

# HOMEWORK SPRING 2025

## APPLIED COMPUTER VISION (CMU-AFRICA)

**Release Date:** Tuesday, February 4<sup>th</sup>, 2025

**DUE:** Friday, February 14<sup>th</sup>, 2025, 11:59 PM CAT

1. **Collaboration policy:** You are permitted to collaborate with fellow students on discussing homework problems; however, the submitted work must be composed in your own words and should not be copied from any external sources. It is essential to document the individuals with whom you discussed the assignment by listing their names at the top of the first page of your homework solutions. While we highly recommend using LaTeX to type your report, it is not mandatory.
2. **Submitting your work:** Assignment will be submitted through Canvas. Please ensure that you include all the necessary files to reproduce your output. If you believe a specific environment is required to run your code, kindly submit a readme file detailing the procedures for setting up this environment. Submit a notebook and a report pdf file separately on canvas. Do not zip them!
3. **Getting Help:** Please use office hours and Piazza to ask any questions related to this assignment.
4. **Refrain from using ChatGPT:** These problems are easy and can be easily solved with the use of generative AI's like ChatGPT or Github Copilot. However, you will not learn by using these tools. Not learning will impact your performance outside of the classroom and you will not reflect. Ask questions on Piazza about anything you don't understand, the TA's and instructors will respond to you as fast as possible. Moreover, you will not have access to internet during the exam.

## Learning Objectives

1. To understand and implement fundamental computer vision techniques such as object counting, line detection, image compression, and data augmentation.
2. To apply these techniques to real-world problems and explore their practical significance.
3. To develop proficiency in using OpenCV and other image processing tools for solving computer vision tasks.
4. To analyze and document the impact of different image processing methods on various applications.
5. To enhance problem-solving skills by designing efficient algorithms for image manipulation and enhancement.

# Introduction

In this assignment, we explore key computer vision techniques, including object counting, line detection, image compression, and data augmentation. We will apply these techniques to practical problems and discuss their significance in real-world applications.

## Question 1: Counting Objects - Inventory Management Systems

At a busy logistics warehouse, workers struggle to count packages manually, leading to errors and delays. To solve this, a computer vision system is deployed. Cameras capture images of stacks of boxes, and an object detection algorithm uses edge detection and contour analysis to count them automatically. This improves efficiency, ensuring that shipments are tracked accurately and reducing the risk of lost inventory. You have just been hired as a computer vision engineer and your task is to help them solve this problem. One of the naive ways is to find contours of objects and counting them. In this section, we will be implementing this naive method.

1. Load an image using OpenCV.
2. Convert the image to grayscale for easier processing.
3. Apply Gaussian blur to reduce noise and improve edge detection.
4. Use Canny Edge Detection to extract contours.
5. Find contours in the image and count them as individual objects.



(a) Figure 1



(b) Figure 2

Although the above method might work in some cases, you are expected to explore other image processing techniques to improve the object counting algorithm. We present two types of scenarios, one where all the objects are separated from each other (see Fig. 1) and one where some or all of the objects are touching each other (see Fig. 2). In all cases, the scene background will be white. Equal marks will be awarded for

the ability of the application to count the number of objects in the first (separated) and second (touching) scenarios. We recommend you try to include some **morphological operations in the image processing to improve your counting algorithm**.

## Tasks

1. Implement the naive object counting algorithm above.
2. Experiment with different parameters of the filters you use and various image processing techniques and report how they help improve your counting algorithm.
3. Real world experimentation: Put some items on a table with a plain background, take a picture using your smart phone and count the number of items.
4. Report on the difficulties faced by the naive counting algorithm in real world example and how it could be improved.
5. Suggest other real world scenarios where counting objects could be helpful.

## Question 2: Finding lines on a chessboard and straightening the image - Robot-Assisted Chess Training

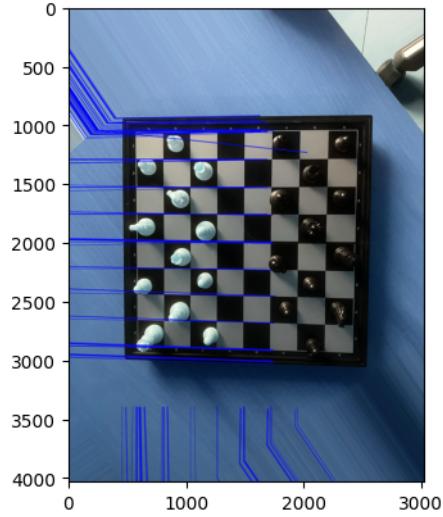
Image quality is a very important input to machine learning models. In this part of the assignment, we will explore how we can automatically calibrate some images to help get better performance from an AI model. After completing the computer vision course, you develop an AI-powered chess coach that needs to analyze board positions from a camera feed. However, due to varying camera angles, the chessboard often appears tilted. To fix this, the system detects the board's grid lines using the Hough Transform and calculates the necessary rotation. By automatically straightening the board, your AI can correctly identify piece positions and suggest optimal moves to players. We will use the following naive algorithm to straighten the chess board.

1. Convert the image to grayscale. (Ofcourse after reading the image).
2. Apply Gaussian Blur and contrast enhancement using CLAHE.
3. Use Canny Edge Detection to identify edges in the image.(Experiment with Sobel)
4. Apply the Hough Transform to detect lines in the image.
5. Store and analyze angles of detected lines to determine the dominant orientations.
6. Apply image rotation to correct the perspective.

In Fig 3 and Fig 4 we can see some of the expected results. You are provided with 3 different images to experiment with.



(a) Figure 3



(b) Figure 4

## Tasks

1. Implement the straightening algorithm above.
2. Experiment with different parameters of the filters you use and various image processing techniques and report how they help improve your straightening algorithm.
3. Real world experimentation: Find an object with which has lines like a square, rotate it and use your implementation to automatically straighten it.
4. Report on the difficulties faced by the naive straightening algorithm in the real world example and how it could be improved.
5. Suggest other real world scenarios where aligning and image can be helpful.

## Question 3: Image Compression with Fourier Transform

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or **Frequency domain**, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the **spatial domain image**.

The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression. Fourier transform breaks down an image into its frequency domain. Within this domain, we can eliminate unimportant frequencies, effectively removing data from the image. By reconstructing the image with fewer frequencies, we achieve a “compressed” version of the image.

You work in a space agency and receives thousands of high-resolution images from Earth observation satellites every day. Storing and transmitting these massive images is expensive and slow. You decided to use Fourier-based image compression to remove less significant details while preserving critical features like coastlines and roads. This reduces data size, allowing faster transmission and efficient storage without losing essential geographic information.

## Tasks

1. Convert the image to gray scale (After reading the image start with low frequency image).
2. Compute the Fast Fourier Transform (FFT) to analyze frequency components.
3. Apply a threshold to retain significant frequency components while discarding others.
4. Apply inverse Fourier Transform to reconstruct a compressed image.
5. Using a high frequency image, compress it and compare the compression performance with low frequency image.
6. Find a high resolution satellite image online, compress it using your algorithm and report on its performance.

## Question 4

Data augmentation is a crucial technique in deep learning. By artificially expanding the training dataset, it enhances model performance and generalization. For instance, in image classification, we can apply transformations like rotation, flipping, and zooming to create diverse examples. This variety prevents overfitting and allows models to learn robust features. Data augmentation finds relevance in various applications, including object detection, image recognition.

## Tasks

1. Please complete the function in the starter notebook which accepts an image and a type of augmentation to perform. Use the parameters described in the comment section of each augmentation.
2. Despite data augmentation being good, not all transformations are valid, and it is therefore important to consider the problem you are trying to solve. Considering the problem of facial recognition, state if the transforms given in the starter notebook will be valid or not.
3. Add any two transforms not included in the starter code that are relevant to face recognition model.

Good Luck