IPCV Practical No : 4

18 September 2025 07:55

Aim
To study and implement different image transformation techniques using OpenCV.

Requirements
- Python 3.x
- OpenCV (cv2)
- NumPy (numpy)
- Matplotlib (matplotlib.pyplot)
- Input image

Theory
Image transformation means **changing the shape, size, or orientation of an image**. It is one of the most important parts of image processing because it helps in aligning, resizing, rotating, or changing the viewpoint of an image.
OpenCV provides built-in functions for common transformations:

1. Translation (Shifting)
- Moves an image from one place to another (left, right, up, down).
- Useful when you want to reposition an image in a bigger frame.

2. Scaling (Resizing)
- Changes the size of the image, either enlarging (zoom in) or shrinking (zoom out).
- Used in applications like thumbnail creation, object detection at multiple sizes, etc.

3. Rotation
- Rotates an image around its center or any point.
- Used in document scanners, photo editing apps, and aligning tilted images.

4. Affine Transformation
- Skews or tilts the image but keeps parallel lines parallel.
- Needs 3 points from the original and transformed image.
- Example: turning a square into a parallelogram.

5. Perspective Transformation
- Changes the viewpoint of an image (as if looking from another angle).
- Needs 4 points mapping.
- Example: converting a rectangular document photo into a properly aligned top-down view.

Advantages of Image Transformations
1. Easy manipulation of images (resize, rotate, shift).
2. Helps in pre-processing for machine learning / pattern recognition.
3. Corrects tilted or skewed images.

Limitations of Image Transformations
1. Large transformations can cause image blur or loss of quality.
2. Black borders or empty pixels may appear at image boundaries.
3. Complex transformations may distort the image if not handled carefully.

Steps to Perform
1. Import required libraries (cv2, numpy, matplotlib).
2. Load input image.
3. Perform **translation** using cv2.warpAffine().
4. Perform **scaling** using cv2.resize().
5. Perform **rotation** using cv2.getRotationMatrix2D().
6. Perform **affine transformation** using cv2.getAffineTransform().
7. Perform **perspective transformation** using cv2.getPerspectiveTransform().

8. Display and compare results.

Expected Output

- **Translated Image** → Image shifted left/right/up/down.
- **Scaled Image** → Image zoomed in/out.
- **Rotated Image** → Image rotated by given angle.
- **Affine Image** → Tilted/skewed image (parallelogram effect).
- **Perspective Image** → Transformed to new viewpoint (3D effect).

Conclusion

Image transformations (translation, scaling, rotation, affine, perspective) are useful for **geometric corrections, computer vision, object detection, and augmented reality applications**.