Coursework Assignment

1 Assignment Overview

This assignment will involve you designing, building, testing and critiquing a machine learning system for learning sequences of images. There is also a secondary extension task detailed below.

This assignment is designed to ensure you can demonstrate achieving the learning outcomes for this module, which are:

- Comprehend and apply the key aspects of a range of recent machine learning techniques.
- Work independently to propose and implement appropriate machine learning solutions for a specified problem and dataset, demonstrating critical awareness of potential challenges.
- Communicate clearly using text and figures an implemented machine learning solution covering the core concepts, rationale for design decisions and critical evaluation.
- Critique an implemented system demonstration a systematic approach to quantitative evaluation.

1.1 Primary Task

You will develop a model that can *interpolate* and *extrapolate* sequences of images. This is a common task in various image-based fields, e.g., in-painting [2], generating intermediary images [1], and trajectory modelling [3]. You will also demonstrate your model's robustness to noise and missing data, by corrupting your dataset with additional type(s) and level(s) of noise and missing data. The aspect is worth 80% of the marks for this assignment.

1.2 Secondary Task

You will transfer your model developed in the Primary Task to perform the same task on a smaller out-of-sample dataset. This aspect is worth 20% of the marks for this assignment.

2 What to hand in?

1. A report that comprises a *maximum* of 10 pages and 2000 words, including captions but excluding references. We expect several pictures, diagrams, flowcharts and plots to be included.

   - A summary and justification for all the steps in your sequence prediction system. Explaining diagrammatically is very welcome.

- Results of your experiments. This should include some discussion of qualitative (example based) and quantitative (number based) comparisons between different approaches that you have experimented with.
- A summary of how your noise corruption / missing data system works with several example results.

2. A .csv file that contains the predicted images for each sequence. Please make sure you run this on the right data and submit in the correct format to avoid losing marks.

3. Either .ipynb files or .py files containing annotated code for all data preprocessing, model training and testing.

3 How will this be graded?

The breakdown of marks for this assignment are given below:

20 Marks **Accuracy and robustness of image sequence prediction**
These marks are allocated based on the performance of the image sequence prediction method. This will be evaluated on the held out test set, which is a subset of the full sequence that is not seen in the training set. The test images are provided in the links below file and the error on the predicted images will be calculated after submission. Marks will be awarded for average accuracy and robustness.

30 Marks **Outline of methods employed**
Justifying and explaining design decisions for sequence prediction model. This needs to be in depth, but we do not expect you to regurgitate the contents of the lecture notes/papers. You should state clearly:

- What methods you have used, with what (hyper)parameters and why.
- Any image pre-processing steps you have used, and why.

For top marks, you should clearly demonstrate a creative and methodical approach for designing your system, drawing ideas from different sources and critically evaluating your choices. Explaining using diagrams and/or flowcharts is very welcome.

25 Marks **Analysing results and noise / missing data robustness**
Critically evaluate the results produced by your system on test data. You should include quantitative (number based) and qualitative (example based) comparisons between different approaches that you have tried. Marks for clearly identifying reason(s) for your model's robustness / weakness to noise and missing data.

Quantitative measures include image similarity metrics between the predicted and true (test) data; and using boxplots or other plots to compare methods. Please note that we are interested in your final prediction results, rather than how the cost function changes during training (though you can include a plot on the latter if you like).

A detailed qualitative analysis would investigate and identify systematic failure cases and biases, providing visual examples, and proposing potential solutions.

20 Marks **Out-of-sample analysis**
Briefly outline/provide a diagram/flowchart explaining how this was performed. Provide qualitative and quantitative results and contrast them with the results from the Primary Task.

5 Marks **Code annotation** is for annotating sections of the training/validation/testing code with what they do. To get maximum marks, explain each algorithmic step (not necessarily each line) in your notebook/.py files.

General points on the report

- Read things! Provide references to anything you find useful. You can take figures from other works as long as you reference them appropriately.
- Diagrams, flowcharts and pictures are very welcome, make sure you label them properly and refer to them from the text.
- All plots should have correctly labelled axes.

4 What resources are provided for me?

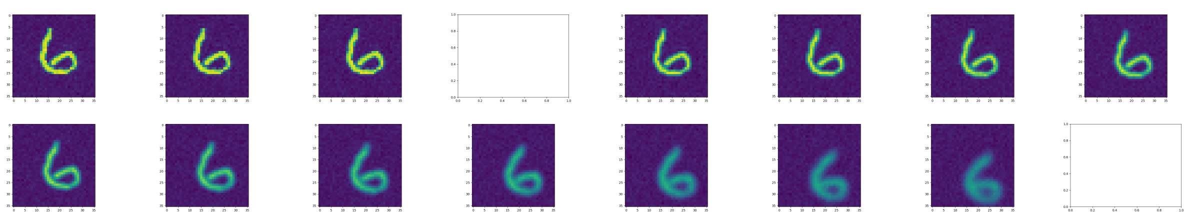

Figure 1: Example sequence of images in the "n6" training data. Note that image 4 and 16 in the sequence are removed (these are the test data for this sequence).

We provide two datasets, which are derived from the MNIST dataset:

1. The "n6" dataset, comprised of samples of the digit 6, which is for the Primary Task.
2. The "n3" dataset, comprised of samples of the digit 3, which is for the Secondary Task.

The original "n6" dataset (before train/test split) was comprised of 400 independent sequences of 16 images. Rotation, translation, and intensity-based transformations were applied autoregressively at each step along the sequence, and independent white noise was added at each step. There are outliers in the dataset.

You are provided with train and test datasets, obtained from the original "n6" dataset. The training set corresponds to a subset of 14 images from the total sequence of 16 images for each sample. The same 2 images (indices 3 and 15, indexing from 0) in every sequence are reserved for the test set. The data are in long format, e.g., the training set has length = 5600, which corresponds to 400×14 sequences. The images are 36×36 pixels and are flattened to an array of length = 1296.

The "n3" dataset is the same as above, except there are only 100 independent sequences.

The train and test data can be downloaded as csv files from the links in the table below.

Contents	filetype	links
"n6" training images	train_data_n6.csv	link
"n6" test images	test_data_n6.csv	link
"n3" training images	train_data_n3.csv	link
"n3" test images	test_data_n3.csv	link

The data can be read by:

```
import pandas as pd

# Load the data using pandas
data_train = pd.read_csv(‘‘train_data_n6.csv’’, header=None)
# .shape == (5600, 1296)
data_test = pd.read_csv(‘‘test_data_n6.csv’’, header=None)
# .shape == (800, 1296)
```

And similarly for the Secondary Task.

4.1 Notes on using Colab

Either you can complete this project using Google Colab, which gives you a few hours of computing time completely free of charge, or you can use your personal/lab machine. If you are using Google Colab, try and familiarise yourself with some of its useful features. To keep your saved models, preprocessed data etc., you can save it to Google Drive following the instructions here. You can also directly download a file you make in Colab using the code below:

```
from google.Colab import files
files.download(filename)
```

If you refactor code into extra .py files, these should be stored in your Google Drive as well, or on Box such that they are easy to load into your Colab worksheet.

4.2 What library functionality can I use?

You're free to use fundamental components and functions from libraries, e.g., PyTorch, TensorFlow, to solve this assignment. You are **not** allowed to use a generative AI tool, e.g., ChatGPT, to write the code for you. You are allowed to use pre-trained models, provided you explain your reasoning and justify their use.

In terms of sourcing additional labelled data, this is **not** allowed for this assignment. This is because in real-world commercial projects you will typically have a finite dataset, and even if there are possibly useful public datasets available, their license normally prohibits commercial use.

On the other hand, data augmentation, which effectively synthesises additional training examples from the data that you have, is highly encouraged. If you use this, please try and add some text or a flow-chart of this process in your report.

5 Top Tips for Success

- Remember Occam's razor, complexity should not be added unnecessarily. The more complicated your system the more things to explain/justify etc.
- Start with a simple achievable goal and use that as a baseline to test against. Keep track of early models/results to use as points of comparison.
- Remember that even if it doesn't work well, having a go at the extension tasks is worth a few marks.
- More broadly, think about the task itself, e.g., interpolation, extrapolation, and what properties you want your model(s) to have.

6 Further reading

The module lectures and seminars covered various approaches for modelling sequences and images. There is also a vast literature on tasks related to interpolation and extrapolation using images.

References

[1] Sauer, A., Schwarz, K., & Geiger, A. (2022). StyleGAN-XL: Scaling StyleGAN to large diverse datasets. ACM SIGGRAPH.

[2] Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., & Norouzi, M. (2022). Palette: Image-to-image diffusion models. ACM SIGGRAPH.

[3] Antelmi, L., Ayache, N., Robert, P., & Lorenzi, M. (2019). Sparse Multi-Channel Variational Autoencoder for the Joint Analysis of Heterogeneous Data. PMLR.