

**COMPUTER VISION USING OPENCV**

**23SDAD10E**

**LAB WORKBOOK**

**Course Coordinator :** **Dr. Sandeep Reddy Chitreddy**

**STUDENT ID: 2310080078 ACADEMIC YEAR: 2025-26**

**STUDENT NAME: EELAPANTI MYTHRI**

**SEM/YEAR: 3RD year Odd sem**

**DEPARTMENT: AI & DS**

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **List of Experiments** | **Session** | **Page No.** |
| 1 | Experiment 1: |  |  |
| 2 | Experiment 2 : |  |  |
| 3 | Experiment 3 : |  |  |
| 4 | Experiment 4 : |  |  |
| 5 | Experiment 5 : |  |  |
| 6 | Experiment 6 : |  |  |
| 7 | Experiment 7 : |  |  |
| 8 | Experiment 8 : |  |  |
| 9 | Experiment 9 : |  |  |
| 10 | Experiment 10 : |  |  |
| 11 | Experiment 11 : |  |  |
| 12 | Experiment 12 : |  |  |
| 13 | Experiment 13 : |  |  |
| 14 | Experiment 14 : |  |  |
| 15 | Experiment 15 : |  |  |
| 16 | Experiment 16 : |  |  |
| 17 | Experiment 17 : |  |  |
| 18 | Experiment 18 : |  |  |
| 19 | Experiment 19 : |  |  |
| 20 | Experiment 20 : |  |  |
| 21 | Experiment 21 : |  |  |
| 22 | Experiment 22 : |  |  |
| 23 | Experiment 23 : |  |  |
| 24 | Experiment 24 : |  |  |
| 25 | Experiment 25 : |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S. No** | **Date** | **Experiment Name** | **Pre-**  **Lab**  **(10M)** | **In-Lab (25M)** | | | **Post-**  **Lab**  **(10M)** | **Viva**  **Voce**  **(5M)** | **Total**  **(50M)** | **Faculty**  **Signature** |
| **Program/**  **Procedure**  **(5M)** | **Data and**  **Results**  **(10M)** | **Analysis &**  **Inference**  **(10M)** |
| **1.** |  | Experiment 1 : |  |  |  |  |  |  |  |  |
| **2.** |  | Experiment 2 : |  |  |  |  |  |  |  |  |
| **3.** |  | Experiment 3 : |  |  |  |  |  |  |  |  |
| **4.** |  | Experiment 4 : |  |  |  |  |  |  |  |  |
| **5.** |  | Experiment 5 : |  |  |  |  |  |  |  |  |
| **6.** |  | Experiment 6 : |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S. No** | **Date** | **Experiment Name** | **Pre-**  **Lab**  **(10M)** | **In-Lab (25M)** | | | **Post-**  **Lab**  **(10M)** | **Viva**  **Voce**  **(5M)** | **Total**  **(50M)** | **Faculty**  **Signature** |
| **Program/**  **Procedure**  **(5M)** | **Data and**  **Results**  **(10M)** | **Analysis &**  **Inference**  **(10M)** |
| **7.** |  | Experiment 7 : |  |  |  |  |  |  |  |  |
| **8.** |  | Experiment 8 : |  |  |  |  |  |  |  |  |
| **9.** |  | Experiment 9 : |  |  |  |  |  |  |  |  |
| **10.** |  | Experiment 10 : |  |  |  |  |  |  |  |  |
| **11.** |  | Experiment 11 : |  |  |  |  |  |  |  |  |
| **12.** |  | Experiment 12 : |  |  |  |  |  |  |  |  |
| **`13.** |  | Experiment 13 : |  |  |  |  |  |  |  |  |
| **14.** |  | Experiment 14 : |  |  |  |  |  |  |  |  |
| **15.** |  | Experiment 15 : |  |  |  |  |  |  |  |  |
| **16.** |  | Experiment 16 : |  |  |  |  |  |  |  |  |
| **17.** |  | Experiment 17 : |  |  |  |  |  |  |  |  |
| **18.** |  | Experiment 18 : |  |  |  |  |  |  |  |  |
| **19.** |  | Experiment 19 : |  |  |  |  |  |  |  |  |
| **20.** |  | Experiment 20 : |  |  |  |  |  |  |  |  |
| **21.** |  | Experiment 21 : |  |  |  |  |  |  |  |  |
| **22.** |  | Experiment 22 : |  |  |  |  |  |  |  |  |
| **23.** |  | Experiment 23: |  |  |  |  |  |  |  |  |
| **24.** |  | Experiment 24: |  |  |  |  |  |  |  |  |
| **25.** |  | Experiment 25: |  |  |  |  |  |  |  |  |

Computer Vision using OpenCV

Method 1 to read images:

import matplotlib.pyplot as plt

import matplotlib.image as img

image=img.imread('Cat.jpg')

plt.imshow(image[:,:,:])

plt.show()

To read image from terminal using Pillow Library, first create a virtual environment named mycvenv using below command:

python3 -m venv mycvenv

Activate virtual environment using one of below commands:

For Mac: source myenv/bin/activate

For Windows: .\myenv\Scripts\activate

Install Pillow library using below code:

pip3 install Pillow

Now place the below code in a python file and run it using python3 from terminal

from PIL import Image

img=Image.open('Cat.jpg')

img.show()

Install Opencv using below command

pip install opencv-python

import cv2

image1=cv2.imread('Cat.jpg')

type(image1)

cv2.imshow("Image",image1)

cv2.waitKey(0)

cv2.destroyAllWindows()

Home Assignment 1: Place KLH logo on the Cat image to the top right corner

import cv2

# Read the main image and logo

image = cv2.imread('Cat.jpg')

logo = cv2.imread('klh.jpg')

# Resize logo if needed (optional, here resized to 200x100)

logo = cv2.resize(logo, (200, 100))

# Get dimensions

image\_height, image\_width = image.shape[:2]

logo\_height, logo\_width = logo.shape[:2]

# Define the position for the top-right corner

top\_left\_y = 0

top\_left\_x = image\_width - logo\_width

# Place the logo on the image

image[top\_left\_y:top\_left\_y + logo\_height, top\_left\_x:top\_left\_x + logo\_width] = logo

# Show the result

cv2.imshow("Image with Logo", image)

cv2.waitKey(0)

cv2.destroyAllWindows()

Reading video from Webcam:

#Reading video from Webcam:

import cv2

cap= cv2.VideoCapture('Earth.mp4')

if (cap.isOpened() == False):

    print("Error opening video file")

while (cap.isOpened()):

    ret, frame = cap.read()

    if ret == True:

        cv2.imshow('Frame', frame)

        if cv2.waitKey(25) & 0xFF == ord('q'):

            break

    else:

        break

cap.release()

cv2.destroyAllWindows()

Home Assignment 2: Place KLH logo on the video recorded using Webcam

import cv2

# Open webcam

video = cv2.VideoCapture(0)

# Read logo image

logo = cv2.imread('klh.jpg')

logo = cv2.resize(logo, (200, 100))  # Resize if needed

if not video.isOpened():

    print("Error reading video stream")

    exit()

# Get video frame dimensions

frame\_width = int(video.get(cv2.CAP\_PROP\_FRAME\_WIDTH))

frame\_height = int(video.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))

# Resize logo to fit frame better, optional

logo\_height, logo\_width = logo.shape[:2]

# Top-right corner placement

top\_left\_y = 0

top\_left\_x = frame\_width - logo\_width

# Video writer setup

size = (frame\_width, frame\_height)

result = cv2.VideoWriter('filename.avi', cv2.VideoWriter\_fourcc(\*'MJPG'), 10, size)

while True:

    ret, frame = video.read()

    if ret:

        # Overlay the logo on the frame

        frame[top\_left\_y:top\_left\_y + logo\_height, top\_left\_x:top\_left\_x + logo\_width] = logo

        # Write and show the frame

        result.write(frame)

        cv2.imshow('Frame with Logo', frame)

        # Exit on 's' key

        if cv2.waitKey(1) & 0xFF == ord('s'):

            break

    else:

        break

# Release everything

video.release()

result.release()

cv2.destroyAllWindows()

Running mp4 video through OpenCV

import cv2

cap= cv2.VideoCapture('Earth.mp4')

if (cap.isOpened() == False):

    print("Error opening video file")

while (cap.isOpened()):

    ret, frame = cap.read()

    if ret == True:

        cv2.imshow('Frame', frame)

        if cv2.waitKey(25) & 0xFF == ord('q'):

            break

    else:

        break

cap.release()

cv2.destroyAllWindows()

Resizing using OpenCV:

import cv2

img = cv2.imread('messi5.jpg')

half = cv2.resize(img, (0, 0), fx = 0.4, fy = 0.4)

bigger = cv2.resize(img, (1024, 1024))

print(img.shape)

print(half.shape)

print(bigger.shape)

cv2.imshow("Image1",img)

cv2.waitKey(0)

cv2.destroyAllWindows()

cv2.imshow("Image2",half)

cv2.waitKey(0)

cv2.destroyAllWindows()

cv2.imshow("Image3",bigger)

cv2.waitKey(0)

cv2.destroyAllWindows()

Live Video Capture using IP Webcam

import cv2

import numpy as np

url = "http://192.168.0.101:8080/video"

cp = cv2.VideoCapture(url)

while(True):

    camera, frame = cp.read()

    if frame is not None:

        cv2.imshow("Frame", frame)

    q = cv2.waitKey(1)

    if q==ord("q"):

        break

cv2.destroyAllWindows()

Image Rotation:

import cv2

img = cv2.imread('messi5.jpg')

# dividing height and width by 2 to get the center of the image

height, width = img.shape[:2]

# get the center coordinates of the image to create the 2D rotation matrix

center = (width / 2, height / 2)

# using cv2.getRotationMatrix2D() to get the rotation matrix

rotate\_matrix = cv2.getRotationMatrix2D(center=center, angle=80, scale=0.5) #####change angle 45,80....

print(rotate\_matrix)

# rotate the image using cv2.warpAffine

rotated\_image = cv2.warpAffine(src=img, M=rotate\_matrix, dsize=(width, height))

cv2.imshow("Image",img)

print("Image Rotation")

cv2.imshow("RImage",rotated\_image)

cv2.waitKey(0)

Image Transformations and Flipping

import cv2

import numpy as np

image = cv2.imread('Cat.jpg')

rows, cols = image.shape[:2]

# Shear matrix (x-shear)

shear\_matrix = np.float32([[1, 0.5, 0],

                           [0,   1, 0]])

# Apply affine transform

sheared\_image = cv2.warpAffine(image, shear\_matrix, (int(cols \* 1.5), rows))

h\_flip = cv2.flip(image, 1)

v\_flip = cv2.flip(image, 0)

both\_flip = cv2.flip(image, -1)

cv2.imshow("Original", image)

cv2.waitKey(0)

cv2.imshow("Sheared (Skewed)", sheared\_image)

cv2.waitKey(0)

cv2.imshow("HFlip", h\_flip)

cv2.waitKey(0)

cv2.imshow("VFlip", v\_flip)

cv2.waitKey(0)

cv2.imshow("HVFlip", both\_flip)

cv2.waitKey(0)

cv2.destroyAllWindows()

Image Gray scaling

import cv2

img = cv2.imread('Cat.jpg', 0)  # Load in grayscale

cv2.imshow('Grayscale Cat', img)  # Window title and image

cv2.waitKey(0)  # Wait for a key press

cv2.destroyAllWindows()  # Close all OpenCV windows

Increase Brightness without using OpenCV:

import cv2

import numpy as np

# Read grayscale image

img = cv2.imread('Cat.jpg', 0)

# Add 20 to each pixel but cap at 255

img1 = np.clip(img + 20, 0, 255).astype(np.uint8)

# Show original and modified images

cv2.imshow("Original", img)

cv2.imshow("Increased Brightness", img1)

cv2.waitKey(0)

cv2.destroyAllWindows()

import cv2

import numpy as np

# Load grayscale image

img = cv2.imread('Cat.jpg', 0)

# Create a new image to store the result

bright\_img = np.zeros(img.shape, dtype=np.uint8)

# Get image dimensions

rows, cols = img.shape

# Increase brightness by 20 using 2 for loops

for i in range(rows):

    for j in range(cols):

        new\_value = img[i, j] + 20

        if new\_value > 255:

            new\_value = 255

        bright\_img[i, j] = new\_value

# Show the result

cv2.imshow('Brightened Image', bright\_img)

cv2.waitKey(0)

cv2.destroyAllWindows()

Contrast of an Image:

import cv2

import numpy as np

def contrast\_stretch(img):

    a, b = 20, 230  # new range

    rmin, rmax = np.min(img), np.max(img)

    stretched = (img - rmin) \* ((b - a) / (rmax - rmin)) + a

    return np.uint8(stretched)

img = cv2.imread('Cat.jpg', 0)

cv2.imshow("Original", img)

cv2.waitKey(0)

stretched\_img = contrast\_stretch(img)

cv2.imshow("Contrast Stretched", stretched\_img)

cv2.waitKey(0)

cv2.destroyAllWindows()

Binary Representation:

import cv2

img = cv2.imread('Cat.jpg', 0)

\_, thresh = cv2.threshold(img, 127, 255, cv2.THRESH\_BINARY)

cv2.imshow("Thresholded", thresh)

cv2.waitKey(0)

cv2.destroyAllWindows()

BGR to HSV

import cv2

# Read the image from file

image = cv2.imread('Cat.jpg')

# Convert the image from BGR to HSV color space

hsv\_img = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

hsv\_img

Tracking using Color in HSV Domain:

import cv2 as cv

import numpy as np

cap = cv.VideoCapture(0)

if not cap.isOpened():

    print("Cannot open camera")

    exit()

while True:

    ret, frame = cap.read()

    if not ret:

        print("Can't receive frame. Exiting ...")

        break

    hsv = cv.cvtColor(frame, cv.COLOR\_BGR2HSV)

    lower\_blue = np.array([110, 50, 50])

    upper\_blue = np.array([130, 255, 255])

    mask = cv.inRange(hsv, lower\_blue, upper\_blue)

    res = cv.bitwise\_and(frame, frame, mask=mask)

    cv.imshow('frame', frame)

    cv.imshow('mask', mask)

    cv.imshow('res', res)

    if cv.waitKey(1) & 0xFF == 27:  # ESC key

        break

cap.release()

cv.destroyAllWindows()

Convolution using filter2D : Negative values allowed

import cv2

import numpy as np

a=np.array([[1,2,3],[4,5,6],[7,8,9]],dtype=np.uint8)

b=np.array([[-1,0,1],[-2,0,2],[-1,0,1]],dtype=np.float32)

print(a)

print(b)

b1=cv2.flip(b,-1)

c=cv2.filter2D(src=a,ddepth=cv2.CV\_32F,kernel=b1,borderType=cv2.BORDER\_CONSTANT)

print(c)