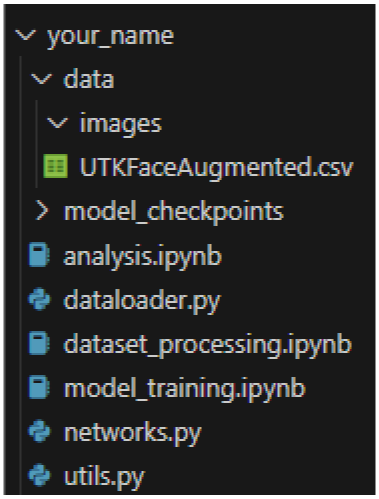
ECEGR4750 Introduction to Machine Learning
Fall 2023 Final Project

This project is modelled after the type of take-home projects often used in machine learning interviews, and so will aim to test how well you can apply what you have learned in this course.
Your goal will be to investigate the assigned dataset, train several models, and write a report analyzing and comparing your models. You will need to package up and submit your working code via GitHub.

A. Submission Guidelines

Please read the following instructions carefully, as there will be no re-submissions and late submissions will not be accepted.

You must use this following structure for your project to keep it organized:



```
⌄ your_name
  ⌄ data
    ⌄ images
    UTKFaceAugmented.csv
  > model_checkpoints
  analysis.ipynb
  dataloader.py
  dataset_processing.ipynb
  model_training.ipynb
  networks.py
  utils.py
```

I will provide you with the data file. Unzip it and place it in your directory as seen above. When I grade your submissions, I will place the data directory like this in your code and will expect your code to work.
You may modify this structure if you find it insufficient for your needs. If you do, please provide a README file in markdown with your submission that describes the purpose of each file.

For submission, upload your folder to GitHub as a separate repository. DO NOT UPLOAD THE DATA. Name your repo as `ecegr4750_fall2023_lastname`. Upload your report separately to Canvas as a PDF named `ecegr4750_fall2023_lastname.pdf`.

B. Code Guidelines

dataset_processing.ipynb

This dataset is the UTKFace dataset as found here https://susanqq.github.io/UTKFace/. I have added some additional features in the CSV file `UTKFaceAugmented.csv`. This file also contains the name of the image that corresponds to those features.

You will need to process and prepare the data for machine learning purposes. You should be familiar by now with the best practices to do so. Images require preparation as well. There are plenty of resources out there explaining the variety of ways to do it. I did mine in just a few lines of code – so no need to go crazy with it. You can open an image using the PIL library and go from there.

Save your processed data into your working directory, but do not submit it. I will re-generate your dataset when I grade by executing your `data_processing.ipynb` notebook.

dataloader.py

The CustomDataloader class we have written together is not sufficient for multimodal training. You are required to make a new dataloader class that can handle a multimodal dataset.

Because images are memory-intensive, you will not be able to store the entire image dataset in memory. This means you must write your multimodal dataloader to *stream* data from disk to your model. In essence, every time a batch is generated, you need to load, process, and return image data on demand. This will let you scale your dataset to massive scale, while paying the cost of featurizing your data at training time. If you do not do this, I will be unable to load your dataset to test your training code.

If you cannot complete this part of the final project, proceed with training on any data you can hold in memory so that you can still receive credit at later stages of the project.

model_training.ipynb, networks.py

You will train three models to predict age.

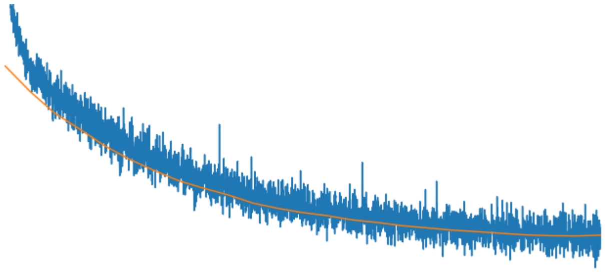
1. Classic Model
   Train a non-neural network model to predict age given the features found in `data/UTKFaceAugmented.csv`. Select the appropriate model for your task.
2. Neural network
   Train a neural network model to predict age given *only* images found in `data/images/`.
3. Multimodal neural network
   Train a neural network model to predict age given *both* the features found in `data/UTKFaceAugmented.csv`*and* the images in `data/images/`.

This dataset is larger than others you have worked on, so training takes longer. GIVE YOURSELF TIME TO TRAIN. If you start on the last day, you will not have time to complete this project. For reference, once I found good hyperparameters, I could train the multimodal neural network in about one hour on my home computer's rather old CPU.

For both neural network models, please checkpoint your model and include it in your submission, as I will not be able to train everyone's model myself. For saving and loading, look here:
https://pytorch.org/tutorials/beginner/saving_loading_models.html

Some model training tips for you:

- Large batch sizes speed up training significantly.
- Larger and deeper models train slower but will improve performance.
- Learning rate is the most critical hyperparameter to test.
- torch.hstack is a handy function to be aware of for the multimodal network
- Identify good hyperparameters by training on a small subset of your training data first so you can rapidly iterate. You will save a lot of time that way.
- For reference, my training curve looks like this. Yours may be different.



analysis.ipynb

To write your report, you will need to generate several figures on your results. Use this notebook as a place for that analysis. Do not put dataset processing or neural network model training in this notebook. I will be grading this by attempting to load in your model checkpoints, run inference on the dataset, and analyze the results. For your classic model, feel free to re-fit the model for the purposes of the analysis as it is relatively painless to repeat. Be mindful that your method is deterministic so that I can easily re-generate your results when I grade.

utils.py

Helper functions, global variables, and any frequently re-used code should be kept here. If at any point you find yourself re-writing highly similar code, then think how you could turn it into a function. Reduce, reuse, recycle!

C. Report Guidelines

Given a dataset containing images of faces and some associated features about that person, apply machine learning to predict their age. Analyze your trained models and provide a recommendation as to the best model to use. Support your claims with evidence from your findings.

The following sections are required, but feel free to add as you see fit. Draw inspiration from the academic papers you read for your paper presentation assignment.

Introduction

- Describe the dataset and introduce your work at a high level.
- Describe any biases within the dataset and describe the steps you took to manage them. Explain how biases may impact your models, and the implications of using biased models for age prediction.
    - Illustrate your findings with at least 1 figure.

Dataset processing

- Describe the steps you took to prepare the data for machine learning, including:
  - How you transformed each modality of data.
  - Any additional feature engineering.
- Describe whether you will frame this as a classification or a regression problem (either can be right) and explain your rationale and the risks associated with each strategy. If you took any specific steps to re-label the data, describe them here.

Models

- For each of your 3 models, do the following:
  - Describe the method itself as you would to a machine learning audience.
  - Provide a visual aid or formula to show how your method takes inputs and produces outputs (the block diagram style is great for neural networks).
- When you fit each of your models, provide the following:
  - An explanation of the most important hyperparameters you tuned for that model.
  - The loss curves when you fit the final version of the model. This may also be useful to illustrate how certain hyperparameters affect training.
  - Any notable challenges you solved to speed up training, address poor training behavior, or otherwise make progress on the problem.

Results

- Analyze the performance of each method.
  - Ensure that calculated metrics are shown together for easy side-by-side comparisons.
  - Identify the parts of your dataset where each model performed strongest and weakest. Provide visual examples from the dataset to illustrate your point.
- Describe how each model compares against a random baseline. How much improvement was there?
- Make a recommendation for one of your models and describe your rationale for this decision.

Conclusion

- Summarize your findings.
- Propose at least 3 possible ways to improve your results.

D. Grading

Late submissions will not be accepted. There will be no re-submissions!

Code Grading (120 pts total)

dataset_processing.ipynb	20 pts
dataloader.py	10 pts
networks.py	10 pts
model_training.ipynb	20 pts
analysis.ipynb	20 pts
Code readability	40 pts

Some pointers

- TEST YOUR CODE BEFORE SUBMISSION AND THEN TEST IT AGAIN
- Non-functioning code will result in an automatic 0 for that section. If you are unsure how to complete any section, at least write functioning code so that I can give you constructive feedback.
- If you elect to alter the format provided, include a README.md document explaining *how* I should run your code. Otherwise, I will assume I can simply unzip your submission, drop the data directory in, and start running your notebooks.
- Don't include featurized data in your submission. I will re-generate it by running your notebook.

Code readability feedback

Because code quality is (somewhat) subjective, I will provide *all* feedback on your code readability ahead of time. If you make a mistake that is not covered in the below feedback, you will not be penalized for it. Here is a collection of feedback I have left throughout the quarter. Each one will be -10 to code readability. Use this as a checklist before submission. Should be an easy and automatic 40 pts!

- Ensure all snake_case variable names are entirely lowercase. Use consistent rules in your naming schemes (don't use x_train and train_y, for example).
- Unused variables. Every variable that is declared should be used.
- Commented out lines of code. Remove it before submission.
- Convert re-used code into functions. It is really easy to bloat a notebook. When in doubt, create a function and put it in your utils.py script. Even one-time use code that *could* be used elsewhere can be handy to convert into a function.
- Usage of markdown cells. Clearly declare the purpose of each series of cells in your notebook using markdown. This improves the flow of reading your notebook dramatically.
- Figure axes. Always label and title all figures. If you generate a figure for the report, ensure the fontsize is large enough to be readable in the final document.
- Excessive redundant / non-descriptive comments. Use comments to explain *why* you made a decision in your code, not to describe *what* your code is doing. Your code should be readable enough on its own to follow it.
- Be explicit in using certain common functions like sklearn.model_selection.train_test_split. Set a random seed and declare your exact split size.
- Formatting and spacing. Break up code into cells and avoid any excessively inconsistent spacing between chunks of code.
- Unused imports. Ensure all imports are declared in the very first cell of your notebook.

Coding aids

I anticipate that most of the entire project should be familiar to you. Nevertheless, if you copy-paste code from any external sources (which is totally ok), just make sure to include the source of that code in an inline comment. Nowadays, I and most other software engineers use GitHub Copilot or GPT4 on a near daily basis to assist in our code. Getting used to incorporating it into your workflow is essential. Regardless of the source, make sure it falls within our code readability guidelines.

Report Grading (80 pts total)

Introduction	10 pts
Dataset processing	5 pts
Models	20 pts
Results	40 pts
Conclusion	5 pts

What I'm looking for in the report

I am looking for thorough and thoughtful writing in every section. Write this report for a machine-learning knowledgeable audience, like a prospective employer you'd like to impress. My goal is to evaluate your ability to critically analyze the problem set before you.

I encourage you to bring in external resources as appropriate to help you analyze the dataset and each model. This includes research papers, examples of other models you have found, and any and all relevant findings in facial recognition or age prediction problems.

Like in an interview situation, I am more willing to forgive certain mistakes in a submission if the candidate has clearly put in an effort. In other words – if you do the minimum as set forth in the above guidelines, you must be perfect. If you go "above and beyond", you don't.

Formatting advice

Write your report to be clear and easy to read. Every figure should be legible and have a caption. References should be cited (I'm not picky about style). There is no word count – but I'm expecting you will need a few pages.

How good do my models have to be?

You should try to get your model to be the best it can be while training for no more than an hour. No matter its performance, I expect that the models are trained to convergence (so if your loss curve has not plateaued in an hour, update your hyperparameters) with reasonable hyperparameters. A better performing model will make for a better comparison in the report. That being said, I care even more about your ability to characterize your model's performance – a converged model can still be a poor predictor.