CEC 2025 Programming Competition Package

Joya Boutros and Orion Wiersma

Table of Contents

Background

On August 7th 2023, Destiny Rennie, a young Mi'kmaq woman died tragically of fungal meningitis, a disease which causes inflammation in the brain and spine. Fungal meningitis has a mortality rate of 50% and should be treated at least within 24 hours of diagnosis. However on July 27th 2023, when Destiny entered Soldiers Memorial Hospital (Middleton, NS), she was prescribed antibiotics and sent home with a CT scan scheduled for the following day. After her symptoms continued to deteriorate, she was taken to the hospital by ambulance on July 31st. According to the PATH legal team "by August 2nd, doctors suspected meningitis and ordered treatment, but delayed administering it for 8.5 hours." She was finally airlifted to the QEII Hospital (Halifax, Nova Scotia), however by that time the "QEII doctors determined nothing could be done." Destiny's story is one of many examples illustrating the ramifications of overwhelmed hospitals and staff, and the impact that has on patient satisfaction.

With an aging population, especially dealing with the aftermath of Covid-19, the current healthcare system is undergoing all time stress. Wait times for hospital emergency rooms across Nova Scotia can range anywhere from 1-7 hours depending on the location. Most hospitals run close to maximum capacity, putting stress on both the patients and health care workers. Even after diagnosis, many patients can end up waiting upwards of 100 days for MRI or CT scans. In order to circumvent this, attempts to automate the triage system as well as other important medical tasks have been made.

In the fall of 2024 the Nova Scotia Government pledged $42 million dollars to help implement AI tools in its health care system. The first stage of this project beginning in the fall of 2024 was used to increase information access to patients, regarding wait times and other aspects of supposed

conditions. In the fall of 2025 the project aims to automate patient diagnosis for X-ray findings, in order to provide radiologists with preliminary imaging findings for chest X-rays. Using machine learning models to compare with thousands of anonymous images, this could be of great help in decreasing wait times for imaging services.

The new JBOW Hospital (Sexton Campus, NS), is taking a modern stance on patient diagnosis. With the overwhelming amounts of patients requiring imaging, JBOW Hospital is in need of a program which can interpret MRI scans of the brain, to aid diagnosis and treatment for people undergoing brain injury. Using the grant given by the government, the JBOW Hospital is having a competition to decide which program to implement this idea. However, developing these models can lead to unforeseen circumstances, where does the liability go when scans are interpreted incorrectly? Should the program favor earring on the side of caution when interpreting scans and risk wasting the doctors time? What role *should* automation and AI play in the medical field?

If you would like to learn more about Destiny Rennie's story, you can go through this CBC article.

Problem

Problem Statement

Congratulations! JBOW Hospital has selected you as one of the select engineers to test and implement a modern program to help aid with our diagnosis and treatment of patients. For this challenge, we will require you to make a program that can determine whether a brain tumor is present following an MRI scan. You have been given access to a database of thousands of brain images with the files located in the github repository, labeled “yes” and “no” depending on whether a tumor is present (“yes” means that there is a tumor and the brain is considered unhealthy while “no” would be considered the healthy brain). Your task is to make a program which can detect when a tumor is present in an MRI scan.

Deliverables & Constraints

In order to complete this challenge, you will need to create a way to process the images, and create a model which can detect if a tumor is present. To test your program, 1,000 augmented MRI images of the brain (both healthy and unhealthy) will be run through your program by the competition directors. Your program should correctly attribute whether a scan has a tumor and put the data into a csv file (excel, google sheets or any other similar file format are also acceptable). While not *necessary* for diagnosis, it would be helpful if we could have specific information regarding the tumor, such as localization and classification in order to better inform treatment.

The success of your program will be based on the number of MRI scans that are accurately detected. However, more scrutiny will be applied to programs which fail to successfully diagnose the MRI scans that actually do have a tumor (ie. False Positive or Type 2 Error). The following truth table was created as a guiding factor to aid in the assessment in these scans:

Table 1: Truth Table

Model Output (Probability)	Class (Tumor/No Tumor)	Confidence of Tumor Present
1.00 - 0.95	Tumor	High Confidence
0.95 - 0.80	Tumor	Considered Positive
0.80 - 0.75	Tumor	Lower Confidence
0.75 - 0.65	Tumor	Adjusted Threshold
0.65 - 0.55	Tumor	Adjusted Threshold
0.55 - 0.50	No Tumor	Traditional Threshold
0.50 - 0.45	No Tumor	Below Threshold
0.45 - 0.30	No Tumor	Low Confidence

In addition to this, a simple way to judge the ethical considerations of your code could be by using a **confusion matrix**. Here the amount of false positives and false negatives can be tracked.

The GitHub general “main” repository has an example template for an expectation of data output on an excel sheet, however, this template does not have to be strictly followed - be creative! That being said, we do require data to be output as a csv, xlsx, or Google Sheet file for testing purposes. More on this can be found in the testing information folder!

Image Dataset Specifications:

This dataset comprises a comprehensive collection of augmented MRI images of brain tumors, organized into three distinct folders: 'yes', 'no', and 'CEC_test'.

- yes: Contains 8,727 PNG images of brain tumors.
- no: Contains 9,310 PNG images that do not exhibit brain tumors.
- CEC_test: Contains 1 PNG image as a mock test image. Only Directors have access to all 1000 test images in this folder.

Total images: 18,037.
All images have the same dimensions and are augmented.
Colour mode: Grayscale.

Yes (Tumor Present) No (No Tumor Present)

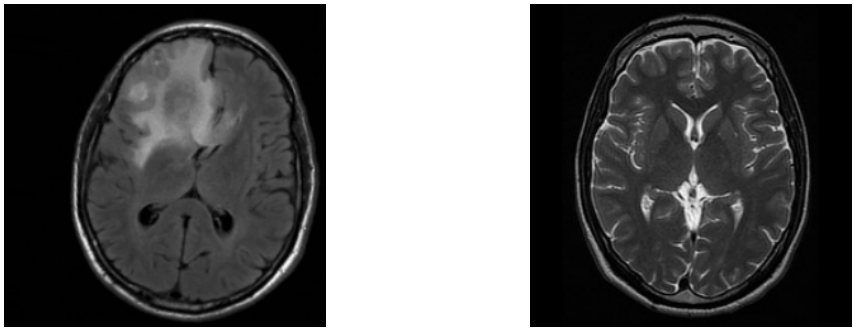

Figure 1. Explanation of Image Naming Conventions

Time constraints

All teams shall complete the challenge within 8 hours. The start time will be announced to all competitors after the competition briefing and question period. The design period shall immediately commence upon the delivery of the competition case document. Updates on the time remaining will be announced **three (3) hours, one (1) hour, thirty (30) minutes, and ten (10)** minutes before the end of the allotted time through

Deliverables

As part of Programming 2025, the following deliverables will be required:

1. Code
2. Presentation

Code

All code must be in each team's specific GitHub repository's "main" branch before the 8-hour deadline in order for it to be considered for judging. Basic instructions must be provided in a README.md on how to compile and run your code. This includes:

- How to run your code
- The language and version your code uses
- A list of required packages (i.e. Pandas, NumPy)
- Model files (with links if not located on git branch) that need to be downloaded to run your code. See \Testing Information\specific_model_file_download.md within the GitHub repo for more information on this.
- If needed, what OS your code should be run on. Any specifications of this sort not included in your README cannot be assumed to be on the Directors' machine(s).

Any specifications of this sort not included in the README cannot be assumed to be on the Directors' machine(s).

All code and presentations should be submitted on or before the deadlines. Late submissions will not be accepted.

Algorithms and Interface:

Your algorithm should successfully detect the brain tumor for the test images and output the data. While no specific interface is required, data about each tested image should be put into a CSV file (see example CSV file

in GitHub for more info). Any model (if used as a solution) must be trained within the time allotted.

Presentation

The presentation file must be submitted in the specific GitHub repository's "Main" branch before the 8-hour deadline. The presentation itself must be submitted as a .pdf, .ppt, or .pptx file to prevent any changes from occuring after the end of the design period. We recommend that teams reserve adequate time to submit the presentation given that technical difficulties may arise. Any modification of presentation content after the deadline is **strictly forbidden** and may subject a team to disqualification.

Your presentation should outline:

- The ethical considerations you took into creating your code.
- The way you made use of the data provided. (Did you make use of all the images?)
- The design and implementation of your algorithm.

Presentation Schedule

The randomized presentation order will be released **thirty (30) minutes** prior to the first presentation through the Discord . Requests to switch presentation order will be declined to ensure fairness between teams.

Each team will have **twenty (20) minutes** to present their deliverables, **followed by a ten (10) minute question-and-answer period** with the judges. All team members **must** participate in the team's presentation.

Demonstration

You will be expected to give a short demonstration of how your program works. The program will be loaded on one of the Director's laptops

and must run on it. You can assume the Directors have installed any languages and packages you have specified in a README file in your submission. Be sure that your instructions to compile and run your code are detailed enough for the Directors to set up your program. See the testing information folder in the GitHub for more information on how your code will be tested.

Testing Phase:

Your code will be tested by comparing the results of your algorithm to the correct results, with each team getting a percentage value for how accurate their code was. A folder labeled 'CEC_test' will have a set amount of test images (that only the directors have access to) corresponding to 'yes' or 'no'. The file format for these images will be PNG, with the names of each image being 'test_xxx.png'. Note: "x" in this case represents the number of files tested, similar to the yes and no directories. Make sure you have a testing script referencing this folder located at /CEC_2025/CEC_test. This can be done simply by employing environment variables, which you can expect the Directors will have on their machines..

See /Testing Information/setting_up_environment_variable.md in the GitHub for more info.

Model File Download Criterion:

If your code requires specific files for model data (e.g., model weights stored in HDF5, Pickle, JSON format, etc.), please specify these dependencies in the README file. Uploading large files directly to GitHub may cause delays. If this is the case, you can use a public OneDrive, Google Drive link, or another service to provide access. Teams will face penalties if files are not uploaded on time (file modification dates will be checked for OneDrive and Google Drive). **Teams will face penalties ranging from elimination to -5pts depending on**

severity if files are not uploaded on time (file modification dates will be checked for OneDrive and Google Drive). Alternative download methods are allowed, as long as the files are uploaded within the challenge's time constraints and the upload time is visible

Setting Up CEC_2025_dataset Environment Variable:

As a part of making testing more efficient and so that nothing is lost while trying to run test scripts on the directors laptops, it is strongly encouraged that you perform environment variable mapping. See GitHub for instructions on how to do this.

FAQ's

Question: *Are the images Augmented?*
Answer: Yes all images are augmented, the data can be considered 'clean.'

Question: *What images will be used in the testing phase?*
Answer: A separate set of images will be used in the demonstration.

Question: *How many images will be used in the testing phase?*
Answer: Your program will be tested with 1,000 images.

Question: *What are the image specifications in the testing phase?*
Answer: The same specifications as the given data will be used in the testing phase.

Question: *Is a google sheet or excel file required for data analysis/output?*
Answer: Yes, a simple output of raw data in a csv (or similar file format is required). Other interfaces for data analysis are complimentary, and will still be considered when assessing your code in the judging phase.

Question: *Is a Machine Learning model or AI based solution, required or can this be solved in a different manner?*
Answer: A Machine Learning model is not required, any way the competitor chooses to go about solving the problem will be considered.

Question: *Is tumor classification and localization required to complete the challenge?*
Answer: No, all that is required by the competitor is to determine IF there is a tumor in a given image, any classification or localization of the tumor will be considered extra, and will be evaluated in the judging phase.

Question: *If we choose to make a machine learning model, will we get extra time for training the model?*
Answer: All time to complete the challenge must be considered, you have 8 hours to complete the challenge and no extensions will be given.

Question: *Is the use of generative AI allowed?*
Answer: The use of generative AI is permitted. A record of important prompts and responses given is suggested in the submission phase.

Question: *How will directors run our code locally?*
Answer: Using all the files and by installing all the packages specified in the README file, competitors' code will be tested on directors machines locally. Please ensure that your README file is clear.

Judging Matrix

The judges will be using the following rubric to evaluate the scores of each team. This matrix will determine how well your team meets the objectives and requirements of the design challenge and ultimately, the top contending

teams. **All generic competition rules** can be found in the CEC 2025 Rule Book. The general desired outcome would be for each team to create a program with at least 65% accuracy, penalties will apply for those falling below the benchmark. Similarly those able to make a program which reaches industry standard: >=95% accuracy will receive 5 bonus points.

Evaluation Rubric

Category	Sub-Item	Evaluator	Weight
Strategy and Algorithm	1. Simplicity 2. Ingenuity 3. Ability to Achieve Desired Outcome	Judges	/5 /10 /20
Code	1. Structure 2. Consistency 3. Readability 4. Efficiency	Judges	/5 /5 /5 /10
Data Output	1. Ease of Use 2. General Aesthetics 3. Creativity	Judges	/5 /5 /5
Presentation	1. Design Process and Justification 2. Design Critique 3. Body Language, Respect and Professionalism 4. Visual Aids 5. Response to Questions	Judges	/5 /5 /5 /5 /5
Bonus	1. Data output with accuracy >=95%	Directors	+5
Penalties	See the below penalties matrix.	Judges	

Total	/100

Penalty Matrix

Scoring Penalties	
Plagiarism	**Elimination**
School or Regional Apparel During Presentation	**Elimination**
Disclosure of School or Region in Presentation Files/Documents	**Elimination**
Verbal Disclosure of School or Region During Presentation	**Elimination**
Documents Received After Deadline	**-50 points per instance**
Absent Team Member	**-25 points per instance**
Failure to hit minimum benchmark of 65% accuracy	**-10 points**
Entering Presentation Room Before Allotted Time (after first offense)	**-10 points per instance**
Late Model Files Submission (extreme extenuating circumstances)	**-5 points per instance**

Communications and Language

Q&A

Only the Programming Directors shall be permitted to answer questions on behalf of the CEC Programming challenge during the competition period. All questions and inquiries along with their responses will be posted publicly to the Discord in the CEC Discord, in both English and French. Questions may be asked up until time or end of the challenge.

Technical Assistance

To ensure fairness, the Programming Directors have developed multiple documents in GitHub that will assist in technical matters. If that does not suffice, teams may additionally request help from Programming Directors to help with issues which are strictly technical issues in nature.

Programming Directors reserve the right to refuse assistance or questions if they believe that it poses an unfair advantage to other teams, or that the question asked may disrupt the integrity of the competition.

Presentation Language

Teams can choose to present in either **French** or **English**. To ensure fairness and due to limited translation resources, teams must indicate if they will speak **English or French** for any amount of their presentation at check-in on the first day of competition.

For teams that select **French** as their presentation language, professional live-interpretation services will be provided to judges who are not bilingual during the presentation.

References

Dr S. Saroja, Shubham Joshi, May 29, 2024, "Augmented MR Images of Brain Tumor", IEEE Dataport, doi: https://dx.doi.org/10.21227/9p7v-ed03.

K. Nadeau, "Lawsuit filed against Nova Scotia Health after Mi'kmaq woman's death from fungal meningitis," *CBC News*, Jan. 10, 2023. [Online]. Available: https://www.cbc.ca/news/canada/nova-scotia/lawsuit-ns-health-mikmaq-woman-death-fungal-meningitis-1.7287879.

Dami