

ARI2129 - Group Project 2024

Part B - Technical Paper



**L-Università
ta' Malta**

Name: **Andrea F. Lucas, Thomas Koppens & Sean D. Muscat**

Course: ARI2129 - Artificial Intelligence (AI)

ID No: 0279704L, 029204L & 0172004L

Distribution of Work - Technical Paper

Section	Responsible Person
Project Overview & README.md	Andrea F. Lucas
TensorFlow (Q3-a)	Andrea F. Lucas
PyTorch (Q3-b)	Thomas Koppens
OpenCV (Q4)	Sean D. Muscat

Project Overview

This project focuses on the use of data augmentation techniques with popular machine learning frameworks, including TensorFlow, PyTorch, and OpenCV. Data augmentation is an essential step in the preprocessing pipeline of machine learning models, particularly for image classification tasks, because it improves model generalization by artificially expanding the training dataset.

Question 3: Data Augmentation in Mainstream Machine Learning Frameworks

In Question 3, we look at how data augmentation is implemented with two popular machine learning (ML) frameworks: TensorFlow and PyTorch. The ImageDataGenerator class in TensorFlow allows for the application of a variety of image transformations. Similarly, PyTorch includes the transforms module, which enables advanced image manipulation techniques. By applying these tools to the COTS (Common Objects of a Traveling Scientist) dataset, we show how effective each framework is at generating augmented data. The task entails creating Jupyter Notebooks for both frameworks, and producing multiple augmented images per input. The results are then saved and analyzed using comparative metrics to determine the differences with the original images.

Question 4: Implementing Data Augmentation Techniques Using OpenCV

In Question 4, we look at how to implement data augmentation techniques with OpenCV. Unlike TensorFlow and PyTorch, OpenCV requires a more manual approach to transformations. This question addresses both photometric and geometric distortions, including brightness variation, contrast adjustment, rotation, saturation, shearing and translation. We use these techniques on the same subset of the COTS dataset as in Question 3, resulting in multiple augmented images per input. The goal is to compare OpenCV results to TensorFlow and PyTorch results, which will provide insights into the practical application and effectiveness of custom augmentation implementations.
