

**MANIPAL UNIVERSITY JAIPUR**

School of Computer Science & Engineering

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING**

**AI3231 Computer Vision & Pattern Recognition Lab [3 1 0 4]**

Lab Course

Session: January 2025 – May 2025

Faculty: Dr. Jaydeep Kishore, Dr. Abhishek Kesarwani, Mr. Ravindra Kumar Soni,

Mr. Harish Sharma, Ms. Rishika Singh

### INTRODUCTION:

This course is offered by the Department of AIML, mainly targeting students who wish to pursue a career in Engineering discipline with AIML specialization. This course objectives to understand and use computating tools for image processing, computer vision and pattern recognition. This course introduces students the fundamentals of image processing tools for and provide student knowledge of image restoration, enhancement, segmentation, feature extraction, segmentation, pattern recognition, and other computer vision related tasks. This course also supports the design, implementation, and inference of advanced technologies of Computer vision for real world applications.

### COURSE OUTCOMES:

At the end of the course, students will be able to

1. **Apply** the concepts of image processing and computer vision using computing tools. [L3 Apply]
2. **Understand** and **apply** methods of color space, image transforms and filtering enhancements using computing tools. [L2 Understand**,** L3 Apply]
3. **Apply** computer vision and pattern recognition related concept for edge detection, feature detection, object detection, segmentation, shape detection and tracking method to real world problem. [L3 Apply]
4. **Suggest** a design of computer vision system for real world application with emphasis on implementation, and performance evaluation. [L6 Create]
5. **SYLLABUS**

Introduction to Computer Vision Tools: MATLAB, OpenCV with Python.

Image Processing: Color Space Conversion, Image Transform, Linear, Non-Linear Filtering, Multi resolution. Feature Detection: Edge Detection, Principal Component Analysis, SHIFT, SURF.

Shape Detection and Segmentation: Watershed Algorithm, Shape Matching.

Object Tracking: Frame Differencing, Optical Flow, Background Subtraction, Object Recognition, Face Detection.

# REFERENCE BOOKS

1. R. C. Gonzalez et al., Digital Image Processing Using MATLAB, (2e), Mc Graw Hill India, 2011

2. G.B García et al., Learning Image Processing with OpenCV, (1e), Packt Publishing, 2015

3. A.F. David and J. Ponce, Computer Vision: A Modern Approach(3e), PHI learning 2015.

4. K. Fukunaga, Introduction to Statistical Pattern Recognition, (2e), Academic press, 2013.

5. C.M. Bishop, Pattern Recognition and Machine Learning (1e), Springer, 2011.

# PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES

**[PO.1] Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**[PO.2] Problem Analysis**: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**[PO.3] Design/Development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**[PO.4] Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**[PO.5] Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**[PO.6] The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**[PO.7] Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**[PO.8] Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practices.

**[PO.9] Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

**[PO.10] Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions

**[PO.11] Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**[PO.12] Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

1. **PROGRAM SPECIFIC OUTCOMES (PSO):** At the end of the B Tech CSE AIML program, the student:
2. Graduates will be able to examine the applications of Artificial Intelligence and Machine Learning in real-life problems.
3. Graduates will be able to design and implement intelligent systems for multidisciplinary problems.
4. **Assessment Plan:**

|  |  |  |
| --- | --- | --- |
| Criteria | Description | Maximum Marks |
| Internal Assessment (Summative) | Assignment / File | 20 |
| Continuous Evaluation | 20 |
| MOOC Course (4 week) | 10 |
| Internal Viva | 10 |
| End Term Exam (Summative) | End Term Exam (Closed Book)  Execution + Writeup + Viva + File Submission | 40 |
|  | Total | 100 |
| Attendance (Formative) | A minimum of 75% Attendance is required to be maintained by a student to be qualified for taking up the End Semester examination. The allowance of 25% includes all types of leaves including medical leaves. | |
| Homework/ Assignment/ Activity Assignment (Formative) | There are situations where a student may have to work in home, especially before a flipped classroom. Although these works are not graded with marks. However, a student is expected to participate and perform these assignments with full zeal since the activity/ flipped classroom participation by a student will be assessed and marks will be awarded. | |

1. **LECTURE PLAN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lec.**  **No** | **Topics** | **Session Outcome** | **Mode of Delivery** | **CO** | **Mode Of Assessing CO** |
| 1 | Perform basic Image Handling and Processing operations on the image. | Understanding the basic of Computer Vision and Image processing | Practical | AI3231.1 | Viva, Assignment, MTE / ETE |
| 2 | Geometric Transformation | Understanding the fundamentals of transformation on images | Practical | AI3231.2 | Viva, Assignment, MTE / ETE |
| 3 | Compute Homography Matrix | Understanding the Homography concept | Practical | AI3231.2 | Viva, Assignment, MTE / ETE |
| 4 | Perspective Transformation | Understanding the perspective transformation | Practical | AI3231.2 | Viva, Assignment, MTE / ETE |
| 5 | Camera Calibration | Calibrate the camera and extract the intrinsic and extrinsic parameters of the camera | Practical | AI3231.1 | Viva, Assignment, MTE / ETE |
| 6 | Compute Fundamental Matrix | Calculate the fundamental matrices required for Image Processing | Practical | AI3231.1 | Viva, Assignment, MTE / ETE |
| 7 | Edge Detection, Line Detection and Corner Detection | Apply edge and line detection algorithm on sample images and compare their output | Practical | AI3231.3 | Viva, Assignment, MTE / ETE |
| 8 | SIFT Feature descriptor | To Understand the SHIFT algorithm and compare its output | Practical | AI3231.3 | Viva, Assignment, MTE / ETE |
| 9 | SURF and HOG feature descriptor | To understand and compute the concepts of SURF and HOG algorithm | Practical | AI3231.3 | Viva, Assignment, MTE / ETE |
| 10 | Project based on Computer Vision Applications | Apply CVPR concept to solve real world problem | Practical | AI3231.4 | Viva, Assignment, MTE / ETE |

1. **Course Articulation Matrix: (Mapping of COs with POs)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** |  | **CORRELATION WITH PROGRAM OUTCOMES** | | | | | | | | | | | | **CORRELATION WITH PROGRAM SPECIFIC OUTCOMES** | | | |
|  |  | **PO**  **1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | | **PSO 1** | **PSO 2** |
| AI3231.1 | Apply the concepts of image processing and computer vision using computing tools.  [L3 Apply] | 2 | 2 | 3 | 2 | 2 | 3 | 1 | 2 | 1 | 2 | 1 | 3 | | 3 | 2 |
| AI3231.2 | Understand and apply methods of color space, image transforms and filtering enhancements using computing tools.  [L2 Understand, L3 Apply] | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 3 | 3 |
| AI3231.3 | Apply computer vision and pattern recognition related concept for edge detection, feature detection, object detection, segmentation, shape detection and tracking method to real world problem.  [L3 Apply] | 3 | 3 | 3 | 2 | 2 | 1 | 3 | 1 | 2 | 1 | 1 | 3 | | 3 | 3 |
| AI3231.4 | Suggest a design of computer vision system for real world application with emphasis on implementation, and performance evaluation.  [L6 Create] | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 3 | 3 | 3 | | 3 | 3 |

1. **Low Correlation; 2- Moderate Correlation; 3- Substantial Correlation**

|  |  |  |
| --- | --- | --- |
| Course Coordinator | Student Representative | Head of the Department |

**Assignment No. 1 Date: Jan 2024**

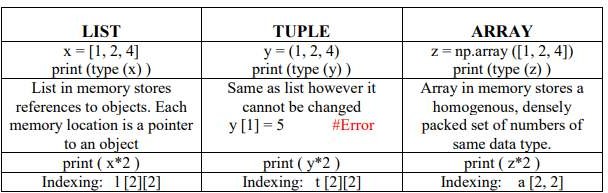
**Aim:** Perform basic Image Handling and processing operations on the image.

**Objectives:** The objective of this lab is to introduce the student to OpenCV, especially for image processing.

* 1. Reading an image in python
  2. Convert Images to another format
  3. Convert an Image to Grayscale
  4. Play a video file

### Some Useful Commands:

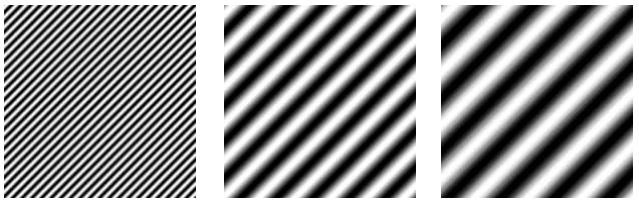
1. To slice a 2D array: x = y [row\_start: row\_ end, col\_start: col\_end]
2. To create a 2D array of zeros using NumPy: my\_array = numpy.zeros ((row, columns), dtype=numpy.uint8)
3. To create a 2D array of ones using NumPy: my\_array = numpy.ones ((row, columns), dtype=numpy.uint8)
4. To check the size of a 2D array: size = numpy.shape(my\_array)
5. To join a sequence of arrays along an existing axis: F = np.concatenate ((a, b), axis=0);
6. To assemble an array from nested lists of blocks. img = np.block ([[np.ones ((2, 2)), np.zeros ((2, 3))], [np.zeros ((2, 2)), np.ones ((2, 3))]]) Note: all the input array dimensions except for the concatenation axis must match exactly
7. Reading an image using OpenCV: my\_image = cv2.imread(<test\_image.jpg=,0) The second argument determines whether the image is read as a grayscale image or a colored image. 0 is used for reading an image as grayscale and while 1 is used for reading in color. If no argument is passed then the image is read as is.
8. Displaying an image using OpenCV: cv2.imshow (<Title of the window=, my\_image). Two more commands that need to be used while displaying an image are cv2.waitKey(x), cv2.destroyAllWindows (). The waitKey () function waits for a key being pressed for x number of milliseconds. If 0 is passed to waitKey () as an argument, it will wait indefinitely for a key press. cv2.destroyAllWindows () closes are the open image windows.
9. Writing an image to disk: cv2.imwrite(<image\_name.jpg=, my\_image)



**Lab Exercise**

The combination of Python (the language), Numpy (the numerical array lib), SciPy (scientific libs), and Matplotlib (the graphical plot lib) will serve as our computational basis to learn image processing and computer vision. Where possible and needed we will use other libraries as well.

1. Reading, displaying, and writing an image using OpenCV.
2. Convert the image to another format using OpenCV.
3. Perform the image resizing using OpenCV.
4. Convert a colored image into a grayscale image using OpenCV.
5. Scaling, rotation and shifting operation on the image using OpenCV.
6. Play a video using OpenCV.
7. Using the following formula f (i, j) = sin (2 π f (i + j)) where i and j indices of a pixel, draw an image with different frequencies (input from the user).



### Reference:

1. [OpenCV Python Tutorial - GeeksforGeeks](https://www.geeksforgeeks.org/opencv-python-tutorial/)
2. [Practical Computer Vision using OpenCV and Python (Basics) | by crossML](https://medium.com/crossml/practical-computer-vision-using-opencv-and-python-basics-cb4907ef694e) engineering | crossml | Medium

**Assignment No. 2 Date: February 2024**

**Aim:** Geometric Transformation

**Objectives:** The objective of this lab is to introduce Geometric Transformation and apply that to images.

* 1. Affine Transformation
  2. Rotation, Translation and Scaling Transformation

### Some Useful Commands:

1. cv2.resize()
2. cv2.warpAffine()
3. cv2.getRotationMatrix2D()
4. cv2.warpPerspective()
5. cv2.getAffineTransform()
6. cv2.getPerspectiveTransform()

### Lab Exercise:

1. Perform scaling, rotation and shifting operations on an image using OpenCV().
2. Perform image reflection on an image using OpenCV().
3. Perform Image shearing on an image using OpenCV().
4. Apply the affine transformation on an image using OpenCV().

### Reference:

1. [Geometric Transformations in Python using OpenCV | coseries](https://www.coseries.com/geometric-transformations-in-python-using-opencv/)
2. [Geometric Transformation OpenCV Python (etutorialspoint.com)](https://www.etutorialspoint.com/index.php/327-geometric-transformation-opencv-python)
3. [Image Geometric Transformation In Numpy and OpenCV | by Daryl Tan | Towards Data](https://www.geeksforgeeks.org/opencv-python-tutorial/) [Science](https://medium.com/crossml/practical-computer-vision-using-opencv-and-python-basics-cb4907ef694e)
4. Geometric Transformations of Images using OpenCV|Image Processing Part−1 | by Ravjot Singh | Analytics Vidhya | Medium
5. Image Transformations using OpenCV in Python − Python Code (thepythoncode.com)

# Assignment No. 3 Date: Feb 2024

**Aim:** Perspective Transformation

### Objectives:

1. Demonstrate the Perspective Transformation in the image.

### Lab Exercise

* 1. For **perspective transformation**, you need a 3x3 transformation matrix. Straight lines will remain straight even after the transformation. To find this transformation matrix, you need 4 points on the input image and corresponding points on the output image. Among these 4 points, 3 of them should not be collinear. Then the transformation matrix can be found by the function **cv2.getPerspectiveTransform()**. Then apply **cv2.warpPerspective()** with this 3x3 transformation matrix.

### Code:

*import cv2*

*import numpy as np*

*from matplotlib import pyplot as plt*

*img = cv2.imread('nature.jpg') rows,cols,ch = img.shape*

*pt1 = np.float32([[50,65],[370,52],[30,387],[390,390]])*

*pt2 = np.float32([[0,0],[310,0],[0,310],[310,310]])*

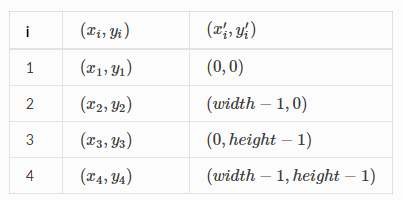
*matrix\_aff = cv2.getPerspectiveTransform(pt1,pt2)*

*dst = cv2.warpPerspective(img,matrix\_aff,(cols,rows))*

*plt.subplot(121),plt.imshow(img),plt.title('Input') plt.subplot(122),plt.imshow(dst),plt.title('Output')*

*plt.show()*

* 1. Write a function <*perspective transform(f, x1, y1, x2, y2, x3, y3, x4, y4, width, height*)” that warps a quadrilateral with vertices at (x1,y1), (x2,y2), (x3,y3) and (x4,y4) to a new image of given width and height. The mapping is given in the following table:

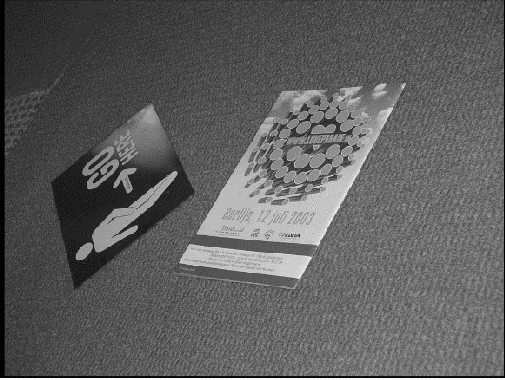


Calculation of the parameters of the perspective transform should be done by your code (do not use the code from OpenCV or other sources).

Be sure that your code works with:

* 1. image of unequal width and height
  2. scalar and color images

In your report present the code and the formulas on which it is based together with some examples showing that it works as intended. You should at least be able to redo the following example (make a reasonable choice for width and height of the new image assuming the flyer is of A4 shape).



**Assignment No. 4 Date: Feb 2024**

**Aim:** Compute Fundamental Matrix

### Objectives:

1. Find fundamental matrix, epipoles, epipolar lines.
2. Plot epipole lines on the images.
3. Find the projection matrix of the second camera position using the fundamental matrix.

### Lab Exercise

1. Compute the Fundamental Matrix using the Least Square Minimization
2. Compute the Fundamental Matrix using the 8-point algorithm.
3. Plotting epipole lines using the Fundamental Matrix computed in (1).

### References:

1. [Python OpenCV: Epipolar Geometry - GeeksforGeeks](https://www.etutorialspoint.com/index.php/327-geometric-transformation-opencv-python)
2. [OpenCV: Epipolar Geometry](https://www.geeksforgeeks.org/opencv-python-tutorial/)
3. [Epipolar Geometry 4 OpenCV-Python Tutorials beta documentation (opencv24-](https://medium.com/crossml/practical-computer-vision-using-opencv-and-python-basics-cb4907ef694e) python-tutorials.readthedocs.io)

### Learning Outcomes:

**Signature of Faculty Member with date:**

**Assignment No. 5 Date: March 2024**

**Aim:** Edge Detection, Line Detection and Corner Detection

### Objectives:

1. Find the edges in the image
2. Corner detection with Harris Corner Detector
3. Line detection

### Lab Exercise

1. Compute the edge detection using Sobel, Prewitt and canny operator.
2. Implement the Harris Corner detector algorithm to determine the corner in the image.
3. Implement the Harris Corner Detector algorithm without the inbuilt OpenCV() function.
4. Detect the line using Hough Transform

**OpenCv functions:** [*cv.Sobel,*](#gacea54f142e81b6758cb6f375ce782c8d) *cv2.filter2D,cv2.cornerHarris(), cv2.HoughLines()*

### References:

1. OpenCV: Image Gradients
2. [OpenCV: Canny Edge Detection](https://www.coseries.com/geometric-transformations-in-python-using-opencv/)
3. [OpenCV: Harris Corner Detection](https://www.etutorialspoint.com/index.php/327-geometric-transformation-opencv-python)
4. [Harris Corner Detector implementation in python 3 Muthukrishnan](https://www.geeksforgeeks.org/opencv-python-tutorial/)
5. [Python | Corner detection with Harris Corner Detection method using OpenCV -](https://medium.com/crossml/practical-computer-vision-using-opencv-and-python-basics-cb4907ef694e) GeeksforGeeks
6. OpenCV: Hough Line Transform
7. Line detection in python with OpenCV | Houghline method - GeeksforGeeks

**Assignment No. 6 Date: March 2024**

**Aim:** SIFT feature descriptors

### Objectives:

1. To understand the concepts of SIFT algorithm
2. To find the key points and descriptors

### Lab Exercise

1. Write a program to compute the SIFT feature descriptors of the image.
2. Write a program to generate a panorama image using SIFT feature descriptor.

**OpenCV functions:** *cv.SIFT\_create(), sift.detect,* [*cv.drawKeypoints*](#ga5d2bafe8c1c45289bc3403a40fb88920)*, sift.detectAndCompute*

### References:

1. Original paper of SIFT [ijcv04.pdf (ubc.ca)](https://www.coseries.com/geometric-transformations-in-python-using-opencv/)
2. [OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform)](https://www.etutorialspoint.com/index.php/327-geometric-transformation-opencv-python)
3. [SIFT Interest Point Detector Using Python - OpenCV - GeeksforGeeks](https://www.geeksforgeeks.org/opencv-python-tutorial/)
4. [GitHub - rohangupta/homography: Implementing Panorama (Image Warping) in](https://medium.com/crossml/practical-computer-vision-using-opencv-and-python-basics-cb4907ef694e) python using OpenCV
5. Implementing SIFT in Python. There’s a lot of content about SIFT… | by Sam Lerner |

Medium

### Learning Outcomes:

### **Signature of Faculty Member with date:**

**Assignment No. 7 Date: April 2024**

**Aim:** SURF and HOG feature descriptors

### Objectives:

1. To understand the concepts of SURF and HOG algorithm
2. To compute the SURF and HOG features

### Lab Exercise

1. Write a program to compute the SURF feature descriptors of the image.
2. Write a program to compute the HOG feature descriptors of the image.
3. Write a program to detect the pedestrians in an image using HOG.

### References:

1. OpenCV: Introduction to SURF (Speeded-Up Robust Features)
2. Histogram of Oriented Gradients explained using OpenCV (learnopencv.com)
3. [Image Recognition using Histogram of Oriented Gradients (HOG) Descriptor](https://www.coseries.com/geometric-transformations-in-python-using-opencv/#%3A~%3Atext%3D%20Writing%20the%20Python%20Script%20for%20Image%20Recognition%2Carrange%20our%20data%20and%20labels%20properly%2C...%20More%20) [(debuggercafe.com)](https://www.etutorialspoint.com/index.php/327-geometric-transformation-opencv-python#%3A~%3Atext%3D%20Writing%20the%20Python%20Script%20for%20Image%20Recognition%2Carrange%20our%20data%20and%20labels%20properly%2C...%20More%20)
4. [Pedestrian Detection using OpenCV-Python - GeeksforGeeks](https://www.geeksforgeeks.org/opencv-python-tutorial/)