COMP 6721 Applied Artificial Intelligence (Fall 2023)

Project Assignment, Part I

Due date (Moodle Submission): Friday, October 27th
Counts for 20% of the course project

Project Introduction: "A.I.ducation Analytics". Welcome to *A.I.ducation Analytics,* where the future of AI-driven academic feedback takes shape. Dive into a world where, as AI lectures unfold complex algorithms, instructors are no longer in the dark! Your innovative system scrutinizes students' facial responses in real-time, distinguishing the curious from the overwhelmed. A sleek dashboard offers the instructor immediate insights—30% engaged, 20% neutral, 10% nearing cognitive overload. Smart AI suggestions nudge adjustments in real-time, ensuring lectures evolve in response to learners' needs. As graduate students on this frontier project, you're not just coding—you're sculpting the next phase of dynamic, AI-enhanced education.

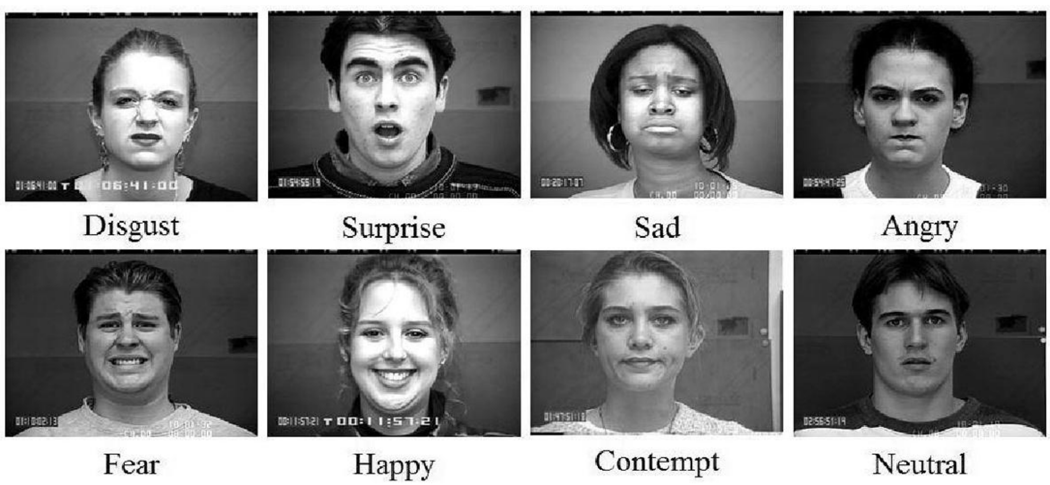

Image from https://www.thoughtworks.com/insights/articles/recognizing-human-facial-expressions-machine-learning

The objective of this project is to develop a Deep Learning Convolutional Neural Network (CNN) using *PyTorch* that can analyze images of students in a classroom or online meeting setting and categorize them into distinct states or activities. Your system has to be able to analyse images in order to recognize four classes. The first three classes are common, mandatory ones that you must detect:

Neutral: A student presenting neither active engagement nor disengagement, with relaxed facial features.

Engaged/Focused: A student evidencing signs of active concentration, with sharp and attentive eyes.

Bored/Tired: A student displaying signs of weariness or a lack of interest. This can be evidenced by droopy eyes or vacant stares.

And for the fourth class, your team has to **pick one** of these three additional classes:

Angry/Irritated: Signs of agitation or displeasure, which might manifest as tightened facial muscles, a tight-lipped frown, or narrowed eyes.

Confused: Signs of difficulty or misunderstanding, which might be evident as a raised eyebrow, a slight frown, or a puzzled expression.

Distracted: The gaze or attention is directed away from the main subject or activity, indicating a diversion of focus.

Course Project Overview. The whole project is split across three parts. Note that these are not stand-alone assignments, but rather build on top of each other to deliver the final, complete project (percentages show the marks attributed to each part for the course project grading item):

Part 1: Data Collection, Cleaning, Labeling & Preliminary Analysis (20%). Here, you will develop suitable datasets that you will later need for training your system. This includes examining existing datasets, mapping them to your classes, performing some pre-processing and basic analysis (see details below).

Part 2: CNN Model Development and Basic Evaluation (35%). After you've learned about CNNs and PyTorch, you will build basic CNN models suitable to detect the above classes. This includes deciding on the number of layers, type of layers (convolutional, pooling, fully connected), and activation functions. You will have to apply basic evaluation metrics: Accuracy, confusion matrix, precision, recall, and F1-score.

Part 3: Bias Analysis, Model Refinement & Deep Evaluation (45%). Addressing bias and refining models is advanced and crucial for AI's ethical applications. You will use visualization tools and metrics to identify bias in your model towards particular classes. Here, you will have to consider demographic fairness to check for racial, age, or gender biases. You will then apply bias mitigation techniques to reduce bias. This can involve oversampling underrepresented classes, employing fairness constraints, or using techniques like adversarial debiasing. Additionally, you have to apply deeper evaluation techniques, in particular k-fold cross-validation.

Your project team. In successful real-world AI projects, collaboration and role specialization go hand-in-hand. While each team member should contribute holistically to the project, designating areas of primary responsibility allows for focused expertise development and ensures every critical aspect of the project is attentively managed. There are three roles:

(I) Data Specialist: This individual will champion the dataset's lifecycle. Key responsibilities encompass sourcing, pre-processing, loading, and exploratory data analysis. As the gatekeeper of data quality and integrity, they will ensure the data is conducive for model training and evaluation. While the Data Specialist will author the dataset-related sections of the report, collaborative brainstorming with the team will enhance data understanding and troubleshooting.

(II) Training Specialist: Entrusted with the heartbeat of the project, this role dives deep into crafting, tuning, and training the Convolutional Neural Network. They'll set hyperparameters, monitor training progress, troubleshoot convergence issues, and collaborate closely with the Data Specialist to understand data nuances that affect training. They will pen the model architecture and training portions of the report, but iterative feedback from the team is crucial for refining the model.

(III) Evaluation Specialist: Once the model is trained, this expert will step in to gauge its prowess. They'll run the model through a gamut of tests, meticulously dissect its predictions, and diagnose areas of strength and weakness. Bias detection, cross-validation, and other advanced evaluation techniques will be their arsenal. Their insights will shape the evaluation section of the report, and their findings will be pivotal in informing the Training Specialist's model refinements.

Remember, while each specialist has a primary domain, the essence of the project lies in the synergy among team members. These designations are not silos but rather focal points. It is through combined insights, shared learnings, and collaborative efforts that the project will truly shine. In particular, this specialization does not mean the designated person does all the work for the specified sub-task, but rather oversees, manages, and distributes sub-tasks to other team members as appropriate.

Project Part I. In this part, you will have to collect suitable training data and perform EDA (exploratory data analysis) as follows:

Training Data: Create datasets for training and testing your AI. You have to provide provenance information, i.e., where you obtained each image in your dataset. You should re-use existing datasets, but again please make sure you properly reference the source of the image datasets (name, author, source, license of the dataset). It is expected that you have a *minimum* of 1500 training images and (additionally) 500 testing images (across all classes), so a minimum of 2000 in total for four classes. Note: This is before applying any data augmentation strategies. Make sure that both your training and testing data sets are *balanced*, i.e., have roughly the same number of images per class.

You must use real training data, i.e., using synthetic, generated data is not permitted.

Data Cleaning: Images can have varied sizes, resolutions, or lighting conditions. Standardize the dataset by resizing images to a consistent dimension, and potentially applying some light processing to increase robustness (e.g., slight rotations, brightness adjustments, minor cropping).

Labeling: If datasets are not pre-labeled or if there's ambiguity, manual labeling might be required. For example, you might have to map single or multiple classes from different datasets to a suitable class for training your system. Platforms like Labelbox[1] can be helpful.

Dataset Visualization using scikit-learn and Matplotlib: Visualize your dataset to ensure you have an even distribution of classes and understand the nature of your data. This is crucial before diving into model training, as having an imbalanced dataset, for instance, can affect model performance. Using Matplotlib, show your class distribution and plot a few sample images together with their pixel intensity distribution (see the report section below for details). Gaining these insights early on will allow you to make informed decisions about any additional preprocessing or cleaning that your dataset might require.

Report. You have to write a report detailing your findings, methodology, and analysis from Project Part I. Adherence to the specified structure and length constraints is crucial for ensuring a clear presentation of your work.

Title page: Presenting your group details such as the team name, team members, student ID numbers, each team member's specialization, and a link to your project repository (e.g., on Github).
Length: 1 page

Dataset: For the dataset you created, provide the following details:

- Overview of the existing datasets that you used: Including total number of images, number per class, and any special characteristics of the dataset (e.g., mostly frontal face shots, diverse backgrounds, etc.)

[1] https://labelbox.com/

- Justification for dataset choices: Explain why these datasets were chosen, their relevance to the project, and any challenges they might present.
- Provenance Information: A detailed table listing where each image (or image batch) was sourced from. This can include links or references to the dataset, the licensing type, and any other relevant information.

Length: ca. 2–3 pages

Data Cleaning: Explain how you cleaned your data:

- Detail the techniques and methods applied for standardizing the dataset, such as resizing and any light augmentation techniques.
- Discuss the challenges encountered during the data cleaning process and how they were addressed.
- Use example images to illustrate the before-and-after cleaning effect to substantiate your claims; For instance, how slight rotations or brightness adjustments impacted an image.

Length: 1–2 pages

Labeling: In this section, detail how you arrived at the labels for your dataset:

- Describe the methods and platforms/tools used for labeling. Discuss any ambiguities encountered and how decisions were made to resolve them.
- If multiple datasets were merged or if classes were mapped, provide details on how this was accomplished and any challenges faced.

Length: 1–2 pages

Dataset Visualization: Using Matplotlib, carry out the following tasks:

- **Class Distribution:** Plot a bar graph showing the number of images in each class. This helps in understanding if any class is overrepresented or underrepresented.
- **Sample Images:** Present a collection of 25 images in a 5×5 grid, with the entire grid sized to fit within the confines of a standard letter-sized page. Ensure that the images are randomly chosen from each class upon every code execution. This visual representation aids in understanding the dataset's content and helps identify any noticeable anomalies or potential mislabelings.
- **Pixel Intensity Distribution:** For the same random images, plot a histogram showing the distribution of pixel intensities. This can provide insights into variations in lighting conditions among images. Note: for color (RGB) images, overlay the intensity distributions of the Red, Green, and Blue channels on a single histogram.

Length: 2–3 pages

Reference Section: Contains citations, in IEEE style, to all pertinent resources consulted during the project. This includes books, websites, online tutorials, forums, and so forth. Remember, even if a resource only served as inspiration, it must be cited. Proper citation is essential for maintaining academic integrity (see below).[2] Ensure that references to any image datasets are also included here.
Length: As required

[2] If you are unsure about proper citation guidelines, see the resources provided by the Library at https://library.concordia.ca/help/citing/index.php

Deliverables. For the completion of Project Part I, ensure you bundle all the necessary items specified below into a single `.zip` or `.tgz` archive for submission:

Python Code. All Python scripts developed for this project:

- This encompasses scripts for data cleaning, data visualization, and dataset processing.
- Note: Only pure Python code (`.py` files) will be accepted. Jupyter notebooks or other formats will not be considered.
- Your code should be well-commented and modular to facilitate easy understanding and evaluation.

Dataset. A file or document detailing the provenance of each dataset/image:

- For publicly available datasets, incorporate only a reference with the necessary details such as dataset name, source, and licensing type (i.e., do not try to submit your whole dataset content here).
- For custom or modified images, ensure you include them alongside any manually crafted metadata.
- Supply 10 representative images from each class within your archive and incorporate a direct link to the full dataset in your repository (e.g., on Github).

README. A comprehensive `readme.txt` or `readme.md` file:

- It must enumerate the contents and describe the purpose of each file in your submission.
- Clearly outline the steps to execute your code for (a) data cleaning and (b) data visualization. If your instructions are incomplete and your code cannot be run you might not receive marks for your work.

Report. Your finalized project report:

- Must be structured adhering to the guidelines provided earlier
- Submit your report as a PDF file.

Originality Form. Include a **single** *Expectation of Originality* form:

- Available at https://www.concordia.ca/ginacody/students/academic-services/expectation-of-originality.html
- This form, attesting to the originality of your work, must be electronically signed by **all** team members.
- If the form is missing, your project will not be marked!

Submission Procedure: You must submit your code electronically on Moodle by the due date (late submission will incur a penalty, see Moodle for details).

Project Contribution and Grading Policy. In the spirit of collaboration and team unity, by default, all team members will receive the same grade for the project. This default policy is based on the expectation that all team members actively contribute, collaborate, and collectively drive the project to completion.

However, we recognize that team dynamics can vary, and there might be instances where contributions are disproportionate. In the event of a dispute regarding contributions:

1. The team must first attempt to resolve the dispute internally. Open dialogue and clear communication are encouraged to ensure all members are aligned on expectations and deliverables.
2. If the internal resolution is unsuccessful and team members believe that the contributions have been significantly uneven, the team should approach their designated TA (acting as their "project manager") with their concerns. The TA will provide guidance and possibly mediate to help resolve the issue.
3. If after the TA's intervention the dispute remains unresolved, the team may ask the TA to escalate the matter to the course instructor.
4. For a dispute to be considered at this stage, the team must provide clear evidence that delineates individual contributions. This evidence should ideally be in the form of version control records, such as Github change logs, commit messages, pull requests, and code reviews. Merely having more commits doesn't necessarily indicate greater contribution; the quality, relevance, and impact of the changes will be evaluated.
5. Along with the evidence, a written statement detailing the nature of the dispute, reasons for the perceived uneven contribution, prior attempts to resolve the matter internally, and the involvement of the TA should be submitted.
6. The course instructor will review the submitted evidence and statements. After consideration, they may adjust individual grades to reflect the contributions more accurately. This decision is final.
7. Any claims towards the contribution made after the final (late) submission deadline for each part will not be considered.

Teams are strongly encouraged to maintain regular communication with their TA throughout the project to preemptively address any potential conflicts and ensure smooth progression. The goal of this policy is not to encourage disputes but to provide a fair mechanism for resolution in the rare event it's needed.

Academic Integrity Guidelines for the A.I.ducation Analytics Project. Upholding the principles of academic integrity is paramount to the learning process. To ensure fairness, clarity, and ethical behavior throughout this project, the following guidelines have been set:

1. **Originality of Work:** All submissions must be the original work of the team members. Copying or adapting work from other teams is strictly prohibited. Using external sources without proper citation is also strictly prohibited (this includes ChatGTP and similar tools, see below). Such actions will be considered academic dishonesty and may lead to a failing grade for the project or other disciplinary actions as deemed appropriate by the university's academic integrity policies. Please make sure you review the academic code of conduct at https://www.concordia.ca/conduct/academic-integrity.html. Not knowing the code is not a valid defence for violating it!
2. **Citing External Sources:** Should you use external sources, such as datasets, code snippets, or any other resources, you must provide clear citations. Ensure that you give appropriate credit by adding the source into your report's reference section as defined above (using the IEEE format). Remember, acknowledging sources not only maintains academic integrity but also highlights your diligence in research.

3. **Usage of Large Language Models (LLMs) like ChatGPT:** Recognizing the relevance of LLMs in modern AI, the use of tools like ChatGPT is permitted with specific restrictions:
    a. LLMs should complement your work, not serve as the main source of your solutions or content.
    b. When using LLMs, both the prompt you provided and the response from the model must be clearly displayed in your report. This ensures transparency and allows for a clear differentiation between student work and LLM-generated content.
    c. While LLMs can offer insights or clarify concepts, relying heavily on them diminishes the learning experience. Aim to understand and articulate in your words, using LLMs as a supportive tool, not a primary crutch.
4. **Collaboration vs. Copying:** While collaboration is encouraged for brainstorming and problem-solving, always ensure that what you submit is your team's authentic work. Sharing code, data, or report content between teams is considered a breach of these guidelines.
5. **Ensure Private Repositories:** You are strongly encouraged to make use of an online repository to coordinate and store your team's work, e.g., using Github or Gitlab. However, it is crucial that you make your repository *private* (make sure you give access to your team's TA). If your repository is public and another team uses your work, both your team (for sharing) and the copying team (for using) can be held accountable for academic misconduct.

Remember, the purpose of this project is to immerse yourself in the world of AI, develop skills, and gain a profound understanding of the challenges and responsibilities that come with the domain. Adhering to these academic integrity guidelines ensures a level playing field and a genuine learning experience for all.