

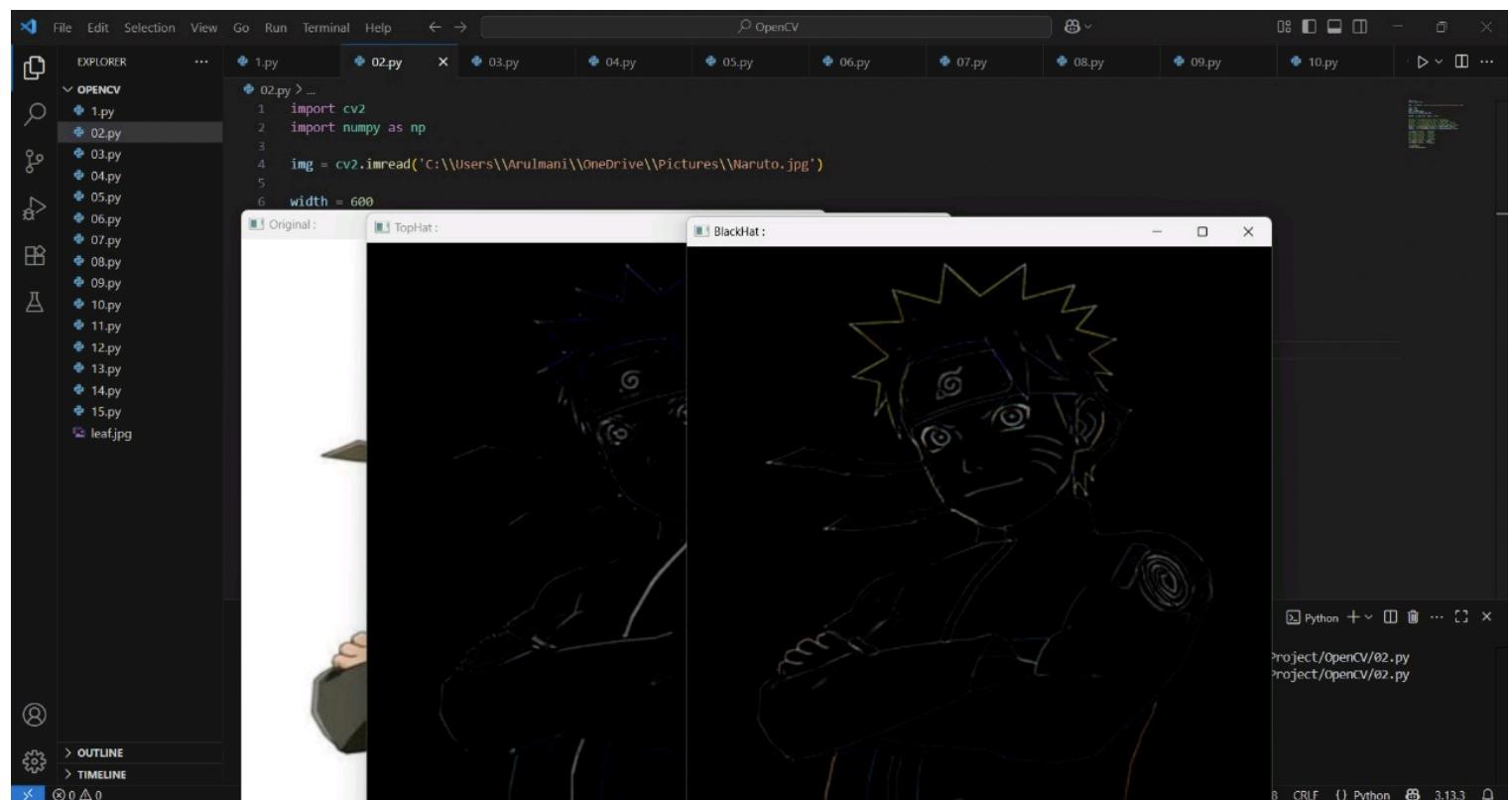
step1 : Writing an image

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing files in the 'OPENCV' folder: 1.py, 02.py, 03.py, 04.py, 05.py, 06.py, 07.py, 08.py, 09.py, 10.py, leaf.jpg. The main area displays the following Python code:

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ OpenCV 🌐 🔍 EXPLORER ... 1.py ✘ 02.py 03.py 04.py 05.py 06.py 07.py 08.py 09.py 10.py 🗑️ ... 1.py > ... 1 import cv2 2 3 img = cv2.imread('C:\\Users\\Arulmani\\OneDrive\\Pictures\\Naruto.jpg',0) 4 5 print("Dimension",img.shape) 6 7 width = img.shape[1] 8 height = 400 9 dim = (width,height) 10 resize = cv2.resize(img, dim) 11 12 cv2.imshow("window",resize) 13 14 cv2.imwrite('leaf.jpg',img) 15 16 cv2.waitKey(0) 17 18 cv2.destroyAllWindows()
```

The terminal tab shows command-line output for files 02.py and 1.py, both indicating a dimension of (500, 300). A preview window titled 'window' shows an image of a character from the anime Naruto. The status bar at the bottom right indicates the code is in Python mode, line 14, column 18, with 3.13.3 version information.

Step2 : Morphological Operations



Step3 : Flipping an image

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files 1.py through 10.py and leaf.jpg.
- Code Editor:** The file 03.py is open, containing the following Python code using OpenCV:

```
import cv2
img = cv2.imread('C:\\Users\\Arulmani\\OneDrive\\Pictures\\Naruto.jpg')
width = 600
height = 350
dimension = (width, height)
resized = cv2.resize(img, dimension)
flip = cv2.flip(resized, 1)
cv2.imshow("Flipped", flip)
flip1 = cv2.flip(resized, 0)
cv2.imshow("Flipped 1", flip1)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- Output Console:** The terminal window shows command-line output:

```
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/02.py
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/02.py
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/03.py
```

- Preview:** Two windows titled "Flipped" and "Flipped 1" show the original image of Naruto Uzumaki and its horizontal flip respectively.
- Status Bar:** Shows file statistics (Ln 3, Col 70), encoding (Spaces:4, UTF-8, CRLF), language (Python), and version (3.13.3).

Step4 : Resizing an image based on the aspect ratio

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar containing files like 1.py, 02.py, 03.py, 04.py (selected), 05.py, 06.py, 07.py, 08.py, 09.py, 10.py, 11.py, 12.py, 13.py, 14.py, 15.py, and leaf.jpg. The main area displays a Python script named 04.py:

```
File Edit Selection View Go Run Terminal Help ← → OpenCV 04.py 01.py 02.py 03.py 04.py 05.py 06.py 07.py 08.py 09.py 10.py ...  
EXPLORER  
OPENCV  
1.py  
2.py  
3.py  
4.py  
5.py  
6.py  
7.py  
8.py  
9.py  
10.py  
11.py  
12.py  
13.py  
14.py  
15.py  
leaf.jpg  
04.py > ...  
1 import cv2  
2  
3 img = cv2.imread("C:\\Users\\Arulmani\\OneDrive\\Pictures\\Naruto.jpg")  
4  
5 print("Dimension of image: ", img.shape)  
6  
7 scale = 50  
8  
9 width = int(img.shape[1] * scale / 100)  
10 height = int(img.shape[0] * scale / 100)  
11  
12 dim = (width, height)  
13  
14 resized = cv2.resize(img, dim, interpolation=cv2.INTER_AREA)  
15  
16 print("Resized image : ",resized.shape)  
17  
18 cv2.imshow("Original", img)  
19 cv2.imshow("Resized", resized)  
20  
21 cv2.waitKey(0)  
22 cv2.destroyAllWindows()
```

Two windows are displayed on the right: "Original" showing the full-size image of Naruto Uzumaki, and "Resized" showing the same image scaled down. The terminal at the bottom shows command-line output:

```
PS C:\Users\Arulmani\Documents\Project\OpenCV & c:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe C:/Users/Arulmani/Documents/Project/OpenCV/04.py  
Dimension of image: (500, 300, 3)  
Resized image : (250, 150, 3)
```

Bottom status bar: Ln 3, Col 70 Spaces:4 UTF-8 CRLF {} Python 3.13.3

Step5 : Adding shapes and text

The screenshot shows a code editor interface with a Python script named `05.py` open. The code uses OpenCV to draw various shapes on a black image and add text.

```
File Edit Selection View Go Run Terminal Help ← → OpenCV 05.py 1.py 02.py 03.py 04.py 06.py 07.py 08.py 09.py 10.py ... EXPLORER OPENCV 1.py 02.py 03.py 04.py 05.py 06.py 07.py 08.py 09.py 10.py 11.py 12.py 13.py 14.py 15.py leaf.jpg 05.py > ... 1 import cv2 2 import numpy as np 3 4 #img = cv2.imread('C:\\\\Users\\\\Arulmani\\\\OneDrive\\\\Desktop\\\\leaf.jpg') 5 img = np.zeros(shape=(600,600,3),dtype=np.uint8) 6 cv2.line(img, (0,0), (150,150), (255,0,0), 2) 7 8 cv2.rectangle(img, (200,150), (250, 300), (0,255,0), 2) 9 10 cv2.circle(img, (300,75), 70, (255, 0, 255), 4) 11 12 pts_polygon = np.array([[100,50], [100,300], [50,300], [50,50], [100,50]]) 13 cv2.polylines(img,[pts_polygon], True,(0,255,255), 2) 14 15 font = cv2.FONT_HERSHEY_SIMPLEX 16 cv2.putText(img, "Hey", (10,500), font, 4, (200,200,200)) 17 18 cv2.imshow("Image", img) 19 cv2.waitKey(0) 20 cv2.destroyAllWindows()
```

The output terminal shows the command-line session:

```
PS C:\Users\Arulmani\Documents\Project\OpenCV & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/05.py
```

The image window displays a black background with a purple circle at the top right, a green rectangle in the center, a blue diagonal line, and the word "Hey" written in white.

Step6 : Resizing an image based on the aspect ratio 2

The screenshot shows a code editor interface with several tabs at the top labeled 01.py, 02.py, 03.py, 04.py, 05.py, 06.py (selected), 07.py, 08.py, 09.py, 09.py, and 10.py. The left sidebar has a tree view under 'OPENCV' containing files 01.py through 10.py and leaf.jpg. The main area displays the contents of 06.py:

```
import cv2
import numpy as np

img = cv2.imread('C:\Users\Arulmani\Documents\Project\OpenCV\leaf.jpg', cv2.IMREAD_COLOR)

column = img.shape[1]
row = img.shape[0]

s = np.float32([(1,0,15), (0,1,15)])
shifted = cv2.warpAffine(img, s, (row, column))

cv2.imshow("Original", img)
cv2.imshow("Shifted", shifted)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Two windows are open below the code: 'Original' and 'Shifted'. Both windows show an image of a character from the anime Naruto, standing with arms crossed. The 'Original' window is larger and centered, while the 'Shifted' window is smaller and positioned to the left of it. The terminal output at the bottom shows the command to run the script and the resulting dimensions of the resized image.

```
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/06.py
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/06.py
Dimension of image: (500, 300, 3)
Resized image : (250, 150, 3)
```

Step7 : Image rotation

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing files in the 'OPENCV' folder: 1.py, 02.py, 03.py, 04.py, 05.py, 06.py, 07.py (selected), 08.py, 09.py, 0.py, and leaf.jpg. The main area displays the following Python code:

```
File Edit Selection View Go Run Terminal Help ← → OpenCV 07.py 1.py 02.py 03.py 04.py 05.py 06.py 08.py 09.py 10.py ...  
EXPLORER  
OPENCV  
1.py  
02.py  
03.py  
04.py  
05.py  
06.py  
07.py  
08.py  
09.py  
10.py  
leaf.jpg  
07.py > ...  
1 import cv2  
2  
3 img = cv2.imread('C:\\Users\\Arulmani\\OneDrive\\Pictures\\Naruto.jpg')  
4 row = img.shape[1]  
5 col = img.shape[0]  
6  
7 center = (row//2, col//2)  
8 angle = 90  
9  
10 r = cv2.getRotationMatrix2D(center, angle, 1)  
11  
12 rotate = cv2.warpAffine(img, r, (row, col))  
13  
14 cv2.imshow("Original", img)  
15 cv2.imshow("Rotated", rotate)  
16  
17 cv2.waitKey(0)  
18 cv2.destroyAllWindows()
```

Below the code, the terminal window shows the execution of the script:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:\Users\Arulmani\AppData\Local\Programs\Python\Python313\python.exe C:/Users/Arulmani/Documents/Project/OpenCV/07.py  
Dimension of image: (500, 300, 3)  
Resized image: (250, 150, 3)  
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe C:/Users/Arulmani/Documents/Project/OpenCV/05.py  
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe C:/Users/Arulmani/Documents/Project/OpenCV/06.py  
PS C:\Users\Arulmani\Documents\Project\OpenCV> & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe C:/Users/Arulmani/Documents/Project/OpenCV/07.py  
In 3. Col 70 Spaces:4 UTF-8 CR LF {} Python 3.13.3
```

Two windows are displayed: 'Rotated' showing the rotated version of the Naruto image, and 'Original' showing the original image.

Step8 : Thresholding

The screenshot shows a Windows desktop environment with a Python code editor and two windows displaying images. The code editor window is titled 'OpenCV' and contains the following Python script:

```
1 import cv2
2 import numpy as np
3
4 img = cv2.imread('C:\\Users\\Arulmani\\OneDrive\\Pictures\\Naruto.jpg')
5 threshold = 200
6
7 _,binary_threshold = cv2.threshold(img, 200, 255, cv2.THRESH_BINARY)
8
9 cv2.imshow("Original", img)
10 cv2.imshow("Binary Threshold", binary_threshold)
11 cv2.waitKey(0)
12 cv2.destroyAllWindows()
```

Below the code editor are two windows titled 'Binary Threshold' and 'Original'. The 'Binary Threshold' window displays a black and white binary thresholded version of the image, where the character's features are highlighted against a white background. The 'Original' window displays the original color image of the character.

At the bottom of the screen, there is a terminal window showing the command line output:

```
PS C:\Users\Arulmani\Documents\Project\OpenCV> &
Dimension of image: (500, 300, 3)
Resized image : (250, 150, 3)
PS C:\Users\Arulmani\Documents\Project\OpenCV> &
PS C:\Users\Arulmani\Documents\Project\OpenCV & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/06.py
PS C:\Users\Arulmani\Documents\Project\OpenCV & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/07.py
PS C:\Users\Arulmani\Documents\Project\OpenCV & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/08.py
PS C:\Users\Arulmani\Documents\Project\OpenCV & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/09.py
PS C:\Users\Arulmani\Documents\Project\OpenCV & C:/Users/Arulmani/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arulmani/Documents/Project/OpenCV/10.py
```

Step9 : Gaussian blur

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows files in the current project, including 1.py, 02.py, 03.py, 04.py, 05.py, 06.py, 07.py, 08.py, 09.py (selected), and 10.py.
- Code Editor:** Displays Python code for applying a Gaussian blur to an image of Naruto. The code reads the image from 'C:\Users\Arulmani\Documents\Project\OpenCV\Naruto.jpg', applies a Gaussian blur with kernel size (5, 5) and sigmaY 0, and shows the original and blurred images.
- Terminal:** Shows the command-line output of the script running in Python 3.13.3, displaying the dimension and resized image information.
- Output:** Shows two windows titled "Gaussian Blur" and "Original". Both windows display the same image of a young Naruto Uzumaki with his arms crossed, one showing the original image and the other showing the image after Gaussian blur processing.

Step10 : Median blur

Step11 : Bilateral filter

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files in the "OPENCV" folder: 1.py, 02.py, 03.py, 04.py, 05.py, 06.py, 07.py, 08.py, 09.py, 10.py, 11.py, 12.py, 13.py, 14.py, 15.py, and leaf.jpg.
- Code Editor:** The file "11.py" is open, containing the following Python code:

```
1 import cv2
2
3 img = cv2.imread('C:\\Users\\Arulmani\\OneDrive\\Pictures\\Naruto.jpg')
4
5 resized = cv2.resize(img, (500, 500))
6
7 d = 7
8 signacolor = 100
9 sigmaspace = 100
10
11 b = cv2.bilateralFilter(resized,
12
13 cv2.imshow("Original", resized)
14 cv2.imshow("Bilateral Filter")
15 cv2.waitKey(0)
16 cv2.destroyAllWindows()
```
- Terminal:** The terminal window shows the command line history:

```
PS C:\Users\Arulmani\Documents\Project
```
- Output:** The output window shows the execution of the script, with multiple instances of "python.exe" and "python" appearing in the log.
- Visual Output:** Two windows are displayed:
 - A window titled "Bilateral Filter" showing the original image of Naruto with his arms crossed.
 - A window titled "Original" showing the same image of Naruto with his arms crossed.
- Status Bar:** The status bar at the bottom right indicates: Ln 3, Col 70, Spaces: 4, UTF-8, CRLF, Python, 3.13.3.

Step12 : Edge detection

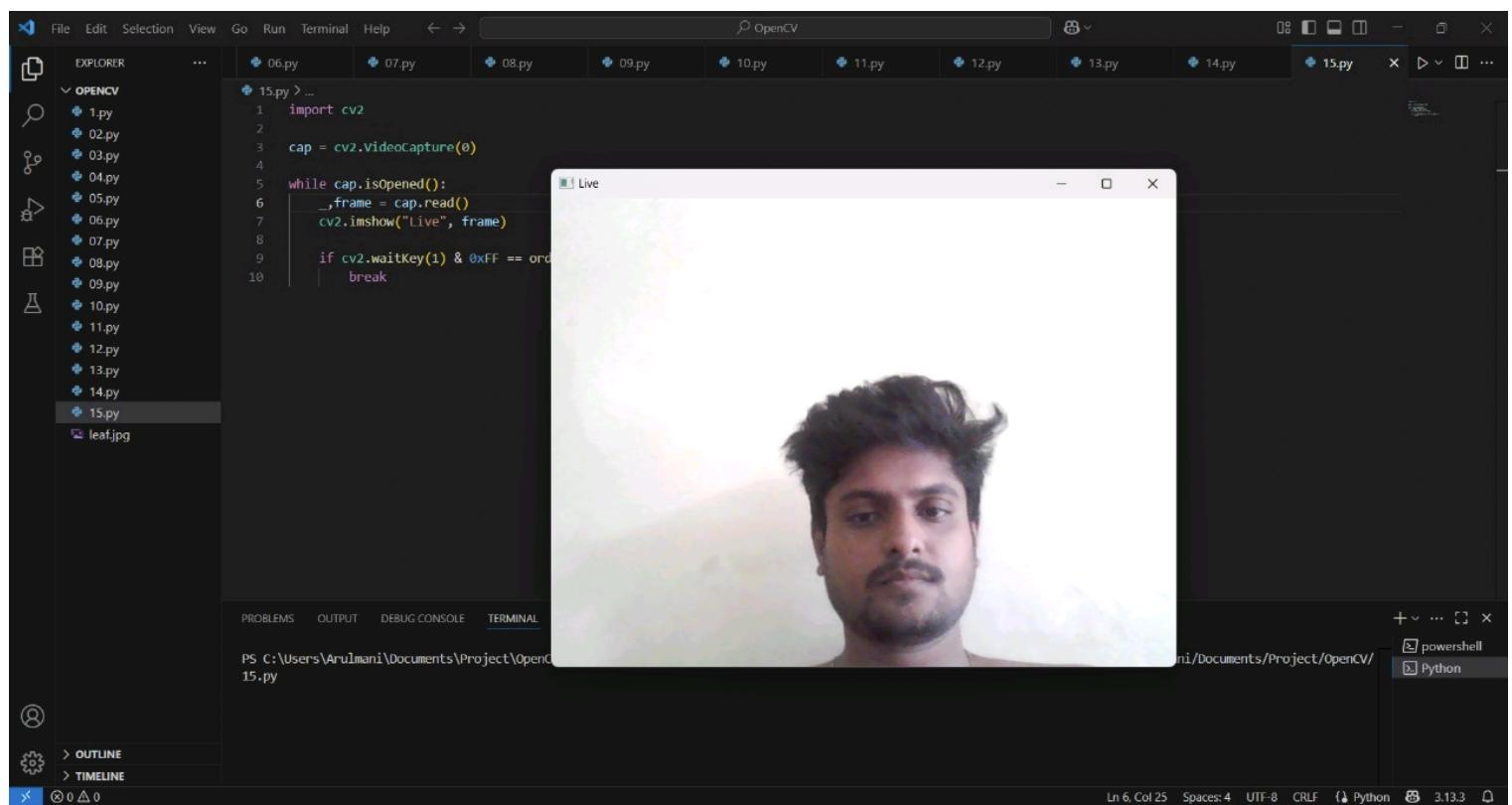
The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files in the "OPENCV" folder: 1.py, 2.py, 3.py, 4.py, 5.py, 6.py, 7.py, 8.py, 9.py, 10.py, 11.py, 12.py, 13.py, 14.py, 15.py, and leaf.jpg.
- Code Editor:** The file "12.py" is open, containing the following Python code:

```
import cv2
import numpy as np

img = cv2.imread('C:\\Users\\Arulmani\\OneDrive\\Pictures\\Naruto.jpg',0)
min_thresh = 100
max_thresh = 200
edge = cv2.Canny(img, min_thresh, max_thresh)
cv2.imshow("Original", img)
cv2.imshow("Canny Edge", edge)
cv2.waitKey(0)
cv2.destroyAllWindows()
```
- Terminal:** The terminal window shows the command PS C:\Users\Arulmani\Documents\Project\OpenCV & c:/Users/Arulmani\ and the output of the Python script.
- Output:** The output window shows the command PS C:\Users\Arulmani\Documents\Project\OpenCV & c:/Users/Arulmani\ and the output of the Python script.
- Windows:** Two windows are displayed: "Canny Edge" showing the edge-detected version of the Naruto image, and "Original" showing the original grayscale image of Naruto.

Step13 : Accessing the webcam



A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "OpenCV". The left sidebar shows an "EXPLORER" view with files like 1.py through 15.py and leaf.jpg. The main editor area displays Python code for accessing a webcam:

```
15.py > ...
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 while cap.isOpened():
6     _,frame = cap.read()
7     cv2.imshow("Live", frame)
8
9     if cv2.waitKey(1) & 0xFF == ord('q'):
10        break
```

The terminal at the bottom shows the command PS C:\Users\Arulmani\Documents\Project\OpenCV\15.py. A preview window titled "Live" shows a video feed of a person's face. The status bar at the bottom right indicates Ln 6, Col 25, Spaces:4, UTF-8, CRLF, Python, and 3.13.3.