Comprehensive Report: OpenCV Week 1

Gangeshwar Lohar

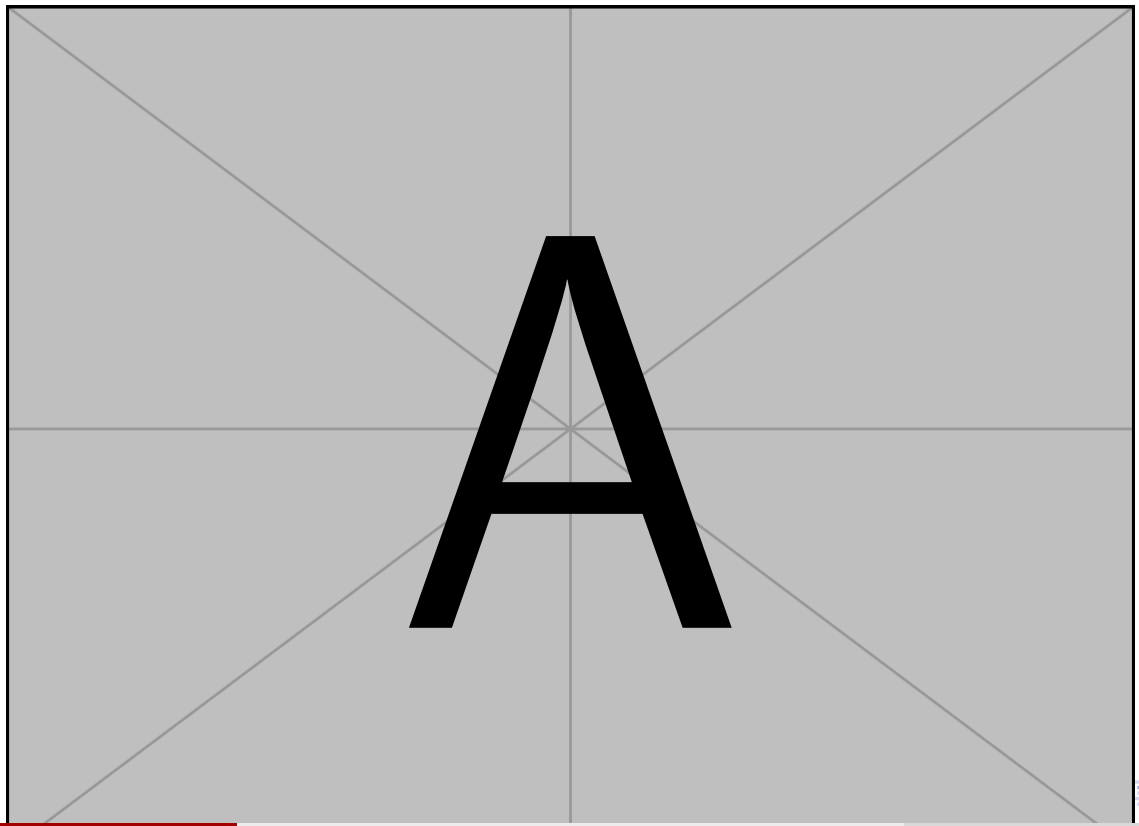
August 21, 2025

Outline

Key Learning: Images are NumPy Arrays!

The most fundamental concept in OpenCV with Python is that images are represented as NumPy arrays.

- **Grayscale Image:** A 2D array of shape ‘(height, width)‘.
- **Color Image:** A 3D array of shape ‘(height, width, 3)‘.
- **Important Note:** OpenCV uses the BGR color channel order by default, not RGB.



Top 5 OpenCV Applications

Our research identified several key areas where OpenCV is critical:

1. Autonomous Vehicles & ADAS
2. Medical Image Analysis
3. Security and Surveillance
4. Robotics and Industrial Automation
5. Augmented Reality (AR)

Code Example: Basic Image Loading

```python
import cv2

# Read an image from disk
# It returns a NumPy array in BGR format
image = cv2.imread('test_image.jpg')

# Convert the image to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Display the image
cv2.imshow('Grayscale Image', gray_image)

# Wait for a key press to close the window
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Listing 1: A simple script to load and display an image.

Summary and Next Steps

This week provided a strong foundation in OpenCV's core functionalities.

Key Takeaways

- OpenCV is a powerful, modular library built around NumPy.
- Understanding the 'highgui' event loop ('waitKey') is crucial for building applications.
- The library has vast applications across many modern industries.

Future Work

The next phase will involve exploring the 'dnn' module for deep learning integration and the 'calib3d' module for 3D reconstruction.