Lab 1: The objective of this lab is to implement preprocessing and feature extraction techniques to analyze an image of a checkerboard.

Learning objectives:

- Utilize existing computer vision libraries (i.e. OpenCV) to:
    - Implement data preprocessing techniques (i.e. augmentation)
    - Implement feature extraction techniques (i.e. SIFT, HOG, FAST)
- Experiment with existing feature extraction techniques and how they are affected by data augmentation

Procedure:

- Download the zip file called *Lab1* and extract it. This zip file contains:
    - An empty python code for you to fill in
    - Two test images for you to practice on
    - This manual for the lab
- You may open the images to view them and see what they look like
- The python code consists of several comments that indicate where you should fill in. Note that there are three functions that you should complete:
    - SIFT (Scale-Invariant Feature Transform)
    - HOG (Histogram of Oriented Gradients)
    - FAST (Features from Accelerated Segment Test)
- For each of the three functions, you are also asked to augment the test image (i.e. rotate/scale it)
- Note that there are existing implementations for the above three techniques in the OpenCV library. **To ensure that you know how to utilize existing Computer Vision libraries, please use the built-in functions to follow the instructions as indicated by the comments. You will then answer the below questions to think critically about the output.**

When you have finished implementing the code, please answer the following questions:

1. For the SIFT technique:
    - How does SIFT visualize the feature matching between two images? How can you use this visualization to verify the correctness of the SIFT algorithm?
    - How robust is the SIFT algorithm to the different sizes of the image?
2. For the HOG technique:
    - How does HOG visualize the features of the image? What features does it focus on (e.g. edges, corners, blobs, etc)?
    - How robust is the HOG technique to the different rotations of the image? Test this technique for a rotation of 0^o, and for two other angles of rotations
3. For the FAST technique:

    - How well does FAST detect the features of the image?
    - How robust is the FAST technique to different angles of rotation? Test this technique for a rotation of 0^o, and for two other angles of rotations.

Deliverables:

- Your code that fills in the blanks as indicated by the comments in *Lab1_empty.py*
- A report that details the results you have found. It should contain the following information:
    - Introduction to the computer vision pipeline
    - A description that highlights your understanding of the data preprocessing and feature extraction techniques
    - Answers to the above questions. Include the results from your code (as specified by the questions)
    - Concluding remark of what you have learned from this lab

Your implemented code should have the following (you will be marked for both):

- Correctness: The code should compile without error
- Style: There should be comments and proper formatting