



國立高雄科技大學

National Kaohsiung University of Science and Technology

Basic Image Processing Using OpenCV on Google Colab

Speaker: Shih-Shinh Huang

Version: v011

Date: August 15, 2021






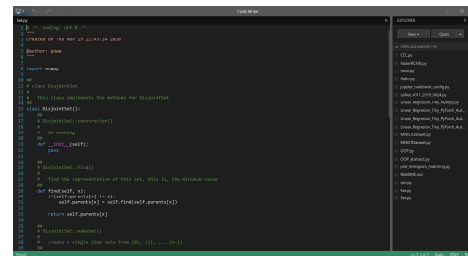
Outline

- Google Colaboartory
 - Why Colab
 - Colab Startup
 - "Hello, Colab!" Program
- Basic Image Processing
 - About OpenCV
 - Image Loading
 - Image Accessing



Google Colaboartory

- Why Colab 
 - What you have to do when you are interested in exploring machine learning (ML)



programming IDE



ML library
(TensorFlow, Pytorch)

hardware accelerator
(GPU/TPU)




hardware driver (cuda)



IDE: integrated development environment



Google Colaboartory

- Why Colab 
 - Colab is a **free** cloud services to encourage machine learning research.
 - free GPU/TPU accelerator
 - Jupyter Notebook for Python programming
 - pre-installed ML libraries: Tensorflow, Pytorch,...

accelerator hardware



hardware driver **Google Account**



programming IDE



ML library





Google Colaboartory

- Colab Startup: Login
 - link to the [Colab](#) official website
 - enter your name / password

The image displays three screenshots illustrating the Google Colab login process:

- Left Screenshot:** Shows the Google Colab welcome page in a browser. A red arrow points from the "Sign in" button in the top right corner to the login form in the middle screenshot.
- Middle Screenshot:** Shows the Google login form. The user is identified as "Hi 世勳" with the email "poww@nkust.edu.tw". A red dashed box highlights the "Enter your password" field, with a red arrow pointing to it from the left screenshot. Below the field is a "Show password" checkbox and a "Next" button. A "Forgot password?" link is also visible.
- Right Screenshot:** Shows the Google Colab interface after login. A red arrow points from the "Sign in" button in the top right corner to the right screenshot.

Red text annotations are present:

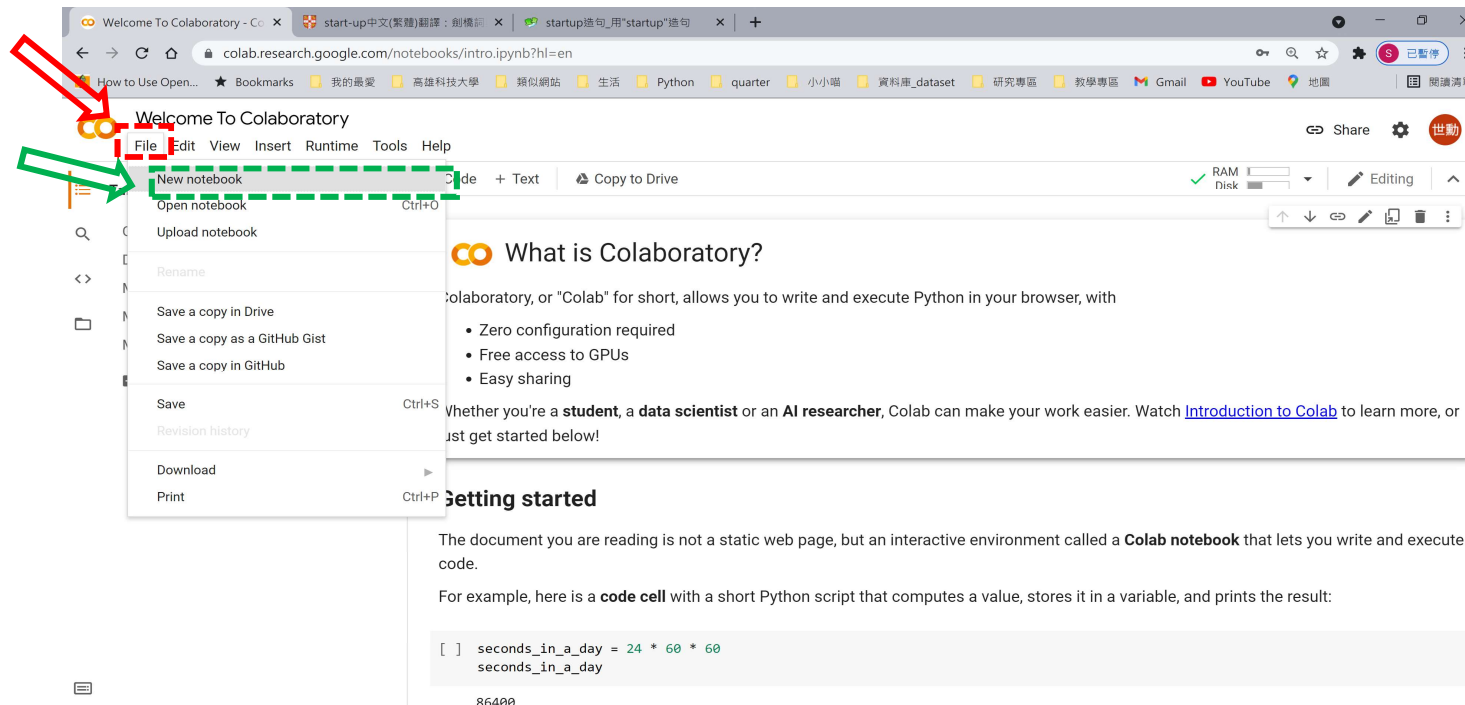
- https://** is written in red below the middle screenshot.
- oogle.com/** is written in red below the right screenshot.

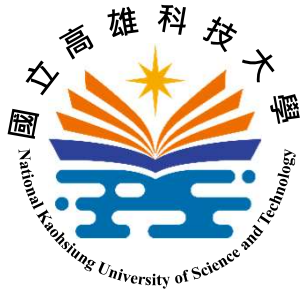
A small "Nkfust.cc" logo is visible in the bottom right corner of the right screenshot.



Google Colaboartory

- Colab Startup: New Notebook
 - create a program file: → File → New notebook

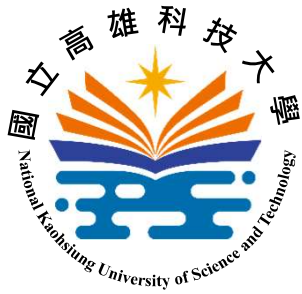




Google Colaboartory

The screenshot shows the Google Colab interface with several components highlighted by red dashed boxes and green arrows:

- File Name:** Located at the top left, above the code editor.
- Main Menu:** Located at the top center, containing 'Runtime', 'Tools', 'Help', and 'Last edited on August 6'.
- Connect Dropdown Menu:** Located at the top right, containing 'Connect', 'Editing', and a user profile icon.
- Table of Contents:** Located in the sidebar on the left, with a green arrow pointing to the '+Code +' button.
- Find and Replace:** Located in the sidebar on the left, with a green arrow pointing to the search icon.
- Code Snippets:** Located in the sidebar on the left, with a green arrow pointing to the '<>' icon.
- Files:** Located in the sidebar on the left, with a green arrow pointing to the folder icon.
- Jupyter Notebook:** The main workspace area, labeled in red text at the bottom center.



Google Colaboartory

The screenshot shows the Google Colab interface with a notebook titled "Untitled0.ipynb". The interface includes a top menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right, there are icons for "Comment", "Share", and "世動" (World), along with RAM and Disk usage indicators and an "Editing" mode selector.

The notebook content consists of two cells:

- A **Text cell** containing the text "This is a Text cell".
- A **Code cell** containing the code `print('This is a Code cell')`. Below the code, the output "This is a Code cell" is displayed and labeled as **Output**.

A **Cell Control Menu** is highlighted with a red dashed box on the right side of the code cell. The menu contains the following icons and their corresponding functions:

- Up arrow: **move cell up**
- Down arrow: **move cell down**
- Comment icon: **add comment**
- Trash icon: **del cell**

At the bottom of the interface, a status bar shows "0s completed at 2:47 PM".





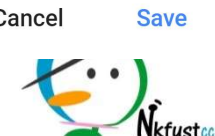
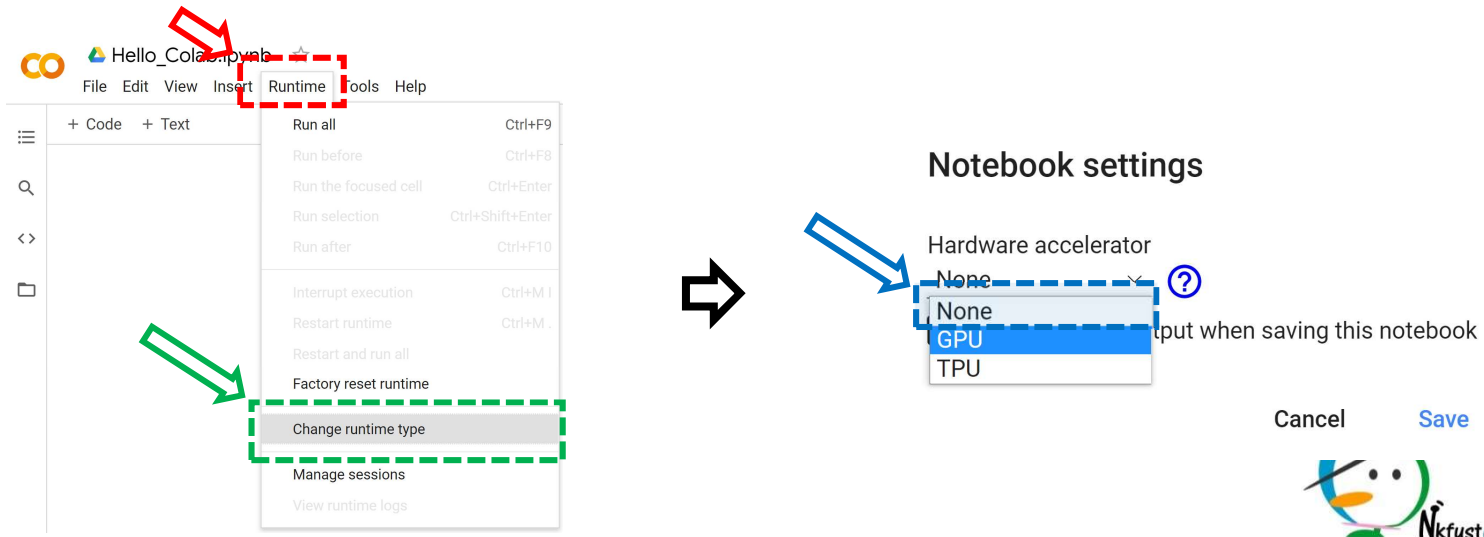
Google Colaboartory

- "Hello, Colab!" Program

- Step 1: rename your file as “Hello_Colab.ipynb”



- Step 2: select the hardware accelerator





Google Colaboartory

- "Hello, Colab!" Program
 - Step 3: write the code to display “Hello, Colab!”

```
print('Hello, Colab!')
```

- Step 4: run the “Hello, Colab!” program

A screenshot of the Google Colab interface. At the top, there are buttons for '+ Code' and '+ Text'. Below, a code cell contains the Python code `print('Hello, Colab!')`. To the left of the code is a play button icon and a green checkmark, with '0s' indicating execution time. Below the code cell, the output 'Hello, Colab!' is displayed. At the bottom, a menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. The 'Runtime' menu is open, showing options like 'Run all' (Ctrl+F9) and 'Run before' (Ctrl+F8). Red arrows point to the play button, the output, and the 'Runtime' menu. A green dashed box highlights the 'Run all' option.

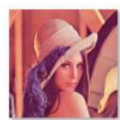


Basic Image Processing

- About OpenCV
 - OpenCV is an open source computer vision and machine learning software library.
 - OpenCV has more than 2,500 optimized algorithms
 - computer vision
 - machine learning
 - OpenCV has interfaces for C++, Python, Java and MATLAB programming languages.

Basic Image Processing

- Image Loading
 - upload a local image file to Google cloud drive
 - read in the image file and display it



lena.bmp
BMP 檔案
768 KB

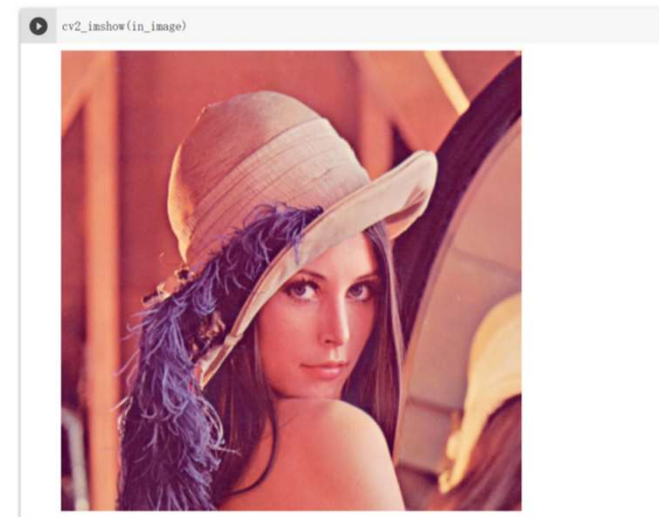
upload



Google
cloud drive

load

Colab





Basic Image Processing

- Image Loading: File Upload
 - mount the Google cloud drive

A composite image illustrating the process of mounting Google Drive. On the left, a 'Files' window shows a 'drive' folder highlighted in green. A large white arrow labeled 'expand' points from this folder to a 'drive' folder highlighted in red in a larger 'Files' window on the right. Below the arrow, a dark grey box contains the text 'Mounting Google Drive...'. The right window also shows a 'MyDrive' folder expanded to reveal subfolders like 'Colab Notebooks' and 'Personal Doc'. A small text box 'Google Drive files notebook to moc' is visible between the two windows.





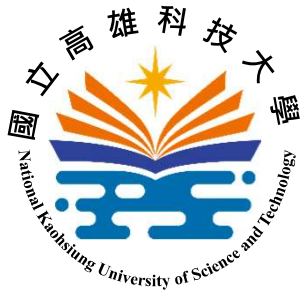
Basic Image Processing

- Image Loading: File Upload
 - select an image file in local disk for uploading

The image shows a file explorer window with a sidebar on the left and a main pane on the right. The sidebar shows a tree view with folders like 'drive', 'MyDrive', 'Colab Notebooks', 'Personal Doc', 'Othercomputers', and 'sample_data'. The 'Colab Notebooks' folder is expanded, showing files 'Untitled0.ipynb' and 'lena.bmp'. The main pane shows a list of files with columns for '名稱' (Name) and '修改日期' (Modified Date). The file 'lena.bmp' is highlighted. A red arrow points from the sidebar to the main pane. A green arrow points from the file name 'lena.bmp' in the main pane to the '開啟(O)' (Open) button in the bottom right corner of the window. The path '/content/drive/MyDrive/Colab Notebooks/lena.bmp' is written in green and red text over the image.

“/content/drive/MyDrive/Colab Notebooks/lena.bmp”

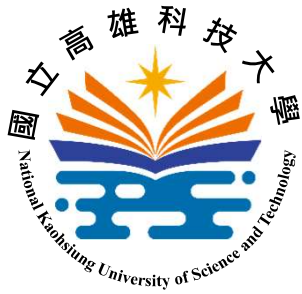




Basic Image Processing

- Image Loading: Coding
 - **Step 1:** import two packages including **opencv** and **google.colab** packages
 - **Step 2:** invoke **imread()** in opencv package to read an image file in Google cloud drive
 - **Step 3:** invoke **cv2_imshow()** in google.colab package to display the image





Basic Image Processing

• Image Loading: Coding

+ Code + Text

Step1: import two packages

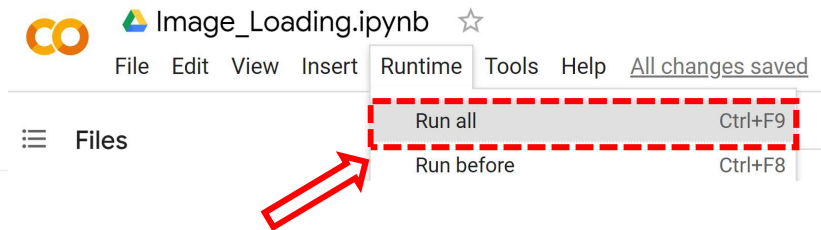
```
[1] import cv2
import google.colab.patches as colab
```

Step2: load an image file in Google cloud drive

```
in_image = cv2.imread("/content/drive/MyDrive/Colab Notebooks/lena.bmp")
```

Step3: display the image

```
[3] colab.cv2_imshow(in_image)
```



Basic Image Processing

- Image Accessing
 - opencv loads an image as a **numpy** array
 - gray image: 2D numpy array
 - color image: 3D numpy array



shape = (512, 512)

 h:height w:width

shape = (512, 512, 3)

 h:height w:width c: channels

Basic Image Processing

- Image Accessing

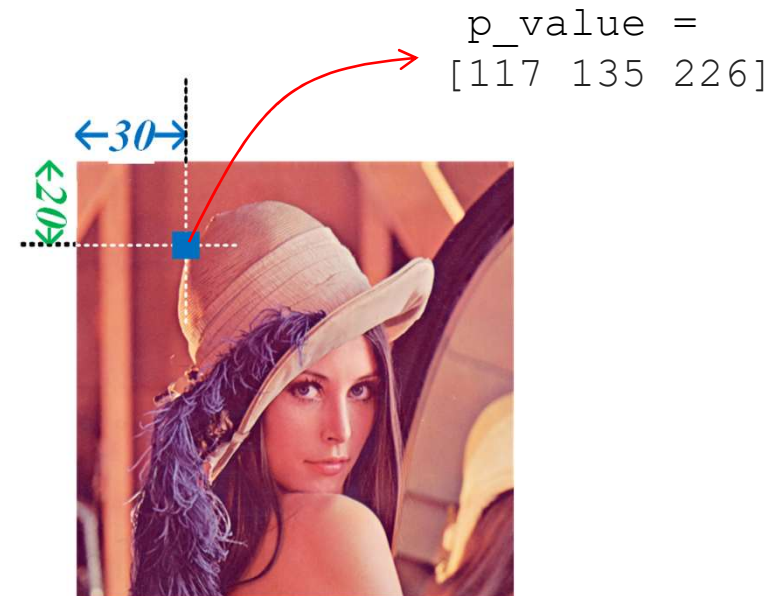
- pixel accessing: single-element indexing

read a pixel

⇒ `p_value = in_image[20, 30]`

write a pixel

⇒ `in_image[20, 30] = [255, 0, 0]`



Basic Image Processing

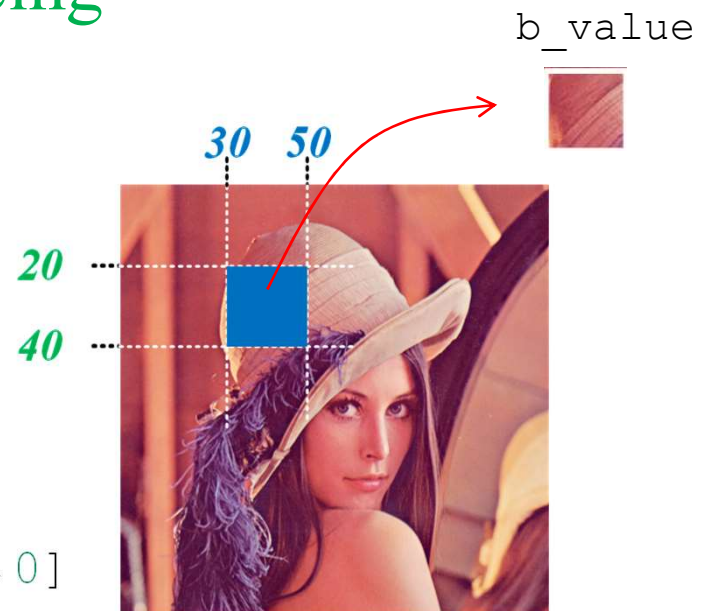
- Image Accessing
 - block accessing: array slicing

read a **block**

⇒ `b_value`
`= in_image[20:40, 30:50]`

write a **block**

⇒ `in_image[20:40, 30:50] = [255, 0, 0]`





thank you

© 1996, 2002 SANRIO CO., LTD.