CM22009 – Machine Learning

Coursework 2: Lossy Compression

Set: 28/02/2025

Due: 21/03/2025, 8pm

Percentage of overall unit mark: 15%

Submission Location: Moodle

Submission Components: Report & Code

Submission Format: 1 x Jupyter Notebook and 1 x PDF Report

Anonymous Marking: Y

1 Overview

This coursework assesses two of the three module Learning Outcomes:

2. Demonstrate understanding of a wide range of machine learning techniques, their strengths and their limitations.
3. Write code in a relevant programming language and employ software libraries to solve problems in machine learning.

1.1 The Task

Compression is the process of encoding data (such as an image) into a new representation which takes less space than the original, this representation can then be decoded to reproduce the original image. There are two types of compression (as shown in Figure 1):

- Lossless compression: A compression method which can decode the compressed representation into a perfect replica of the original data.
- Lossy compression: A compression method which can decode the compressed representation into an almost perfect replica of the original data.

You are tasked with developing a lossy compression model for a given dataset of images (detailed in section 3) using an Autoencoder which is a special type of unsupervised neural network architecture. To do this, you will need to:

1. Research Autoencoders, how they function and how they can be constructed.
2. Develop an Autoencoding neural network to reproduce the images from the dataset to minimize the information lost through the compression whilst considering the compression ratio.
3. Document your experiment in a short report format.

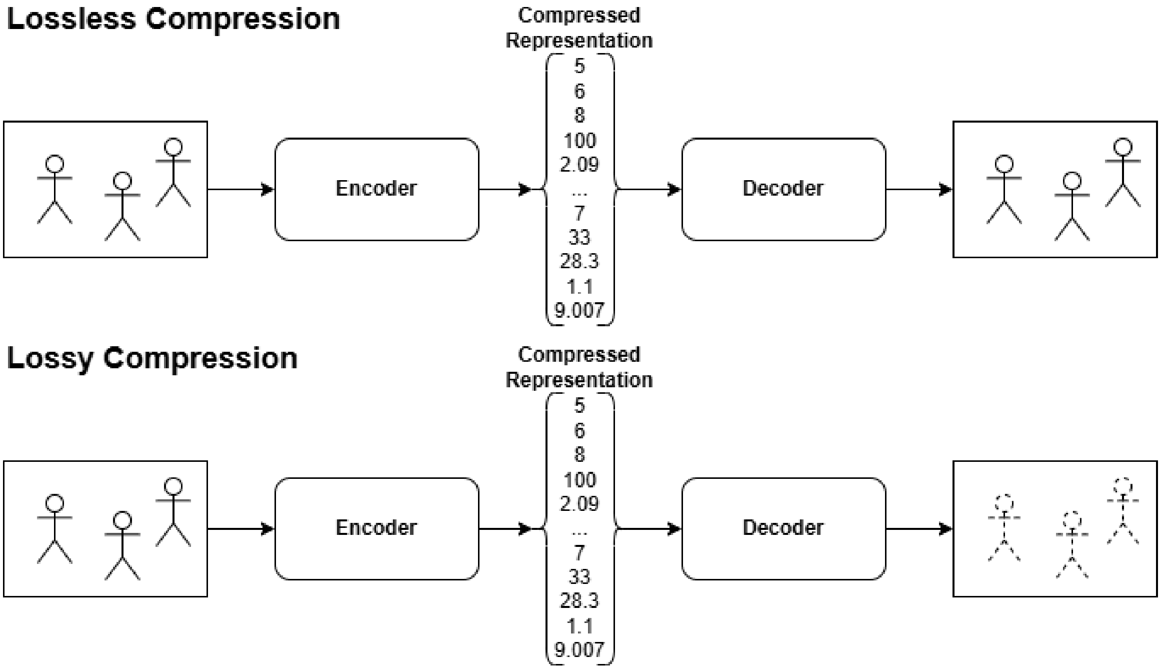

Figure 1. Illustration of lossless and lossy compression.

2 Deliverable

There are two deliverables to this coursework:

Deliverable	Format	Content
Code	.ipynb	A Jupyter Notebook containing the code to reproduce your experiments and final model. The Jupyter Notebook must contain: • Code to load and perform any data engineering on the images. • Code to reproduce your model. • Code to reproduce your evaluation and results.
Report	PDF	A report containing details of your model development, the report must contain the following sections: • Introduction: Brief description of what was done. • Data: Description of the data used and any data engineering performed. • Methodology: Overview of the modelling methodology including architecture, hyperparameter tuning and training regime. • Results: Analysis of the experiments including the performance of the model and an appropriate discussion of said results. • Conclusion: Summary of the conclusions which can be drawn from the results including limitations and future improvements. The report must be submitted in PDF format using a reasonable font (Arial, Calibri etc.) at size 11 minimum. The maximum page count is 5 excluding references.

2.1 Mark Scheme

Your report and code will be assessed using the contributions below:

Component	Criteria	Percentage of overall deliverable mark
Report (80%)	Introduction	0%
	Data	10%
	Methodology	20%
	Results	30%
	Conclusions	10%
	Presentation	10%
Jupyter Notebook (20%)	Presentation & Reproducibility	20%

Table 2. Mark Allocation.

Outlines for what would be expected of a typical submission in each grade classification are as follows:

- First-Class (>70%): Exceptional implementation and experimentation of autoencoding architectures making use of advanced neural network features. The report tells a clear story with a detailed overview and justification of the methodology whilst also containing an appropriate level of detail and discussion of the results.
- Upper-Second Class (60-69%): An effective implementation making use of appropriate libraries and machine learning techniques. The report highlights not only the methodology but also its limitations and flaws highlighting a clear path towards improvement.
- Lower Second Class (50-59%): A considered implementation of an autoencoder using Python and a report which tells a clear story, highlighting at a reasonable level the implementation, results and any appropriate conclusions.
- Pass (40-49%): A successful implementation of an autoencoder using Python and an accompanying report describing the process and results. Some conclusions are given which are drawn from the results.

3 The Images

The dataset provided is in the form of three Numpy files:

- subset_1.npy, subset_2.npy, subset_3.npy – Each file contains at most 400 flattened images each with a resolution of 150 x 225 (x 3 for the RGB channels) pixels, examples are given in Figure 2. Each image is represented as a row consisting of 101,250 columns each representing a pixel component in the original image.

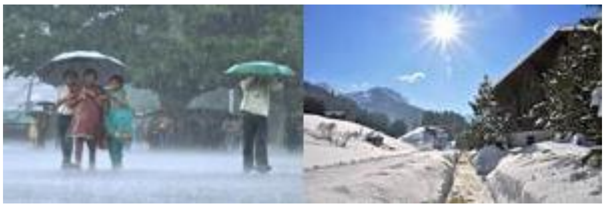

Figure 2. Sample images.

To load the data you should use the Numpy load function, for example:

```
import numpy as np
inputs = np.load(“subset_1.npy”)
```

To display an image at a given row (i) you can use the following code:

```
import matplotlib.pyplot as plt
inputs = np.load(“subset_1.npy”)
i = 500
plt.imshow(np.reshape(inputs[i, :], (150, 250, 3)))
```

4 Feedback

Formative feedback will be available in the labs.

You will receive **summative feedback** on your work within 3 semester weeks of the submission deadline. The feedback will discuss your performance based on the criteria for marking, including what you did well and how specific components/sections could have been improved.

5 Academic Integrity

Your work will be checked to ensure that you have not plagiarised. For more information about the plagiarism policy at the University see: https://library.bath.ac.uk/referencing/plagiarism

Remember that published work that you refer to in your report should be clearly referenced in your text and listed in a bibliography section given at the bottom of your report. For more information see, https://library.bath.ac.uk/referencing/new-to-referencing

This coursework is classified as Type A with regards to the use of Generative AI, hence no generative AI should be used.

6 FAQ

6.1 Is the quality of my models marked?

In short, no. We are more interested in whether you can adequately explain and evaluate your models' performances using the metrics and evaluation methods you choose. There are no direct marks available for the quality of the model.

6.2 Why is the page count so small?

Whether writing a technical report for a client or an academic conference paper concise writing is key.

6.3 Do we have to write the models from first-principles?

No! Please use PyTorch or other neural network libraries.