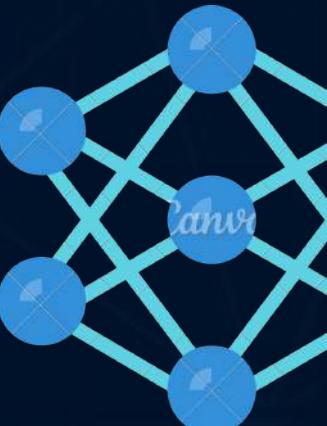
```
cv2.imshow('Face Object Detection', frame)
# Break the loop if 'q' key is pressed
if cv2.waitKey(1) & 0xFF == ord('q'):
```

When press key 'q' it will exist

```
break
# Release the capture and close all OpenCV windows
cap.release()
cv2.destroyAllWindows()
```

## Model (face , eye , etc.)

<https://github.com/opencv/opencv/tree/master/data>



Link Video Basic AI

# Try on your self

## to run program

```
PS C:\Users\bluep\Desktop\AI Learning\Object Detecting_test> python face_detection_opencv.py
```

directory path file already in

python filename.py

อย่าลืม cd ตำแหน่งโฟล์เดอร์ที่ต้อง



TITLE	LAST MODIFIED
data	5/29/23
ch1 colab test.ipynb	10/12/24
ch1_hello.ipynb	10/11/24
Object_Detection_yolov5.ipynb	10/11/24

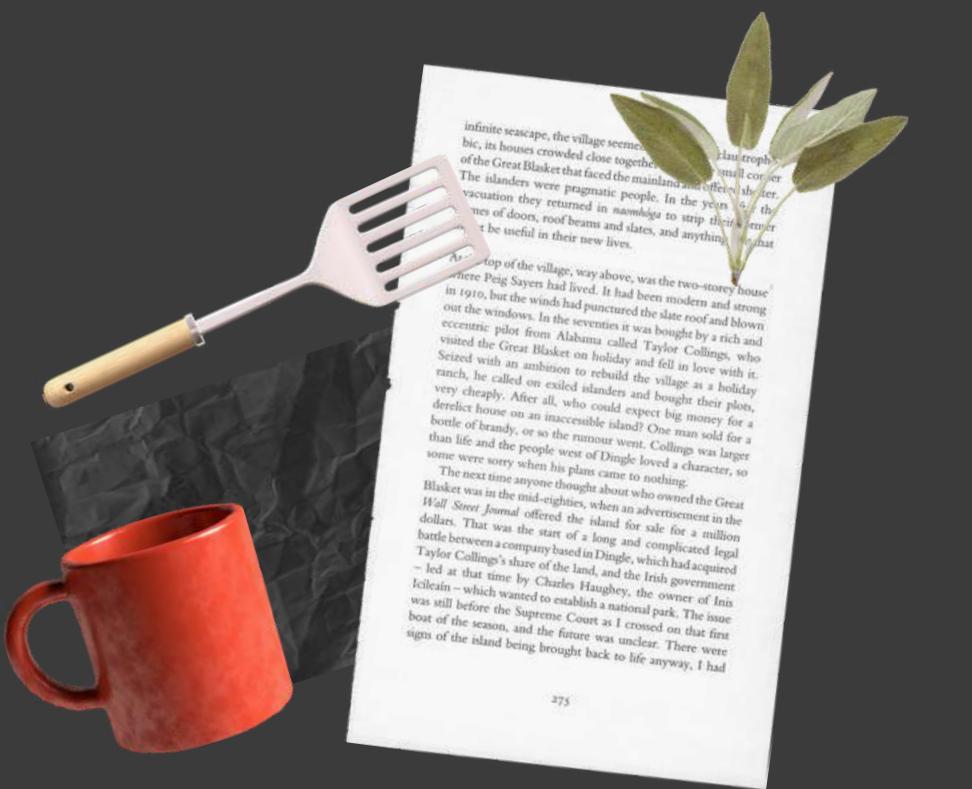
can try on others object example detection on this link : <https://shorturl.at/pHIW6>



# BREAK SLIDES

**LET'S TAKE A BREAK 7 MINUTES!**

- Taking breaks during work is a good thing
- Don't work too hard, get some rest

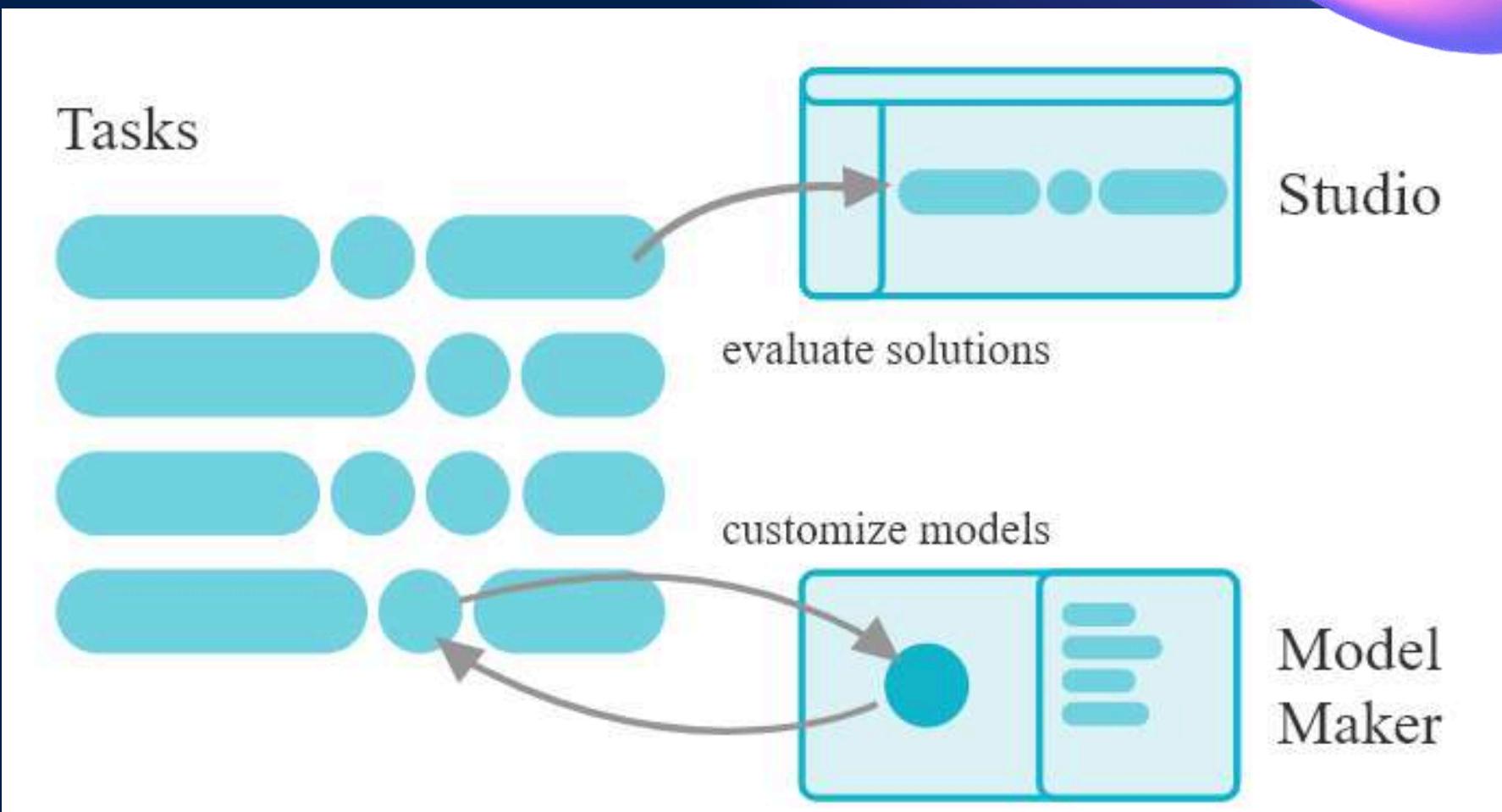


**DO YOU WANT CONTINUE  
DETECTION ANALYZE TYPE >/?<**

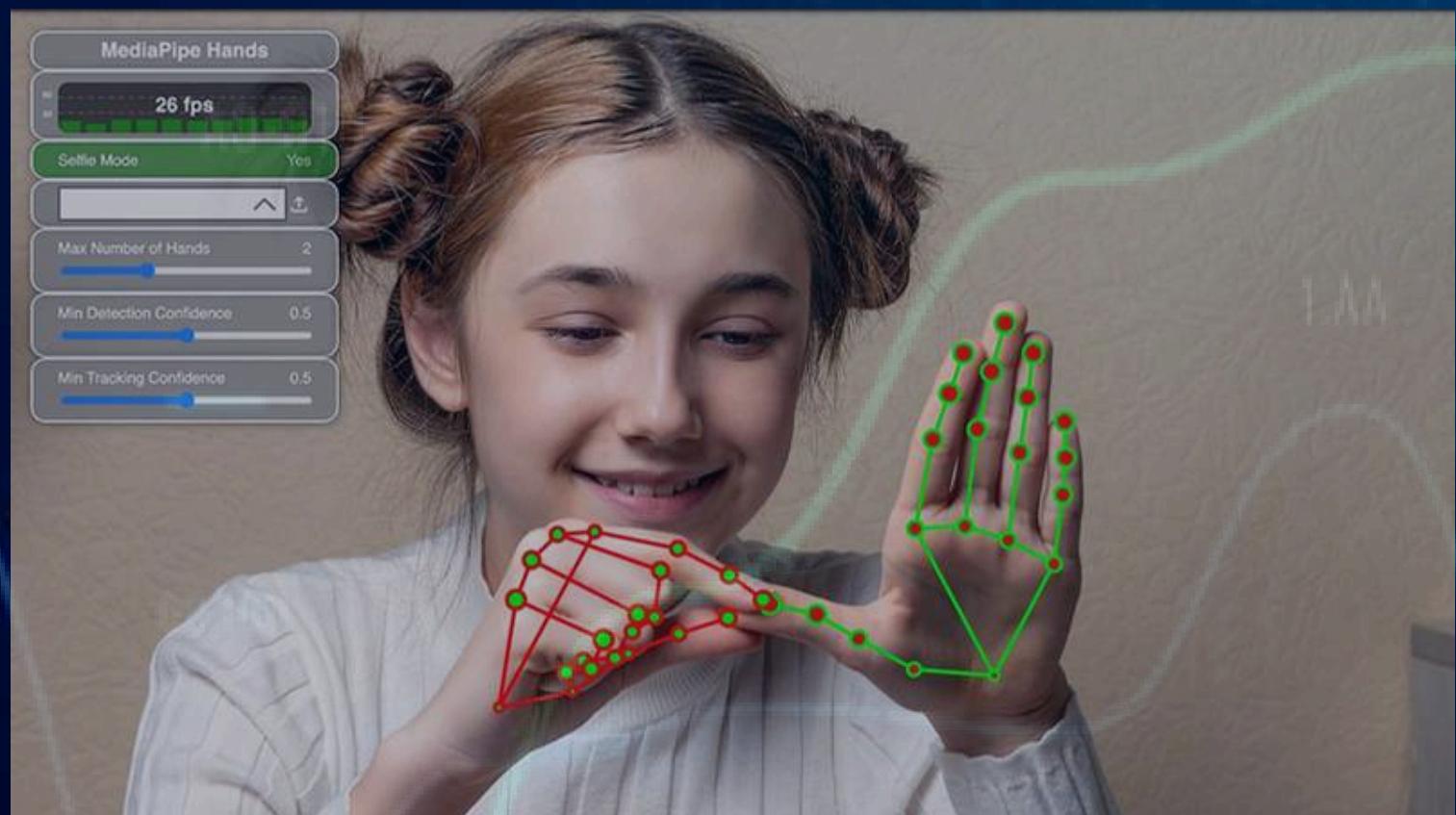
**NEXT PAGE >>**

# Mediapipe is?

- MediaPipe Solutions provides a suite of libraries and tools for you to quickly apply artificial intelligence (AI) and machine learning (ML) techniques in your applications.
- You can plug these solutions into your applications immediately, customize them to your needs, and use them across multiple development platforms. MediaPipe Solutions is part of the MediaPipe open source project, so you can further customize the solutions code to meet your application needs.

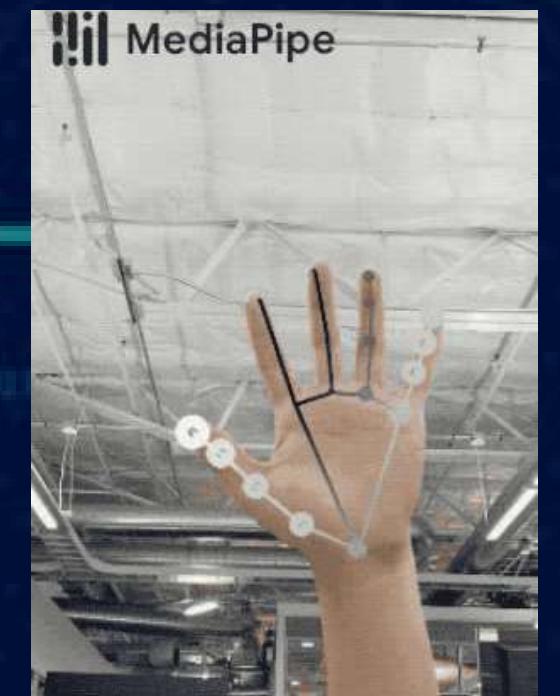


# MediaPipe Hands



Today we will learn to use MediaPipe, which is a Machine Learning Solutions or a ready-made program from Google that can be used to do Hand Tracking accurately. And as fast as real-time tracking.

Because MediaPipe has a number of solutions available, ranging from gesture detection to facial recognition, today we're going to use MediaPipe Hands that detect hands.





## 21 point of finger Landmarks



Hand Tracking starts by sensing the palm with the " Palm Detection Model " and then identifies 21 key hand locations.

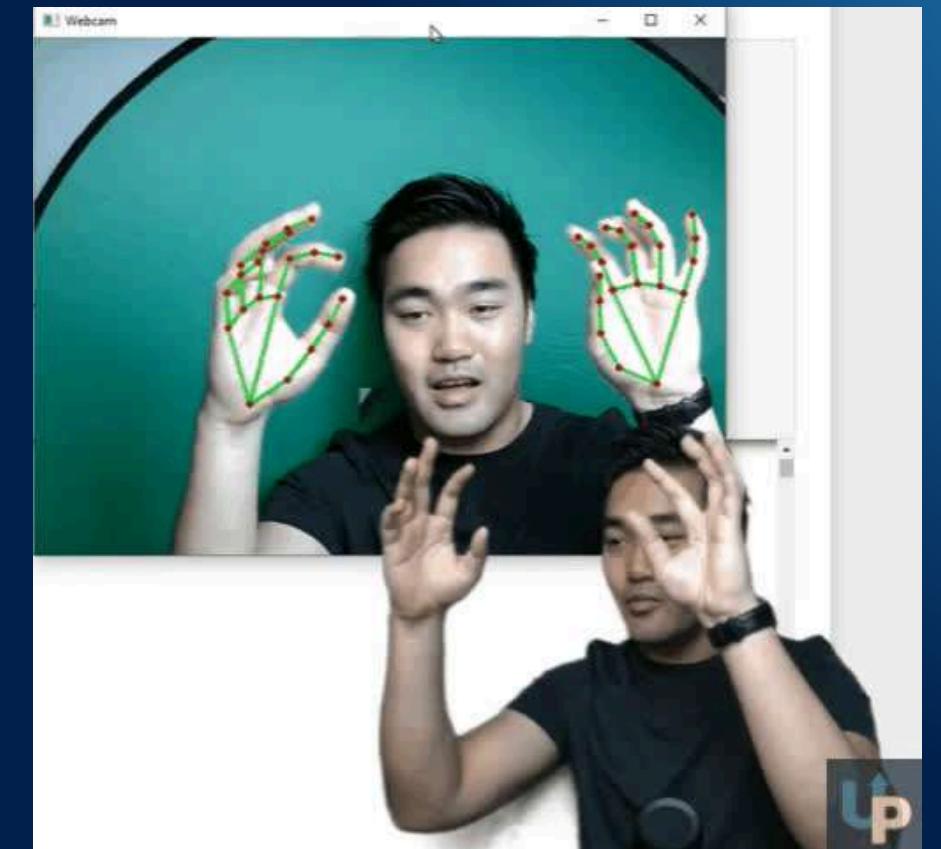
# Hand Tracking

open your command prompt

Q1

```
pip install mediapipe
```

```
pip install opencv-python
```



# Q3

## Hand Track Coding

# Our Vision

```
1 import cv2
2 import mediapipe as mp
3
4 mp_hands = mp.solutions.hands
5 mp_draw = mp.solutions.drawing_utils
6
7 webcam = cv2.VideoCapture(0) → Initialize the webcam
8
9 hands = mp_hands.Hands() → Initialize the Hand model
10
11 while True:
12     success, image = webcam.read() → Read a frame from the webcam
13
14     image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) Convert the frame from BGR to RGB
15
16     results = hands.process(image_rgb) → Process the frame to detect
17         hand landmarks
18
19     if results.multi_hand_landmarks:
20         for hand_landmarks in results.multi_hand_landmarks:
21             mp_draw.draw_landmarks(image, hand_landmarks, mp_hands.HAND_CONNECTIONS)
22
23     cv2.imshow("Webcam", image)
24     cv2.waitKey(1)
```

Print the detected hand landmarks' positions





# BREAK SLIDES

**LET'S TAKE A BREAK 5 MINUTES!**

- Taking breaks during work is a good thing
- Don't work too hard, get some rest



# Hand Count Finger Send To NodeRed

open your command prompt

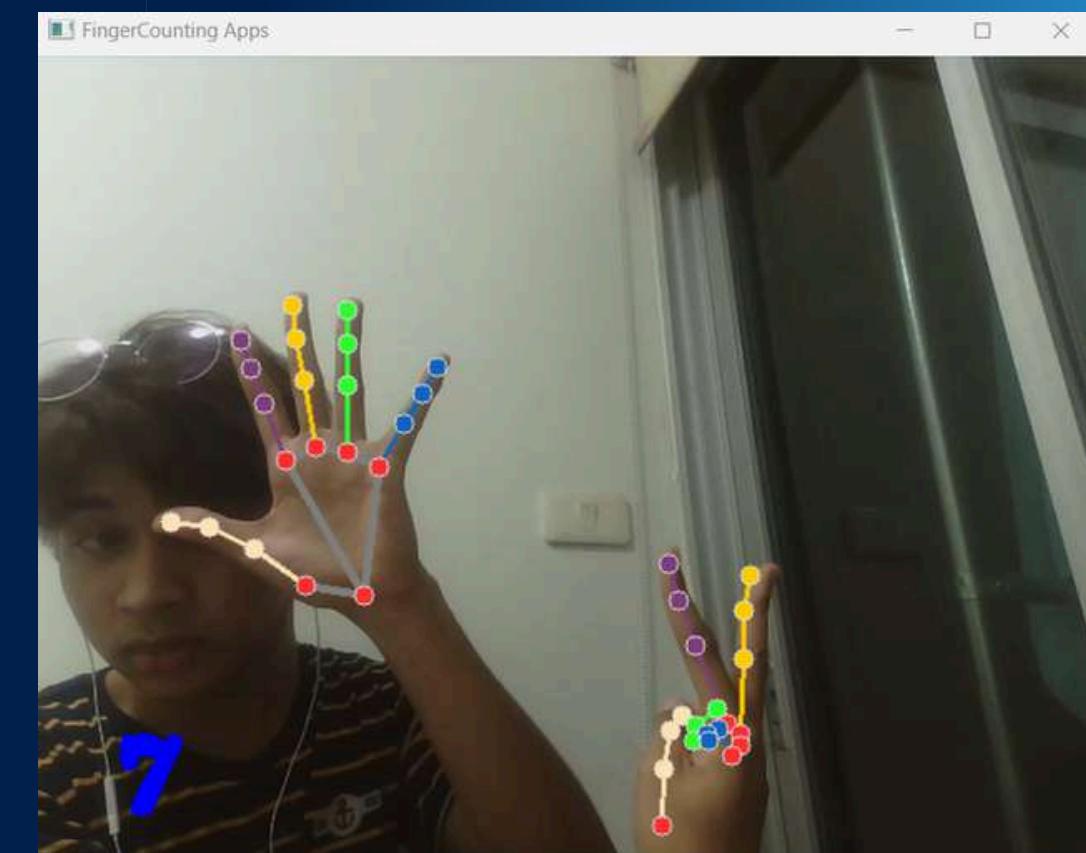
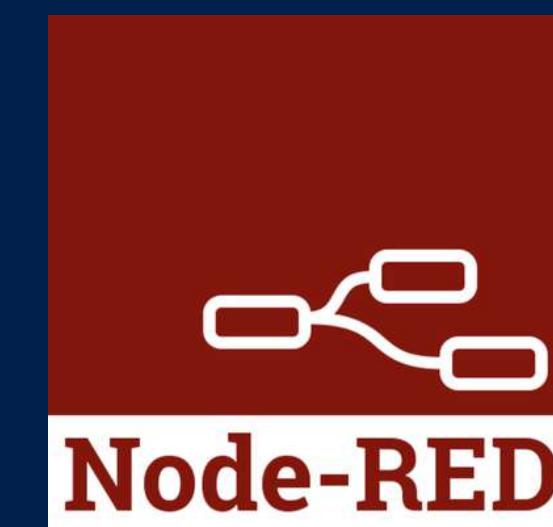
**Q1**

```
pip install mediapipe  
pip install opencv-python
```

**Q2**

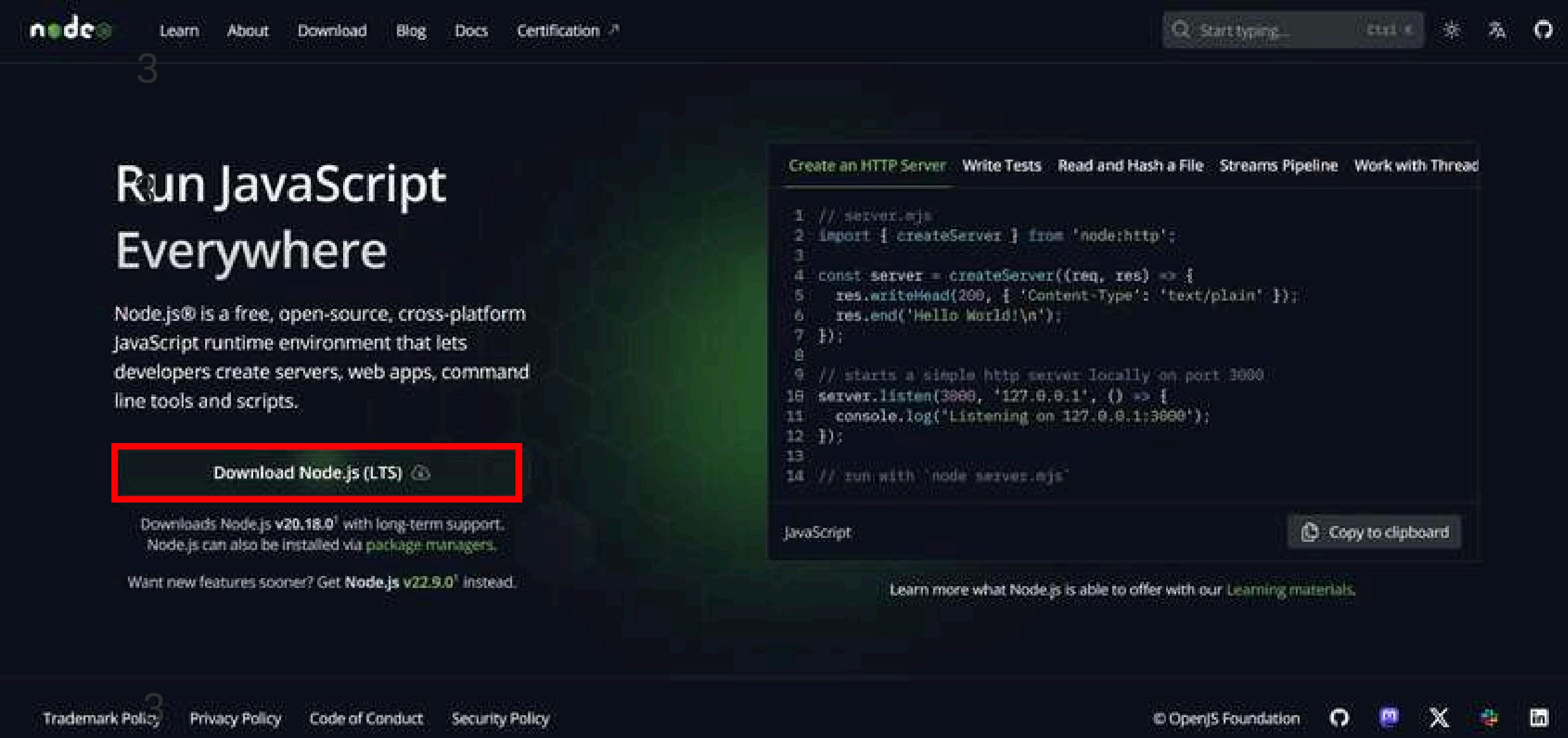
install NodeRed

```
pip install requests
```



# Hand Count Finger Send To NodeRed

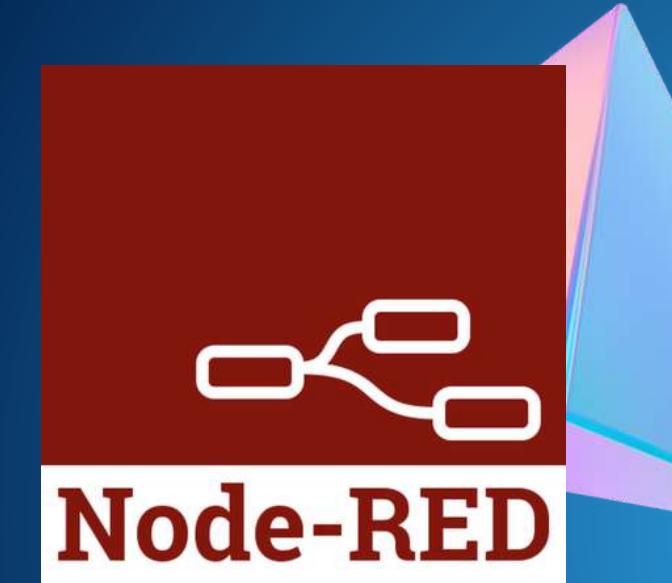
Q2 | install NodeRed



The screenshot shows the official Node.js website. At the top, there's a navigation bar with links for Learn, About, Download, Blog, Docs, and Certification. Below the navigation, there's a large banner with the text "Run JavaScript Everywhere" and a brief description of what Node.js is. A prominent green button labeled "Download Node.js (LTS)" is highlighted with a red border. To the right of the banner, there's a code editor window titled "Create an HTTP Server" containing sample Node.js code for a simple HTTP server. The code is as follows:

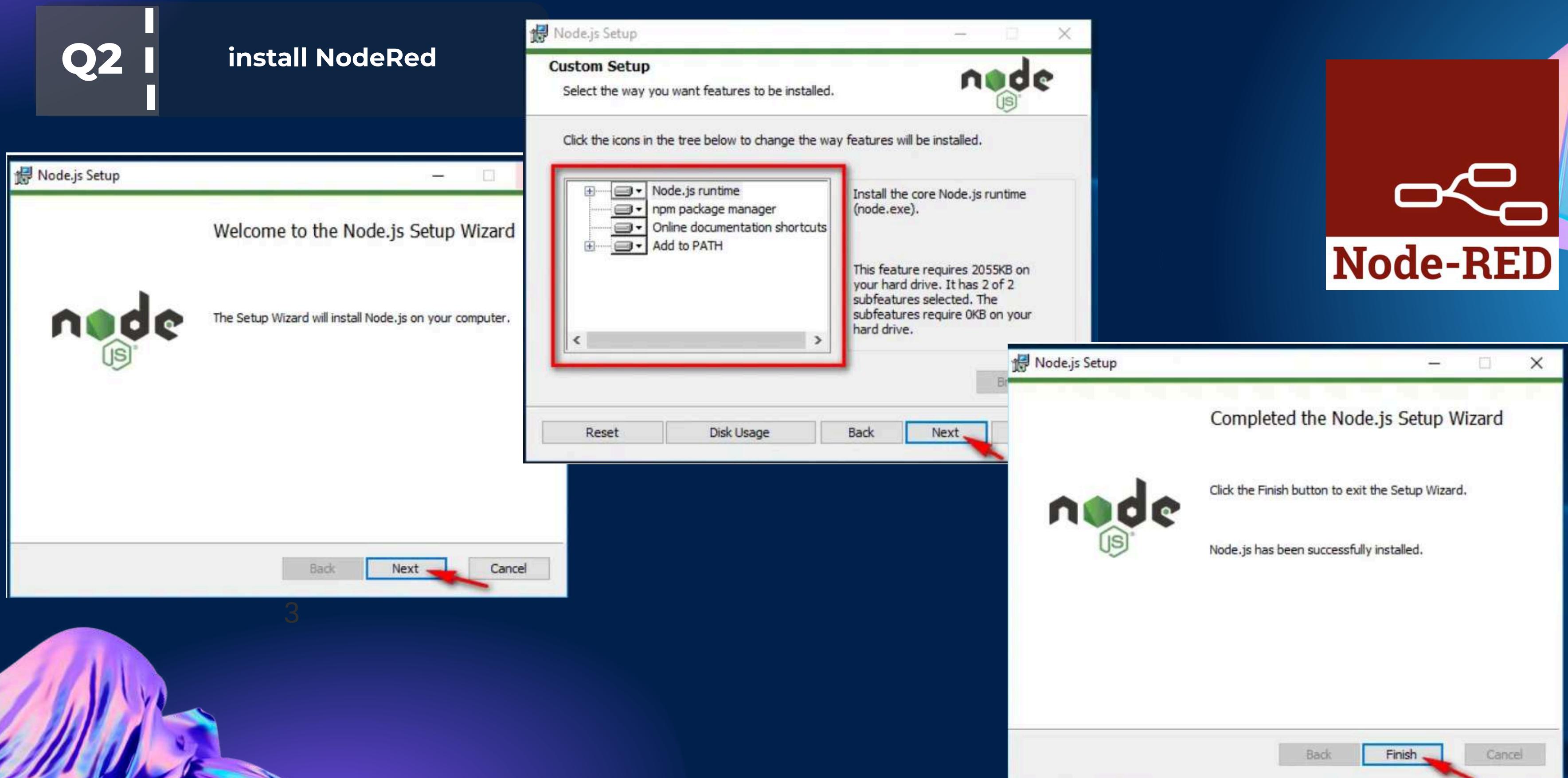
```
1 // server.js
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log("Listening on 127.0.0.1:3000");
12 });
13
14 // run with node server.js
```

Below the code editor, there's a "JavaScript" label and a "Copy to clipboard" button. At the bottom of the page, there are links for Trademark Policy, Privacy Policy, Code of Conduct, Security Policy, and social media icons for GitHub, LinkedIn, and others.



1. Install Node.JS (<https://nodejs.org/en/>)  
Go to the nodejs.org website and choose to download the LTS version.

# Hand Count Finger Send To NodeRed



# Hand Count Finger Send To NodeRed

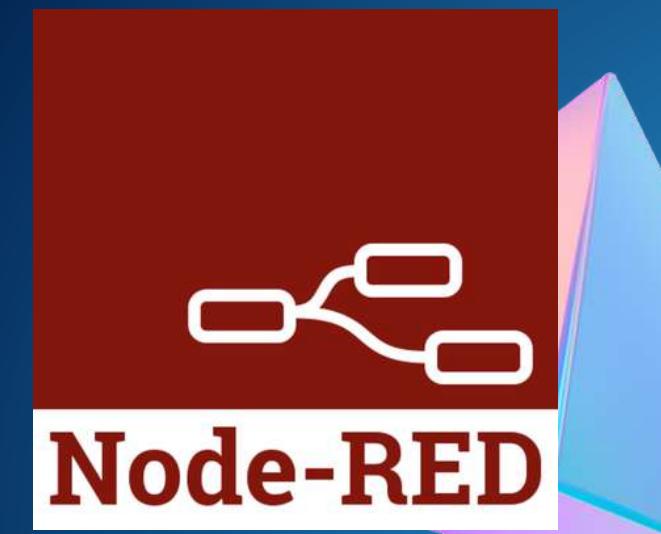
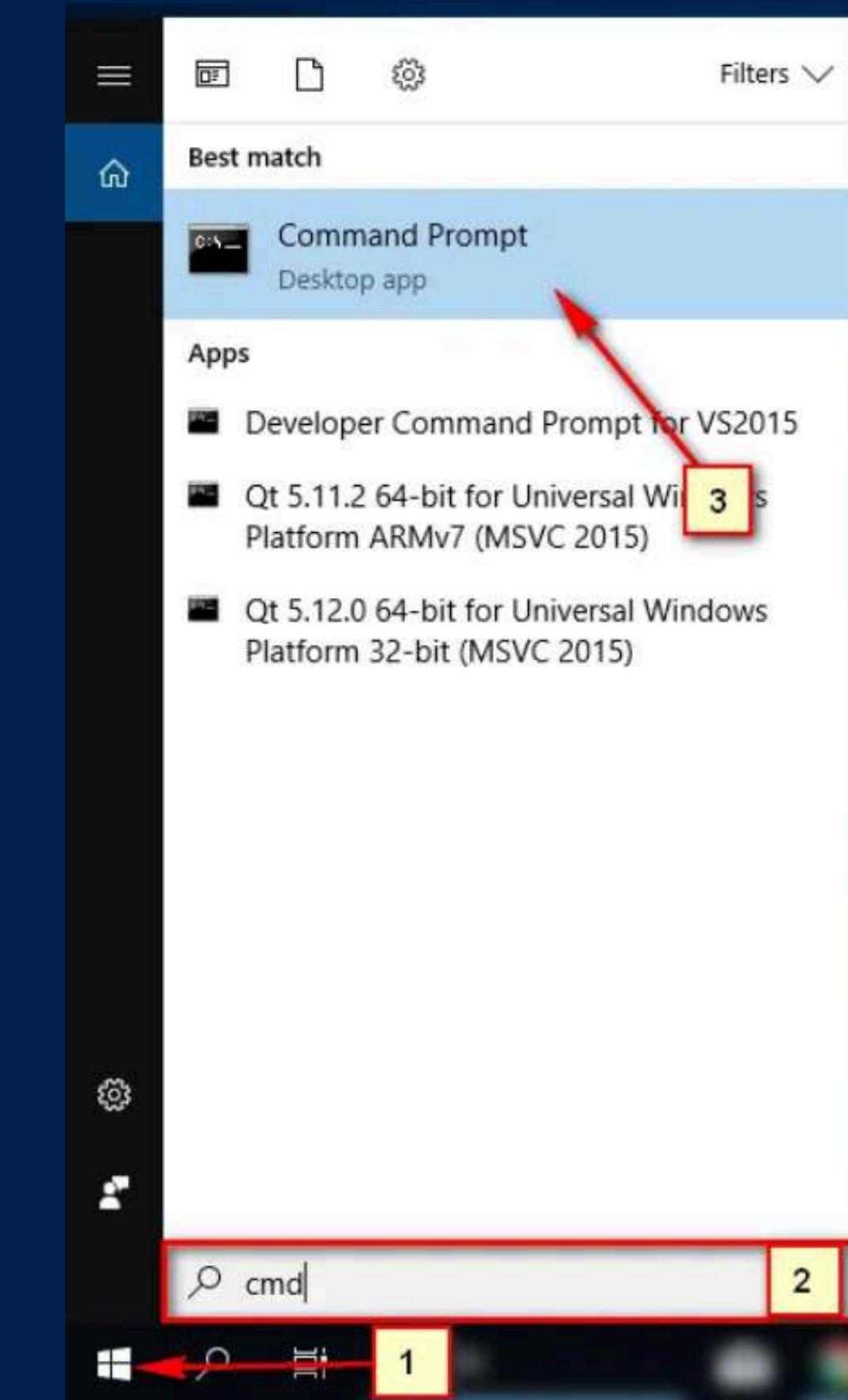


## 2. Check Version NodeJS with CMD

3

- #1. Press the start menu button
- #2.Type cmd to search
- #3.Select Command Prompt

3

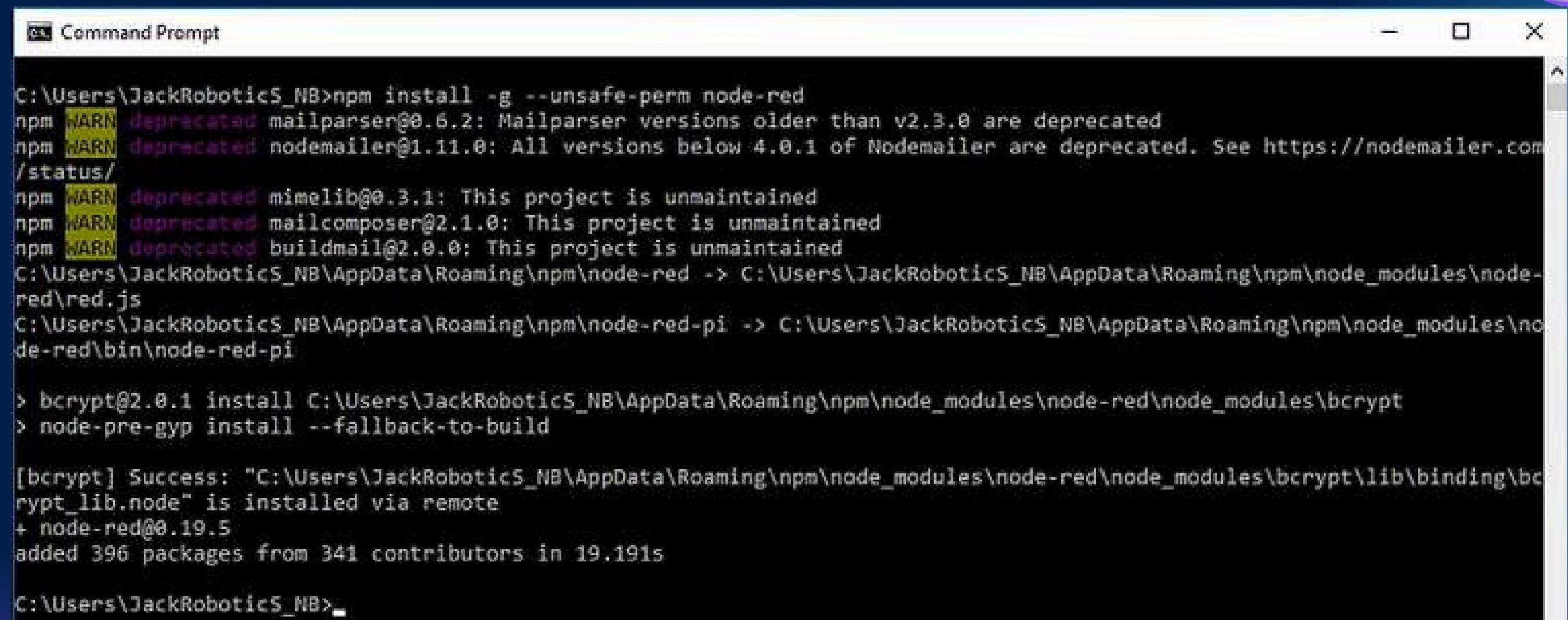


# Hand Count Finger Send To NodeRed

Q2

```
npm install -g --unsafe-perm node-red
```

## 3. Install Node-Red with npm



```
C:\Users\JackRobotics_NB>npm install -g --unsafe-perm node-red
npm [WARN] deprecated mailparser@0.6.2: Mailparser versions older than v2.3.0 are deprecated
npm [WARN] deprecated nodemailer@1.11.0: All versions below 4.0.1 of Nodemailer are deprecated. See https://nodemailer.com/status/
npm [WARN] deprecated mimelib@0.3.1: This project is unmaintained
npm [WARN] deprecated mailcomposer@2.1.0: This project is unmaintained
npm [WARN] deprecated buildmail@2.0.0: This project is unmaintained
C:\Users\JackRobotics_NB\AppData\Roaming\npm\node-red -> C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node-red.js
C:\Users\JackRobotics_NB\AppData\Roaming\npm\node-red-pi -> C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi

> bcrypt@2.0.1 install C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build

[bcrypt] Success: "C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node_modules\bcrypt\lib\binding\bcrypt_lib.node" is installed via remote
+ node-red@0.19.5
added 396 packages from 341 contributors in 19.191s

C:\Users\JackRobotics_NB>
```

When Node-Red is finished installing, it will show the screen as shown below. indicates that the installation is complete

# Hand Count Finger Send To NodeRed

Q2

install NodeRed

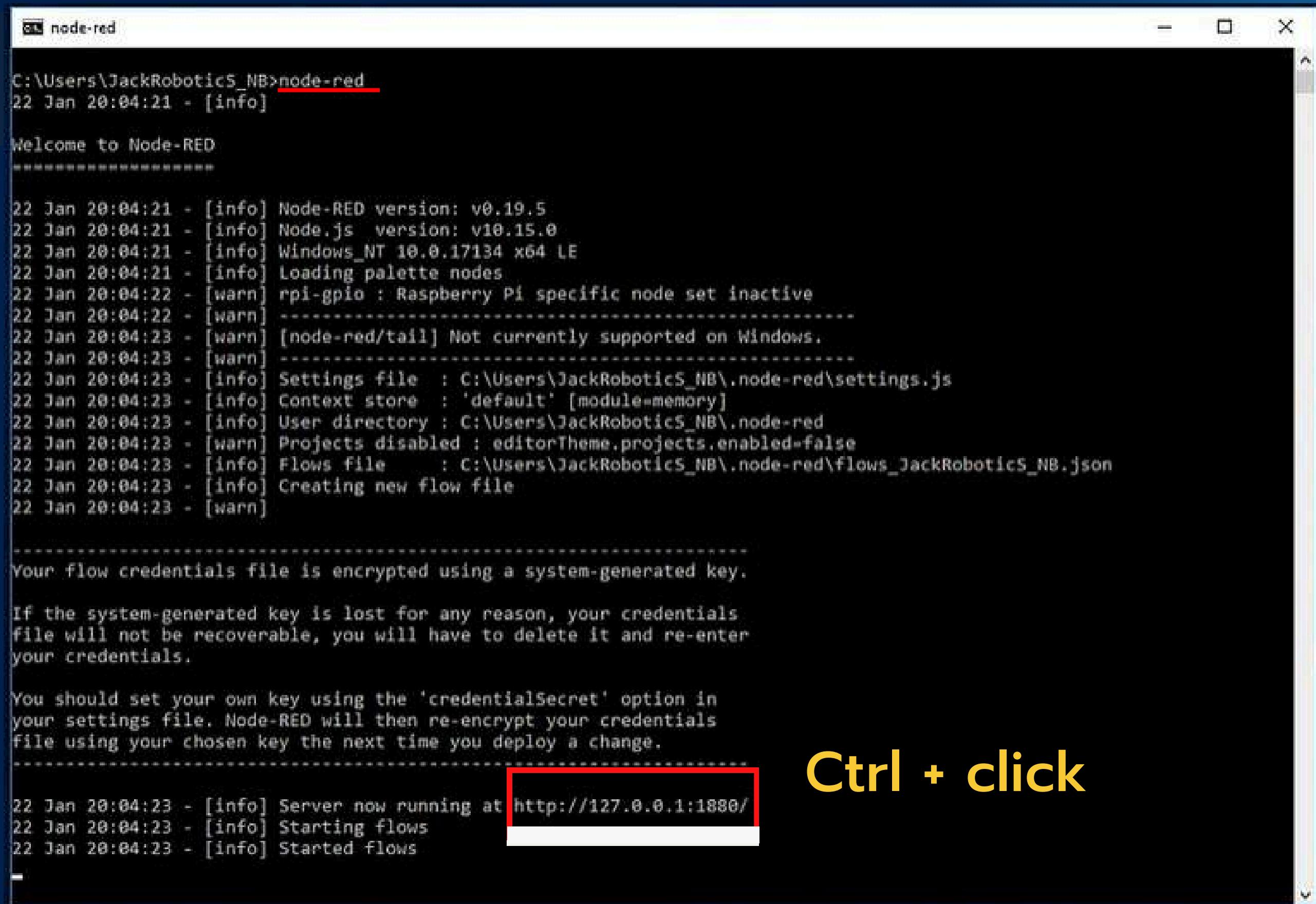
## 4. Run Node-Red

node-red

Type the command node-red

into cmd, it will look like the picture.

3



```
C:\Users\JackRobotics_NB>node-red
22 Jan 20:04:21 - [info]

Welcome to Node-RED

22 Jan 20:04:21 - [info] Node-RED version: v0.19.5
22 Jan 20:04:21 - [info] Node.js  version: v10.15.0
22 Jan 20:04:21 - [info] Windows_NT 10.0.17134 x64 LE
22 Jan 20:04:21 - [info] Loading palette nodes
22 Jan 20:04:22 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
22 Jan 20:04:22 - [warn] -----
22 Jan 20:04:23 - [warn] [node-red/tail] Not currently supported on Windows.
22 Jan 20:04:23 - [warn] -----
22 Jan 20:04:23 - [info] Settings file  : C:\Users\JackRobotics_NB\.node-red\settings.js
22 Jan 20:04:23 - [info] Context store  : 'default' [module=memory]
22 Jan 20:04:23 - [info] User directory : C:\Users\JackRobotics_NB\.node-red
22 Jan 20:04:23 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Jan 20:04:23 - [info] Flows file    : C:\Users\JackRobotics_NB\.node-red\flows_JackRobotics_NB.json
22 Jan 20:04:23 - [info] Creating new flow file
22 Jan 20:04:23 - [warn]

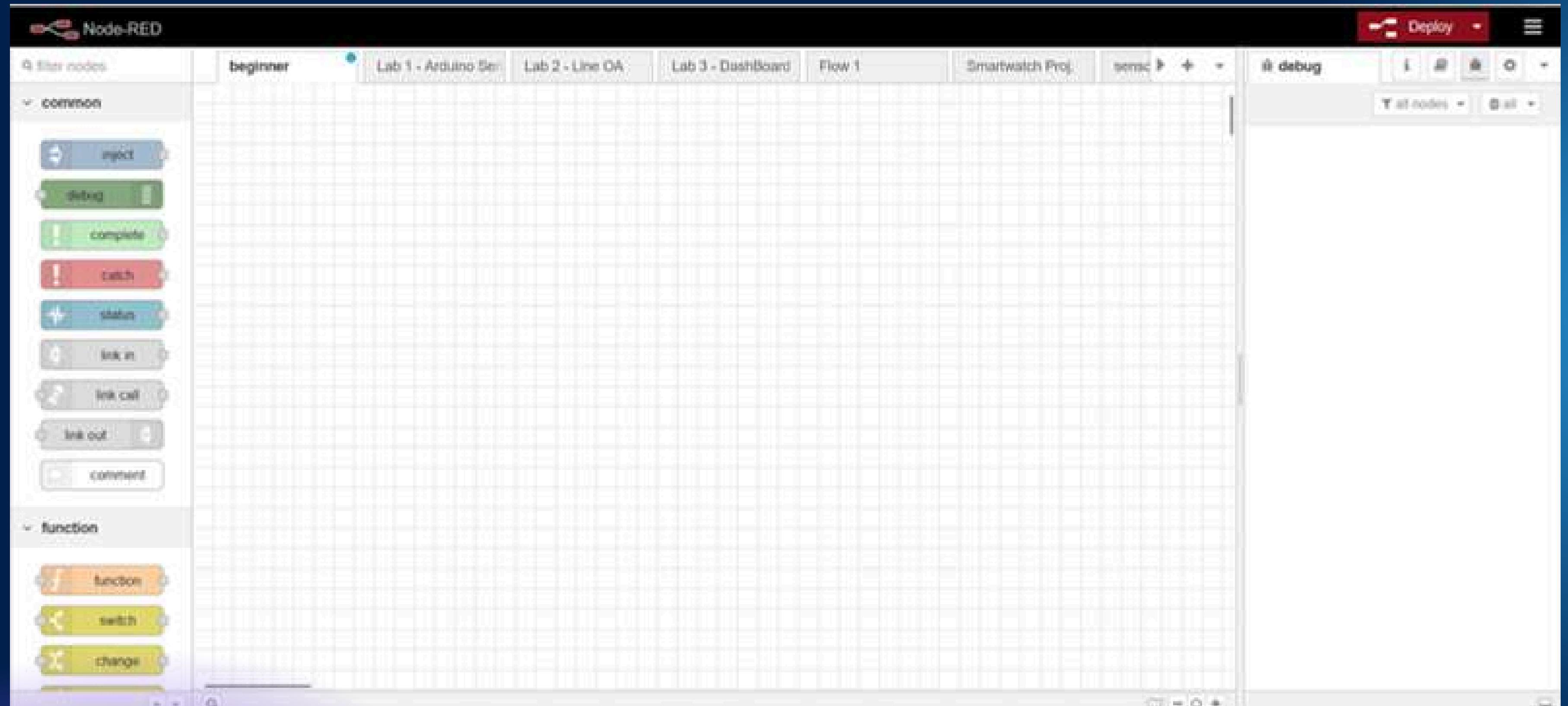
-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.

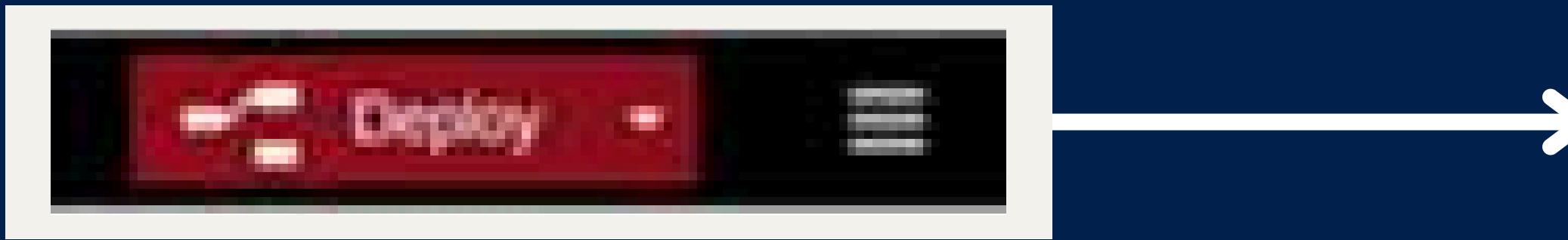
22 Jan 20:04:23 - [info] Server now running at http://127.0.0.1:1880/
22 Jan 20:04:23 - [info] Starting flows
22 Jan 20:04:23 - [info] Started flows
```

Ctrl + click

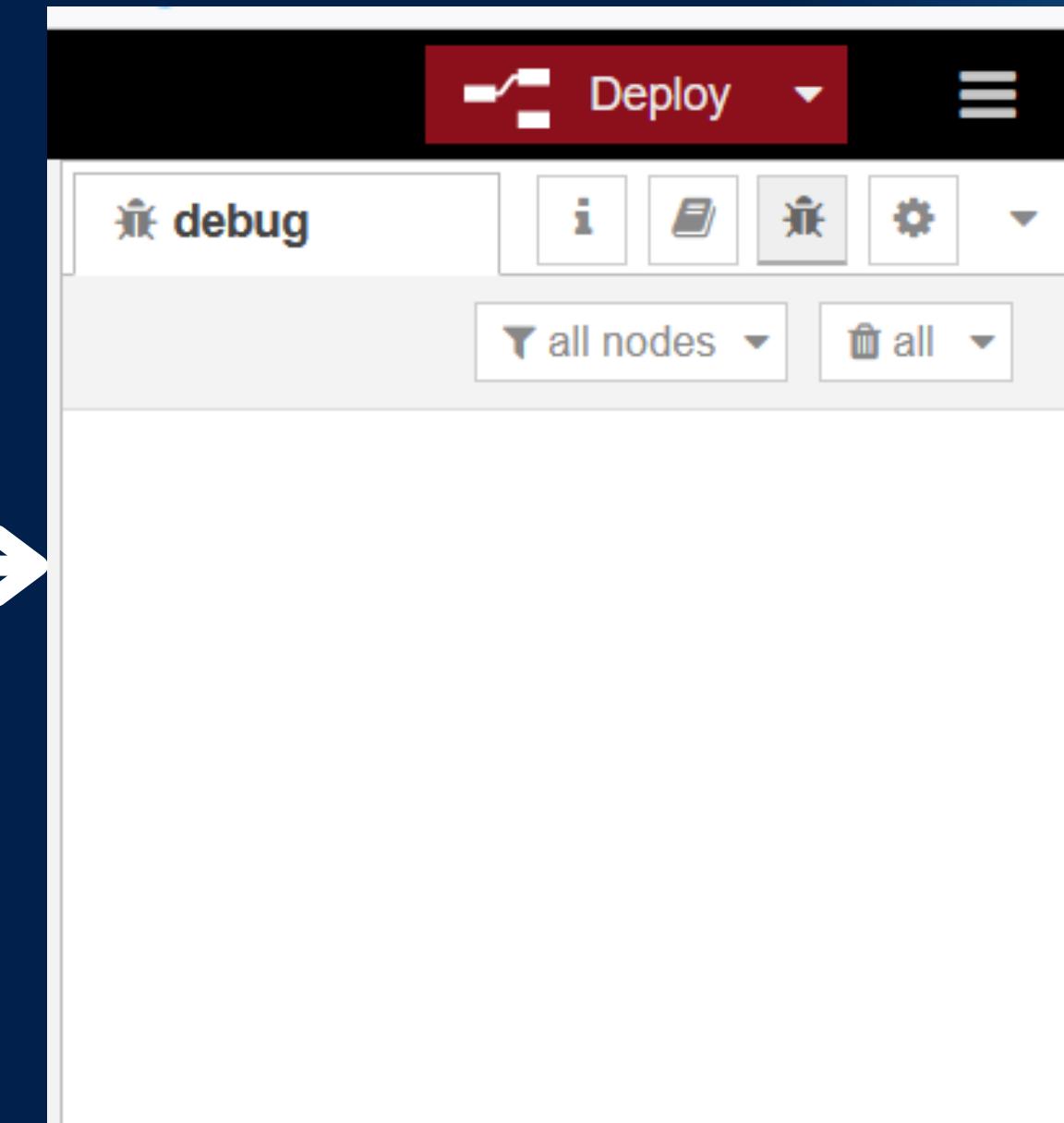


## Workspace **Node-Red**

# Deploy Button



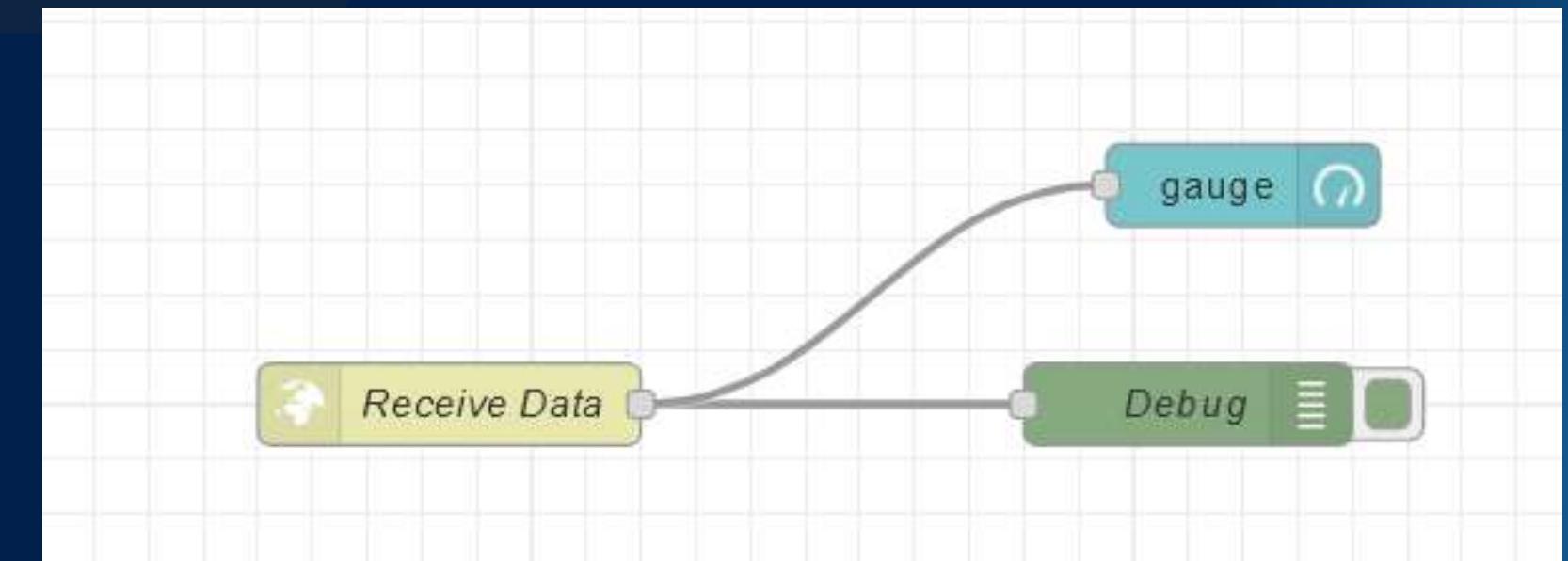
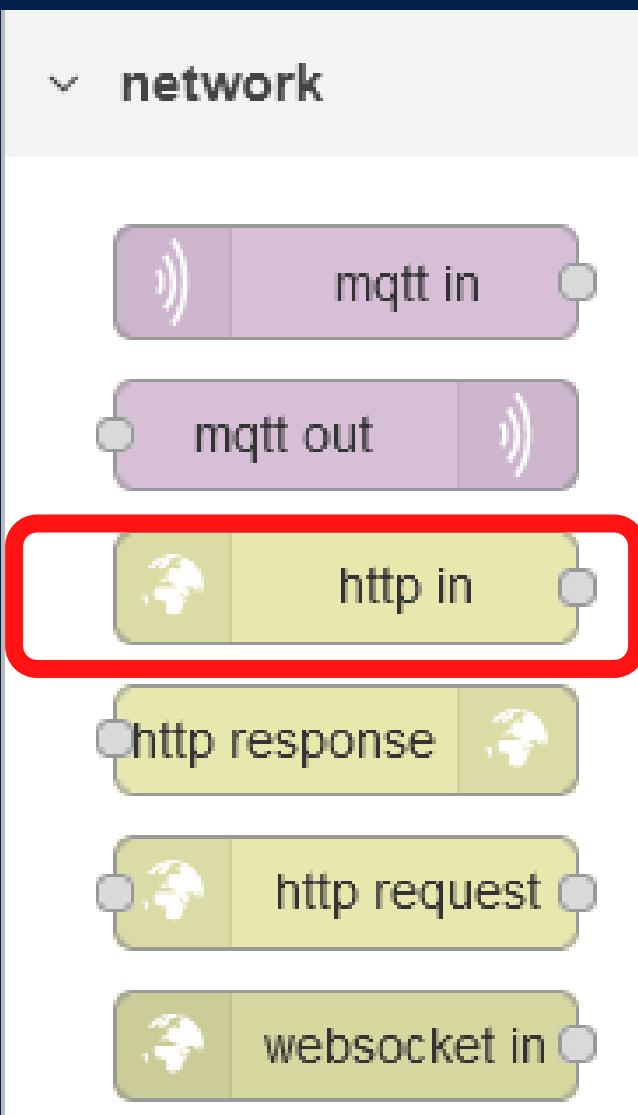
- บุ่มนี้ไว้กดทุกครั้งเวลา อัพเดต **block code**  
หลังแก้ไขเสร็จ



นำไว้ดู Debug เวลาเมื่อค่าเข้ามาจากการส่งข้อมูล  
**Data Value / Struct after processing**

# Q3

## Configure padlet in node red



**double-click in nodes Receive Data**

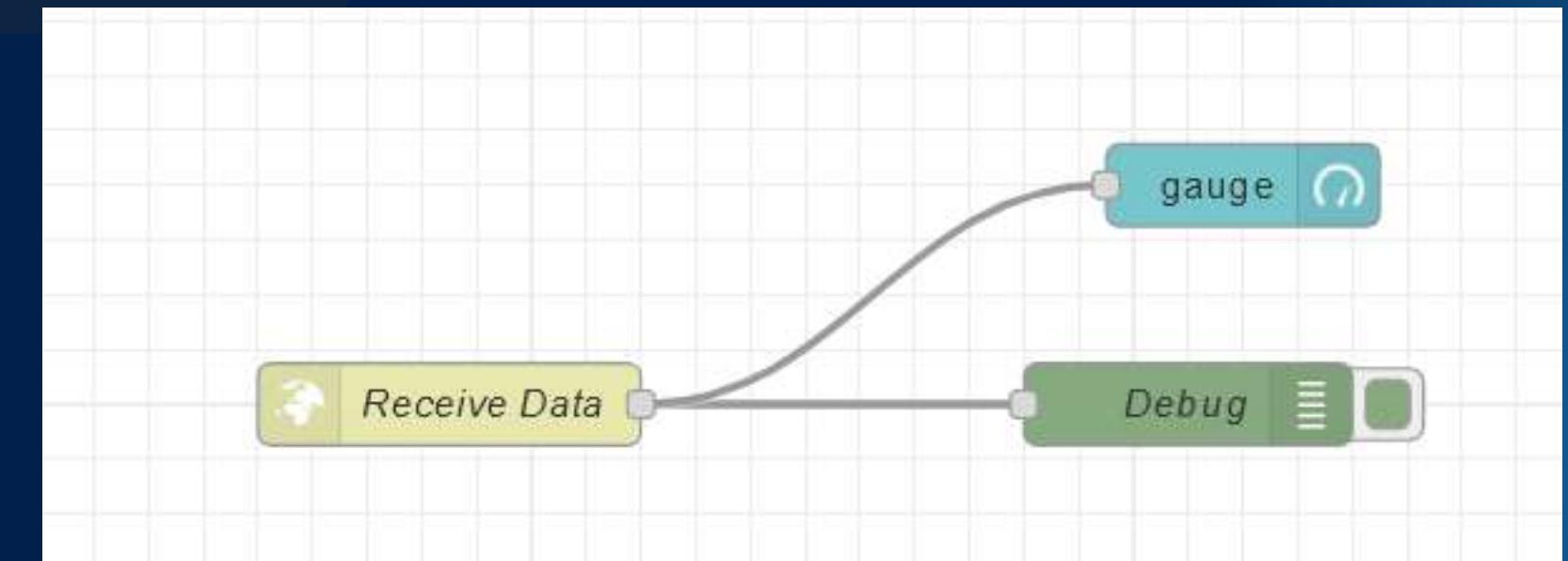
The configuration dialog for the 'Receive Data' node. It includes the following fields:

- Method: POST
- Accept file uploads?
- URL: /receive-data
- Name: Receive Data

A red box highlights the 'URL' input field, with the text 'config your URL you need' displayed in red to its right.

# Q3

## Configure padlet in node red



**double-click in nodes Receive Data**

The configuration dialog for the 'Receive Data' node. It includes the following fields:

- Method: POST
- Accept file uploads?
- URL: /receive-data
- Name: Receive Data

A red box highlights the 'URL' input field, with the text 'config your URL you need' displayed in red to its right.

# Q3

## Hand Count Coding

# Our Wisdom

URL

```
1 import cv2
2 import mediapipe as mp
3 import requests
4 url = 'http://127.0.0.1:1880/receive-data' URL
5
6 mp_drawing = mp.solutions.drawing_utils
7 mp_drawing_styles = mp.solutions.drawing_styles
8 mp_hands = mp.solutions.hands
9
10 capture = cv2.VideoCapture(0)
11 with mp_hands.Hands(
12     model_complexity=0,
13     min_detection_confidence=0.5,
14     min_tracking_confidence=0.5) as hands:
15     while capture.isOpened():
16         success, image = capture.read()
17         if not success:
18             print('Ignored empty webcam\'s frame')
19             continue
20         image.flags.writeable = False
21         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
22         results = hands.process(image)
23
24         image.flags.writeable = True
25         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
26
27         fingerCount = 0
```

```
61         cv2.putText(image, str(fingerCount), (50,450), cv2.FONT_HERSHEY_COMPLEX_SMALL, 3, (255,0,0), 10)
62         cv2.imshow('FingerCounting Apps', image)
```

```
29     if results.multi_hand_landmarks:
30         for hand_landmarks in results.multi_hand_landmarks:
31             handIndex = results.multi_hand_landmarks.index(hand_landmarks)
32             handLabel = results.multi_handedness[handIndex].classification[0].label
33
34             handLandmarks = []
35
36             for landmarks in hand_landmarks.landmark:
37                 handLandmarks.append([landmarks.x, landmarks.y])
38
39             if handLabel == "Left" and handLandmarks[4][0] > handLandmarks[3][0]:
40                 fingerCount = fingerCount + 1
41             elif handLabel == "Right" and handLandmarks[4][0] < handLandmarks[3][0]:
42                 fingerCount = fingerCount + 1
43
44             if handLandmarks[8][1] < handLandmarks[6][1]:
45                 fingerCount = fingerCount + 1
46             if handLandmarks[12][1] < handLandmarks[10][1]:
47                 fingerCount = fingerCount + 1
48             if handLandmarks[16][1] < handLandmarks[14][1]:
49                 fingerCount = fingerCount + 1
50             if handLandmarks[20][1] < handLandmarks[18][1]:
51                 fingerCount = fingerCount + 1
52
53             mp_drawing.draw_landmarks(
54                 image,
55                 hand_landmarks,
56                 mp_hands.HAND_CONNECTIONS,
57                 mp_drawing_styles.get_default_hand_landmarks_style(),
58                 mp_drawing_styles.get_default_hand_connections_style()
59             )
60
```

**Q4**

## FingerCount Send To NodeRed Coding

youtube : <https://www.youtube.com/watch?app=desktop&v=1iq9FxLxBIY>

```
64 if cv2.waitKey(1) == 27: # Check if the ASCII value of the pressed key is 27 (ESC key)
65     break
66 if cv2.waitKey(1) & 0xFF == ord('c'): # Check if the ASCII value of the pressed key is 99 (C key) // 32 is (backspace key)
67 # send http buffer string to http in Node-red not recall.
68 try:
69     response = requests.post(url, data=str(fingerCount), timeout=5) # Set timeout to 5 seconds
70     if response.status_code == 200:
71         print('Data sent successfully')
72     else:
73         print('Error sending data:', response.text)
74 except requests.Timeout:
75     print('Request timed out. Server did not respond in time.')
76 except requests.RequestException as e:
77     print('An error occurred:', e)
78 capture.release()
```

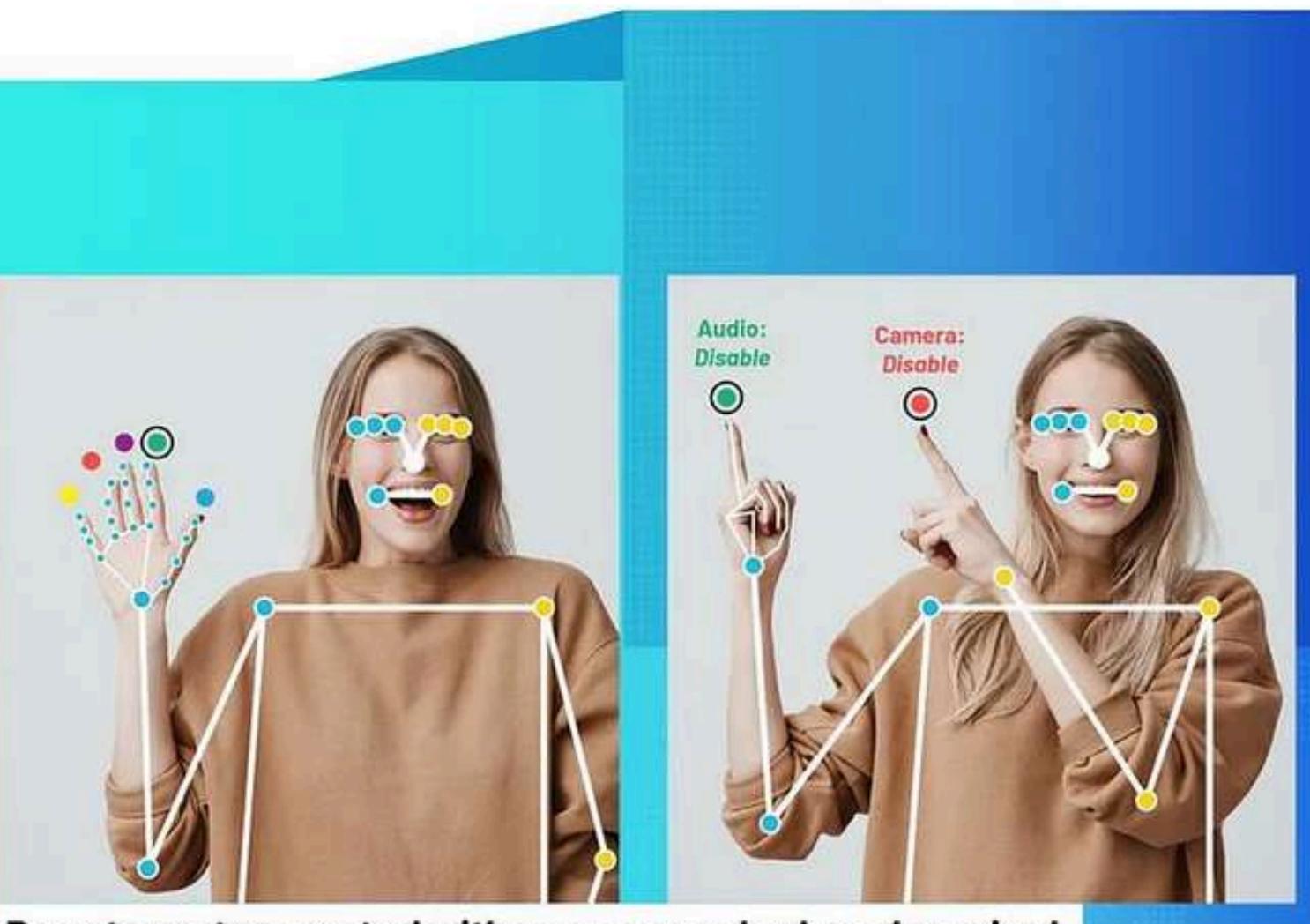


github code :

[https://github.com/TheDBoyZ/  
CodeBotix\\_Camp\\_Image](https://github.com/TheDBoyZ/CodeBotix_Camp_Image)

# Intelligence

## Google's MediaPipe Holistic Interface

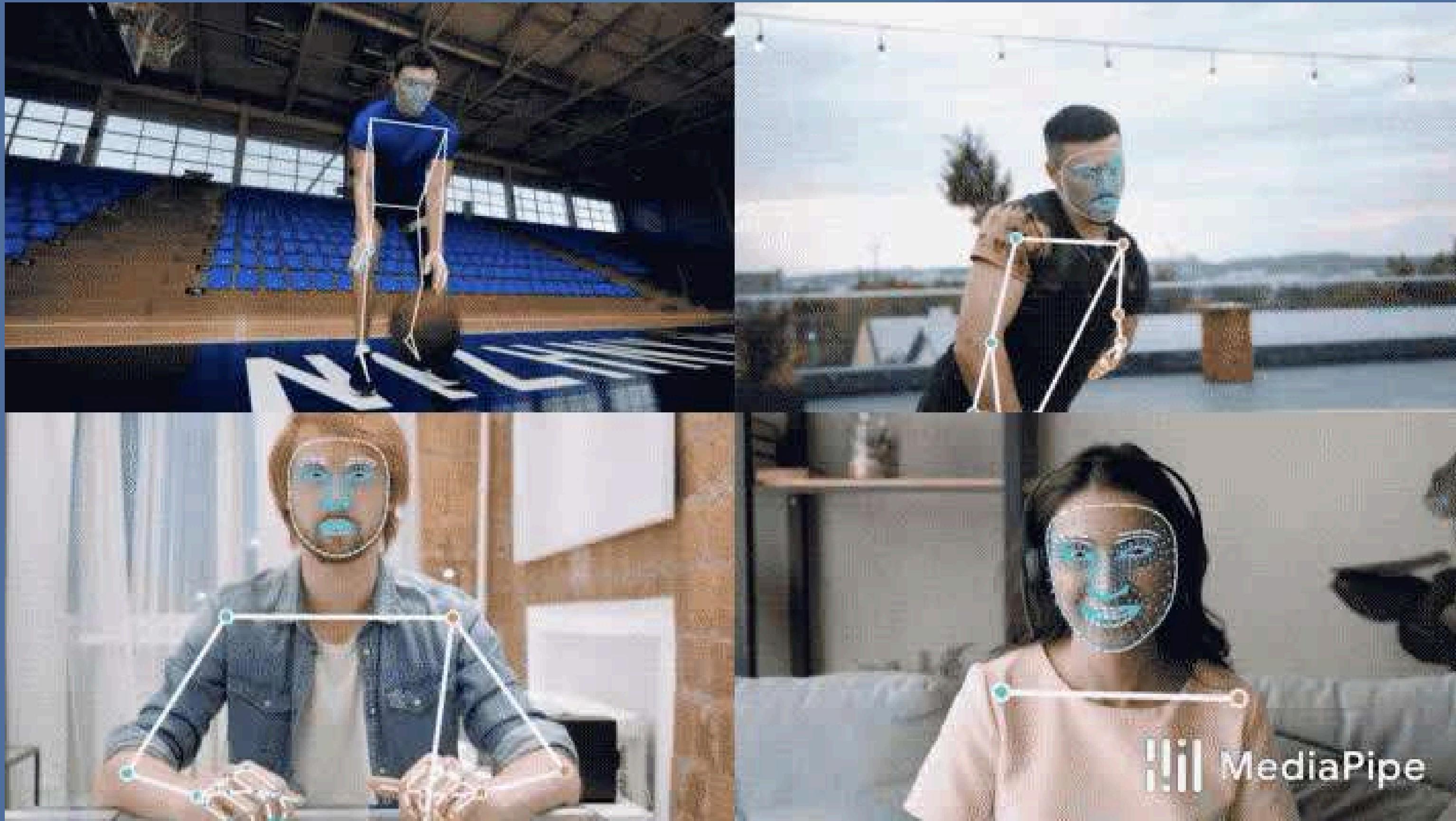


This gesture control and command technique will take us to the next level. And allow us to use new innovations that other devices cannot do

It can be tried on MediaPipe Holistic Interface from Google at:

[https://mediapipe.dev/demo/holistic\\_remote/](https://mediapipe.dev/demo/holistic_remote/)





[https://mediapipe.dev/demo/holistic\\_remote/](https://mediapipe.dev/demo/holistic_remote/)

# Holistic demo

ໂຄມດກາຣໃຊ້ງານຕ່າງໆ

1

1 ນຶ່ງ ມື່ອຊ້າຍ ຮັບ ຂວາ

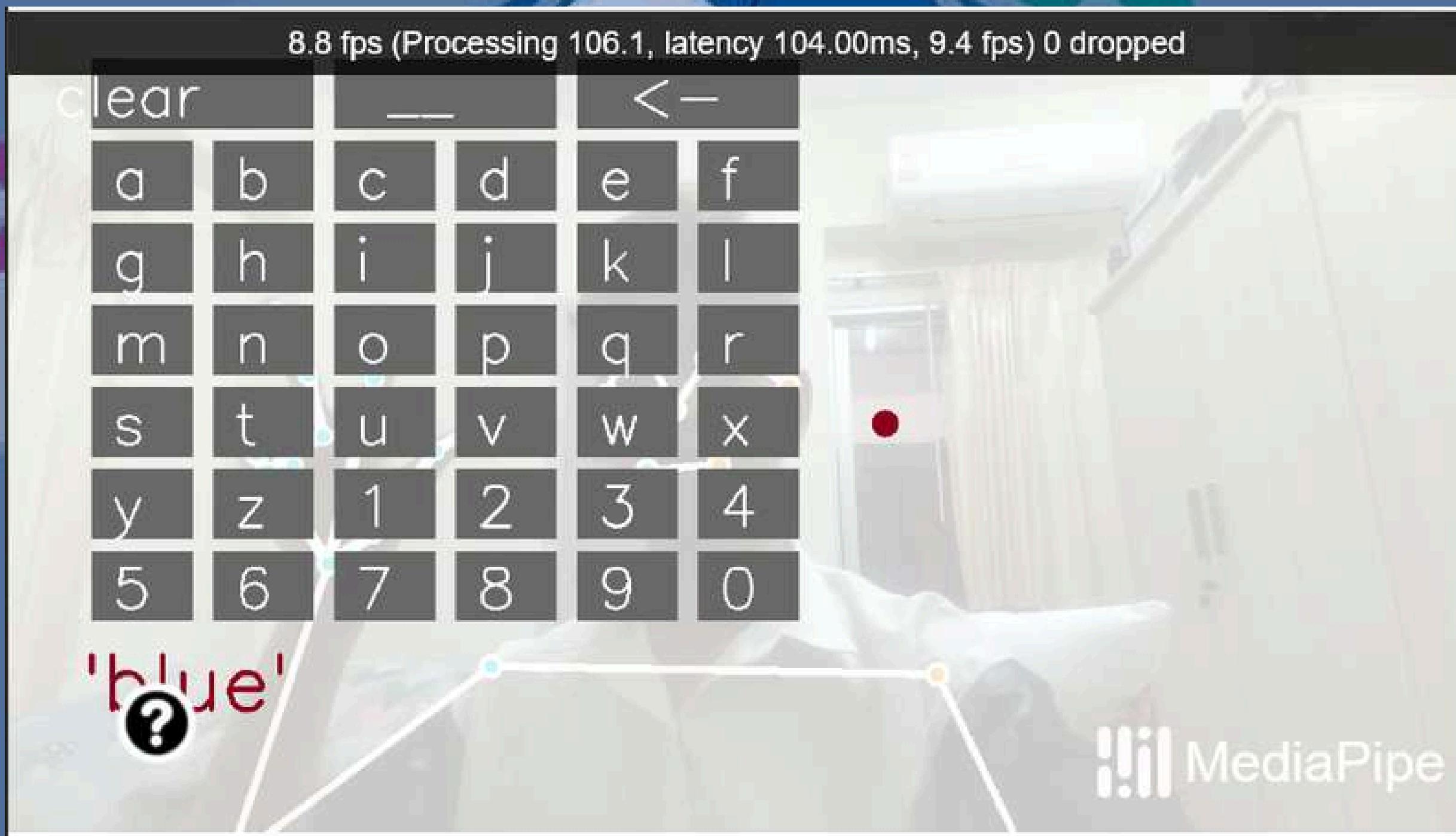
2

3 ນຶ່ງ ມື່ອຊ້າຍ

3

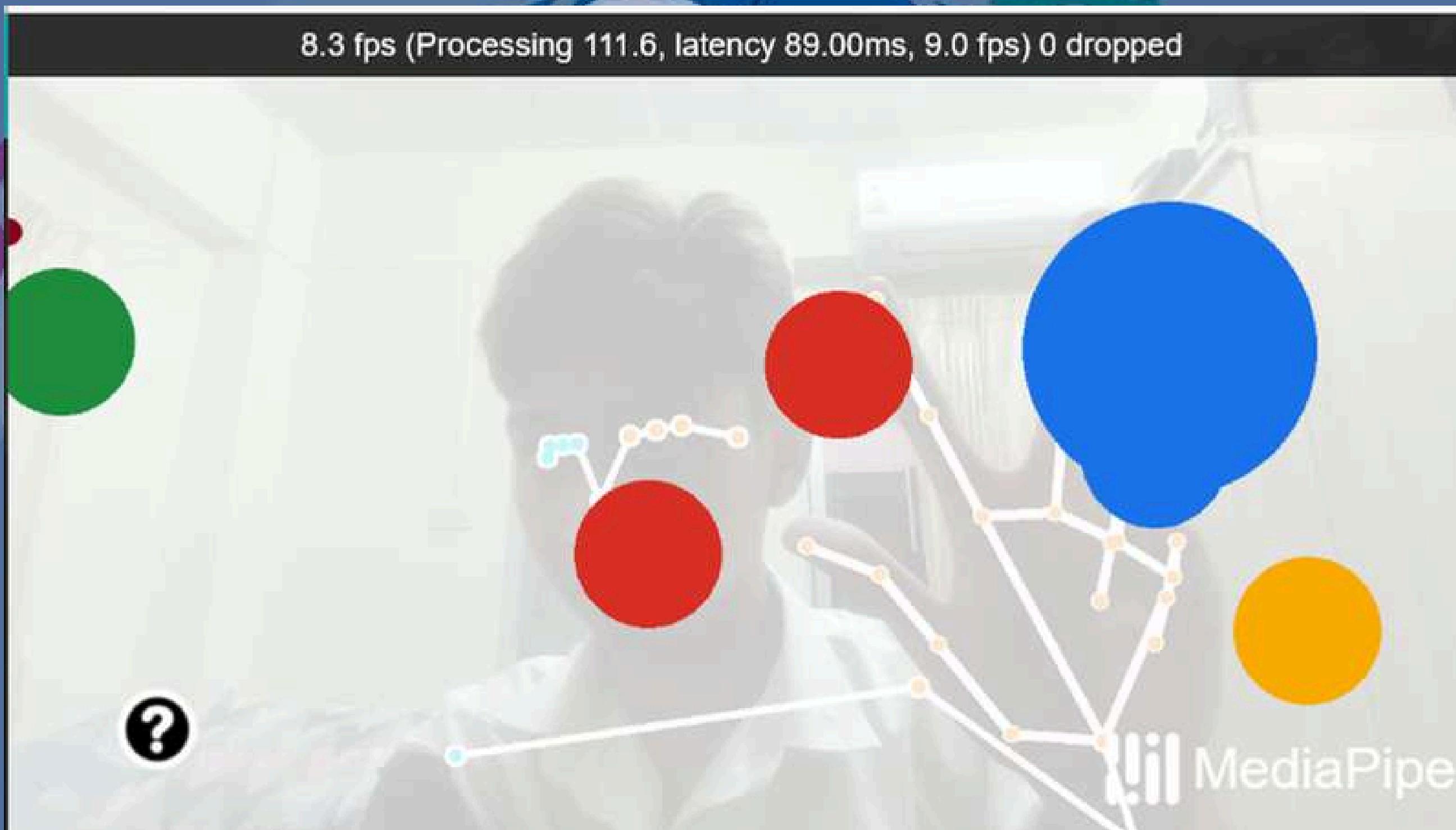
3 ນຶ່ງ ມື່ອຂວາ

# Holistic demo



ลองเขียนชื่อตัวเอง เป็นภาษาอังกฤษ

# Holistic demo



สามารถใช้มือแทน เม้าส์ในการลากกรุ๊ปสีจัดหมวดหมู่ได้

# Holistic demo

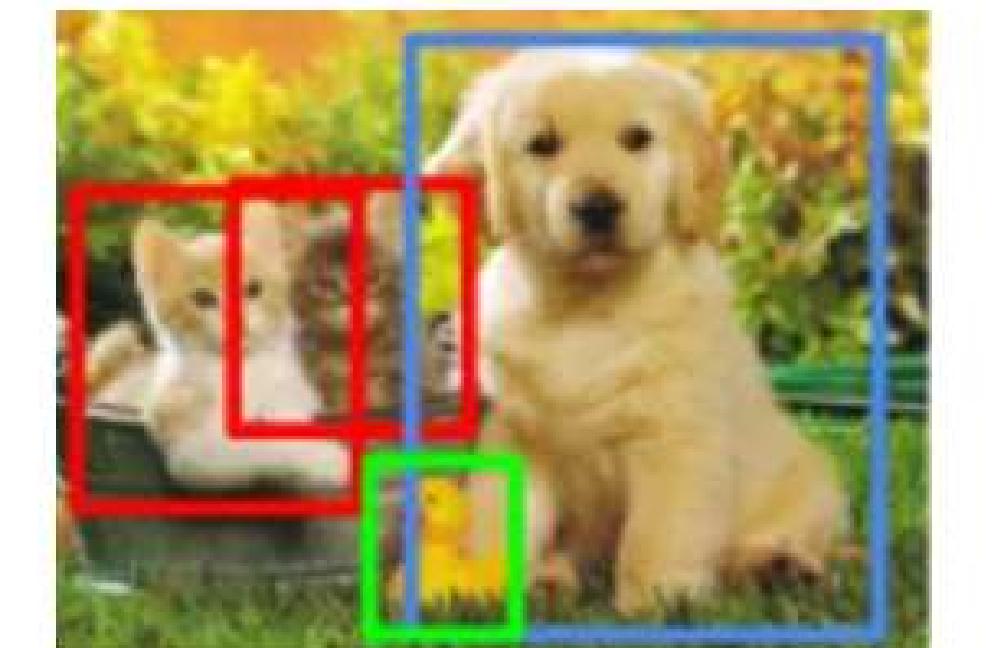


Disable / Enable mode

# IMAGE CLASSIFICATION

เช่นเดียวกับหลักการของ AI เราต้องส่งรูปเข้าไปพร้อมกับผลลัพธ์ว่า นี่คือรูปแนว เพื่อให้มันเรียนรู้เรื่อยๆ และนี่คืองานหลักของสิ่งที่เรียกว่า Image Classification

แล้วแนวมันอยู่ตรงไหนของรูปหละ? ถ้าได้โจทย์มาเป็นแบบนี้ จะเป็นโจทย์อีกแนวที่เรียกว่า Image Detection โดยมันจะบอกถึงตำแหน่งของรูปด้วย



น้องแมวแอนด์เดอะเกิ๊กค์

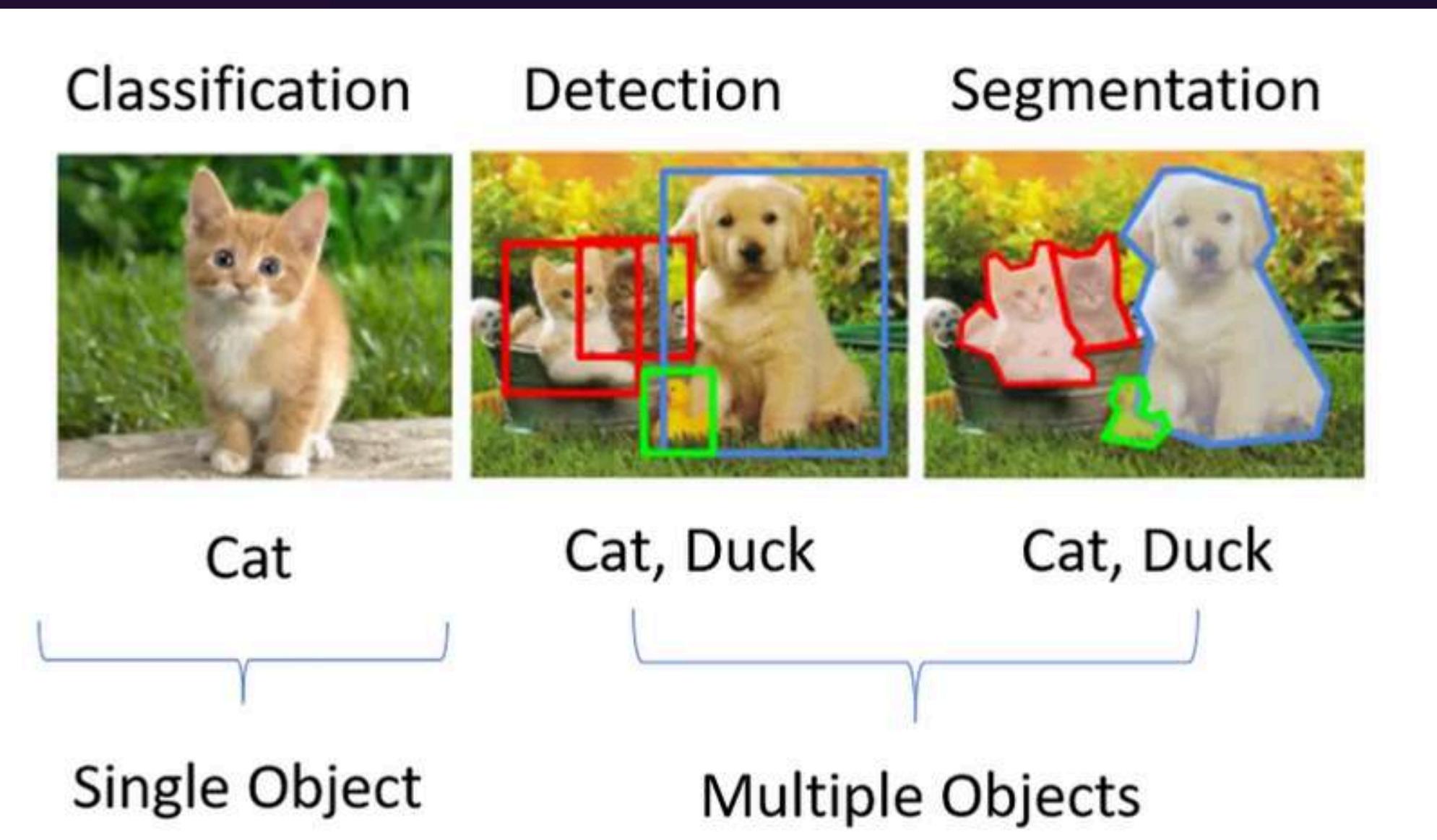
บทความ ความรู้ image classification to AI :

EXIT



# IMAGE CLASSIFICATION

ภาพรวมของโจทย์โดยรวม คือ

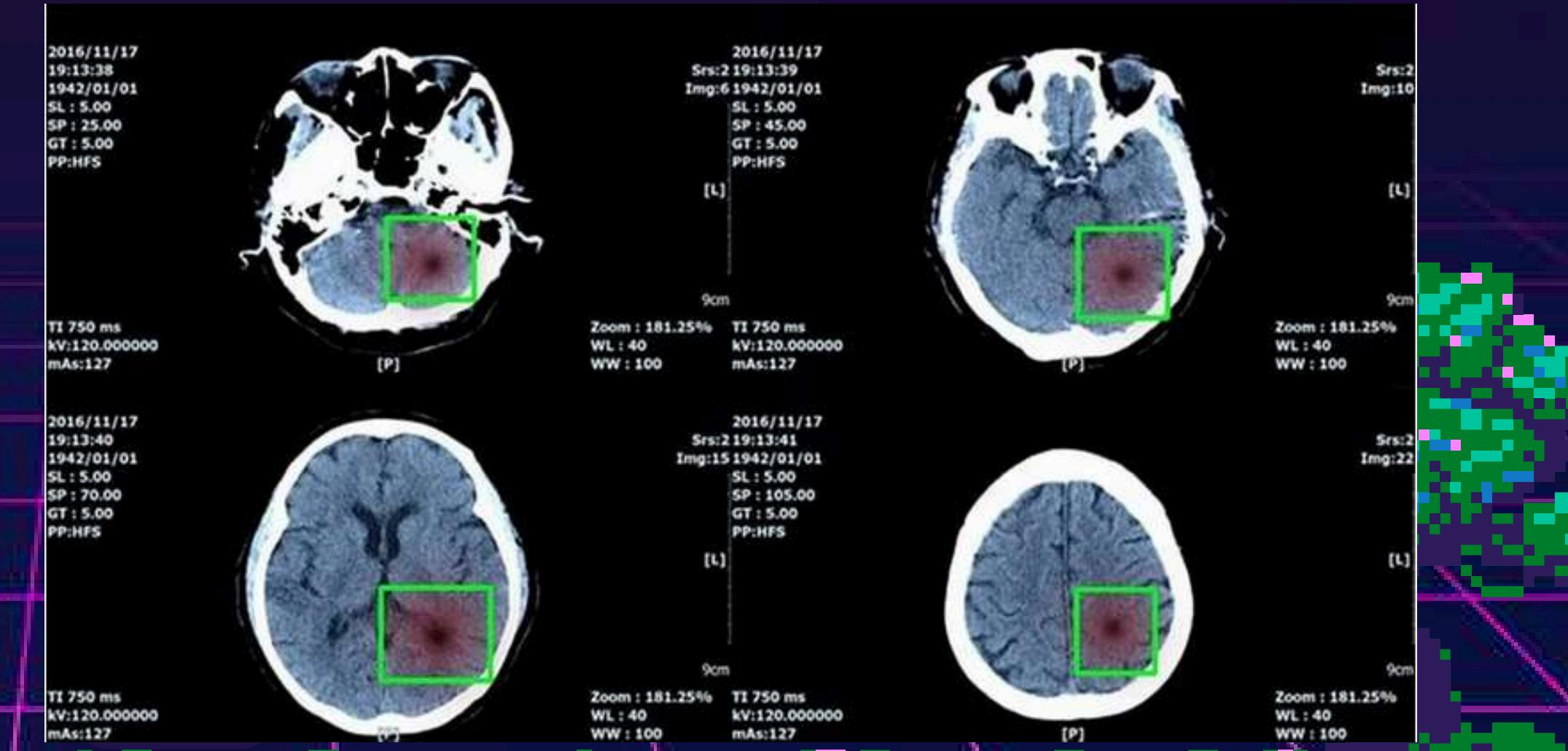


EXIT

# IMAGE CLASSIFICATION

## Medical Sector

บางครั้งเวลาที่มีรูป X-ray ออกมาก ในการตรวจเช็ครูปนี้จะต้องใช้หมอ  
เฉพาะทางที่มีจำกัด ถ้าเรามี AI ค่อยช่วย Focus หรือตีกรอบส่วนที่มีปัญหา  
จะลดเวลาในส่วนนี้มาก รวมถึงอาจจะใช้งานร่วมกับ Image  
Segmentation ได้ จากรูปด้านล่าง เขาวิเคราะห์เรื่องตำแหน่งของเนื้อ  
งอกในสมองครับ

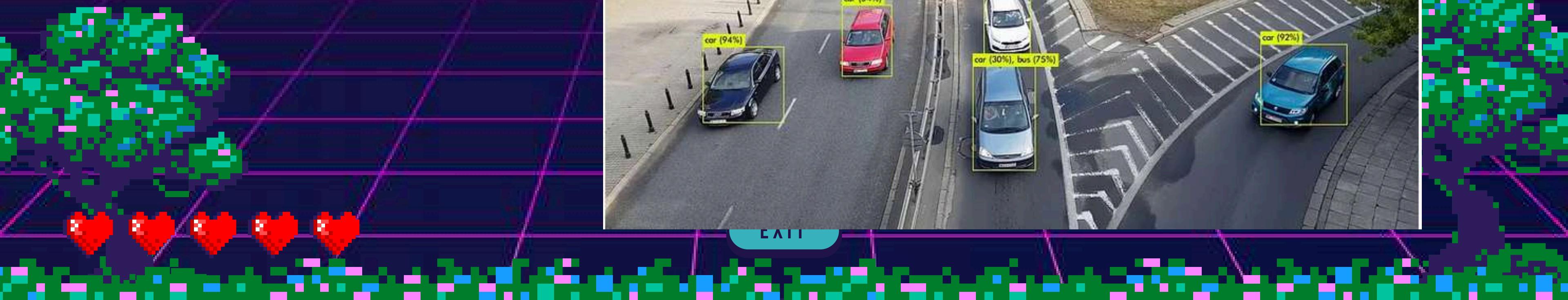
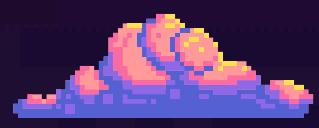
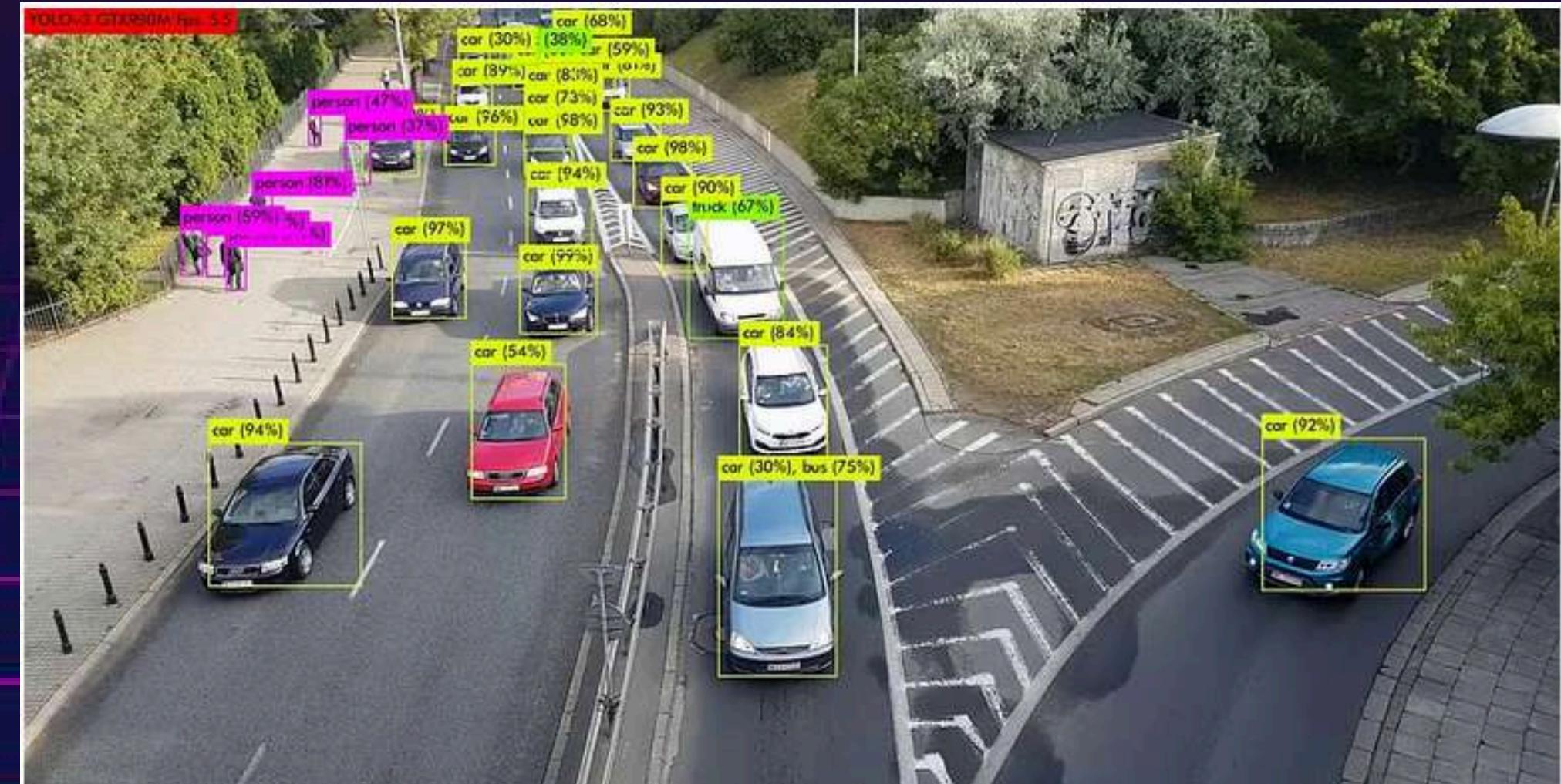




# IMAGE CLASSIFICATION

## Traffic Sector

อีกตัวอย่าง เวลาที่เราขับรถบนท้องถนน ก็จะมีกล้องที่คอยเช็คความเร็วหรืออาจจะตรวจสอบว่ารถติดไหม เราสามารถนำ AI นี้ไปช่วยได้ว่ามีรถบนถนนเยอะหรือไม่แล้วไปเชื่อมกับระบบอื่น (รวมไปถึงรถกำลังจะเข้าเส้นทึบไหม)



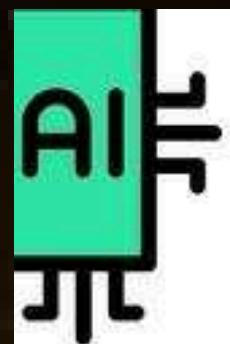


# YOLOv คืออะไร ?



มันคือ 'You only look once' ซึ่งจะเป็นหลักการทำงานของมัน และในปัจจุบันมีถึง Version 13 แล้ว

โดยตัว YOLO นี้คือสถาปัตยกรรมที่ทาง ultralytics ได้ออกแบบไว้เพื่อทำ Image Detection ได้อย่างรวดเร็วและมีประสิทธิภาพ



IMPLEMENTING A MODEL ?

Yolo



# YOLOV គឹវតារ ?

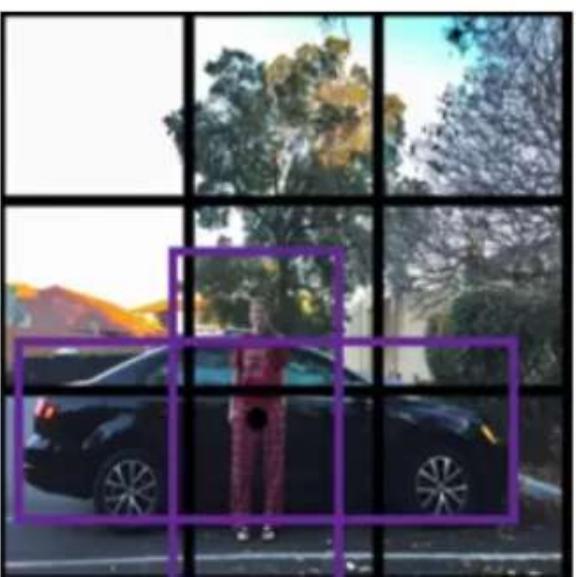


EXIT

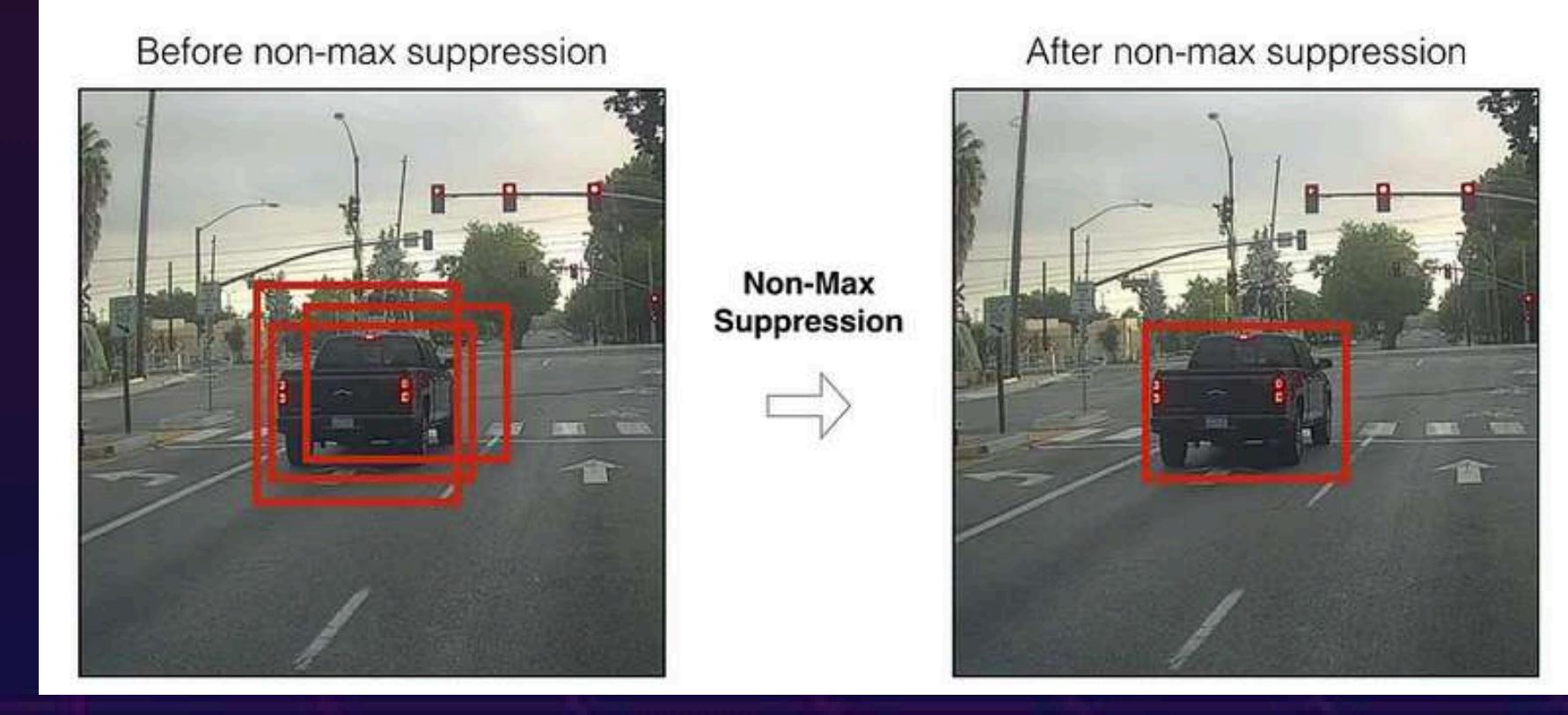
# YOLOV8 คืออะไร ?



การ grid ในแต่ละส่วนของภาพ  
เพื่อดู elements pixel



EXIT



layer ที่ predict ได้  
มีการจัดกระทำ intersection

# ขั้นตอนการพัฒนา YOLO

## 5 ขั้นตอน)

- 1) Image Detection
- 2) การดึงข้อมูลและจัดเตรียม (Data Gathering and Collecting)
- 3) การกำหนดผลลัพธ์ (Data Labeling) และการทำ Image Augmentation )
- 4) การสร้างโมเดลและวัดผล (Modelling)
- 5) การนำโมเดลไปประยุกต์ใช้ (Deployment)

บทความ medium yolo :

EXIT



1.

# ນາສຕິອຕັນ GIT

The screenshot shows the official Git website at [git-scm.com](https://git-scm.com). The page features a large search bar at the top right. On the left, there's a sidebar with links to 'About', 'Documentation', 'Downloads' (which is highlighted in red), 'GUI Clients', and 'Logos'. Below the sidebar, there's a 'Community' section. The main content area has a large heading 'Downloads' and three download links: 'macOS', 'Windows', and 'Linux/Unix'. To the right of the download links is a prominent box for the 'Latest source Release 2.42.0' with a 'Download for Windows' button. A note below the download links states: 'Older releases are available and the Git source repository is on GitHub.' At the bottom of the page is a navigation bar with 'HOME', 'ABOUT', 'DOCS', 'RELEASES', 'LOGOS', and 'CONTACT'.

git --local-branching-on-the-cheap

About Documentation Downloads GUI Clients Logos Community

Downloads

macOS Windows Linux/Unix

Latest source Release  
**2.42.0**  
Release Notes (2023-08-21)  
Download for Windows

Older releases are available and the Git source repository is on GitHub.

EXIT

# การติดตั้ง YOLOV8

2.

ดาวน์โหลด YOLOv8 เราสามารถเริ่มดาวน์โหลดและติดตั้ง Library ที่ต้องใช้

```
1 git clone https://github.com/pithakpong/hello-yolov8.git  
2 cd hello-yolov8  
3 pip install ultralytics  
4 pip install opencv-python
```

EXIT

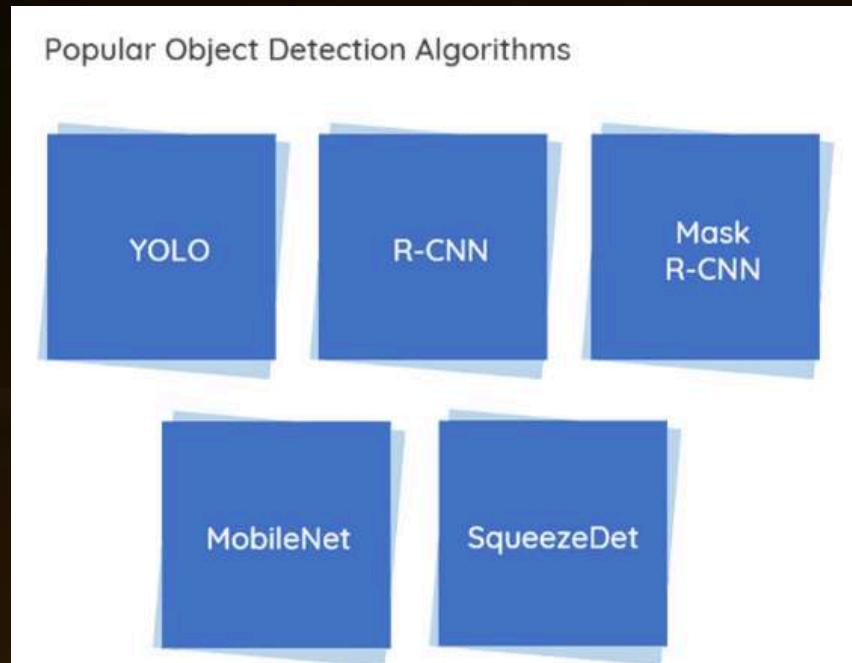


# ตัวอย่างการประยุกต์การใช้ YOLOV8

เพื่อจัดหมวดหมู่ Classification Detect ของ Person ในรูปแบบ

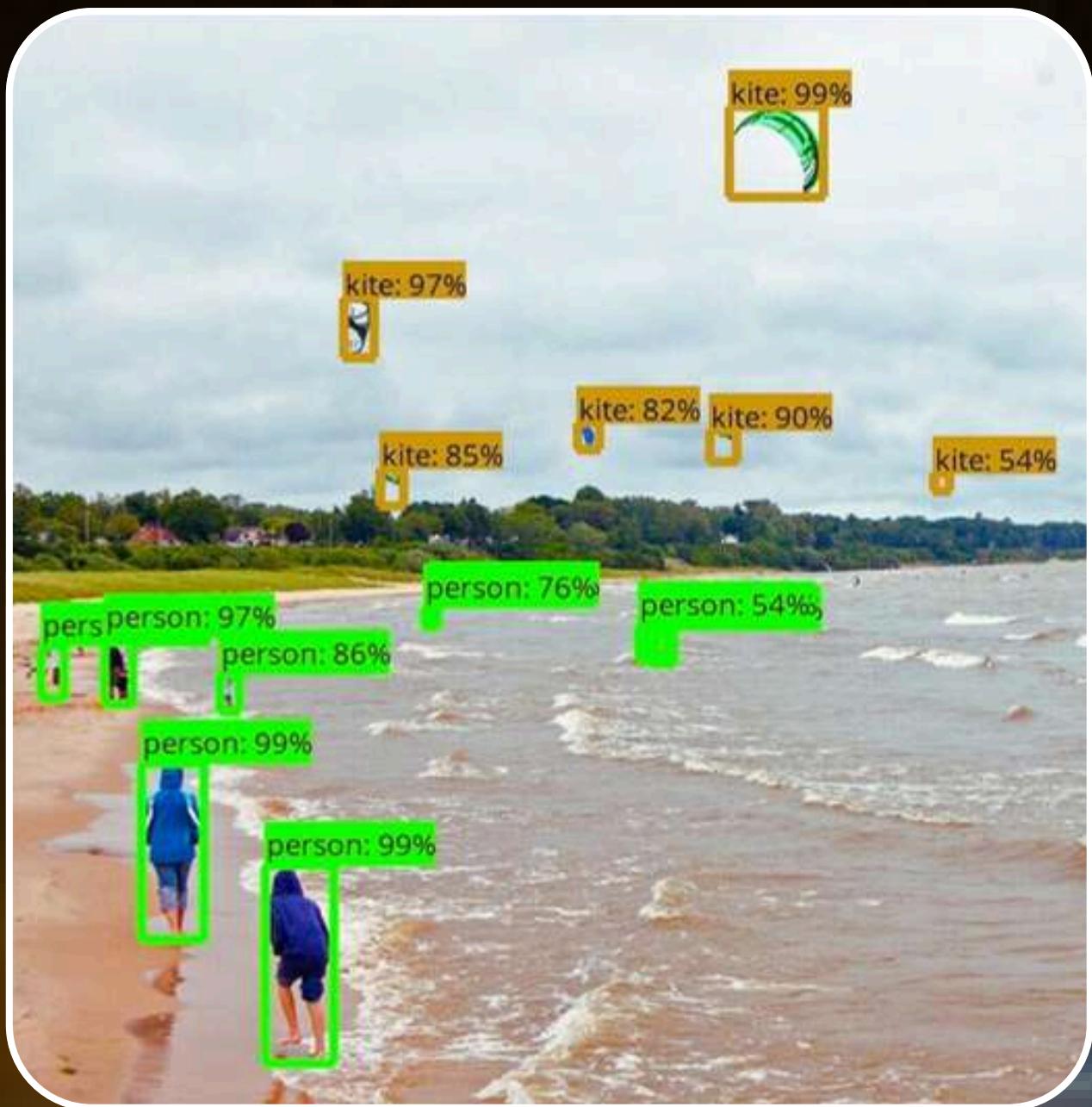
0 - 1 ( 0-100%)

ค่าความน่าเชื่อถือ (confident value)

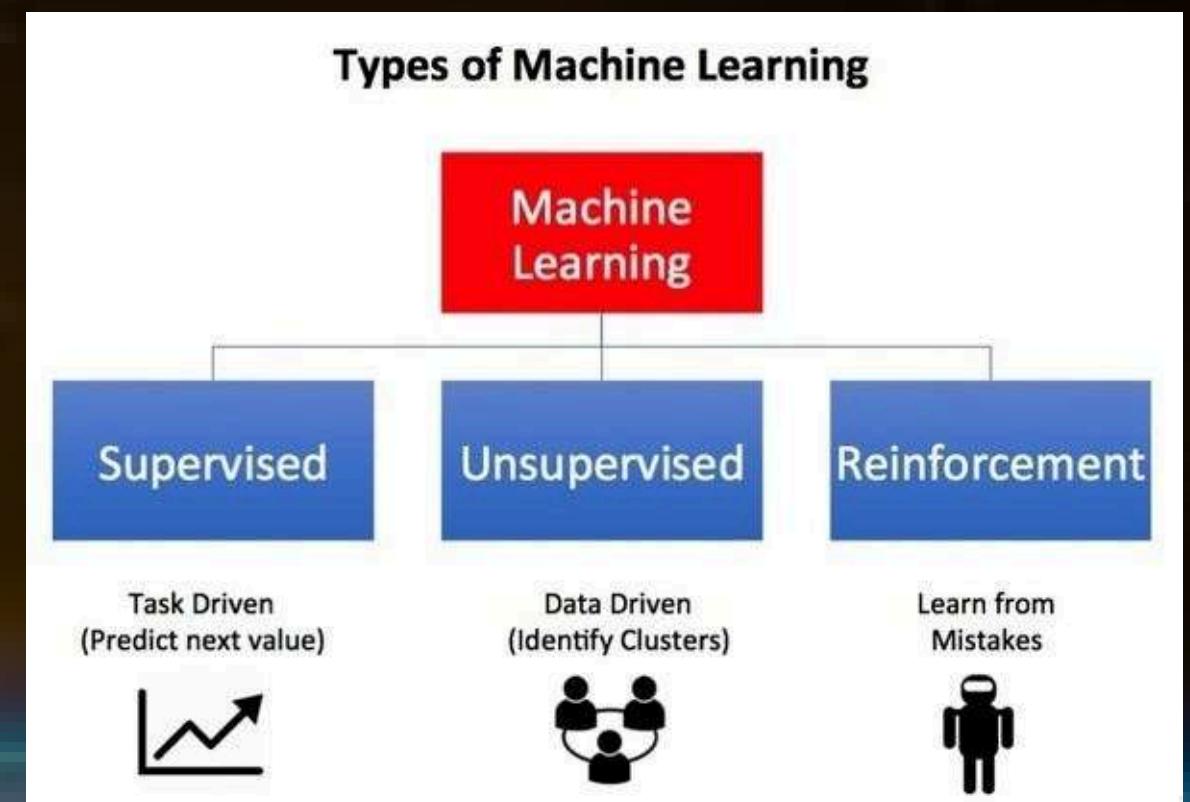


EXIT

# OBJECT DETECTION คืออะไร



Object Detection การตรวจจับวัตถุ คือ เทคโนโลยีในการคอมพิวเตอร์ หลักการที่เกี่ยวข้องกับ Computer Vision และ Image Processing ที่ใช้ในงาน AI ตรวจจับวัตถุชนิดที่กำหนด เช่น มนุษย์ รถยนต์ อาคาร ที่อยู่ในรูปภาพ วิดีโอ



งาน Object Detection การตรวจจับวัตถุในรูปภาพ สามารถเจาะลึกลงไปได้อีกหลายแขนง เช่น การคำนวณ Face Detection ตรวจจับหน้าคน Pedestrian Detection ตรวจจับคนเดินถนน สามารถประยุกต์ใช้ได้หลากหลาย เช่น ใช้ในงานรักษาความปลอดภัย และรถยนต์ไร้คนขับ เป็นต้น

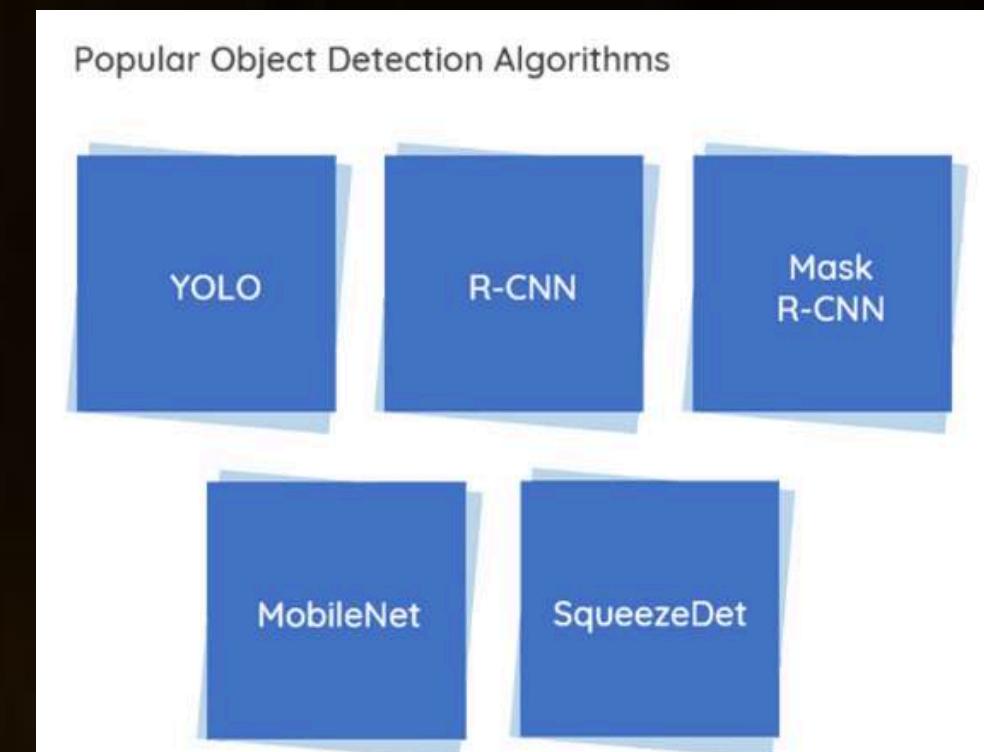
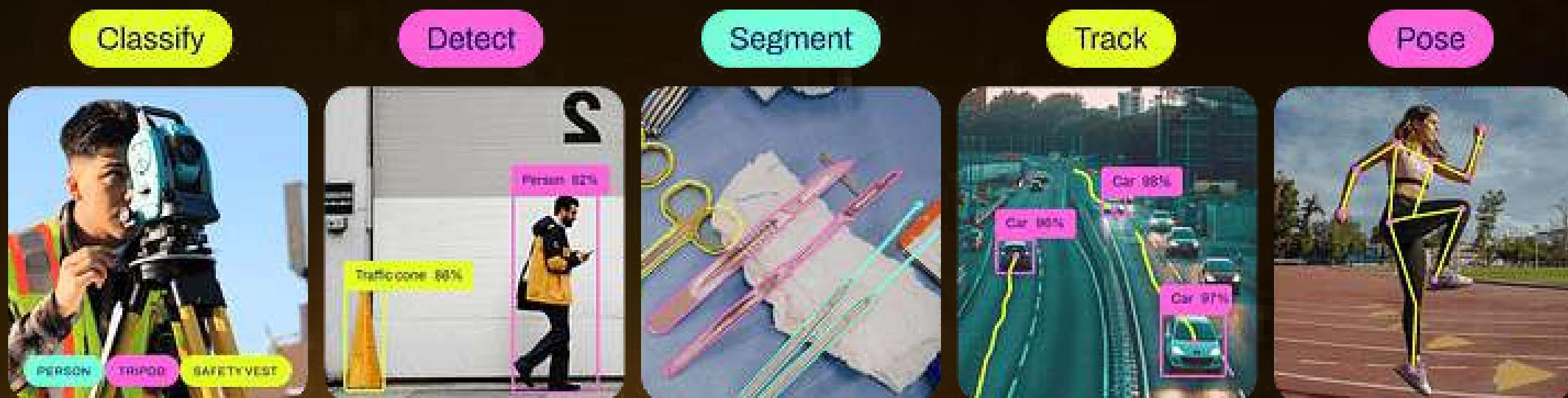


# EXPLORE ULTRALYTICS YOLOV8

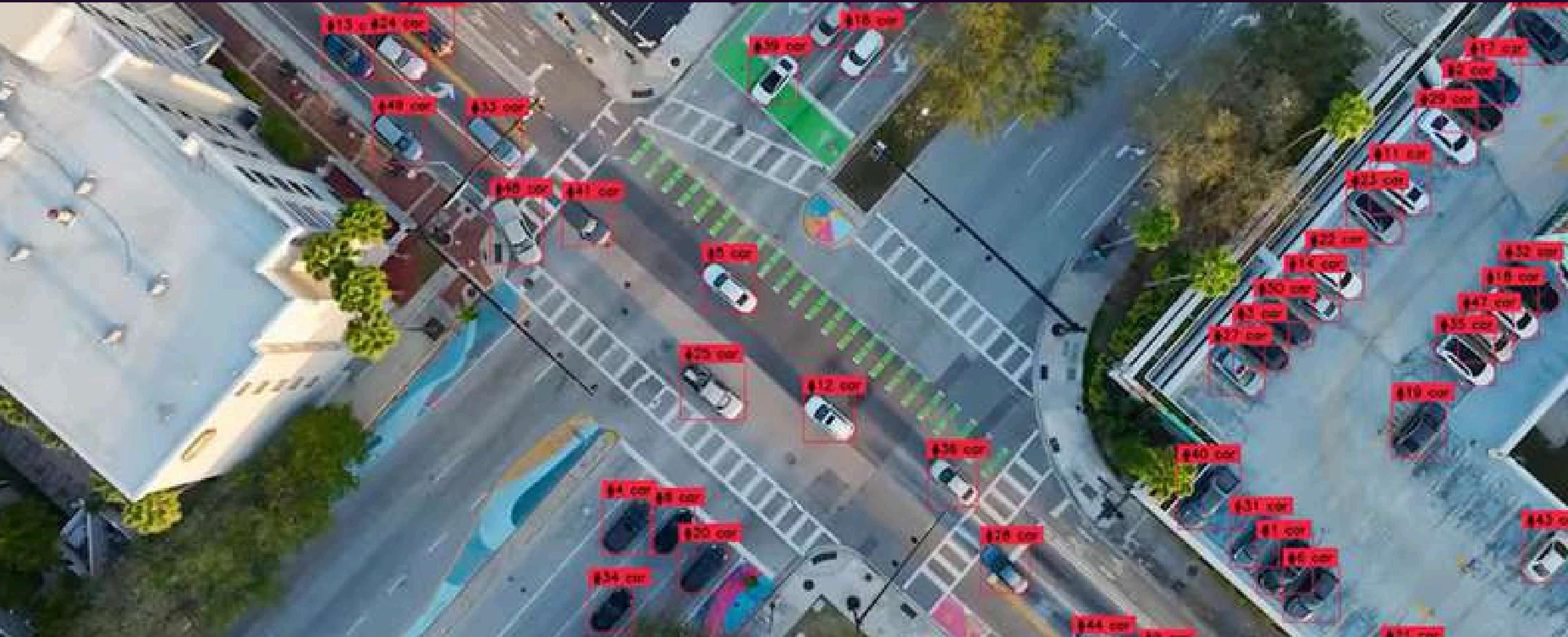
เพื่อจัดหมวดหมู่ Classification Detect ของ Person ในรูปแบบ

0 - 1 ( 0-100%)

ค่าความน่าเชื่อถือ (confident value)



# ตัวอย่างการประยุกต์การใช้ YOLOV8



▶ **DETECTION**

click

# ตัวอย่างการประยุกต์การใช้ YOLOV8



▶ **TRACKING**

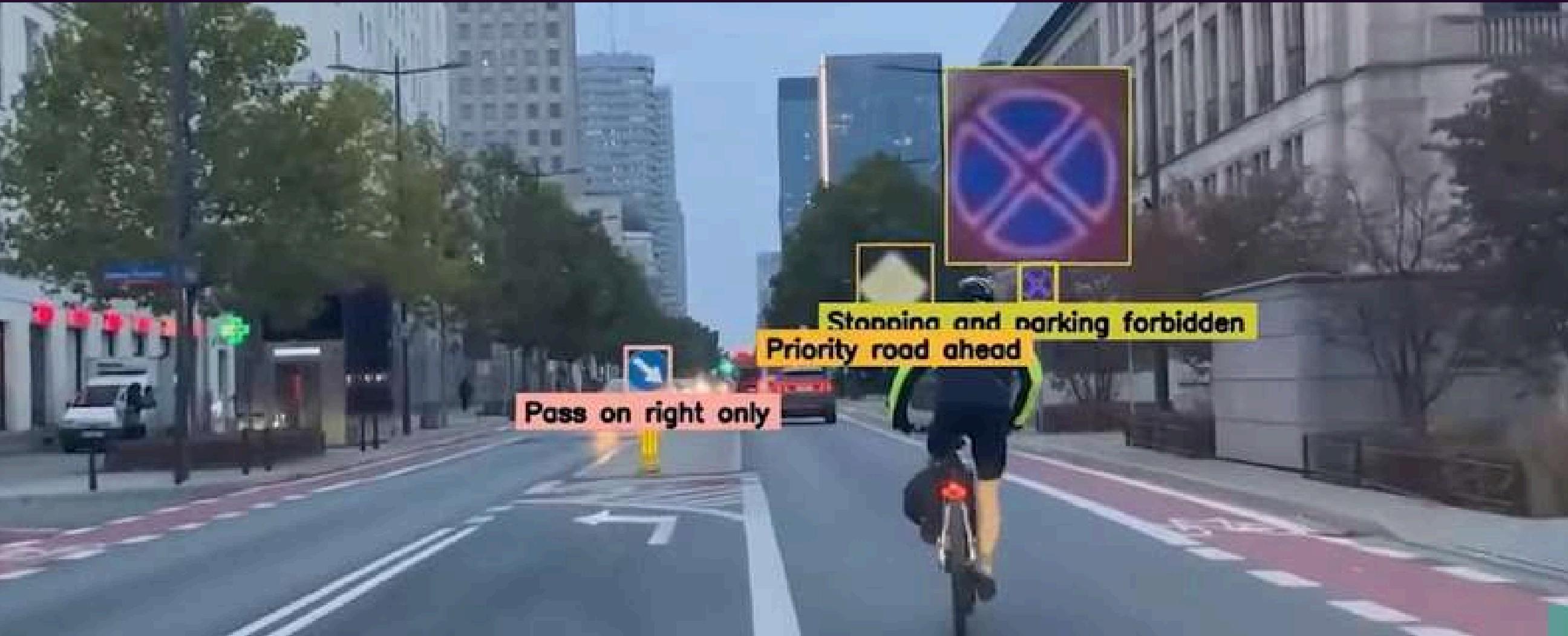
click

# ตัวอย่างการประยุกต์การใช้ YOLOV8



➤ COUNTING  
click

# ตัวอย่างการประยุกต์การใช้ YOLOV8



➤ ANALYSIS

click

**2020**



# ROBOFLOW ACCOUNT

roboflow Products Solutions Resources Pricing Docs Blog Sign In Book a demo Get Started

Everything you need to build and deploy computer vision applications.

Used by over 1 million engineers to create datasets, train models, and deploy to production.

Get Started Request a Demo

Detection Tracking Counting Analysis

# ROBOFLOW WORKSPACE

The screenshot shows the Roboflow Workspace interface. On the left is a sidebar with the Roboflow logo and a user profile for "0490\_TWEEB". The sidebar includes links for "Projects", "Workspaces", "Monitoring", "Deployments", "Explore", "Settings", "Help & Docs", "Notifications", and "Upgrade". The main area is titled "Projects" and displays three projects:

- ABU Ball Cyro**: Edited a year ago, Public, 589 Images + 2 Models.
- ABU Ball Colors**: Edited a year ago, Public, 435 Images + 2 Models.
- ABU Ball Colors - 2**: Edited a year ago, Public, 437 Images + 2 Models.

At the top right of the workspace are buttons for "New Folder" and "New Project", along with color-coded circular icons for "CTO" and "Invite Team".

# ROBOFLOW CREATE PROJECT

Let's create your project.

ABU Ball >  Public: Mouse computer classify No space use '-' instead.

Project Name  Annotation Group  Visibility  Private  Public Licenses  CC BY 4.0 Tool  Traditional  Rapid

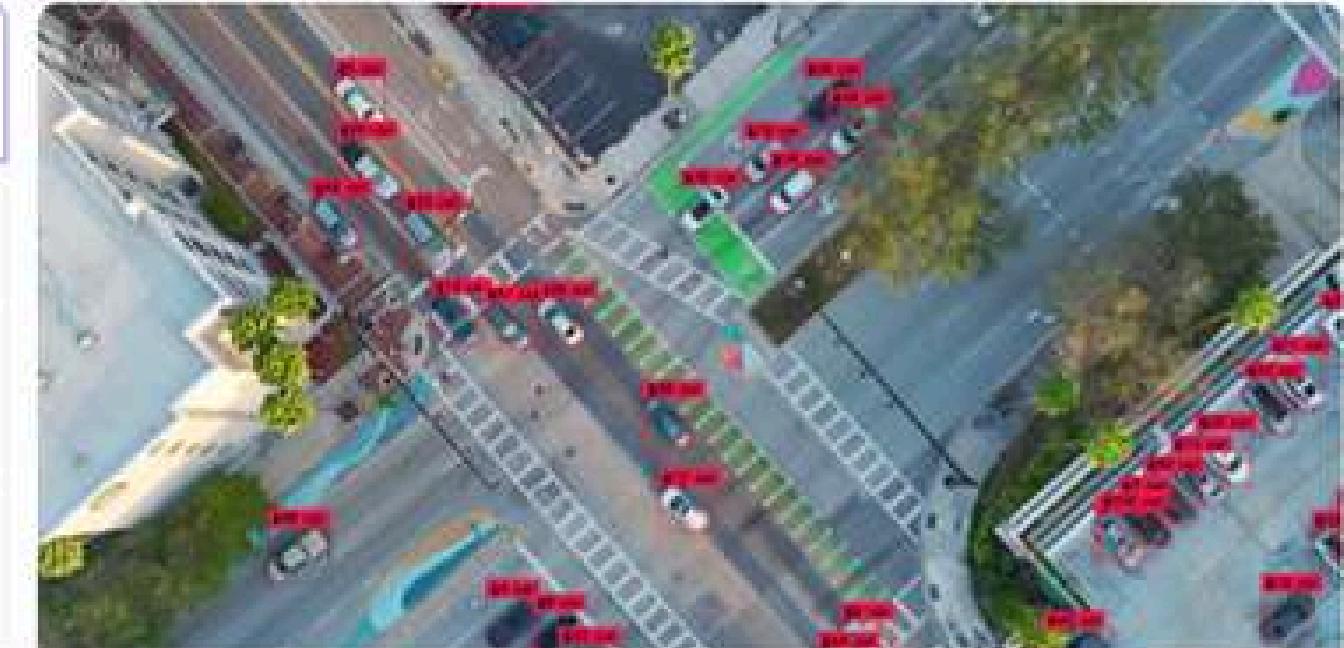
Project Type 1 2

Object Detection  Bounding Boxes  Counts  Tracking  
Identify objects and their positions with bounding boxes.

Classification  Image Labels  Filtering  Content Moderation  
 Single-Label  Multi-Label  
Assign labels to the entire image.

Instance Segmentation  Polygons  Measuring  Odd Shapes  
Detect multiple objects and their actual shape.

Keypoint Detection  Skeleton Structure  Pose Estimation  
Identify keypoints ("skeletons") on subjects.



Cancel Create Public Project

# ROBOFLOW PROJECT

The screenshot shows the Roboflow web interface. On the left, a sidebar lists project components: **Annotations**, **Images**, **Labels**, **Metrics**, **Model**, **Object Detection**, **Project**, **Script**, **Tensorboard**, **Train**, **Visualize**, and **Weights**. The **Annotations** section is currently active. The main area is titled **Upload** and displays a large central upload button with the text "Drag and drop file(s) to upload, or:". Below this are four file selection buttons: **Select File(s)**, **Select Folder**, **Supported Formats** (which is highlighted with a red box), and **Supported Labels**. The **Supported Formats** section lists **Images** (formats: jpg, png, bmp, webp, tiff), **Annotations** (formats: json, tfrecord), **Videos** (formats: mp4), and **PDFs**. To the right of the main upload area, there's a sidebar with sections for **Need images to get started?**, **Upload data from your phone** (with a QR code), **Search on Roboflow Universe** (with a search bar), and **Bulk Upload Images**.

# ROBOFLOW ANNOTATE

Annotations

Sort By: Newest

Unassigned 0 batches

Upload More Images

Annotating 0 items

Dataset 1 job

See all 50 images

Clone on 9/24/2023 from pnu-og7cn/Mouse...

Labeler: 0491\_THEERAMET CHUAPAYUNO

50 Images

File img

Next Page

The screenshot shows the Roboflow Annotate interface. On the left, there's a sidebar with various options like 'Annotate', 'Dataset', 'Versions', 'Analytics', 'Classes & Tags', 'Models', 'Visualize', 'Deploy', and 'Deployments'. The main area is titled 'Annotations' and shows three sections: 'Unassigned' (0 batches), 'Annotating' (0 items), and 'Dataset' (1 job). The 'Dataset' section includes a 'See all 50 images' link and details about the clone (9/24/2023) and labeler (0491\_THEERAMET CHUAPAYUNO). In the center, there's a grid of images labeled 'File img'. A large red arrow points from the 'File img' button to the 'Next Page' button at the bottom right.

# ROBOFLOW ANNOTATE

Annotations → Batch  
Uploaded on 09/24/25 at 1:21 pm  
Uploaded Sep 24, 2025 (1:21 PM)

Upload More Rename

How do you want to label your images?

- Label Myself**  
Label images with our AI labeling tools.
- Label With My Team**  
Split up the labeling work across your team.
- Auto-Label Entire Batch**  
Use your own custom model or a zero-shot model to automatically label your entire batch.
- Hire Outsourced Labelers** Upgrade  
Work with a professional labeling team vetted by Roboflow.

Annotations

Dataset

Upload Data

Annotate

Version

Analytics

Classes & Tags

Models

Visualize

Deploy

Deployments

# ROBOFLOW WORKSPACE

Uploaded on 09/24/25 at 1:21 pm

Start Annotating      

Sort By: Newest

**Progress**

12 Images  
0 Annotated  
12 Unannotated

**Unannotated** (12)   **Annotated** (0)

**Instructions** 

No specific instructions were added when this job was assigned

**Assignment** 

0491\_THEERAMET CHUAI PAYUNG  
Labeler

**Timeline**

Job created via API and assigned it to 69010491@kmit.ac.th  
24/9/2568 13:27:23

  
20240104\_130505.jpg

  
20240104\_130547.jpg

  
20240104\_130505.jpg

  
20240104\_130505.jpg

  
20240104\_130535.jpg

  
20240104\_130547.jpg

  
20240104\_130547.jpg

  
20240104\_130505.jpg

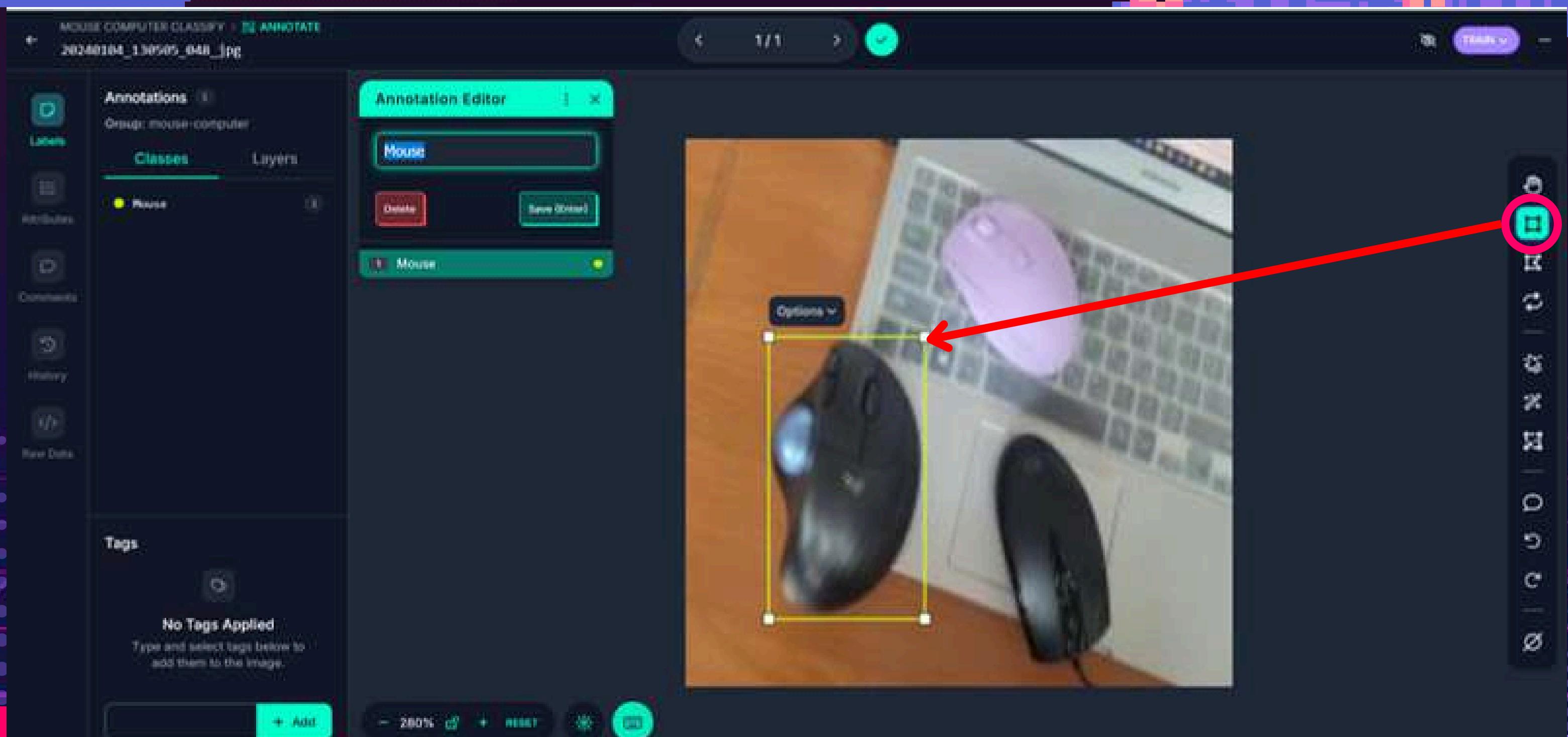
  
20240104\_130529.jpg

  
20240104\_130528.jpg

  
20240104\_130531.jpg

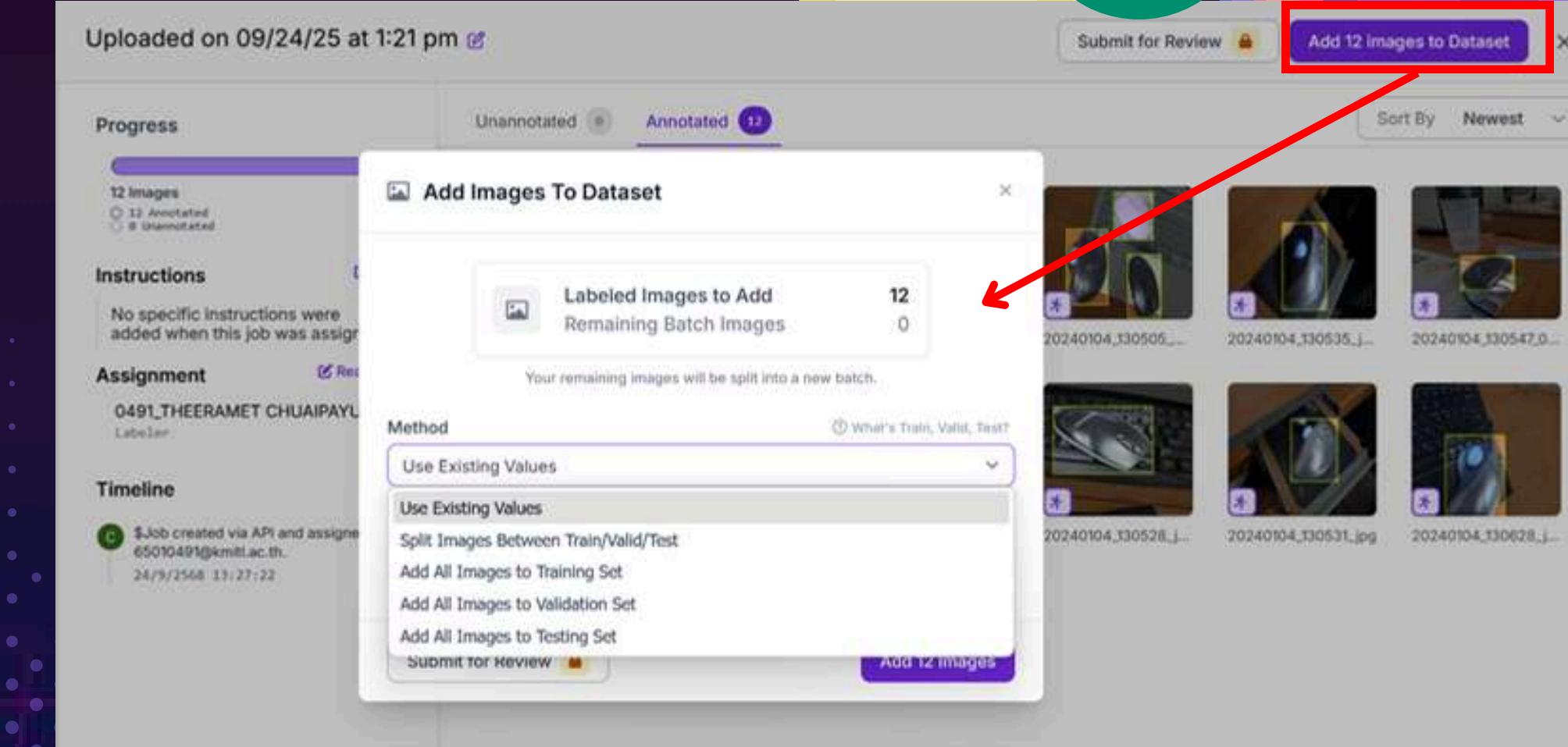
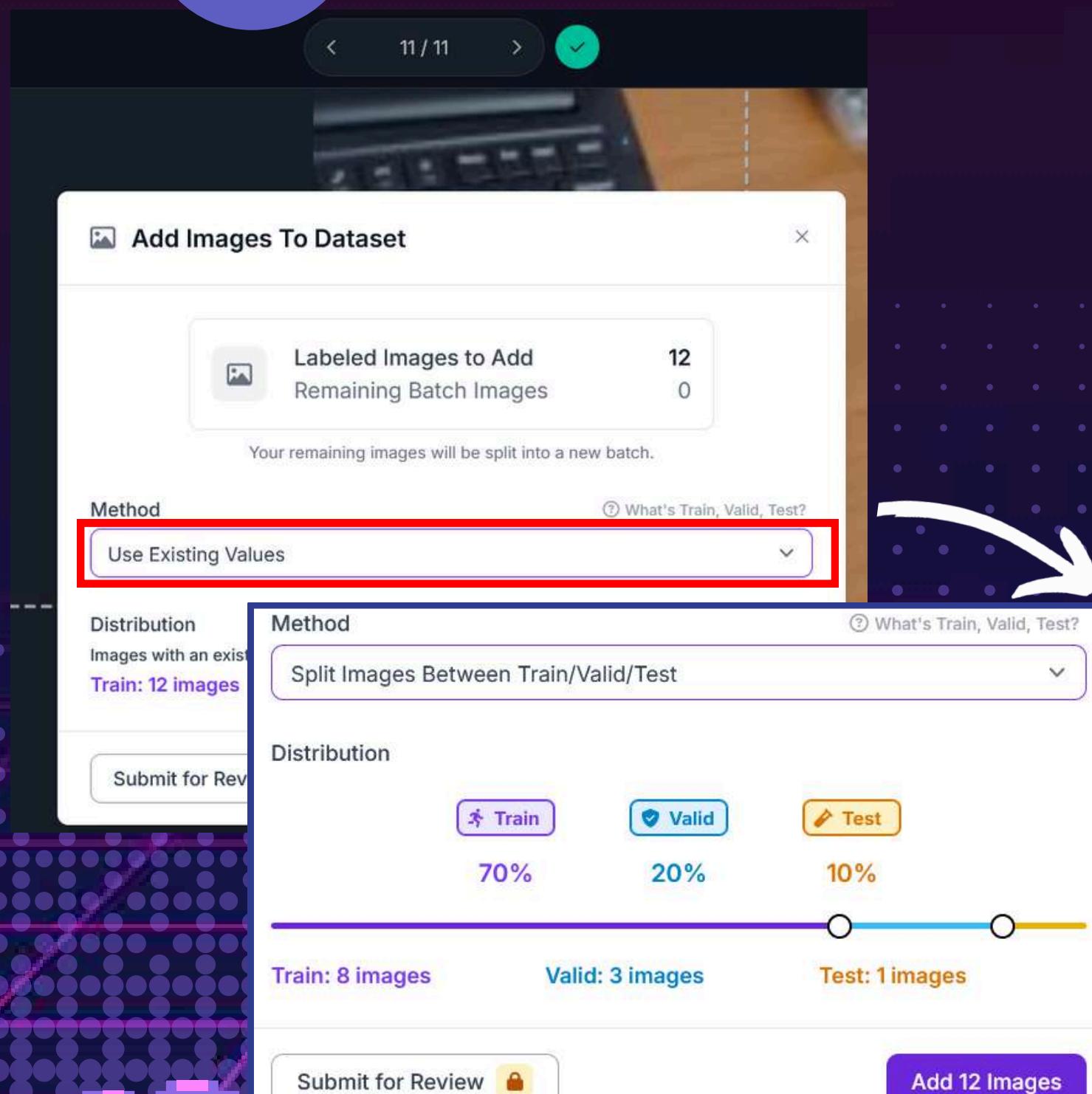
  
20240104\_130505.jpg

# ROBOFLOW CREATE PROJECT



1

# ROBOFLOW SPLIT DATASET



2

# ROBOFLOW MODELS

Models

Fine-tuned 1 Universe 0

Search models... Model Type: All ▾ Sort By: Newest First

MODEL NAME	UPDATED	METRICS	TYPE	DATASET VERSION	LICENSE
Roboflow Instant (v1) ⓘ abu-ball/mouse-computer-classify-v1	24/9/2023 16:25	...	Roboflow Instant	Fine-tuned on your dataset.	...

# ROBOFLOW VERSION

The screenshot shows the Roboflow web interface with a dark-themed header featuring the text "ROBOFLOW VERSION". Below the header is a navigation bar with various icons and links: "View in Universe", "Create New Version" (highlighted with a red box), "Dataset", "Analytics", "Classes & Tags", "Models", "Visualize", "Deploy", and "Deployments". The main content area is titled "Versions" and displays a single version entry: "v1 Roboflow Instant 1 [Eval]" generated on Sep 24, 2025 by Grid. A purple button labeled "Train" is visible on the left side of the version card. Below the version card, a message states "This version doesn't have a model." followed by a description of training options. A "Get More Credits" button and a "How to Upload Custom Weights" link are also present. At the bottom, it shows "62 Total Images" and a "View All Images" link.

+ Create New Version

Versions

v1 Roboflow Instant 1 [Eval]

Generated on Sep 24, 2025 by Grid

This version doesn't have a model.

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

Get More Credits

Available Credits:

How to Upload Custom Weights

62 Total Images

View All Images →

# IMAGE PRE-PROCESS

**Versions**

**Source Images**  
Images: 62  
Classes: 1  
Unannotated: 0

**Train/Test Split**  
Training Set: 48 images  
Validation Set: 10 images  
Testing Set: 4 images

**3 Preprocessing**  
What can preprocessing do?  
Decrease training time and increase performance by applying image transformations to all images in this dataset.

Auto-Orient

+ Add Preprocessing Step

Continue

**Preprocessing Options**

Preprocessing can decrease training time and increase inference speed.

Isolate Objects    Static Crop    Dynamic Crop    Resize    Grayscale

Auto-Adjust Contrast    Tile    Modify Classes    Filter Null    Filter by Tag

Cancel

Auto-Orient

Resize  
Stretch to 640x640

+ Add Preprocessing Step

Continue

# IMAGE AUGMENTATION

**Versions**

Roboflow Instant 1 [Eval]  
v1 62

**Preprocessing**  
Auto-Orient: Applied  
Resize: Stretch to 640x640

**4 Augmentation**  
Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

**Add Augmentation Step**

**Use Previous Augmentations**  
Use augmentations from a previous version.

**Continue**

**Augmentation**

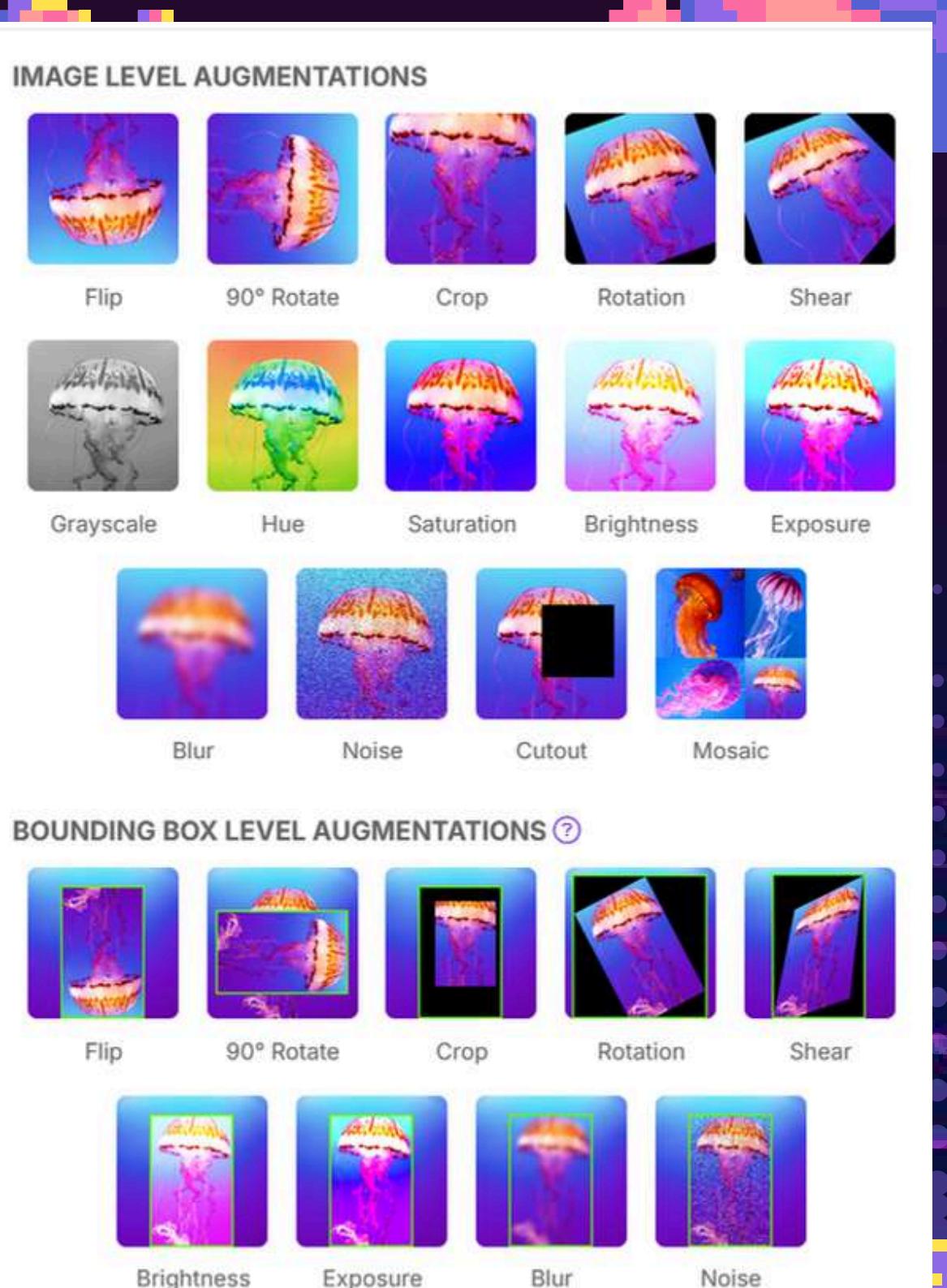
What can augmentation do?  
Create new training examples for your model to learn from augmented versions of each image in your training set.

**Flip**  
Horizontal, Vertical Edit

**Rotation**  
Between -12° and +12° Edit

**Shear**  
±11° Horizontal, ±10° Vertical Edit

**Blur**  
Up to 2.5px Edit



# IMAGE AUGMENTATION

## Versions

**Versions**

**Mouse Four Augment**

Generated on Sep 24, 2025 by 0491\_THEERAMET CHUAIPIYUNG

v2 158 640x640  
Stretch to  
0491\_THEERAMET C...

**Roboflow Instant 1 [Eval]**

v1 62 Grid

+ Create New Version

This version doesn't have a model.

Train an optimized, state of the art model with Roboflow or upload a custom trained model to use features like Label Assist and Model Evaluation and deployment options like our auto-scaling API and edge device support.

Get More Credits

Available Credits: 0

How to Upload Custom Weights

158 Total Images



[View All Images →](#)

Download Dataset Edit

## Download

### Image and Annotation Format

YOLOv8

TXT annotations and YAML config used with YOLOv8.

### Download Options

Download zip to computer

Downloads all images, annotations, and classes.

Show download code

Custom train this dataset using the provided code snippet in a notebook.

Cancel

Continue

## Download

Jupyter

Terminal

Raw URL

Paste this snippet into a [notebook](#) from our model library to download and unzip your dataset:

```
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="REDACTED")
project = rf.workspace("abu-ball").project("mouse-computer-classify-rsliy")
version = project.version(2)
dataset = version.download("yolov8")
```

⚠ Warning: Do not share this snippet beyond your team, it contains a private key that is tied to your Roboflow account.  
Acceptable use policy applies.

Copy Snippet and Open Notebook

# การใช้ YOLOV8

yolov8 size:

Classification	Detection	Segmentation	Kind
yolov8n-cls.pt	yolov8n.pt	yolov8n-seg.pt	Nano
yolov8s-cls.pt	yolov8s.pt	yolov8s-seg.pt	Small
yolov8m-cls.pt	yolov8m.pt	yolov8m-seg.pt	Medium
yolov8l-cls.pt	yolov8l.pt	yolov8l-seg.pt	Large
yolov8x-cls.pt	yolov8x.pt	yolov8x-seg.pt	Huge

```
model = YOLO("yolov8m.pt")
```

```
results = model.predict("video.mp4", save=True)
```

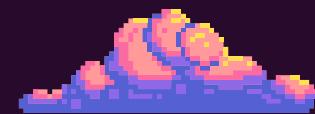
```
result = results[0] box = result.boxes[0]
```

ตัว box มีหลาย properties ประกอบด้วย

- xyxy : พิกัดของ bounding box [x1,y1,x2,y2]
- cls : Id ของ class ที่กำหนด
- conf : ค่าความมั่นใจของผลลัพธ์ที่กำหนด

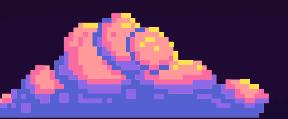
EXIT

# MSFT YOLOV8



yolov8 class:

ID	Label	ID	Label	ID	Label	ID	Label	ID	Label
0	person	1	bicycle	2	car	3	motorcycle	4	airplane
5	bus	6	train	7	truck	8	boat	9	traffic light
10	fire hydrant	11	stop sign	12	parking meter	13	bench	14	bird
15	cat	16	dog	17	horse	18	sheep	19	cow
20	elephant	21	bear	22	zebra	23	giraffe	24	backpack
25	umbrella	26	handbag	27	tie	28	suitcase	29	frisbee
30	skis	31	snowboard	32	sports ball	33	kite	34	baseball bat
35	baseball glove	36	skateboard	37	surfboard	38	tennis racket	39	bottle
40	wine glass	41	cup	42	fork	43	knife	44	spoon
45	bowl	46	banana	47	apple	48	sandwich	49	orange
50	broccoli	51	carrot	52	hot dog	53	pizza	54	donut
55	cake	56	chair	57	couch	58	potted plant	59	bed
60	dining table	61	toilet	62	tv	63	laptop	64	mouse
65	remote	66	keyboard	67	cell phone	68	microwave	69	oven
70	toaster	71	sink	72	refrigerator	73	book	74	clock
75	vase	76	scissors	77	teddy bear	78	hair drier	79	toothbrush



# GOOGLE COLAB TRAINING YOLOV8



## Step 5: Exporting dataset

Once the dataset version is generated, we have a hosted dataset we can load directly into our notebook for easy training, and select the `YOLO v5 PyTorch` dataset format.

A screenshot of the Roboflow web interface. A modal window titled "Your Download Code" is open, showing code snippets for Jupyter, Terminal, and Raw URL. The raw URL snippet is copied to the clipboard. The background shows a list of datasets and a progress bar for generating a dataset.

Paste this snippet into a [notebook](#) from our model library to download and unzip your dataset:

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="REDACTED")
project = rf.workspace("abu-ball").project("mouse-computer-classify-rsliy")
version = project.version(2)
dataset = version.download("yolov8")
```

!mkdir {HOME}/datasets  
%cd {HOME}/datasets

```
!pip install -U albumentations  
!pip install -U ultralytics  
!pip install roboflow
```

```
from roboflow import Roboflow
rf = Roboflow(api_key="riPAXao7CKmIMVdrSkPD")
project = rf.workspace("phattara").project("car-detection-nxsxm-utc5")
version = project.version(1)
dataset = version.download("yolov8")
```

```
# download_dir = "/content/datasets/bo-1"
# version = project.version(1)
# dataset = version.download("yolov8", download_dir) # Specify download directory here!
```

A red arrow points from the "Raw URL" section of the Roboflow snippet to the "Raw URL" section of the Google Colab snippet.

# GOOGLE COLAB TRAINING YOLOV8

```
%cd (HOME)
# yolo task=detect mode=train model=yolov8s.pt data=(dataset.location)/data.yaml epochs=50 imgs=
# yolo task=detect mode=train model=yolov8s.pt data=(dataset.location)/data.yaml epochs=25 imgs=
# yolo task=detect mode=train model=yolov8n.pt data=(dataset.location)/data.yaml epochs=64 imgs=
# yolo task=detect mode=train model=yolov8s.pt data=(dataset.location)/data.yaml epochs=32 imgs=
[Errno 2] No such file or directory: '(HOME)'

/content
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8s.pt to 'yolov8s.pt'...
100% 21.5M/21.5M [00:00<00:00, 264MB/s]
Ultralytics 8.3.138 Python 3.11.12 torch 2.6.0+cu124 CUDA 0 (Tesla T4, 15095MB)
engine/trainer: agnostic_nms=False, amp=True, augment=False, auto_augment=randaugment, batch=8, t
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/ultralytics/Arial.ttf'...
100% 755k/755k [00:00<00:00, 95.7MB/s]
Overriding model.yaml nc=80 with nc=1

from n params module arguments
0 -1 1 938 ultralytics.nn.modules.conv.Conv [3, 32, 3, 2]
1 -1 1 10560 ultralytics.nn.modules.conv.Conv [32, 64, 3, 2]
2 -1 1 29056 ultralytics.nn.modules.block.C2f [64, 64, 1, True]
3 -1 1 73984 ultralytics.nn.modules.conv.Conv [64, 128, 3, 2]
4 -1 2 197632 ultralytics.nn.modules.block.C2f [128, 128, 2, True]
5 -1 1 295424 ultralytics.nn.modules.conv.Conv [128, 256, 3, 2]
6 -1 2 788480 ultralytics.nn.modules.block.C2f [256, 256, 2, True]
7 -1 1 1180672 ultralytics.nn.modules.conv.Conv [256, 512, 3, 2]
8 -1 1 1834080 ultralytics.nn.modules.block.C2f [512, 512, 1, True]
9 -1 1 656896 ultralytics.nn.modules.block.SPPF [512, 512, 5]
10 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
11 [-1, 6] 1 0 ultralytics.nn.modules.conv.Concat [1]
12 -1 1 591360 ultralytics.nn.modules.block.C2f [768, 256, 1]
13 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
14 [-1, 4] 1 0 ultralytics.nn.modules.conv.Concat [1]
```

best.pt คือ pre-train model ที่จะใช้งาน  
จะใช้ epoch ที่เทรนได้ค่า val, mAP ที่ดีที่สุด

Path runs/detect/train/weights  
แล้ว download ไฟล์ลงเครื่อง

# Vscode Install

Q1

Install Python

open your command prompt

Q2

```
pip install opencv-python
```

```
pip install numpy
```

```
pip install ultralytics
```

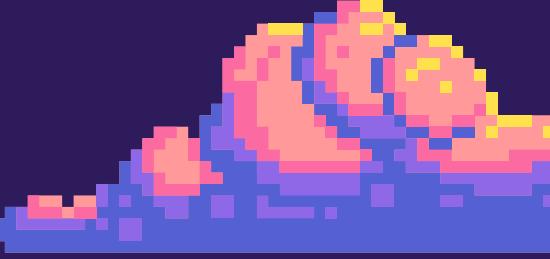


We'll use Visual Studio Code



- Create new file your project (.py file)

# ถ้าจะใช้ GPU (CUDA) ให้ติดตั้ง PyTorch ที่รองรับ GPU



**Q3.1**

```
pip install torch torchvision --index-url https://download.pytorch.org/whl/cu121
```

เปลี่ยน cu121 เป็นเวอร์ชันที่ตรงกับ driver CUDA ของเครื่องคุณ หรือใช้ pip install torch torchvision torchaudio ถ้าใช้ CPU เท่านั้น

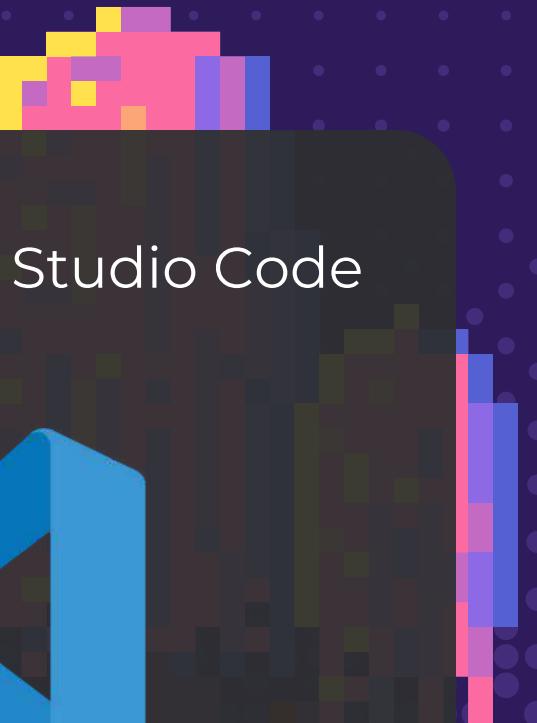
**Q3.2**

```
pip install torch torchvision torchaudio
```

We'll use Visual Studio Code



- Create new file your project (.py file)



# VSCODE REAL-TIME YOLOV8 DETECTION



A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a folder named 'YOLOv8 MODEL (FINAL THOUGH)' containing two files: 'mouse-computer.pt' and 'RealTime\_yolov8.py'. The main editor area displays the following Python code:

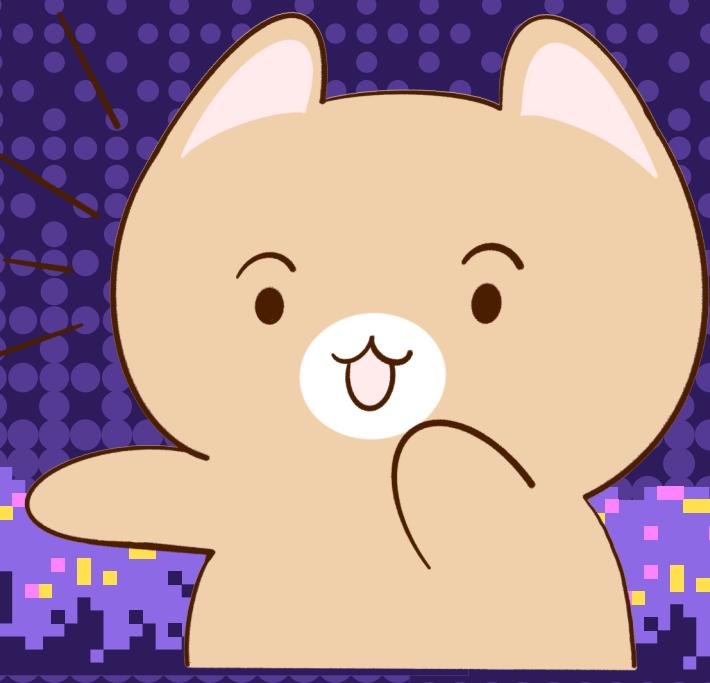
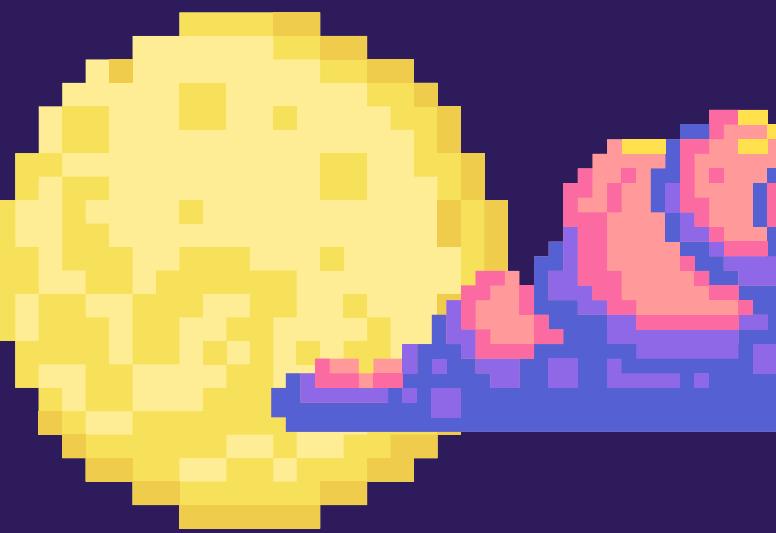
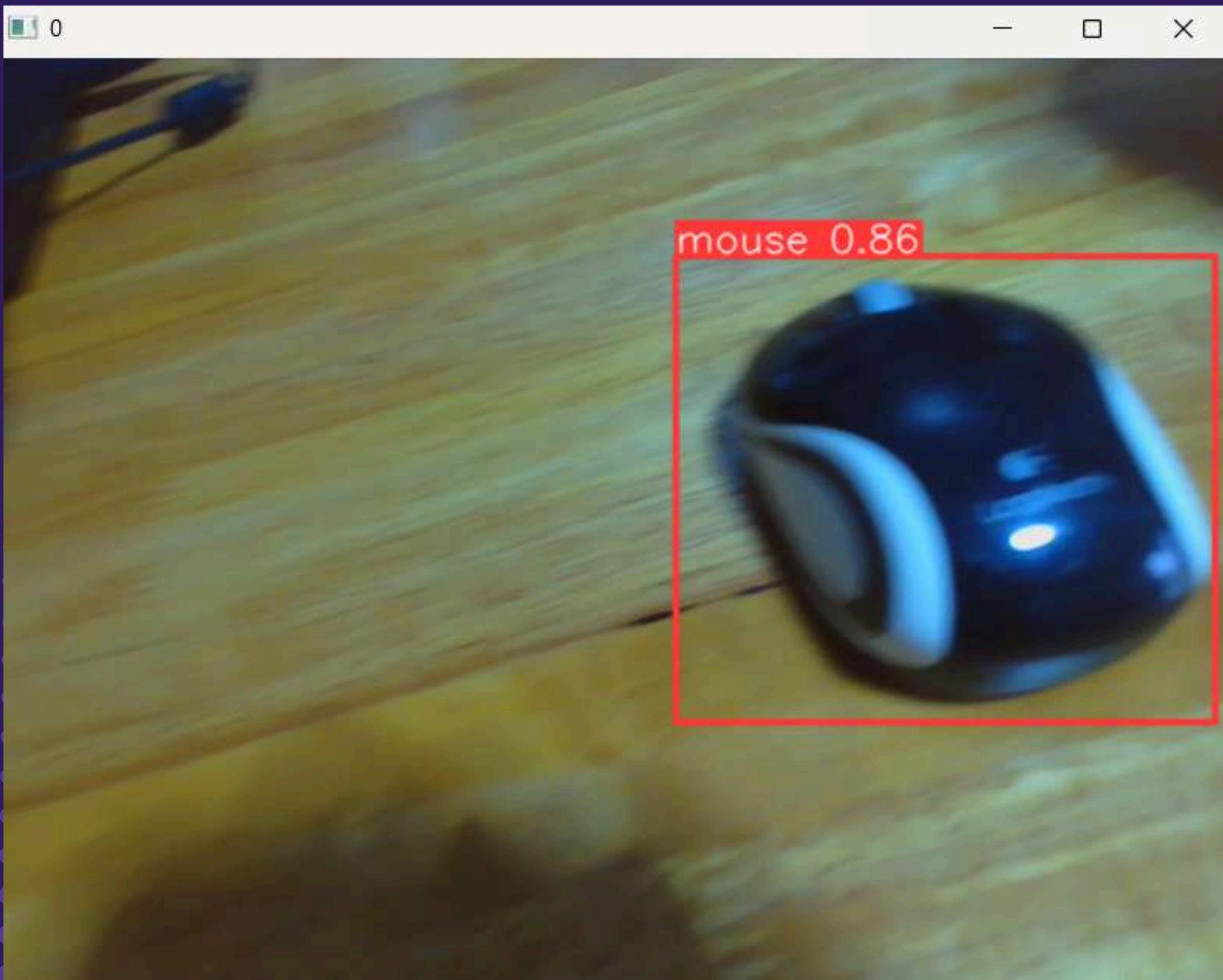
```
from ultralytics import YOLO
from ultralytics.models.yolo.detect import DetectionPredictor
import cv2

model = YOLO("mouse-computer.pt")

result = model.predict(source="0", show=True) # accept(0) for camera, -1 for img/0.jpg/0.jpg/0.jpg
print(result)
```

paste pre-train model ใน folder เดียวกันกับไฟล์ python

# RESULT:

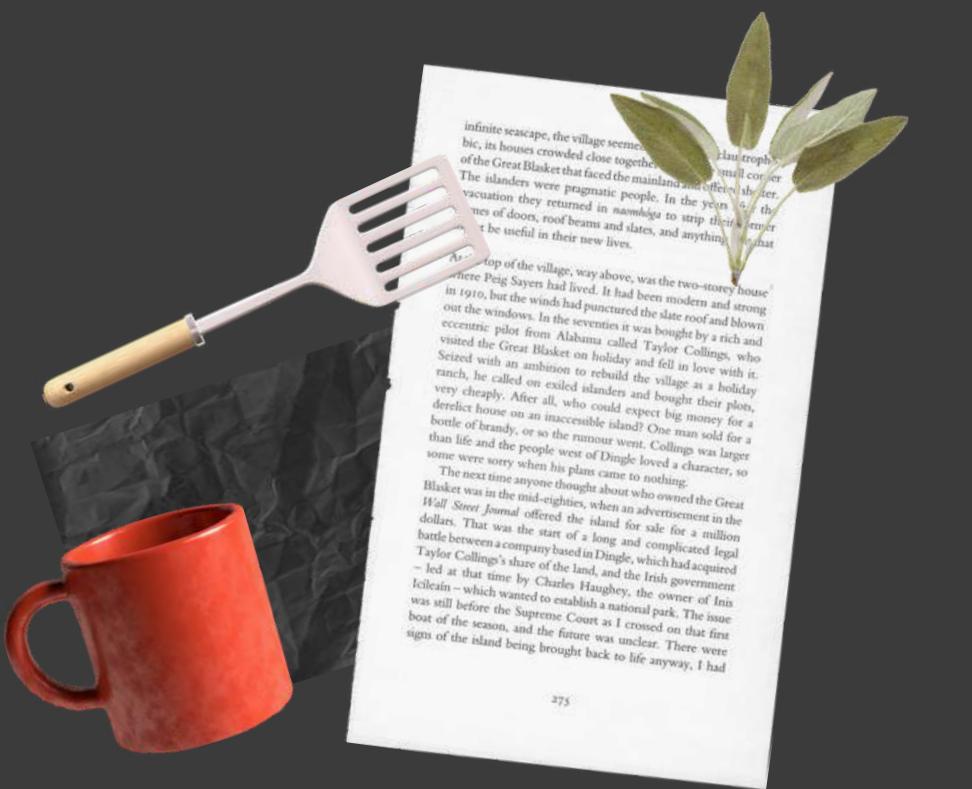




# BREAK SLIDES

LET'S TAKE A BREAK 10 MINUTES!

- Taking breaks during work is a good thing
- Don't work too hard, get some rest



# Anaconda Navigator

Anaconda  
<https://www.anaconda.com>

## Anaconda | The World's Most Popular Data Science Platform

Anaconda is the birthplace of Python data science. We are a movement of data scientists, data-driven enterprises, and open source communities.

**Free Download**

Conda is an open-source p

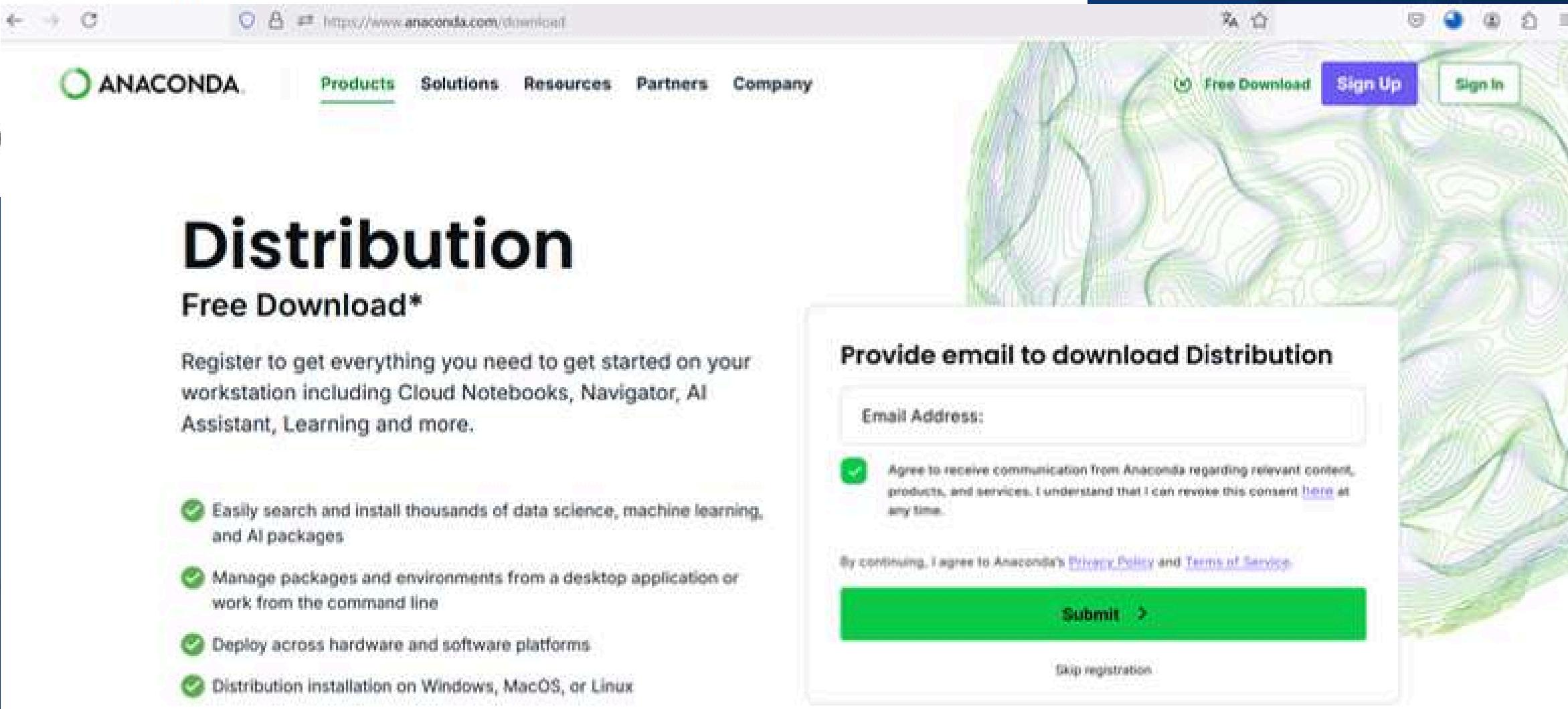
## Distribution

**Free Download\***

Register to get everything you need to get started on your workstation including Cloud Notebooks, Navigator, AI Assistant, Learning and more.

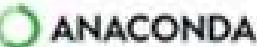
- Easily search and install thousands of data science, machine learning, and AI packages
- Manage packages and environments from a desktop application or work from the command line
- Deploy across hardware and software platforms
- Distribution installation on Windows, Mac OS, or Linux

\*Use of Anaconda's Offerings at an organization of more than 200 employees requires a Business or Enterprise license. See Pricing.



The screenshot shows the Anaconda website's distribution download page. At the top, there's a navigation bar with links for Products, Solutions, Resources, Partners, and Company. Below the navigation, a large image of a topographic map serves as the background for the form. A green button labeled "Free Download" is visible on the left. On the right, there's a "Sign Up" button and a "Sign In" button. The main form area has a white background and contains fields for "Email Address" and a checkbox for accepting communication terms. Below the form, there's a note about privacy and terms, a "Submit" button, and a "Skip registration" link.

# Anaconda Navigator

 ANACONDA Products Solutions Resources Partners Company Sign Up Sign In

## Download Now

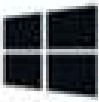
For installation assistance, refer to [Troubleshooting](#).

Download Anaconda Distribution or [Miniconda](#) by choosing the proper installer for your machine. Learn the difference from our [Documentation](#).



### Anaconda Installers

[Download](#)



#### Windows

**Python 3.12**

 64-Bit Graphical Installer (912.3M)



#### Mac

**Python 3.12**

 64-Bit (Apple silicon) Graphical Installer (704.7M)

 64-Bit (Apple silicon) Command Line Installer (707.3M)

 64-Bit (Intel chip) Graphical Installer (734.7M)

 64-Bit (Intel chip) Command Line Installer (731.2M)



#### Linux

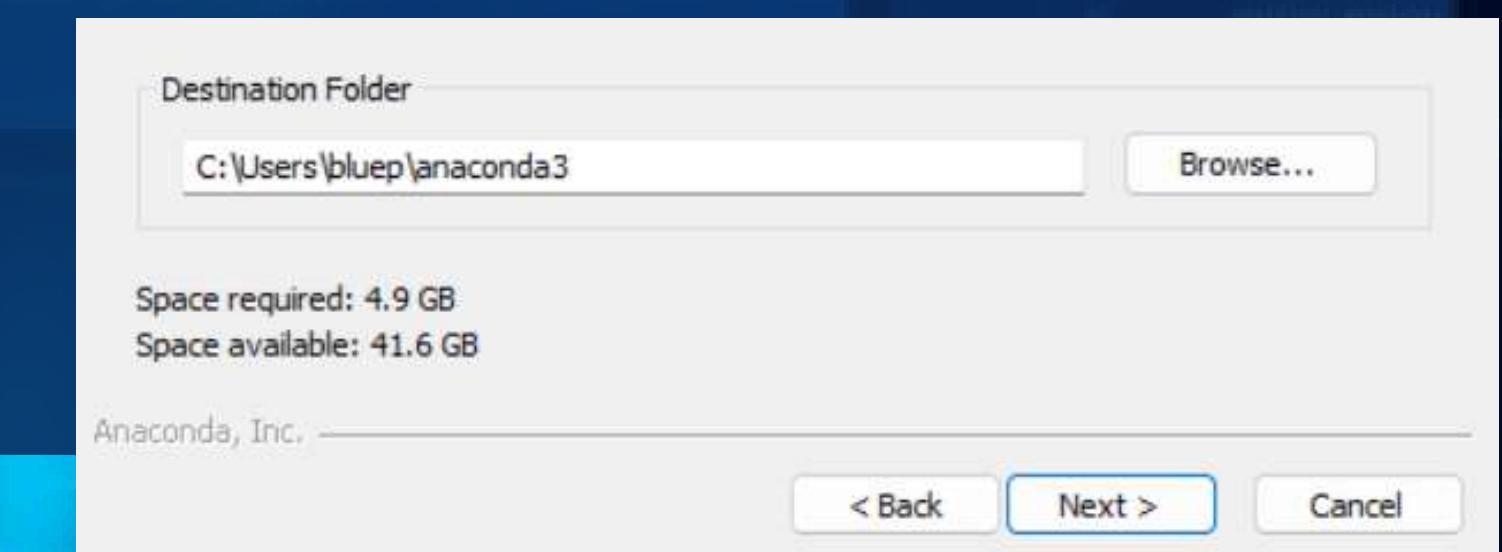
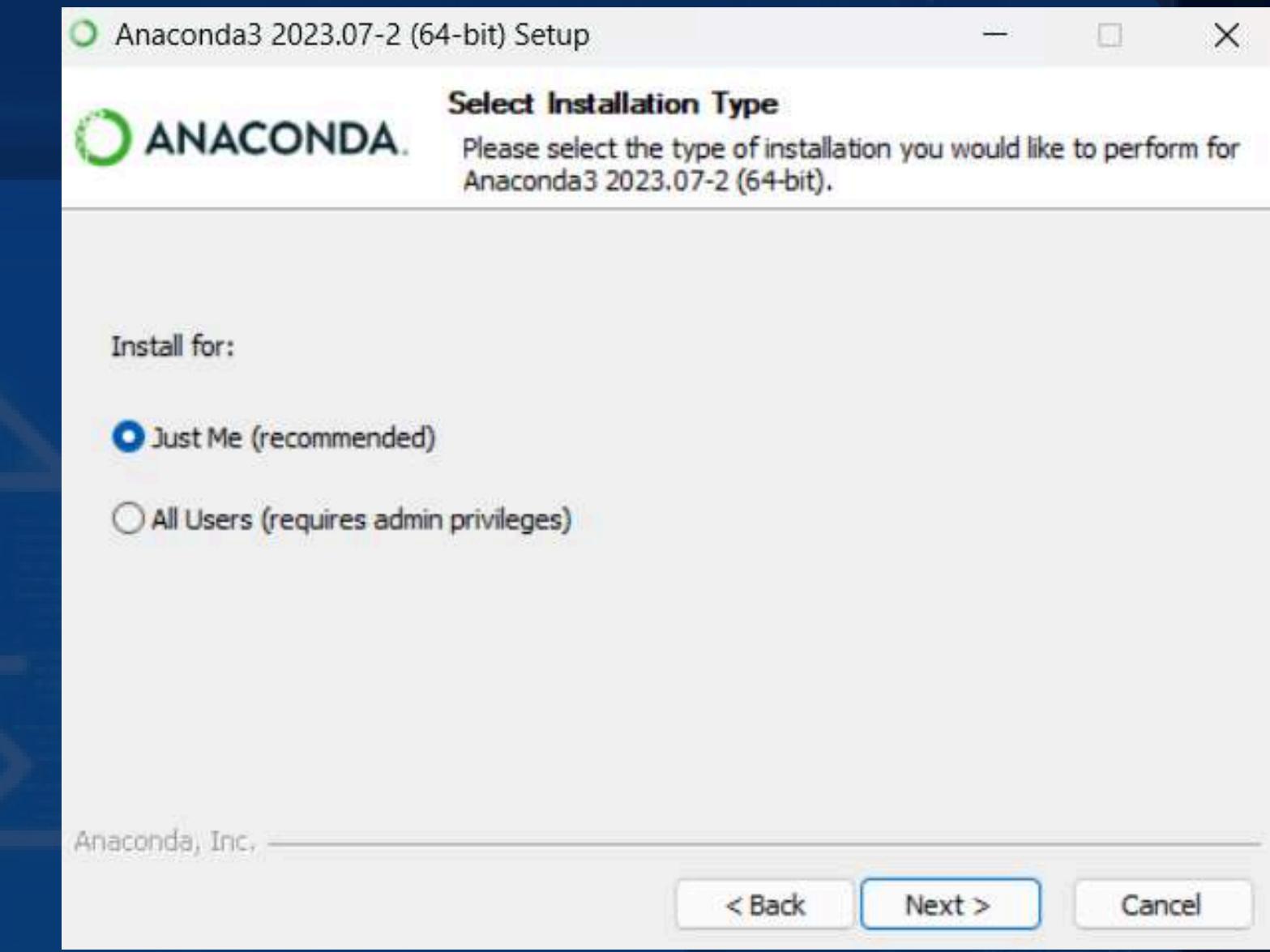
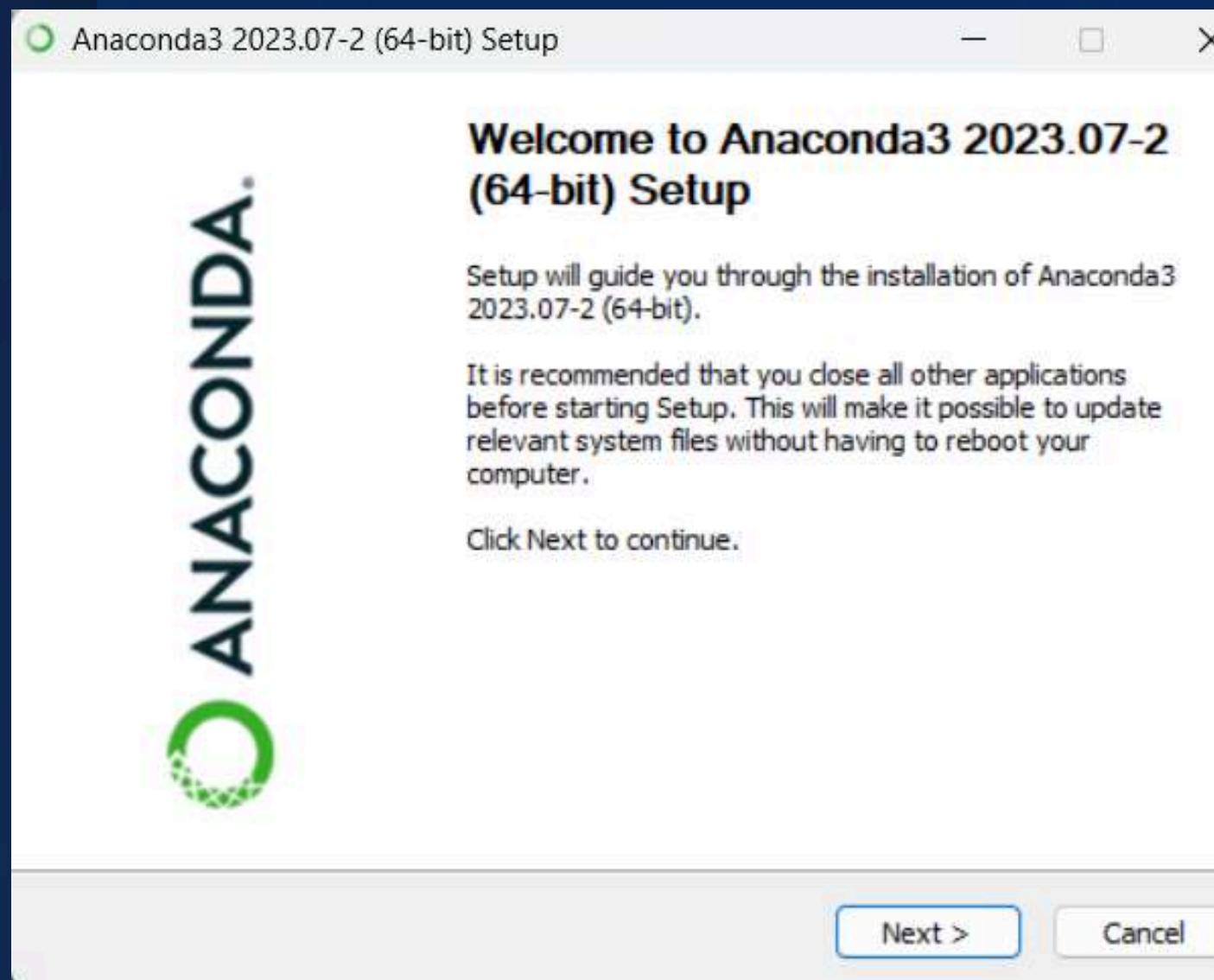
**Python 3.12**

 64-Bit (x86) Installer (1007.9M)

 64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)

 64-bit (Linux on IBM Z & LinuxONE) Installer (425.8M)

# Anaconda Navigator



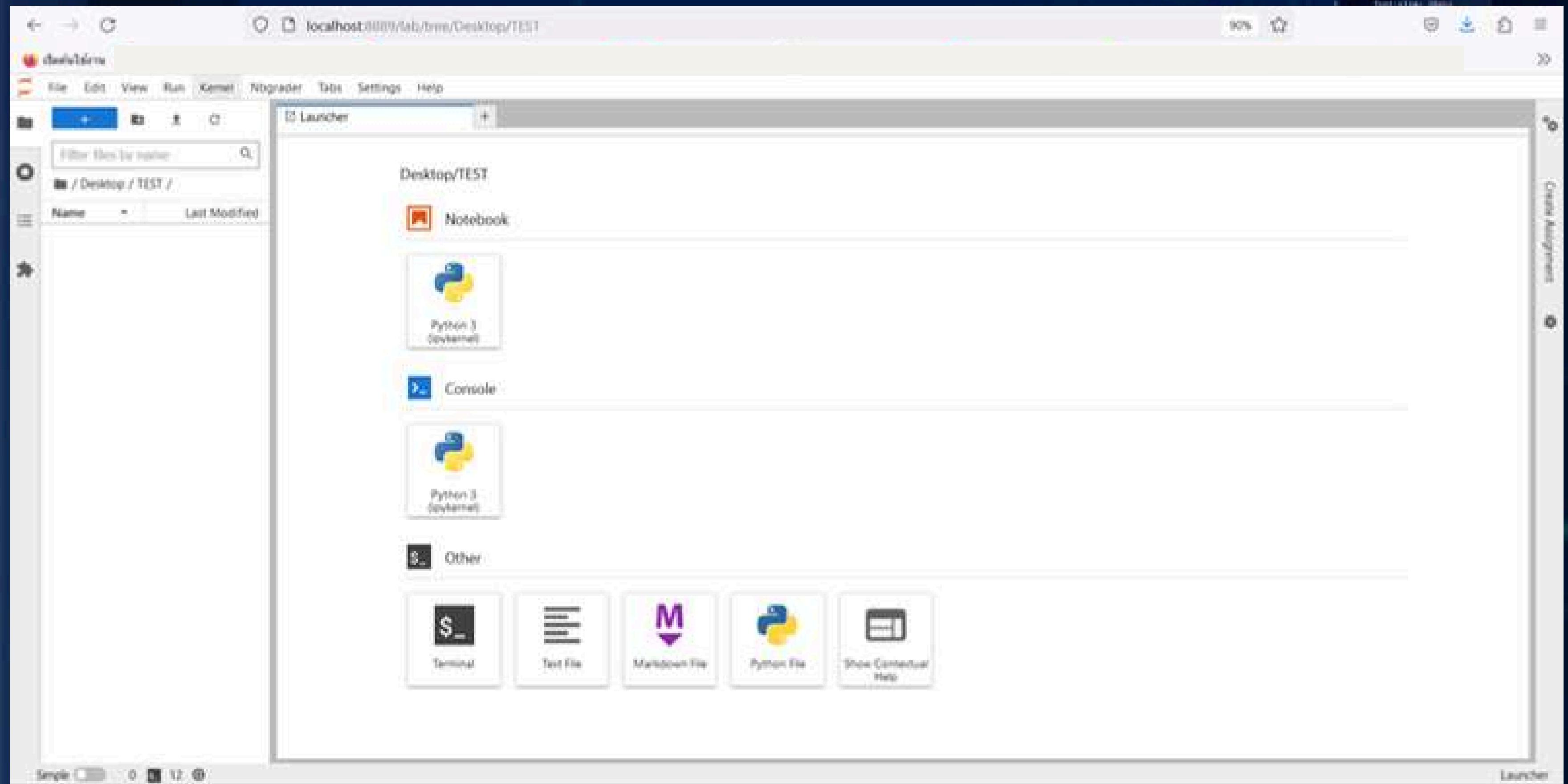
# Anaconda Navigator

The screenshot shows the Anaconda Navigator application window. On the left is a sidebar with navigation links: Home, Environments, Learning, and Community. A prominent feature is the "Anaconda Notebooks" section, which includes a "Create New" button and a "Launch" button. Below this are links for "A Full Python IDE Directly From the Browser", "Documentation", and "Anaconda Blog". Social media icons for Twitter, LinkedIn, and YouTube are at the bottom.

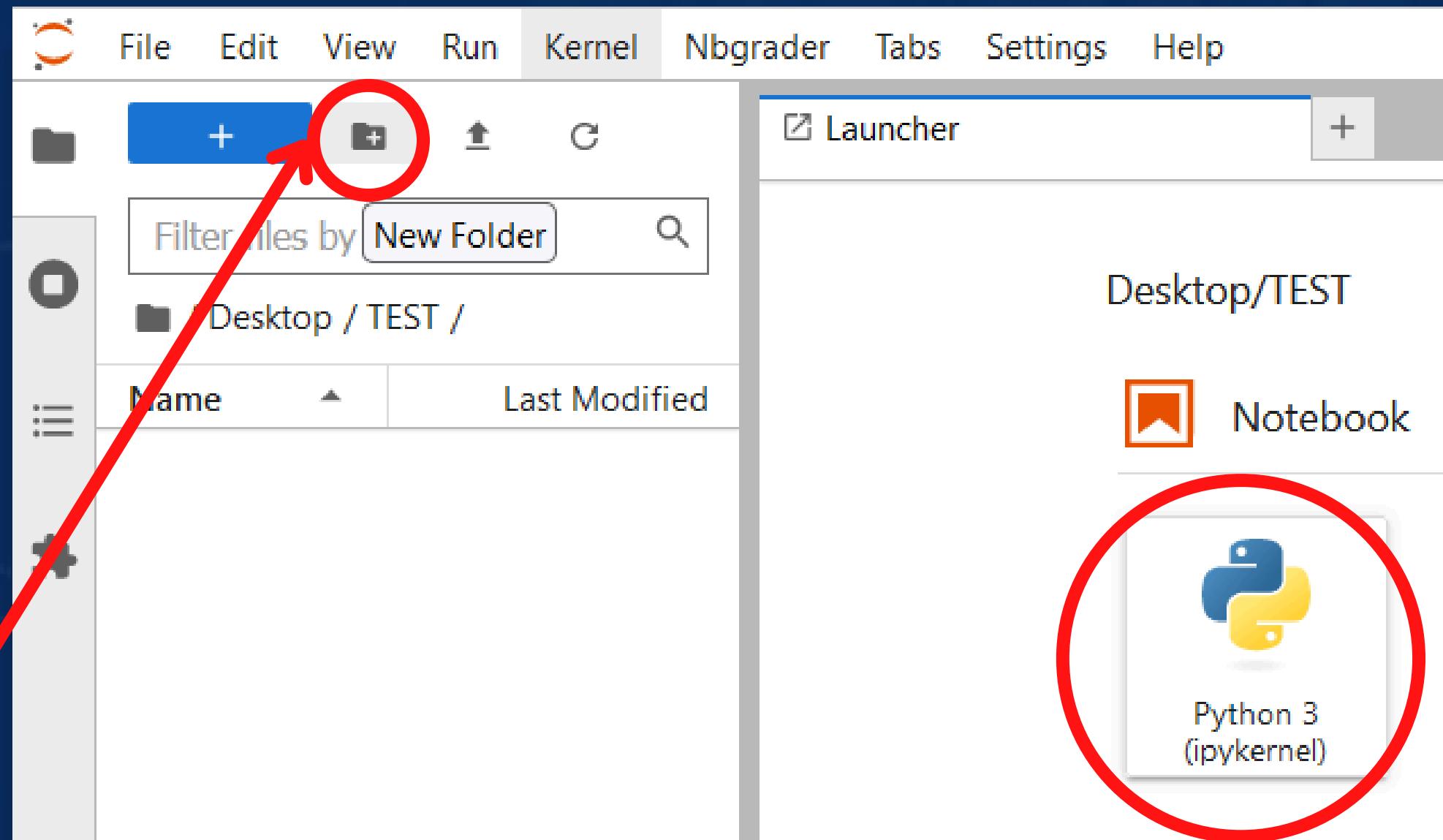
The main area displays a grid of 15 data science tools:

- DataSpell**: An IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in the user-friendly environment. [Install](#) [Learn](#)
- CMDShell Prompt**: Run a command terminal with your current environment from Navigator activated. [Learn](#)
- JupyterLab**: An extensible environment for interactive and reproducible computing, based on the Jupyter notebook and architecture. [Learn](#)
- jupyter**: Web-based, interactive computing notebook environment. Edit and run human-readable code while developing the data analysis. [Learn](#)
- iPowerShell Prompt**: Run a PowerShell terminal with your current environment from Navigator activated. [Learn](#)
- IPy**: IPython cell supports inline figures, proper multicell editing with syntax highlighting, graphical outputs, and more. [Learn](#)
- Spyder**: Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and visualization features. [Learn](#)
- VS Code**: Streamlined code editor with support for development operations like debugging, test running and version control. [Learn](#)
- DataLine**: Kickstart your data science projects in seconds in a pre-configured environment. Every coding assistance for Python, SQL, and R in Jupyter notebooks and kernels. From no-code automation, use DataLine online for free. [Learn](#)
- IBM Watson Studio Cloud**: IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling. [Learn](#)
- ORACLE Cloud Infrastructure**: Oracle Data Science Service. Oracle Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools. [Learn](#)
- Qlucore**: Multidimensional data visualization across files. Explore relationships within and among related datasets. [Install](#)
- Orange 3**: Component-based data mining framework. Data visualization and data analysis for scientific and web-based Python development. [Learn](#)
- PyCharm Professional**: A full-fledged IDE for JetBrains' Python scientific and web Python development. [Learn](#)
- RStudio**: A set of integrated tools designed to help you be more productive with R. Includes R

# Anaconda Navigator



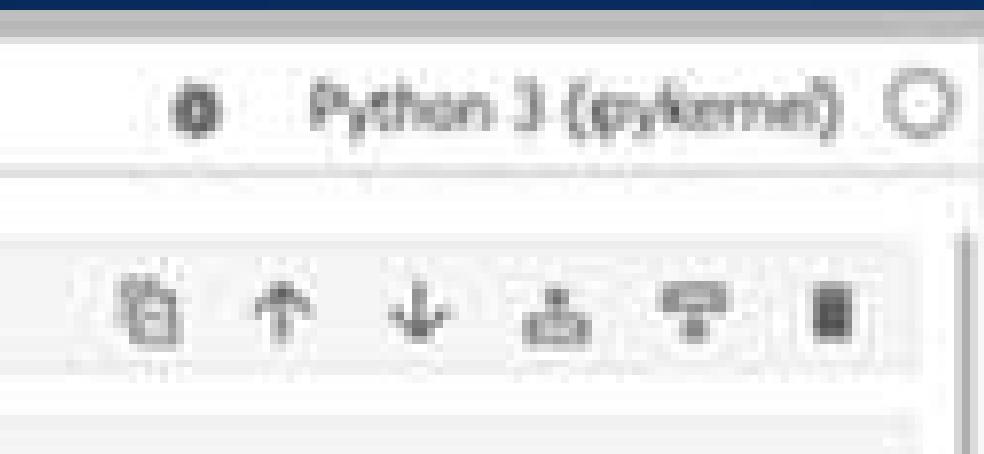
# Anaconda Navigator



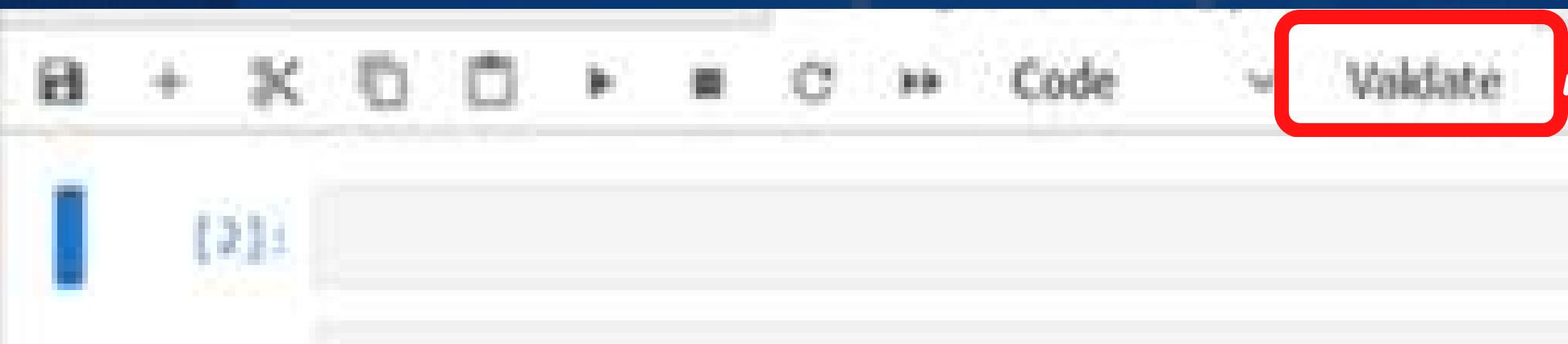
สร้าง Folder ใหม่

สร้าง สร้างไฟล์ .ipynb เพื่อเขียน  
โปรแกรมใน jupyter notebook

# Anaconda Navigator



python 3 (ipykernel) ในการรันโค้ด



ใช้สำหรับการทดสอบ save ไฟล์  
ตรวจสอบความถูกต้องทั้งหมด

# Jupyter Notebook

## image processing



# Jupyter Notebook

## image processing

```
[5]: print("Hello , Theeraset")  
Hello , Theeraset
```

### - ทดสอบ test jupyter Lab 1

```
[6]: a = 4  
[7]: b = 2
```

#### part 1

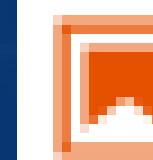
```
[8]: c = a // b  
c  
[9]: 2  
[10]: c = a * b  
c  
[11]: 8
```

#### part2

```
[12]: print(c)  
8  
[13]: b = 10  
[14]: b  
[15]: 10
```

# Jupyter Notebook

## image processing



ch2\_output\_python.ipynb

# Jupyter Notebook

## image processing

```
[1]: 2 ** 5
[1]: 32

comment

[2]: n = 20.5
[2]: n = 20.5

Data types

[3]: a = 3
[3]: type(a)
[3]: int

[4]: b = 3.0
[4]: type(b)
[4]: float

[5]: c = '4'
[5]: type(c)
[5]: str

[6]: d = eval(c)
[6]: type(d)
[6]: str

[7]: name = 'theeraset'
[7]: sur = 'chuipayung'
[7]: name + sur
[7]: "theeraset chuipayung"

[8]: print(name + sur)
```

# Jupyter Notebook

## image processing

```
print
```

```
[17]: v = 2  
      sc = 3.673245  
      name = "Blue"
```

```
[18]: print("Hi, %s a = %d b = %.2f" % (name, n, sc))  
Hi, Blue a = 20 b = 3.67
```

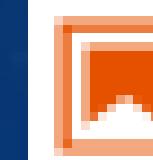
```
[19]: print('Hi {} {:.1f}'.format(name , sc))  
Hi Blue 3.7
```

```
[21]: b = 280903  
      print('data = {:.2f}'.format(b))  
data = 280,903.00
```

```
[ ]:
```

# Jupyter Notebook

## image processing



ch3\_basic\_python.ipynb

# Jupyter Notebook

## image processing

Python เป็นต้น

```
[1]: name = 'theeranet'
name = 'scimath'
print(name)

scimath

[2]: เลือกหน้าผานา (Operators)
+ * / 
% หารเศษ
// หารปั๊เศษ
** ยกกำเนิด

[3]: a = 10

[4]: b = a / 2
b

[5]: 5.0

[6]: 5 // 2

[7]: 2

[8]: 5 % 2

[9]: 1

[10]: 2 ** 3

[11]: 8
```

# Jupyter Notebook

## image processing

### Variable

```
[16]: raha = 100  
gudi = raha / 4  
nagari = gudi  
print(nagari)
```

25.0

```
[17]: 2r = 4  
pass = 9
```

```
[20]: r = 3  
pass_ = 50
```

### Condition

```
[1]: > < >= <= !=
```

```
[25]: a = 3  
b = 7  
c = 9
```

```
[26]: a > b
```

```
[26]: False
```

```
[27]: a < b
```

```
[27]: True
```

```
[28]: a < b and a > c
```

```
[28]: False
```

# Jupyter Notebook

## image processing

```
if
```

```
[31]: img = 32
if img < 30 :
    print('Image off')
elif img >= 30 and img < 40:
    print('Image Process')
else:
    print('Turn on a Visualize Box')
```

```
Image Process
```

### Iteration (Loop):

```
[35]: for count in range(4,6):
    print("number = ",end="")
    print(count)
print()
print("done")

#print
print('data = ',a)

number = 4
number = 5

done
data = 3
```

# Jupyter Notebook

## image processing

while

```
[1]: interest = 0
while interest < 4:
    print("hello naa")
    interest = interest + 1
print("end of sol")
```

hello naa  
hello naa  
hello naa  
hello naa  
end of sol

Function

```
[41]: def area(length):
    area = length * length
    return area

a = area(4)
print('Area = %d' % (a))
print('Area = ', a)
```

Area = 16  
Area = 16

```
[42]: area(5)
```

```
[42]: 25
```

```
[43]: area(6)
```

```
[43]: 36
```

# Jupyter Notebook

## image processing

### Container

#### List

```
[44]: personality = [20, 3, 19.5, 230, 145]
```

```
[45]: type(personality)
```

```
[45]: list
```

```
[47]: personality[0]
```

```
[47]: 20
```

```
[49]: personality[-4:-1]
```

```
[49]: [3, 19.5, 230]
```

```
[51]: for j in personality:  
      print(j)
```

```
20  
3  
19.5  
230  
145
```

```
[52]: # enumerate  
for i,n in enumerate(personality):  
    print(i,n)
```

```
0 20  
1 3  
2 19.5  
3 230  
4 145
```

# Jupyter Notebook

## image processing

```
[53]: data = [[20, 62, 22.4, 189, 123],  
            [32, 47, 11.9, 168, 174],  
            [24, 35, 51.43, 155, 144]]
```

```
[54]: data[0][1]
```

```
[54]: 62
```

```
[55]: data[2][3]
```

```
[55]: 155
```

### Dictionary

```
[60]: book_lish = {'age':27,'ht':171,'bmi':18.4}  
      type(book_lish)
```

```
[60]: dict
```

```
[62]: a = book_lish['ht']  
a
```

```
[62]: 171
```

```
[63]: y = book_lish['bmi']  
y
```

```
[63]: 18.4
```

```
[64]: for i in book_lish.values():  
      print(i)
```

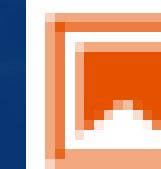
```
27
```

```
171
```

```
18.4
```

# Jupyter Notebook

## image processing



ch4\_library.ipynb



# Jupyter Notebook

## image processing

Library

การจัดการ Library

- Anaconda Navigator -pip (Package Installer for Python): <https://pip.pypa.io/en/stable/>
- conda: <https://docs.conda.io/projects/conda/en/latest/commands.html>

ตรวจสอบว่าติดตั้งแล้วหรือยัง

```
[2]: !pip show pandas
```

```
Name: pandas
Version: 1.5.3
Summary: Powerful data structures for data analysis, time series, and statistics
Home-page: https://pandas.pydata.org
Author: The Pandas Development Team
Author-email: pandas-dev@python.org
License: BSD-3-Clause
Location: c:\users\bluep\anaconda3\lib\site-packages
Requires: numpy, python-dateutil, pytz
Required-by: datashader, holoviews, hvplot, seaborn, statsmodels, xarray
```

```
[3]: !conda list pandas
```

```
# packages in environment at C:\Users\bluep\anaconda3:
#
# Name           Version        Build  Channel
pandas          1.5.3           py310h4ed8f06_0
```

```
[4]: !pip show matplotlib
```

```
Name: matplotlib
Version: 3.7.0
Summary: Python plotting package
Home-page: https://matplotlib.org
```

# Jupyter Notebook

## image processing

```
[5]: !pip show mlxtend  
WARNING: Package(s) not found: mlxtend
```

### Pandas

read CSV

```
[3]: import pandas as pd  
[4]: ds = pd.read_csv('data/advertise.csv')  
[5]: ds
```

```
[5]:   Unnamed: 0    TV  Radio  Newspaper  Sales  
0          1  230.1    37.8      69.2  22.1  
1          2   44.5    39.3      45.1  10.4  
2          3   17.2    45.9      69.3   9.3  
3          4   151.5    41.3      58.5  18.5  
4          5   180.8    10.8      58.4  12.9  
..        ..    ..    ..    ..    ..  
195       196   38.2     3.7     13.8   7.6  
196       197   94.2     4.9      8.1  9.7  
197       198  177.0     9.3      6.4  12.8  
198       199  283.6    42.0     66.2  25.5  
199       200  232.1     8.6      8.7  13.4
```

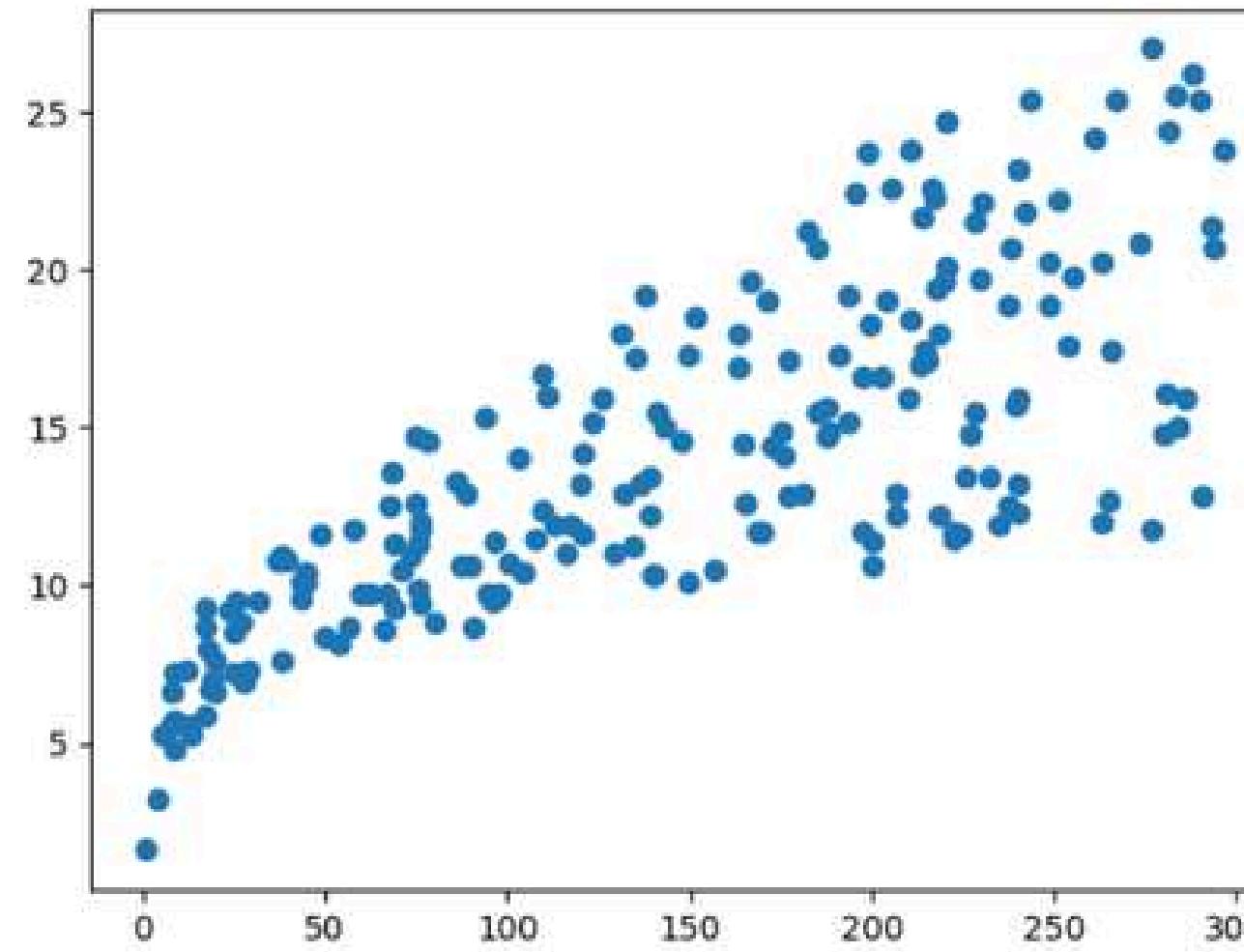
# Jupyter Notebook

## image processing

plot

```
[6]: import matplotlib.pyplot as plt
```

```
[7]: plt.scatter(ds.TV, ds.Sales)  
plt.show()
```



```
[8]: pip --version
```

```
pip 22.3.1 from C:\Users\bluep\anaconda3\lib\site-packages\pip (python 3.10)
```

Note: you may need to restart the kernel to use updated packages.

# Jupyter Notebook

## image processing

### การจัดการ Library

```
[9]: import pydotplus
```

pip (Package Installer for Python)

<https://pip.pypa.io/en/stable/>

```
[10]: pip install numpy
```

Requirement already satisfied: numpy in c:\users\bluep\anaconda3\lib\site-packages (1.23.5)  
Note: you may need to restart the kernel to use updated packages.

```
[11]: !pip show mtcnn
```

WARNING: Package(s) not found: mtcnn

```
[12]: !pip uninstall mtcnn
```

WARNING: Skipping mtcnn as it is not installed.

### conda

```
[13]: conda list pandas
```

# packages in environment at C:\Users\bluep\anaconda3:  
#  
# Name Version Build Channel  
pandas 1.5.3 py310h4ed8f06\_0

Note: you may need to restart the kernel to use updated packages.

```
[14]: !conda install numpy
```

```
[15]: # pip install mtcnn
```

# Jupyter Notebook

## image processing

```
[1]: # pip install mtcnn  
!pip uninstall mtcnn  
  
[2]: !conda info  
  
# https://conda-forge.org/docs/user/introduction.html  
  
# https://conda.anaconda.org/conda-forge/
```

### \* Numpy 1

```
[3]: import numpy as np  
data = [4, 7, 1, 2, 9, 3, 8, 16]  
np.mean(data)  
  
[4]: np.mean([4, 7, 1, 2, 9, 3, 8, 16])  
  
[5]: np.std(data)  
  
[5]: 4.575751304430781  
  
[6]: np.min(data)  
  
[6]: 1  
  
[7]: np.max(data)  
  
[7]: 16  
  
[8]: np_a = np.array([data])  
type(np_a)  
  
[8]: numpy.ndarray
```

# Jupyter Notebook

## image processing

```
[21]: U = [[6, 7, 4, 5, 1],  
          [2, 8, 3, 6, 4],  
          [1, 3, 2, 9, 6],  
          [8, 9, 1, 7, 2]]  
  
[22]: npu = np.array(U)  
npu  
  
[23]: array([[6, 7, 4, 5, 1],  
           [2, 8, 3, 6, 4],  
           [1, 3, 2, 9, 6],  
           [8, 9, 1, 7, 2]])  
  
[24]: npu.shape  
  
[25]: (4, 5)  
  
[26]: npu[0,1]  
  
[27]: 7  
  
[28]: npu[2,4]  
  
[29]: 6  
  
[30]: npu[:, :]  
  
[31]: array([[6, 7, 4, 5, 1],  
           [2, 8, 3, 6, 4],  
           [1, 3, 2, 9, 6],  
           [8, 9, 1, 7, 2]])  
  
[32]: npu.T  
  
[33]: array([[6, 2, 1, 8],  
           [7, 8, 3, 9],  
           [4, 3, 2, 1],  
           [1, 5, 6, 0]])
```

# Jupyter Notebook

## image processing

```
[35]: array([[6, 2, 1, 8],  
           [7, 8, 3, 9],  
           [4, 3, 2, 1],  
           [5, 6, 9, 7],  
           [1, 4, 6, 2]])  
  
[36]: npu.reshape(1,-1)  
  
[36]: array([6, 2, 1, 8, 7, 3, 9, 4, 3, 2, 1, 5, 6, 9, 7, 1, 4, 6, 2])  
  
[37]: npu.reshape(-1,1)  
  
[37]: array([[6],  
           [7],  
           [4],  
           [5],  
           [1],  
           [2],  
           [8],  
           [3],  
           [6],  
           [4],  
           [1],  
           [3],  
           [2],  
           [9],  
           [6],  
           [8],  
           [9],  
           [1],  
           [7],  
           [2]])
```

# Jupyter Notebook

## image processing

Random and unique

```
[36]: np.random.rand(4)

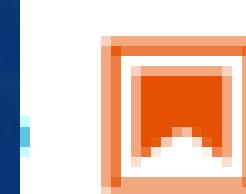
[38]: array([0.85592474, 0.00815198, 0.04118109, 0.78303578])

[40]: mu, sigma = 3.2, 0.5
      s = np.random.normal(mu, sigma, 1000)
```

```
[ ]:
```

# Jupyter Notebook

## image processing



ch5\_matplotlib.ipynb

# Jupyter Notebook

## image processing

Matplotlib

ใช้ภาษา Python

[2]: `!conda list matplotlib`

```
# packages in environment at C:\Users\bluep\anaconda3:
#
# Name           Version      Build Channel
matplotlib      3.7.0        py310haa95532_0
matplotlib-base 3.7.0        py310h4ed8f06_0
matplotlib-inline 0.1.6       py310haa95532_0
```

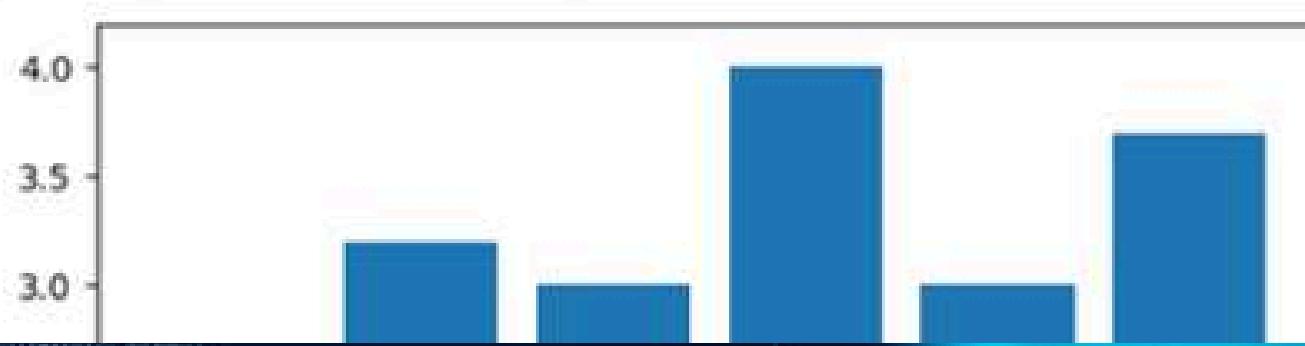
[5]: `import matplotlib.pyplot as plt`

```
from matplotlib import pyplot as plt # เรียกชื่อแบบนี้ได้
```

Bar

[6]: `x = [1,2,3,4,5,6]`  
`y_data = [1,3.2,3.0,4,3,3.7]`

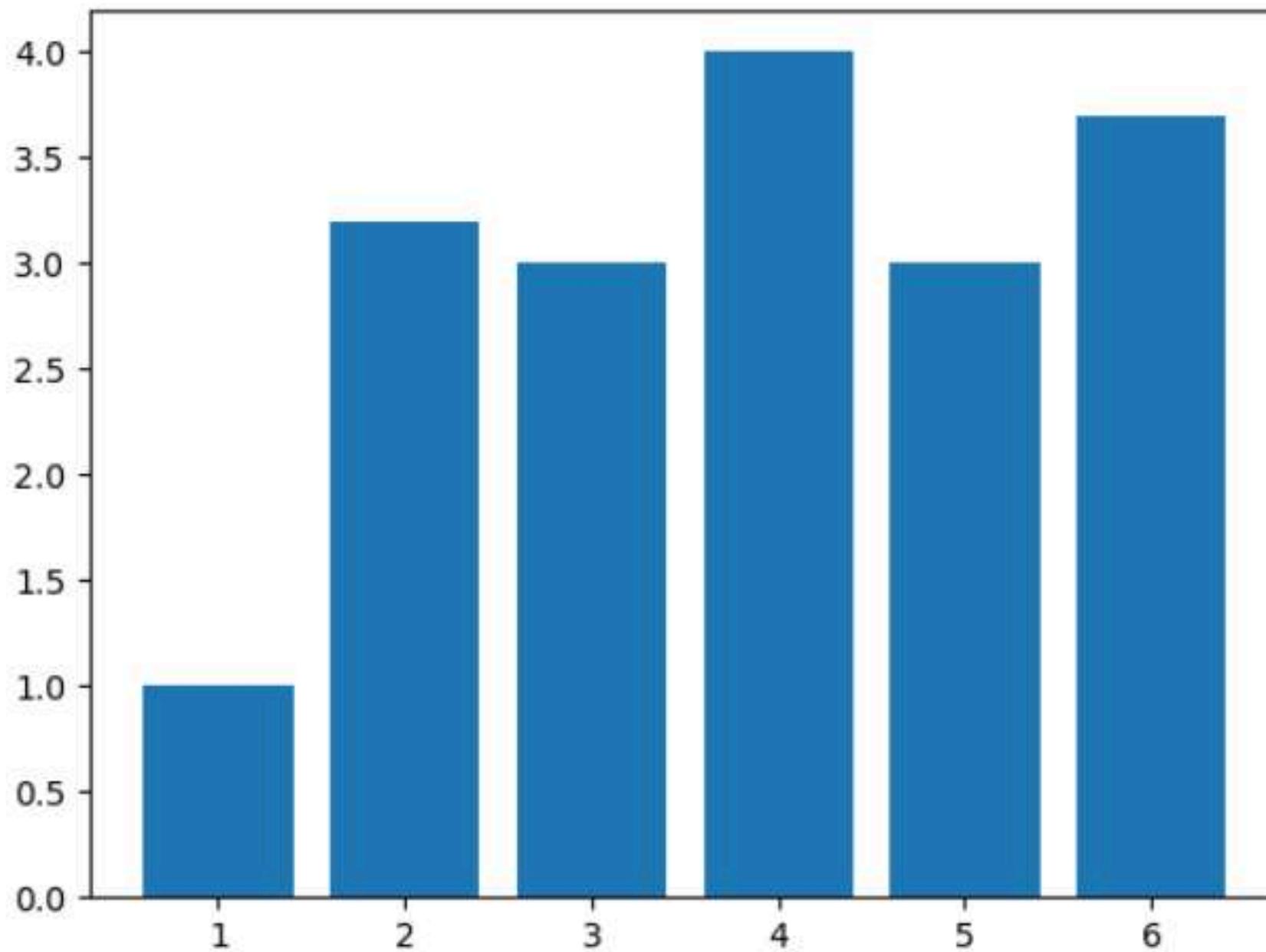
[7]: `plt.bar(x, y_data)`  
`plt.show() # แสดงผลกราฟ x บน y ล่างๆ`



# Jupyter Notebook

## image processing

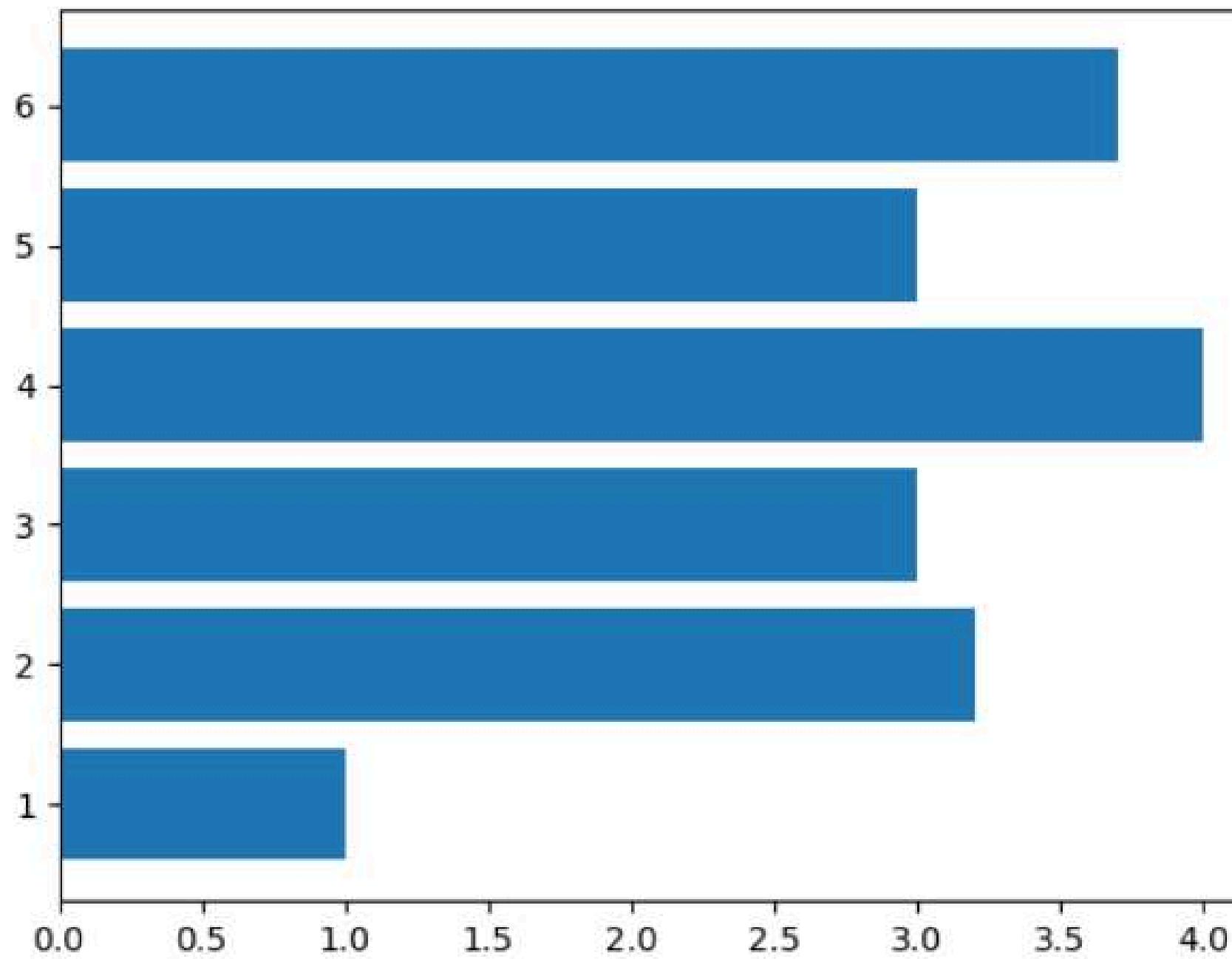
```
[7]: plt.bar(x, y_data)
plt.show() # พล็อตกราฟแกน x แกน y และดึง
```



# Jupyter Notebook

## image processing

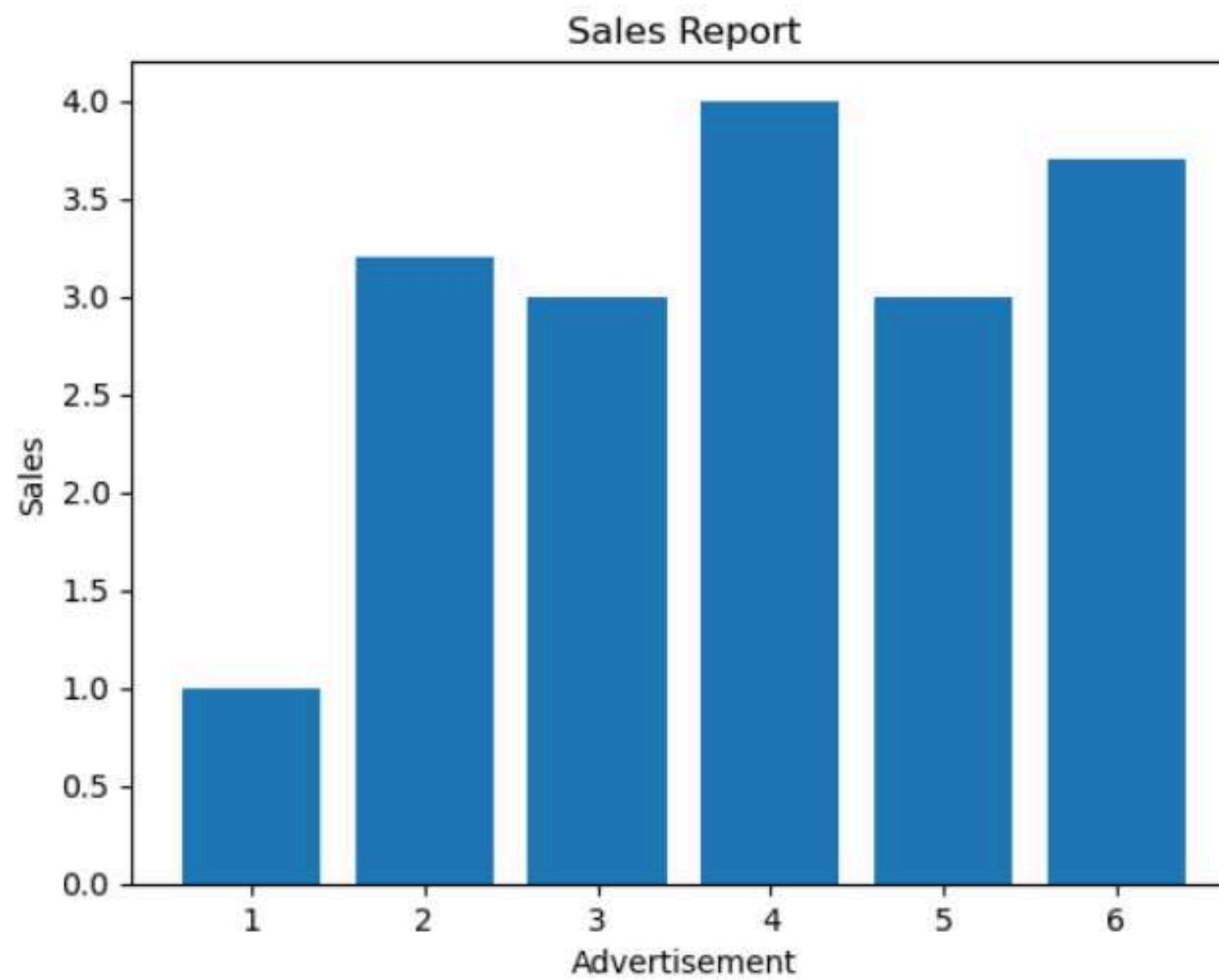
```
[8]: plt.barh(x, y_data) # พล็อตกราฟแนวนอน  
plt.show()
```



# Jupyter Notebook

## image processing

```
[11]: plt.title('Sales Report')
plt.bar(x, y_data)
plt.xlabel('Advertisement')
plt.ylabel('Sales')
plt.show()
```

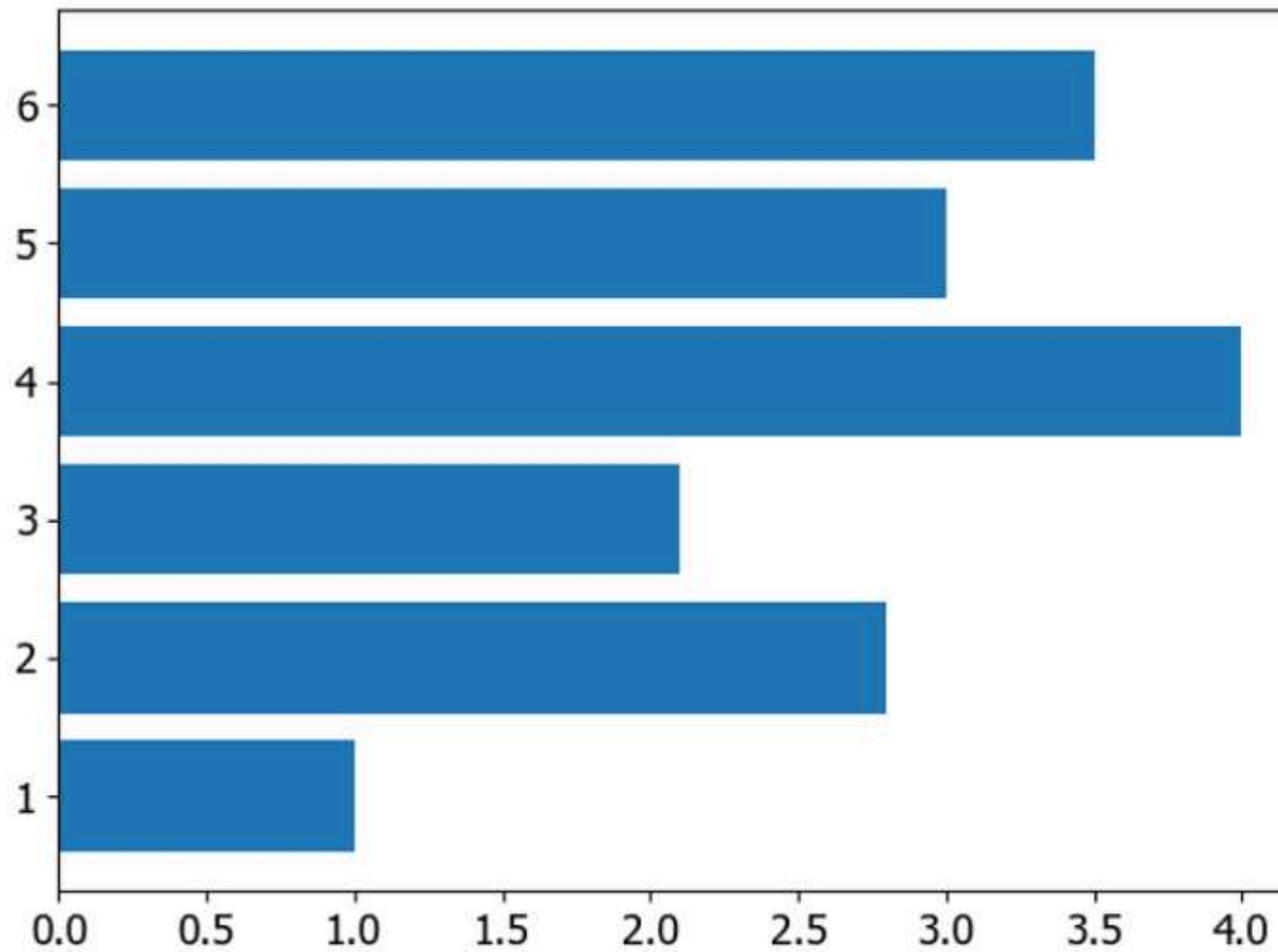


# Jupyter Notebook

## image processing

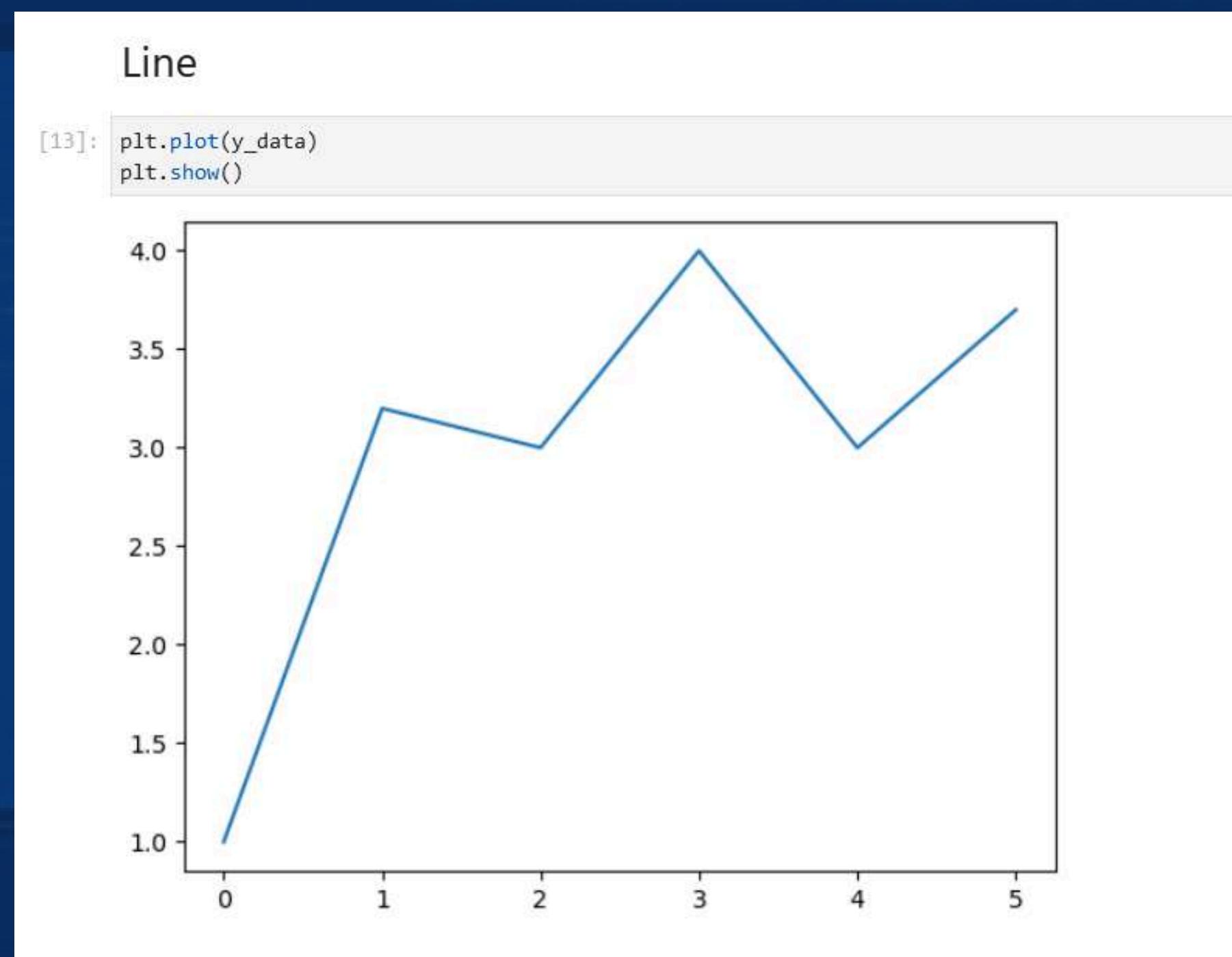
save

```
[19]: plt.barh(x, ydata) # พล็อตกราฟแท่ง  
plt.tight_layout()  
plt.savefig('plot1.png', dpi=100)  
plt.savefig('plot1.pdf', dpi=100)  
plt.show()
```



# Jupyter Notebook

## image processing

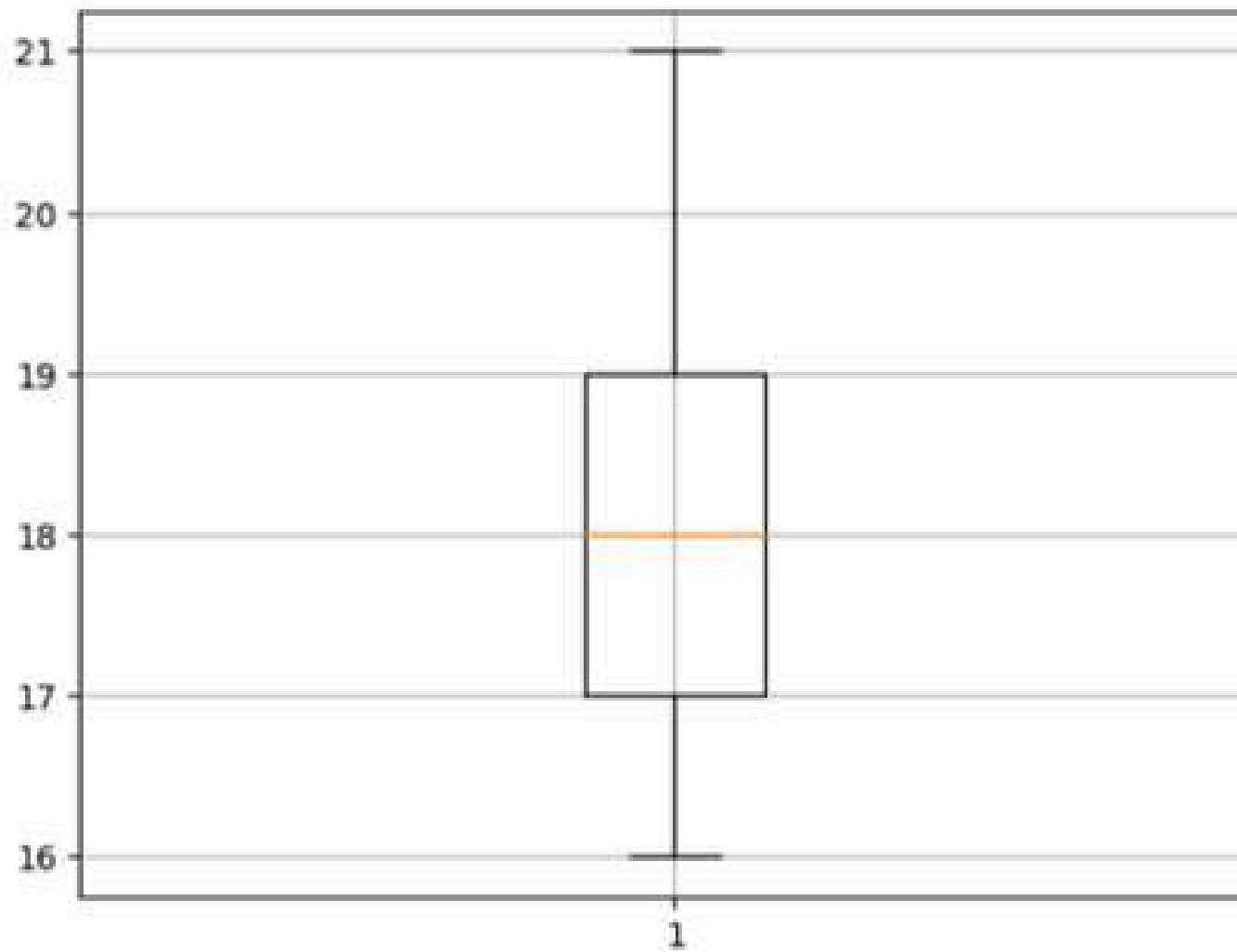


# Jupyter Notebook

## image processing

Box plot

```
[15]: age = [16, 17, 17, 19, 20, 20, 19, 19, 18, 18, 17, 16, 19, 19, 19, 18, 17, 21]
plt.grid()
plt.boxplot(age)
plt.show()
```



```
[16]: import pandas as pd
ds = pd.Series(age)
ds.describe().round(2)
```

# Jupyter Notebook

## image processing

(Annotation) การเขียนป้ายรายละเอียด

```
[17]: # เขียนรายละเอียดที่ต้องการ (annotate)

xs = [1, 2, 3, 4, 5, 6] # x

ydata = [1,2.8,2.1,4,3,3.5] # y1
ydata2 = [2.2, 1.6, 2.3, 2, 2.3, 2.1] # y2

plt.rcParams.update({'font.size':14, 'font.family':'tahoma'})
plt.figure(figsize=(6, 4)) # ขนาดกราฟ

plt.grid(linestyle='--', axis='y')
plt.bar(x, ydata, color='c', width=0.5, label='ລາຄາຕົວເທິງ')
plt.bar([x + 0.2 for x in xs], # ນັ້ນມີການກົດ
        ydata2, width=0.5, color='orange', alpha=0.8, label='ລາຄາສຳເນົາ')

plt.plot(x, ydata, color='b', marker='s', markersize=8) #plot ນັ້ນມີການ
plt.plot(x, ydata2, color='r', marker='o', markersize=8)

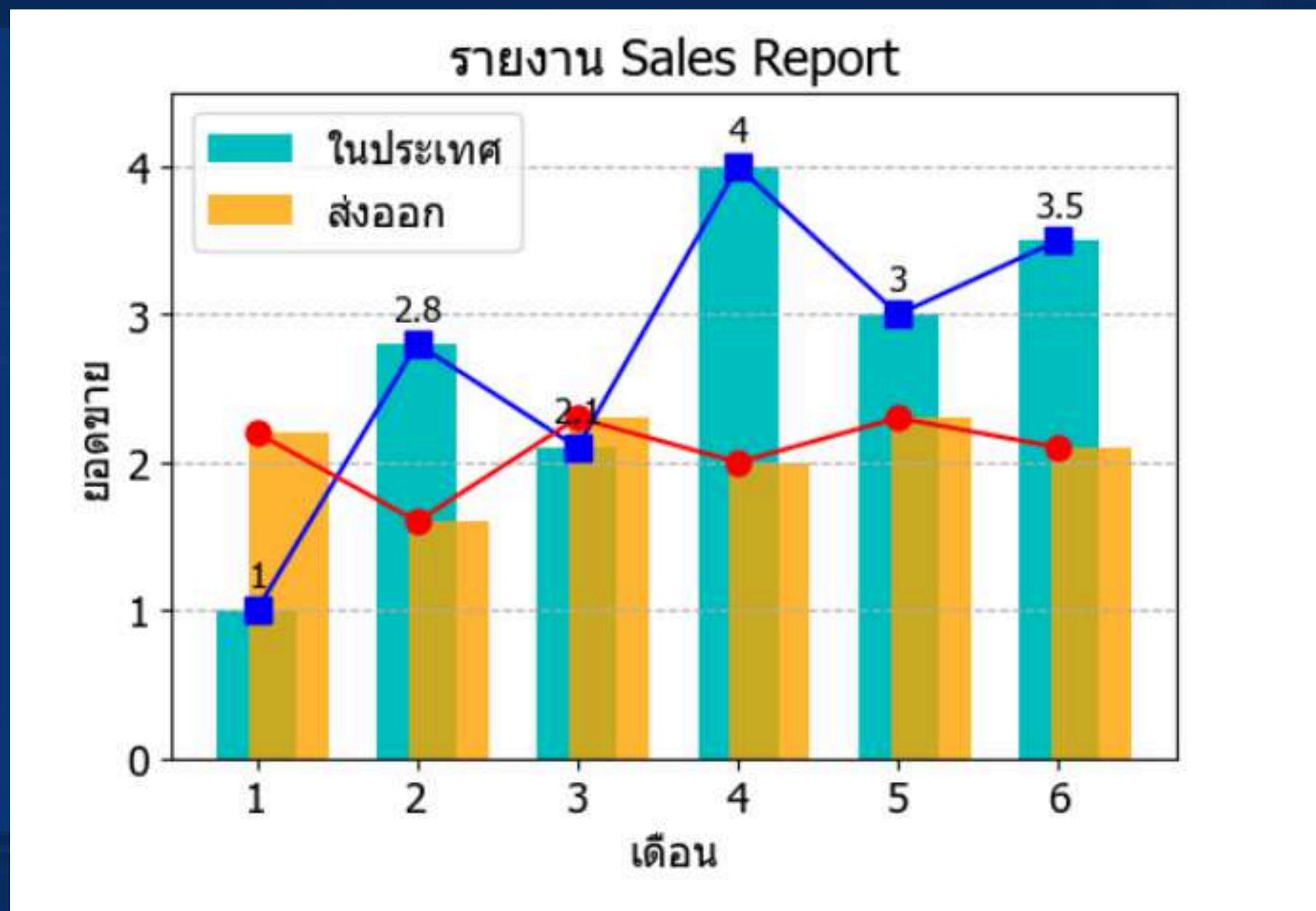
plt.xlabel("ເພື່ອນ")
plt.ylabel("ຍອດຂາຍ")
plt.title("ຮາມວານ Sales Report")

for tx,ty in list(zip(x, ydata)):
    plt.annotate(ty, # ນັ້ນມີການ label
                 (tx, ty), # point ສໍາເລັດ label
                 textcoords="offset points", # position ນັ້ນມີການ
                 xytext=(0, 8), # ສະບັບຈາກ points (x,y)
                 ha='center',
                 fontsize=12) # font

plt.legend()
plt.ylim(0, 4.5) #ນັ້ນກັດ ນັ້ນນັ້ນ y
plt.show()
```

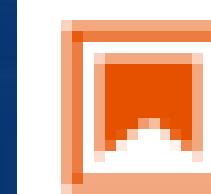
# Jupyter Notebook

## image processing



# Jupyter Notebook

## image processing



ch6 seaborn.ipynb



# Jupyter Notebook

## image processing

### Seaborn

```
[1]: !conda list seaborn

[11]: import seaborn as sns
import matplotlib.pyplot as plt

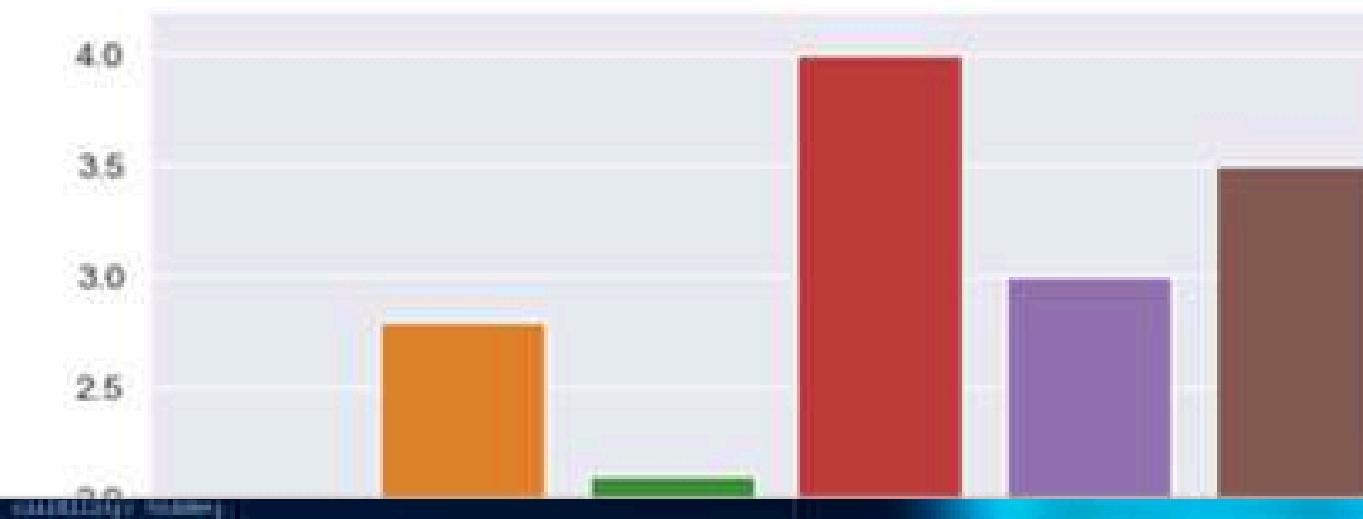
[40]: xdata = [1,2,3,4,5,6]          # សម្រាប់អាមិពណ៌នី x (ដែលមានប័ណ្ណក្នុងការបង្កើតរូប)
      ydata = [1,2.8,2.1,4,3,3.5]    # សម្រាប់អាមិពណ៌នី y (x និង y មានជូនបាននៅក្នុងការបង្កើតរូប)

sns.barplot(xdata, ydata)

-----
TypeError                                 Traceback (most recent call last)
Cell In[40], line 4
      1 xdata = [1,2,3,4,5,6]          # សម្រាប់អាមិពណ៌នី x (ដែលមានប័ណ្ណក្នុងការបង្កើតរូប)
      2 ydata = [1,2.8,2.1,4,3,3.5]    # សម្រាប់អាមិពណ៌នី y (x និង y មានជូនបាននៅក្នុងការបង្កើតរូប)
----> 4 sns.barplot(xdata, ydata)

TypeError: barplot() takes from 0 to 1 positional arguments but 2 were given

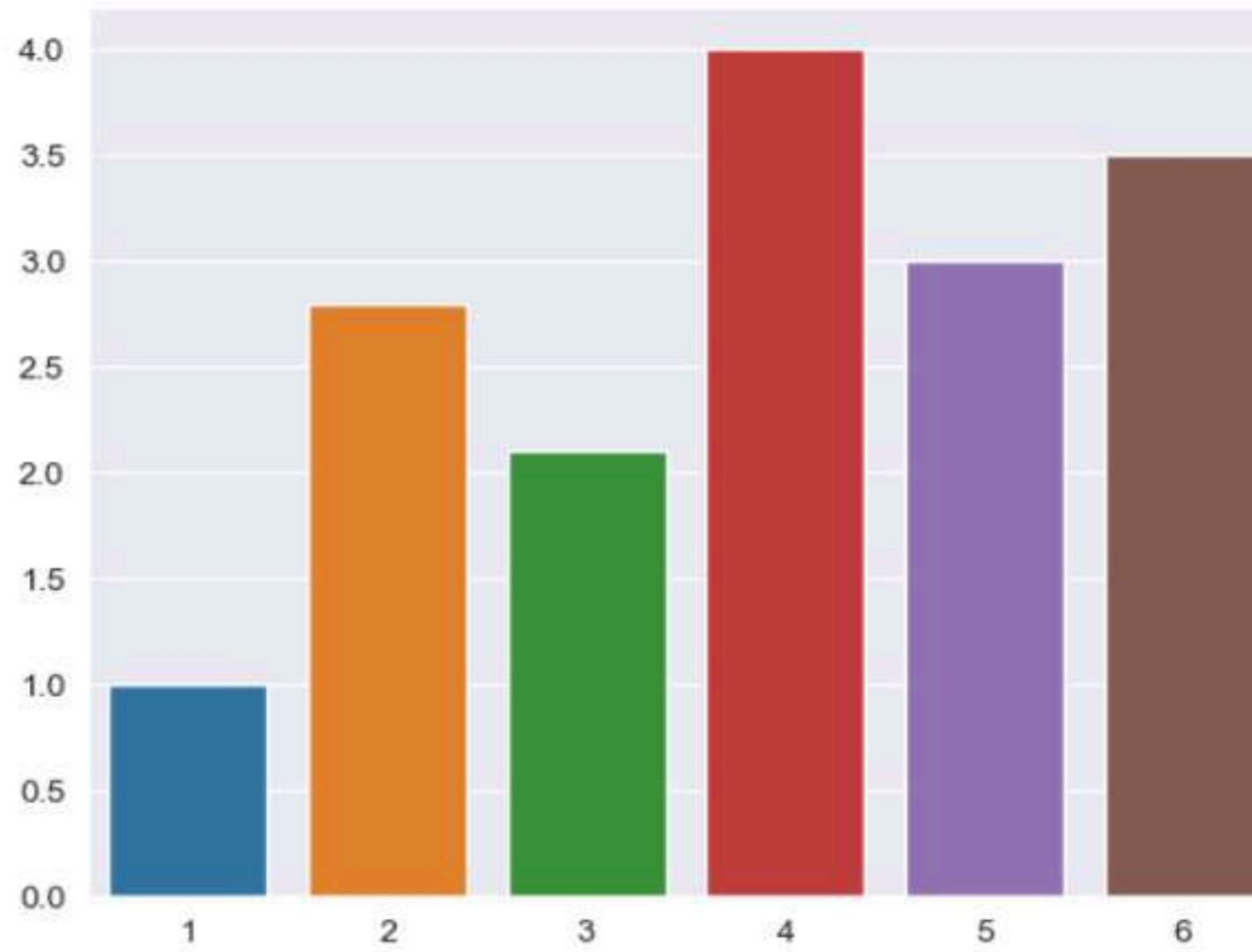
[22]: sns.barplot(x=xdata, y=ydata)
plt.show()
```



# Jupyter Notebook

## image processing

```
[22]: sns.barplot(x=xdata, y=ydata)  
plt.show()
```



# Jupyter Notebook

## image processing

```
[23]: sns.lineplot(x=xdata, y=ydata)
plt.savefig('snsplot.png', dpi=100)
# plt.show()
```

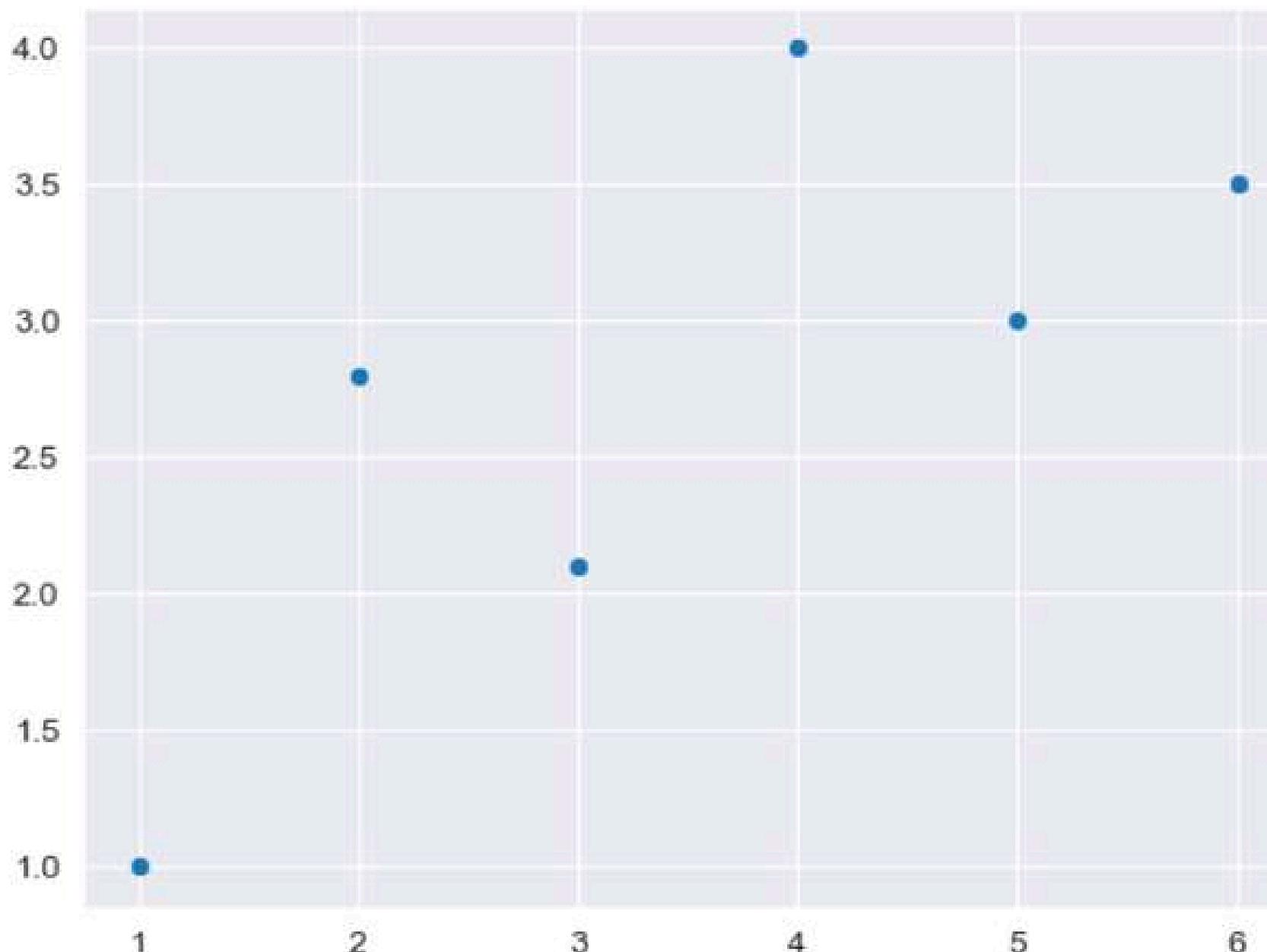


# Jupyter Notebook

## image processing

```
[24]: sns.scatterplot(x=xdata, y=ydata)
```

```
[24]: <Axes: >
```



# Jupyter Notebook

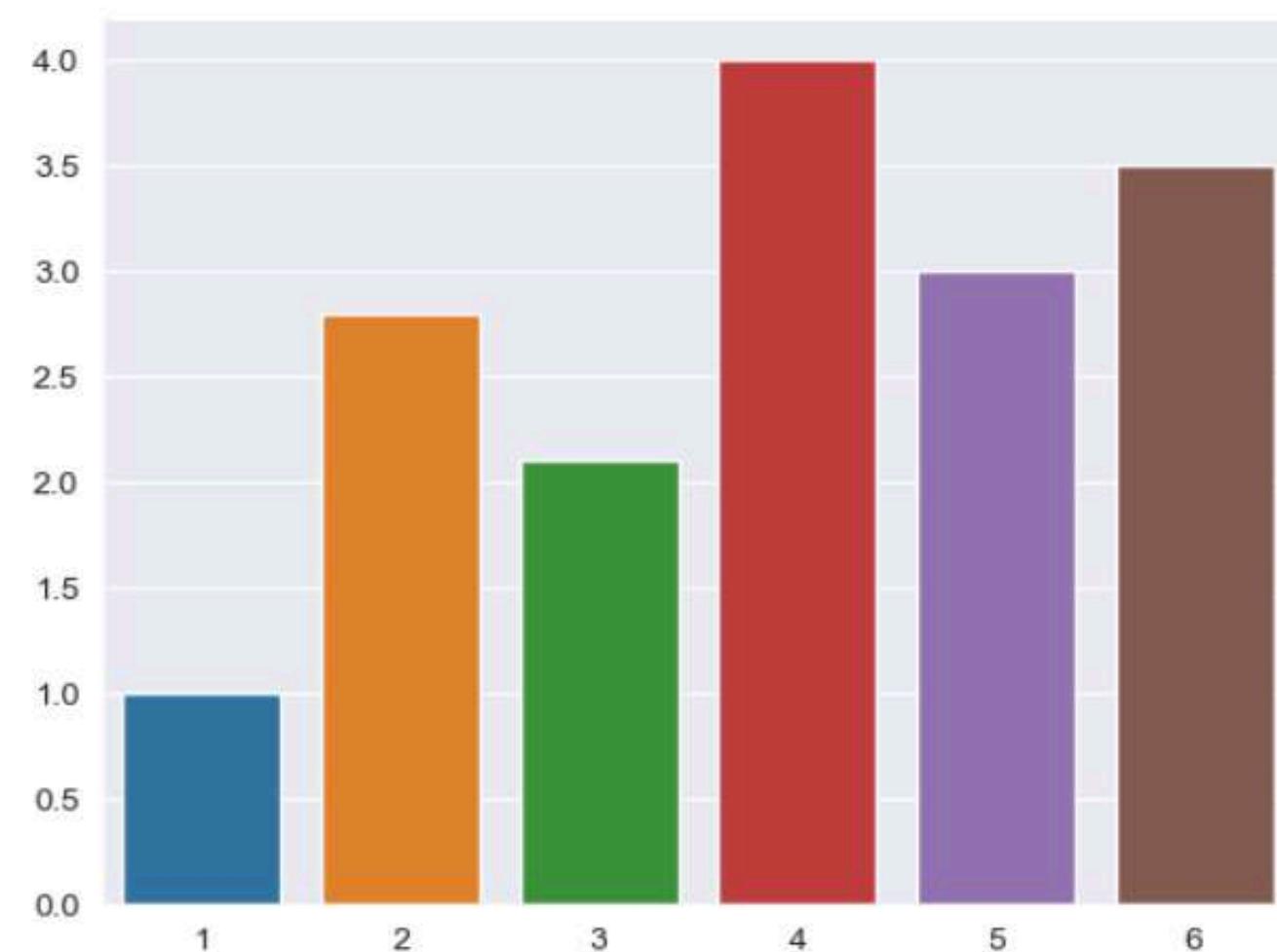
## image processing

### Style

```
[25]: sns.set_style('darkgrid')
# sns.set_style('dark')
# sns.set_style('whitegrid')
# sns.set_style('white')
# sns.set_style('ticks')

sns.barplot(x=xdata, y=ydata)
```

```
[25]: <Axes: >
```

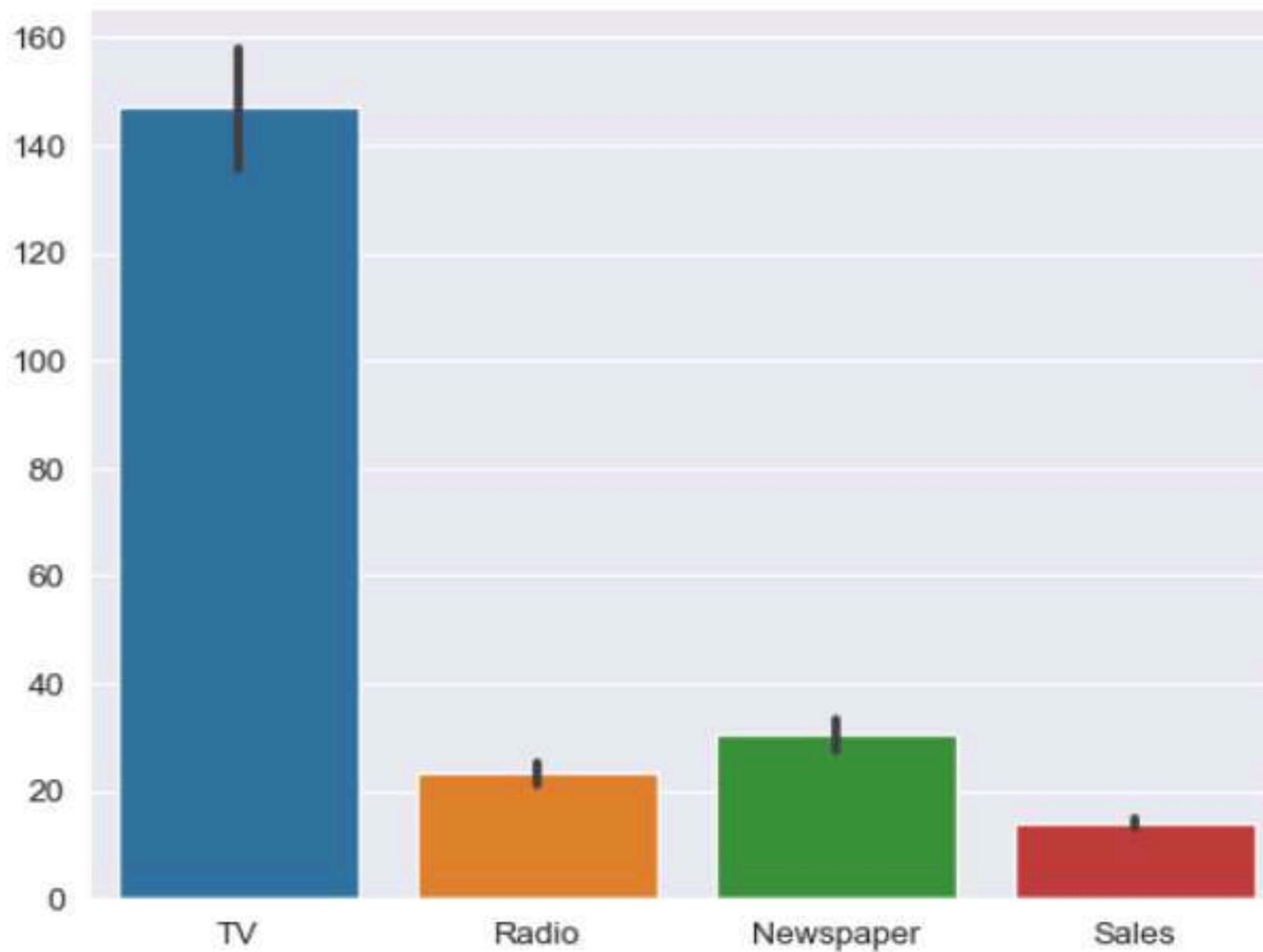


# Jupyter Notebook

## image processing

```
[30]: sns.barplot(data=df)
```

```
[30]: <Axes: >
```



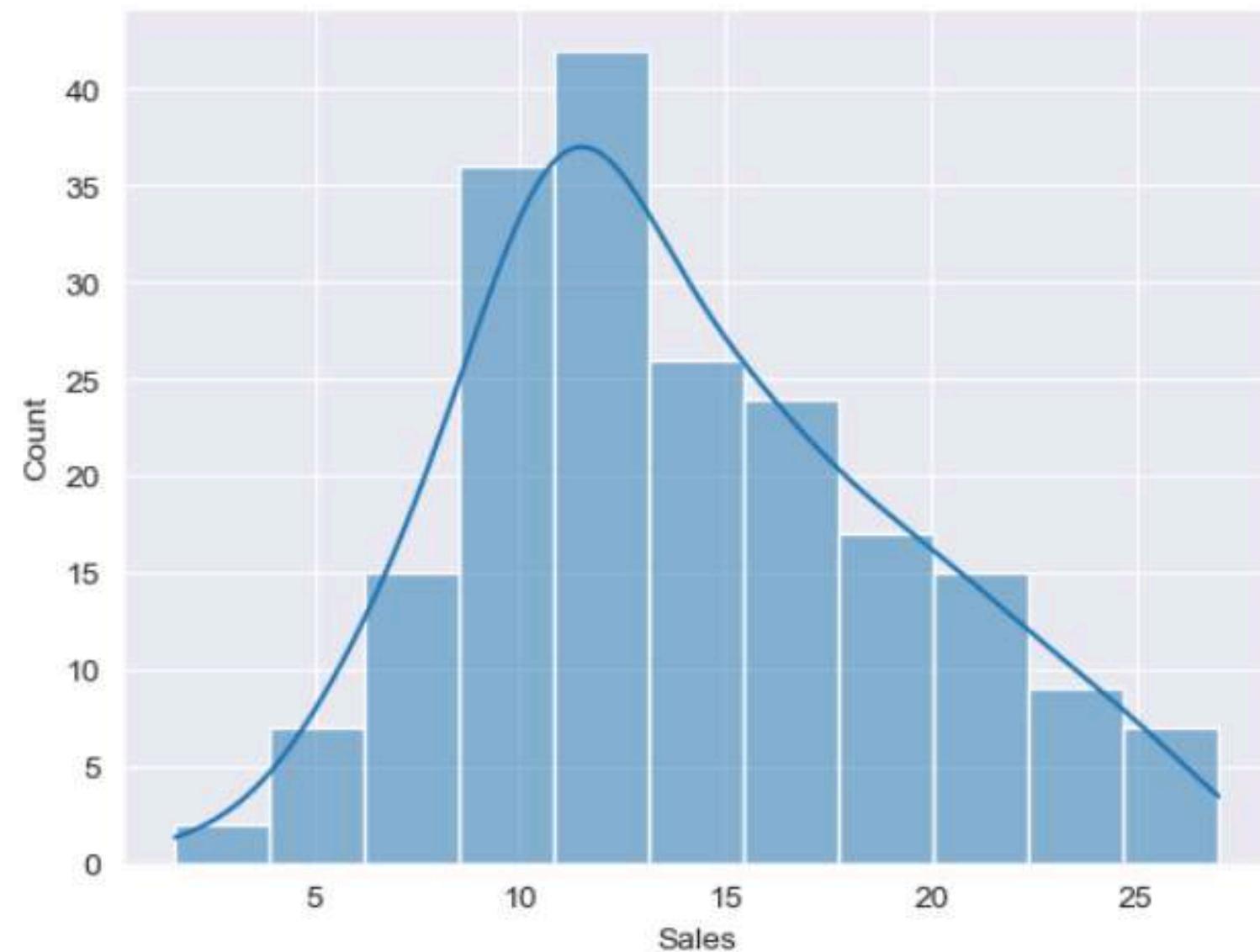
# Jupyter Notebook

## image processing

Histogram

```
[32]: sns.set_style('darkgrid')

sns.histplot(df.Sales, kde=True)
# sns.distplot(df.Sales, kde=True)
plt.show()
```



# Jupyter Notebook

## image processing

### KDE

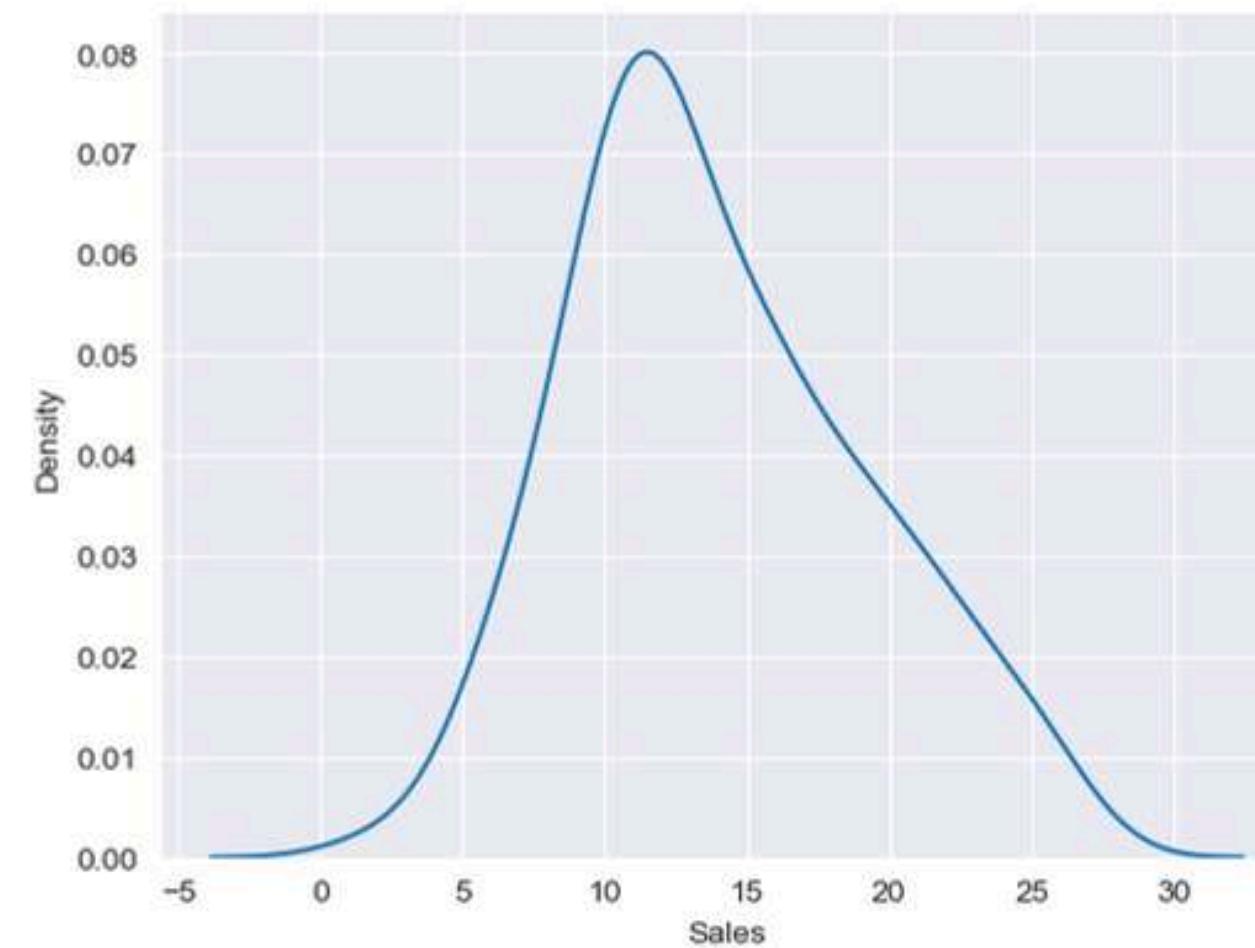
A kernel density estimate (KDE) plot (for visualizing the distribution of observations in a dataset)

```
[33]: sns.kdeplot(df.Sales, shade=False)
```

```
C:\Users\bluep\AppData\Local\Temp\ipykernel_14356\4156251864.py:1: FutureWarning:  
  'shade' is now deprecated in favor of 'fill'; setting 'fill=False'.  
  This will become an error in seaborn v0.14.0; please update your code.
```

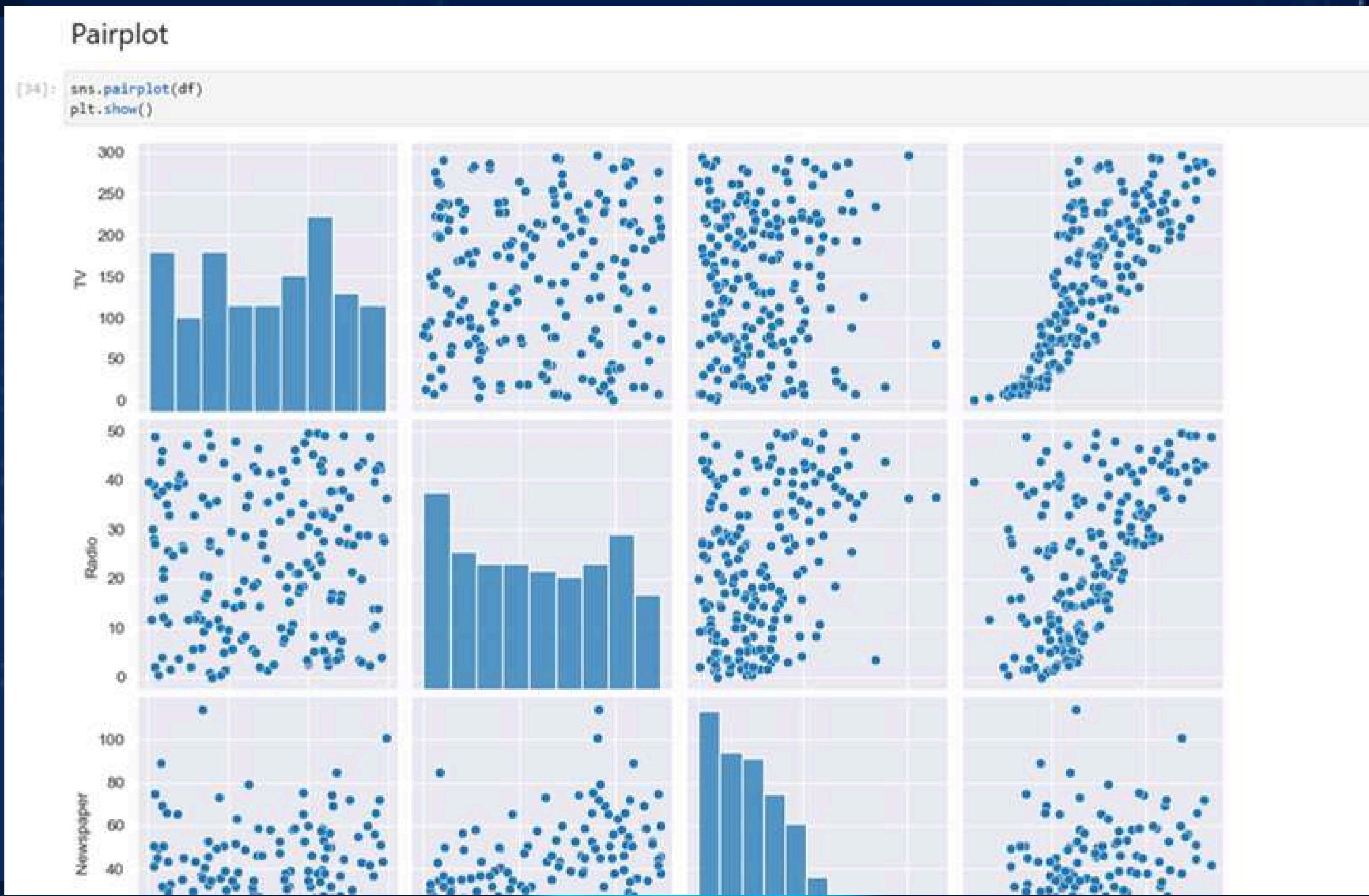
```
sns.kdeplot(df.Sales, shade=False)
```

```
[33]: <Axes: xlabel='Sales', ylabel='Density'>
```



# Jupyter Notebook

## image processing

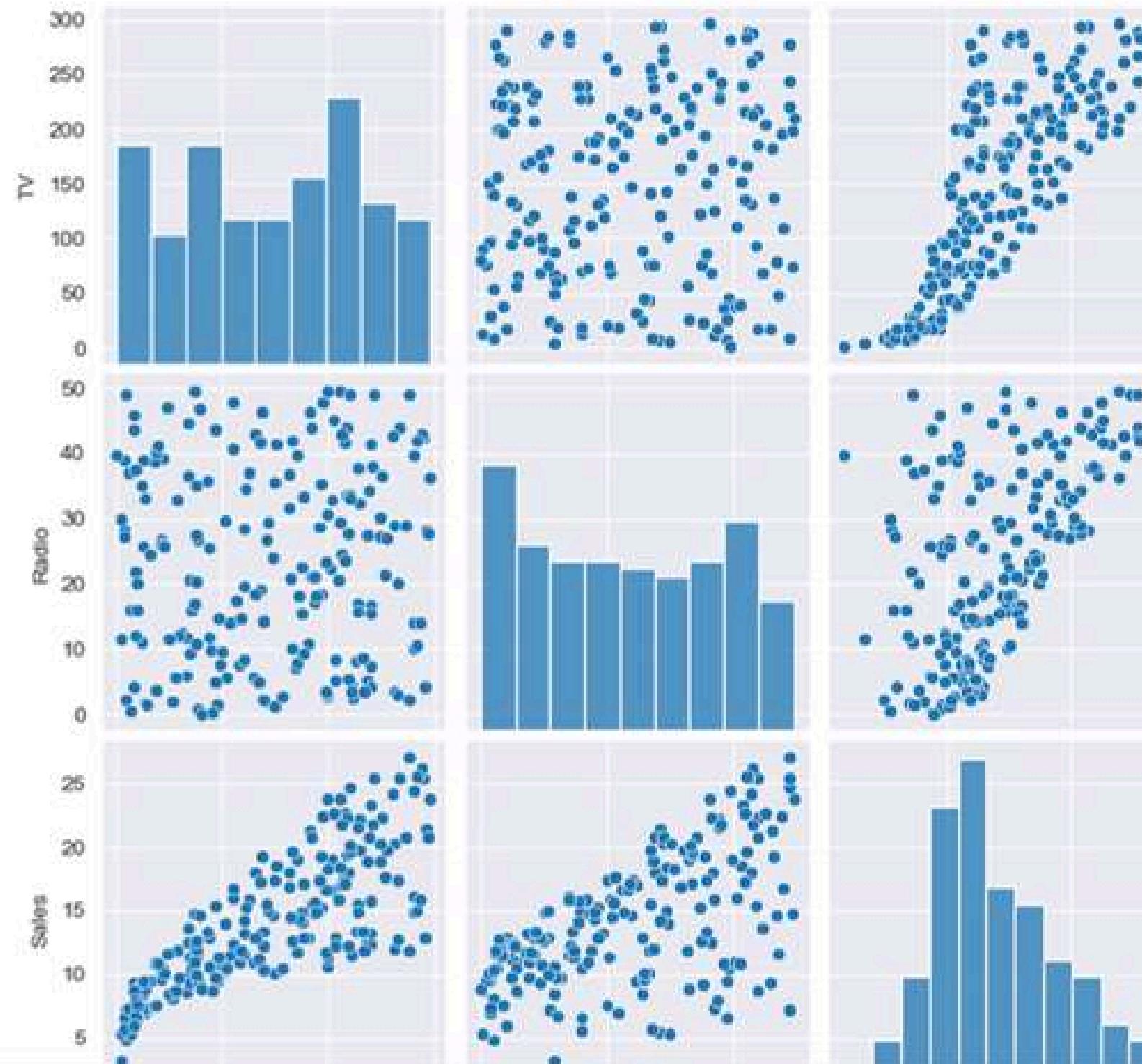


# Jupyter Notebook

## image processing

```
[35]: columns = ['TV', 'Radio', 'Sales']  
sns.pairplot(df[columns])
```

```
[35]: <seaborn.axisgrid.PairGrid at 0x2273672b220>
```



# Jupyter Notebook

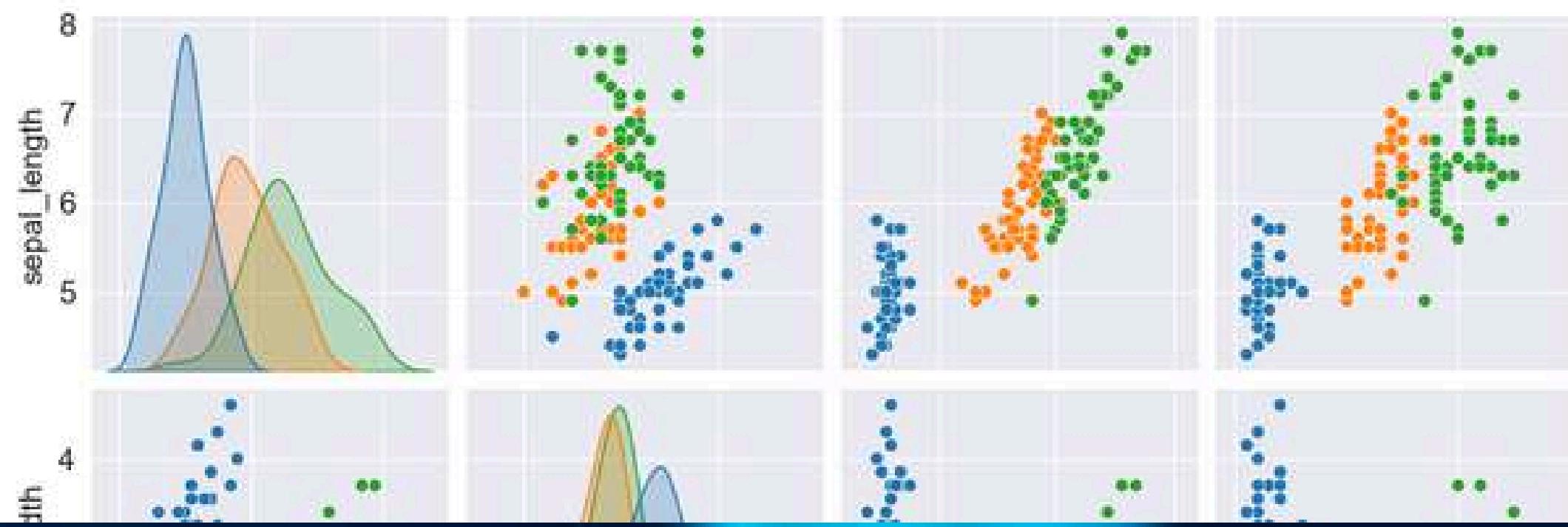
## image processing

### Iris dataset

```
[36]: df = sns.load_dataset('iris')
df.head()
df.tail()
```

```
[36]:   sepal_length  sepal_width  petal_length  petal_width  species
  145          6.7         3.0         5.2         2.3  virginica
  146          6.3         2.5         5.0         1.9  virginica
  147          6.5         3.0         5.2         2.0  virginica
  148          6.2         3.4         5.4         2.3  virginica
  149          5.9         3.0         5.1         1.8  virginica
```

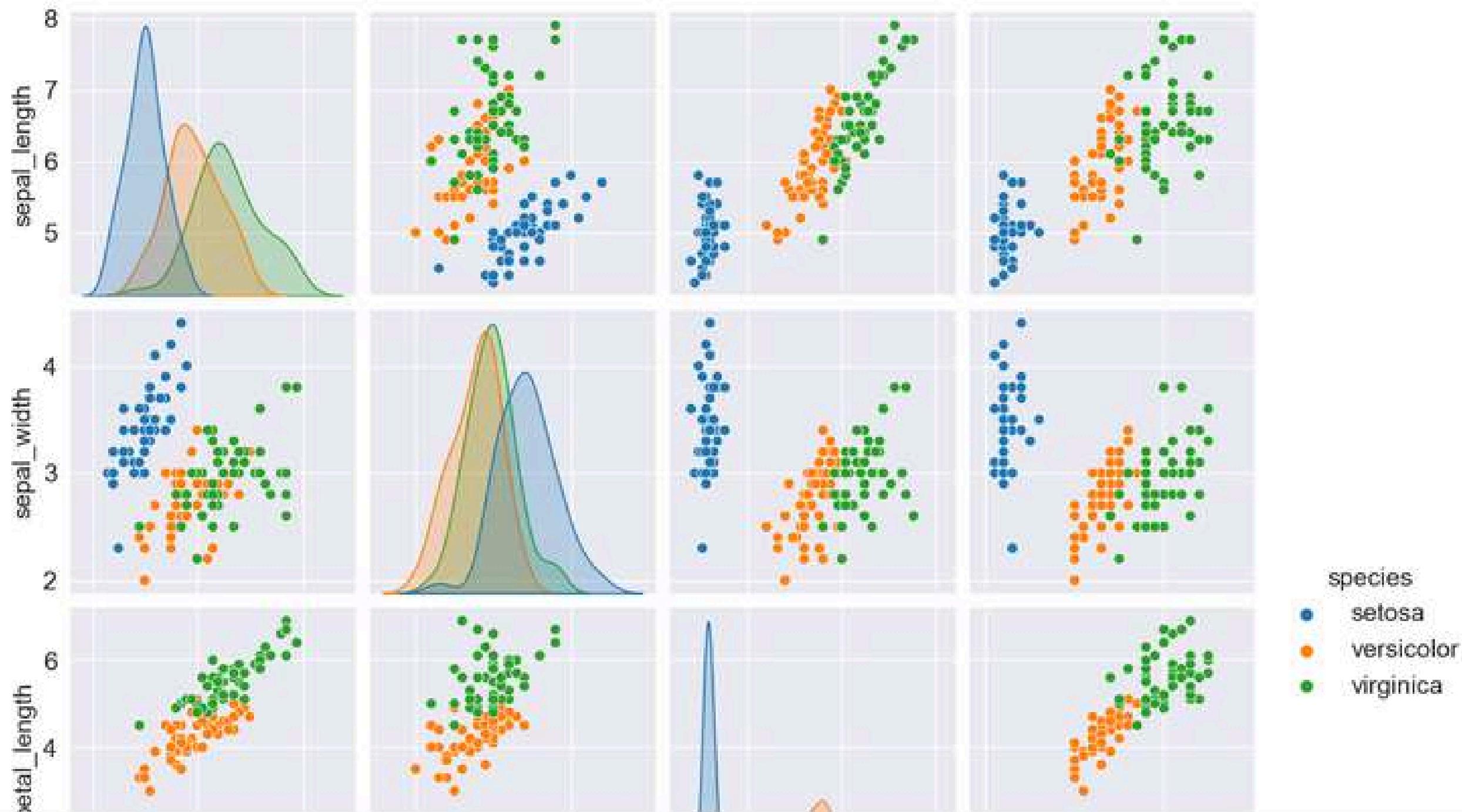
```
[37]: plt.rcParams.update({'font.size':16})
#sns.pairplot(df)
sns.pairplot(df, hue='species')
plt.show()
```



# Jupyter Notebook

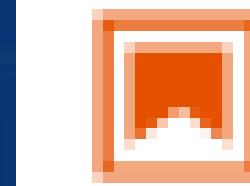
## image processing

```
[33]: plt.rcParams.update({'font.size':14})
sns.pairplot(df)
sns.pairplot(df, hue='species')
plt.show()
```



# Jupyter Notebook

## image processing



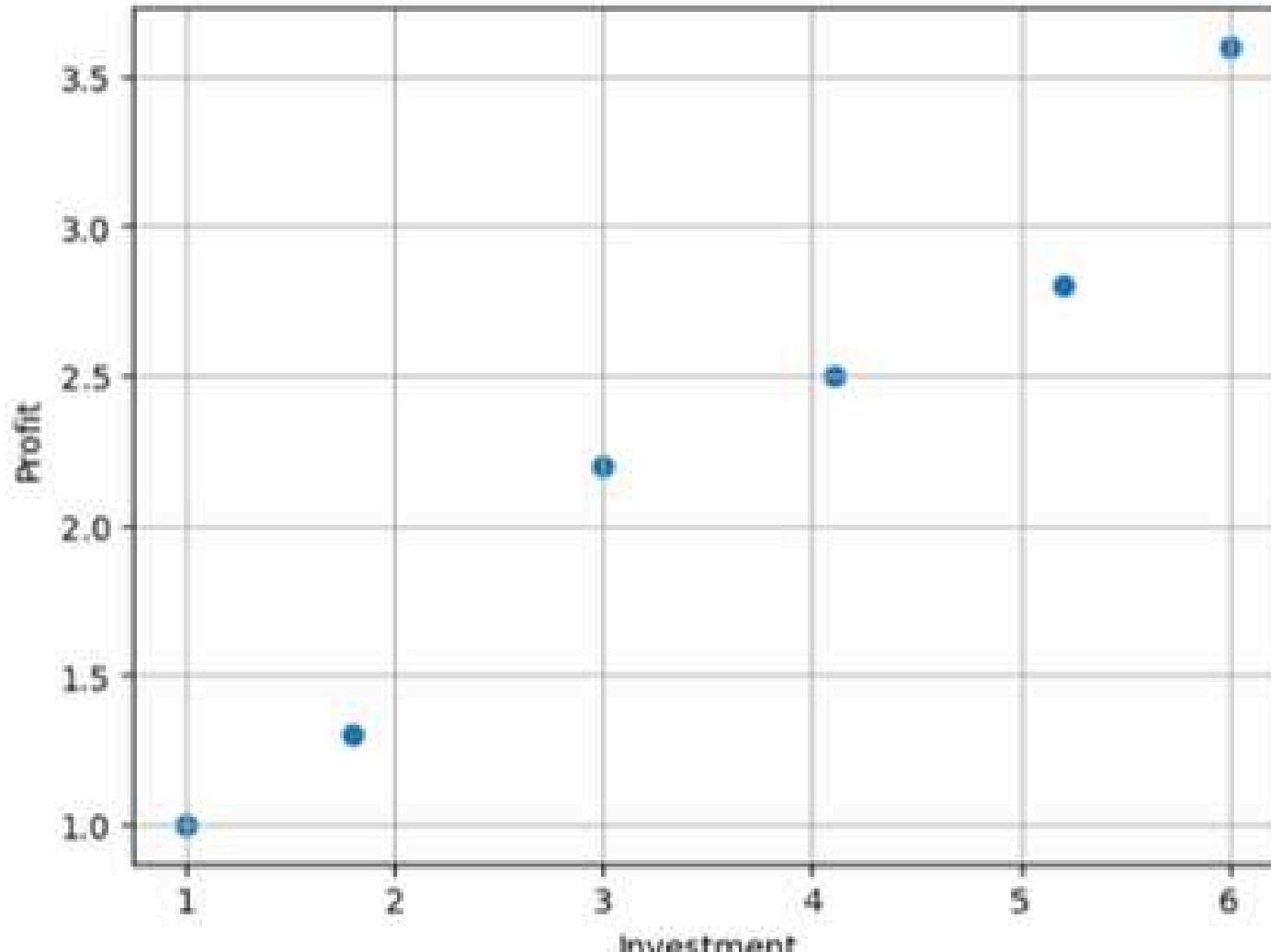
ch7\_model.ipynb

# Jupyter Notebook

## image processing

ข้อมูลปกติ Original Programming

```
(1): import matplotlib.pyplot as plt  
  
x_data = [1.0, 1.8, 3.0, 4.1, 5.2, 6.0]  
y_data = [1, 1.3, 2.2, 2.5, 2.8, 3.6]  
  
plt.xlabel('Investment')  
plt.ylabel('Profit')  
plt.scatter(x_data, y_data)  
plt.grid()  
plt.show()
```



A scatter plot showing the relationship between Investment (X-axis) and Profit (Y-axis). The X-axis ranges from 1 to 6, and the Y-axis ranges from 1.0 to 3.5. The data points show a positive linear trend.

Investment	Profit
1.0	1.0
1.8	1.3
3.0	2.2
4.1	2.5
5.2	2.8
6.0	3.6

# Jupyter Notebook

## image processing

### Machine Learning (w/ Scikit-learn)

```
[3]: !conda list scikit-learn

# packages in environment at C:\Users\bluep\anaconda3:
#
# Name           Version      Build  Channel
scikit-learn    1.2.1        py310hd77b12b_0
scikit-learn-intelex 2023.0.2  py310haa95532_0

[4]: import matplotlib.pyplot as plt
      import numpy as np
```

### Dataset

```
[6]: x_data = [1.0, 1.8, 3.0, 4.1, 5.2, 6.0]
y_data = [1, 1.3, 2.2, 2.5, 2.8, 3.6]

x = np.array(x_data)      # Numpy Array
y = np.array(y_data)
```

```
[7]: x
[7]: array([1. , 1.8, 3. , 4.1, 5.2, 6. ])
```

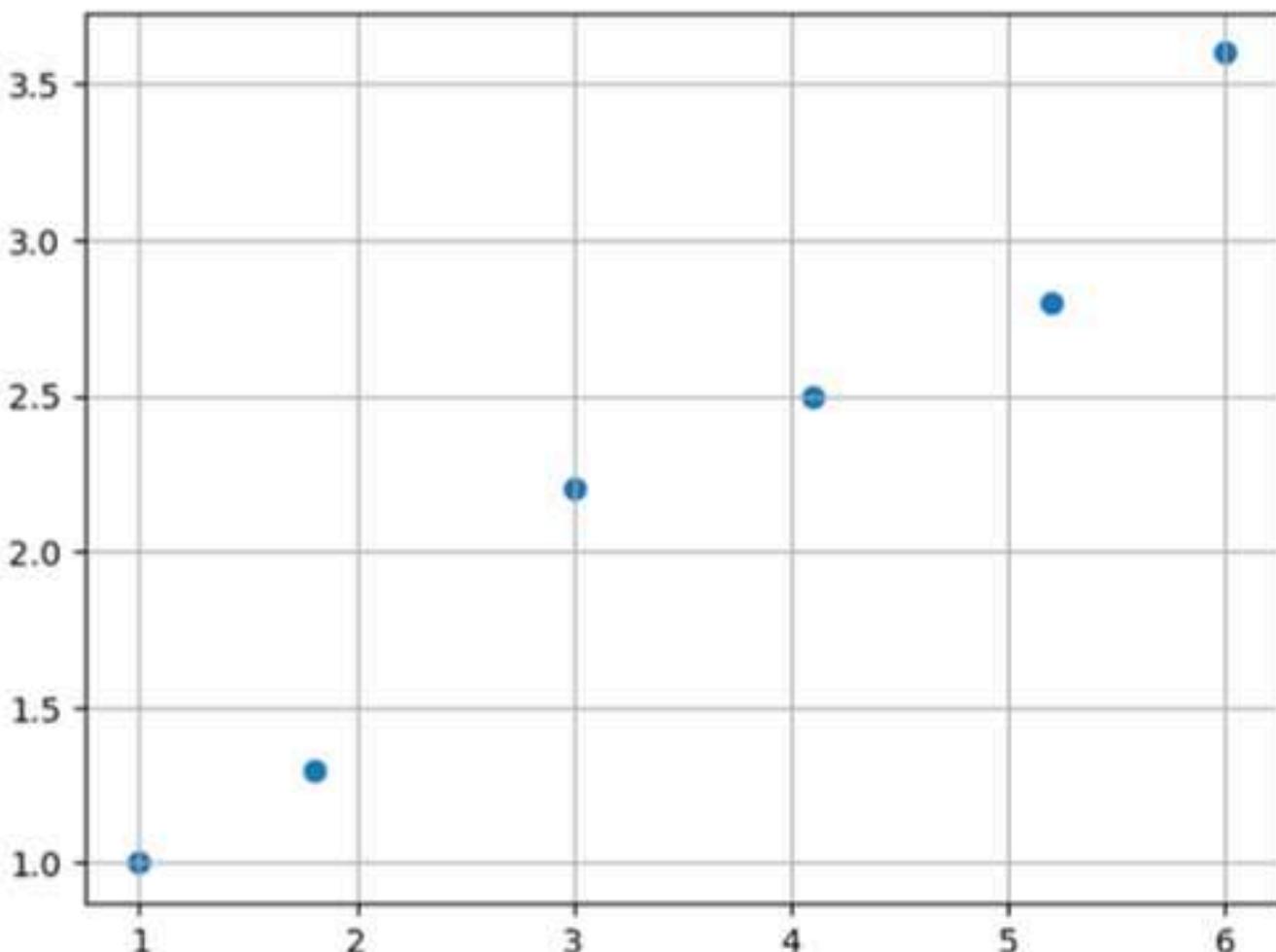
```
[8]: y
[8]: array([1. , 1.3, 2.2, 2.5, 2.8, 3.6])
```

# Jupyter Notebook

## image processing

### Data Visualization

```
[9]: plt.scatter(x,y)  
plt.grid()  
plt.show()
```



```
[10]: x = x.reshape(-1, 1)  
x
```

```
[10]: array([[1. ],  
           [1.8],  
           [3. ]])
```

# Jupyter Notebook

## image processing

### Model & Train

```
[11]: from sklearn.linear_model import LinearRegression  
  
[12]: model = LinearRegression()  
model.fit(x, y)  
  
[12]: * LinearRegression  
LinearRegression()
```

### Predict ทำนาย

```
[13]: new_input = 2.5  
model.predict([[new_input]])  
  
[13]: array([1.73670696])  
  
[22]: x_input = [[2.0],  
               [2.5],  
               [3],  
               [7.0]] # ไม่จำเป็นต้องเรียงตาม  
y_predict = model.predict(x_input)  
y_predict  
  
[22]: array([1.49246448, 1.73670696, 1.98094944, 3.93488926])  
  
[24]: x_input = [2.0, 2.5, 3, 7.0]  
x_input = np.array(x_input).reshape(-1, 1)  
  
x_input  
  
[24]: array([[2. ],  
           [2.5],  
           [3],  
           [7. ]])
```

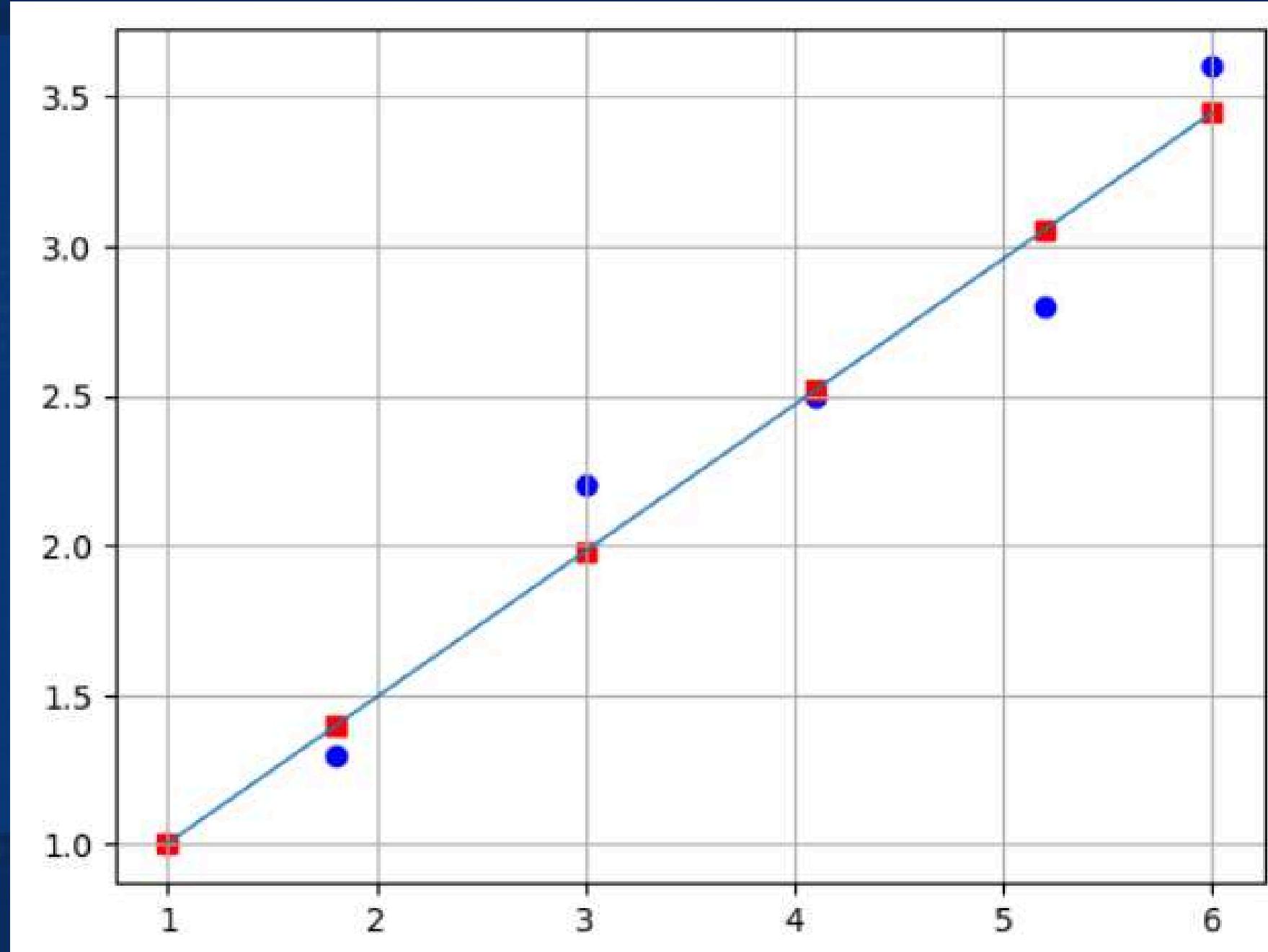
# Jupyter Notebook

## image processing

```
[28]: # គិតថាបានលក្ខណៈ input នាមអតិថិជន  
for i, y in enumerate(y_predict):  
    print('x={0} y={1:.4f}'.format(x_input[i], y))  
  
x=[2.] y=1.4925  
x=[2.5] y=1.7367  
x=[3.] y=1.9809  
x=[7.] y=3.9349  
  
[27]: predict = model.predict(x)  
  
# plt.rcParams['figure.figsize'] = 4, 3 #  
  
plt.grid()  
plt.scatter(x_data, y_data, color='b')      # ម៉ោងទីតា (real data points)  
plt.plot(x_data, predict, linewidth='1')       # ការពារ (prediction)  
plt.scatter(x_data, predict, color='r', marker='s')  
# plt.savefig('modell.png', dpi=100)  
plt.show()
```

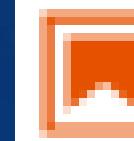
# Jupyter Notebook

## image processing



# Jupyter Notebook

## image processing



[ch8\\_use\\_model.ipynb](#)

# Jupyter Notebook

## image processing

การใช้โมเดล Model to use

```
[2]: import joblib
import numpy as np

model = joblib.load('model_math.pkl')

[3]: model.coef_, model.intercept_      # m and b
      (array([0.48848496]), 0.5154945733698053)

[4]: x_input = [2.0, 2.5, 3, 5.0, 7.0]
x_input = np.array(x_input).reshape(-1, 1)

y_predict = model.predict(x_input)
y_predict

[4]: array([1.49246448, 1.73670696, 1.98094944, 2.95791935, 3.93488926])

[5]: # แสดงผล output แบบหน้าจอ
for i, y in enumerate(y_predict):
    print('x={:.1f} y={:.4f}'.format(x_input[i], y))

x=[2.] y=1.4925
x=[2.5] y=1.7367
x=[3.] y=1.9809
x=[5.] y=2.9579
x=[7.] y=3.9349
```



# Thank You!

**End of Session Image Processing**