CS 175: Projects in AI

FRONT COVER

CS 175 Technical Memorandum

Challenge Title: Multi-label Image Classification

Authors: Thien Vu, Jake Leue, Darren Hoang, Diya Mirji, Nadia Ahmed (Team Mentor)

Team Name: Supervise Me

June 2023

Collaborators: NASA GeneLab: BPS Mice Microscopy Dataset, Lauren Sanders, PhD)

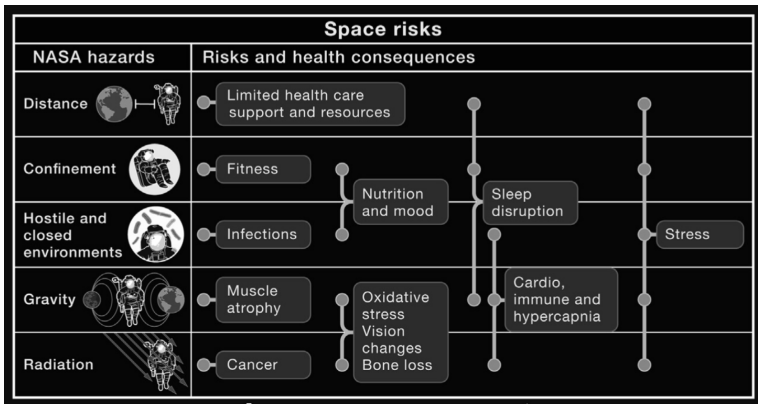
Introduction

Astronauts experience exposure to ionizing radiation or “galactic cosmic rays” which have been linked to health hazards such as DNA damage, central nervous system effects and immune system effects. The exposure to radiation leads to DNA damage within cells of astronauts. The effects can be mitigated if we can predict the damage done on cells by specific types of radiation.

The data images we will be utilizing come from the BPS Mice Microscopy Dataset of damaged mice cells. The damage to the cells are visible through fluorescent imaging of DNA damage markers as radiation-induced foci that create a linear pattern as tracks, which the Machine Learning model can learn to classify the images by their particle type or particle type and dosage.

In an effort to contribute to solving the problem, we’ve built Machine Learning models that are capable of analyzing images of patterns of DNA damage and classifying them by particle type or both particle type and dosage.

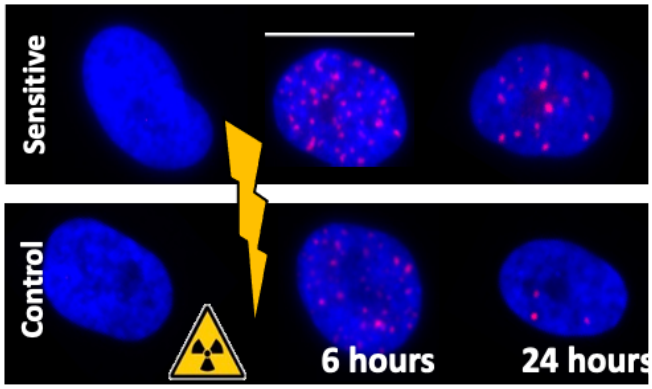
Challenge



Hazards and Health Consequences for Astronauts in Space

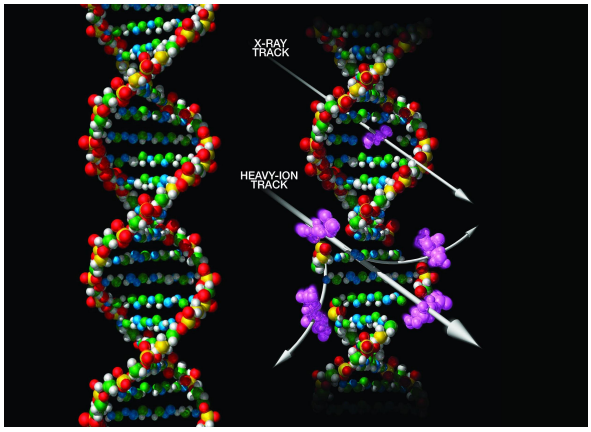
When astronauts experience spaceflight, there are numerous health risks due to five key hazards: distance, confinement, hostile and closed environments, gravity, and radiation. As one of the biggest problems, the consequences of ionizing radiation from space include DNA damage, central nervous system effects, and immune system effects. These issues can lead to cancer, oxidative stress, vision changes, bone loss, hypercapnia, and stress.

Before attempting to create a solution, it is important to first research space radiation. It is composed of solar particle events and galactic cosmic rays. The galactic cosmic rays contain about eighty-seven percent of protons, twelve percent of Helium, and one percent of high mass-energy particles through Iron. These particles are reproduced for simulations at NASA Space Radiation Laboratory in Brookhaven National Lab through a particle accelerator. Although the particles seem negligible, the galactic cosmic rays create significant damage to the DNA. The DNA impairment can be visible through fluorescent imaging to display radiation-induced foci as markers of the damage. A linear pattern of the radiation-induced foci are formed by heavy-ion tracks breaking through the DNA strands.



Fluorescent Cell Images with Radiation-Induced Foci

Machine Learning models may be able to model cell data to comprehend and predict the effects of these heavy-ion tracks on astronauts in space. In order to prevent putting humans at risk, it is possible to leverage data from other living systems' biology for a similar level of understanding on radiation effects. For instance, mice are genetically and physiologically similar to humans.



Heavy-Ion Tracks' Damage on DNA Strands

The NASA laboratory has generated a mouse culture fibroblast dataset accommodating five inbred strains and ten collaborative cross strains of mice. The mice cells have undergone three dosages (high, medium, and low) for each radiation particle type: Iron and X-ray. Images of these radiated cells are collected four, twenty-four, and forty-eight hours post irradiation.

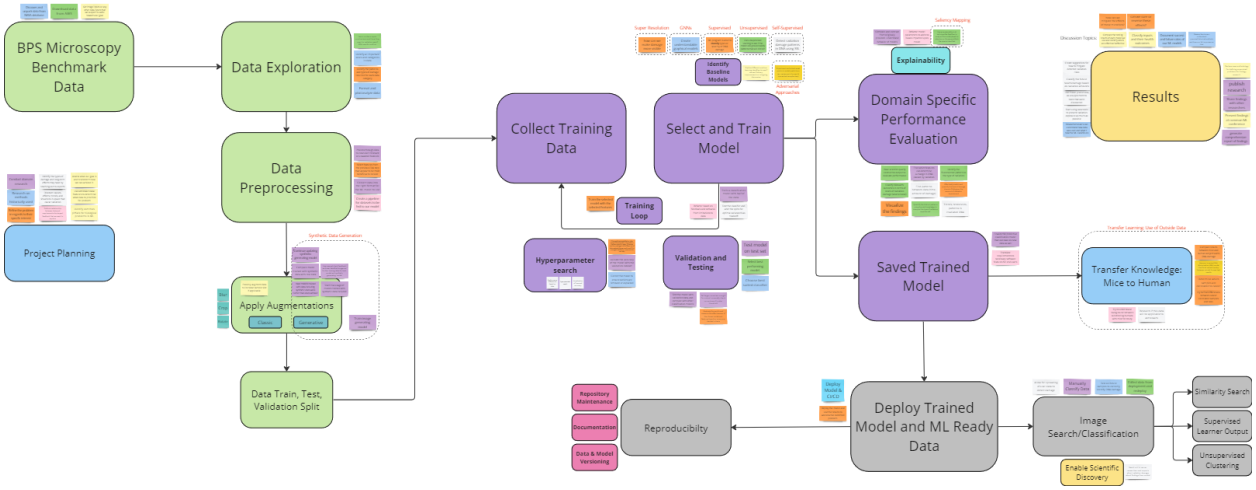
With the help of this data from the mice, we are able to find patterns in the visuals of the cell images to classify the types of radiation damage through artificial intelligence.

Procedure

Design Sprint

The design and ideation phase of this project was conducted as a class cohort, in which we all collaborated in order to overview the challenge, identify the problem, understand the data, and use different sprint techniques in order to gather the information, and decide on the direction of each of our projects.

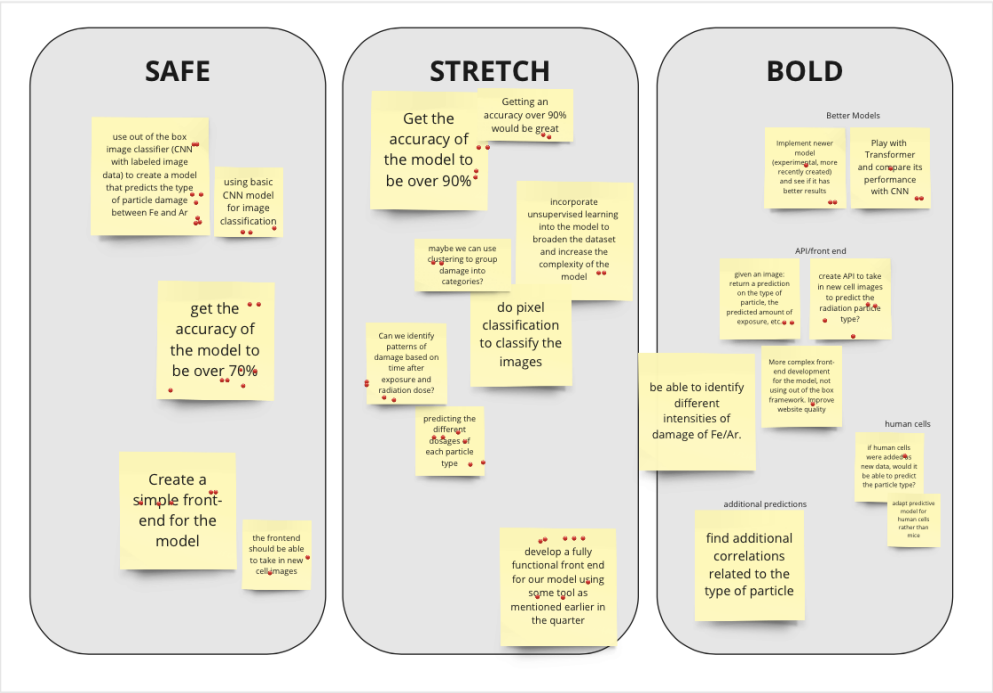
We mapped out the process from start to finish, allowing us to have a well defined plan for how to approach each step of developing our model.



Project Pipeline Created by Class

Our class consulted domain experts to achieve a better understanding of the domain space, as they could overview the most important ideas for our research as well as highlight which aspects of our project were technically or practically important.

We additionally compiled all of our questions and concerns on how we might accomplish certain tasks, which allowed us to identify our key obstacles. After doing so each student designed a proposed end-to-end solution, each of which was reviewed by our peers who identified the best parts of our solutions as well as leaving questions addressing any ambiguities in our proposals. Taking all of the information that we gathered through these sprint techniques, we as teams created our safe, stretch, and bold goals for our project.



Brainstorming Team Safe, Stretch, and Bold Goals

Data Description

The Biological and Physical Sciences (BPS) Microscopy Benchmark Training Dataset is provided by NASA through Dr. Lauren Sanders. The BPS Dataset is a public data repository hosted in an AWS S3 bucket which is a cloud-based storage service from Amazon. It consists of fluorescent microscopy images of individual nuclei from mouse fibroblast cells irradiated by Fe particles or X-rays with fluorescent foci indicating DNA damage. The dataset consists of 77,177 data instances, each instance containing an image, dosage amount, particle type, and hours post exposure. The dosage amounts for Iron particles are 0 (low), 0.3 (medium), and 0.82 (high). The dosage amounts for X-ray particles are 0 (low), 0.1 (medium), and 1.0 (high). The imaging times post-exposure are four, twenty-four, and forty-eight hours.

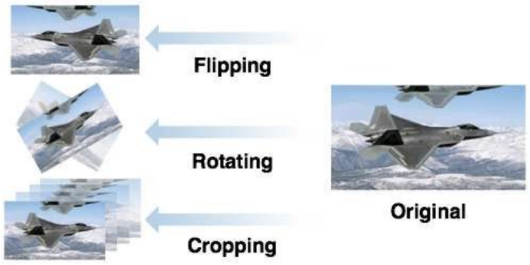
Label Attribute	Values	Total Data Instances
Radiation Type	Iron(“Fe”) or X-ray(“X-ray”)	77,177 Images
Radiation Dose	Iron Doses: [0, 0.3, 0.82] or X-ray Doses: [0. 0.1, 1.0]	
Imaging Time Post-Exposure	4, 24, 48 hours	

The script data_utils.py retrieves a meta.csv that contains the attributes of the image tiff name, the radiation dosage, the particle type, and hours post exposure for each image in the

dataset. Using this metadata, bps_dataset.py selectively chooses an image file of interest from the S3 bucket. The script, bps_datamodule.py will download the selected images as data. Since our project predicts the radiation particle type of the damaged cell, we utilize the particle type attribute. Later, we use particle type and radiation dosage to perform multi-label classification as well.

Data Preprocessing

The data processing we perform on the cell images are normalization of the image array values, resizing, vertical and horizontal flipping, rotating, a random crop, and tensorification of each image. These transformations and augmentations help reduce noise from the image and allows the machine learning model to focus on the linear tracks of the DNA damage foci.



Examples of Image Transformations

Methodology

Convolutional Neural Networks (CNNs) were chosen as the framework for our project due to its strengths in processing and analyzing visual data. As our task focused on the image classification, CNNs are well-suited to learn meaningful features from the images. The architecture of CNNs, with their convolutional and pooling layers, shrinks the dimensionality of data significantly which enables them to effectively capture spatial relationships in images. Fully Connected Neural Networks (FCN) on the other hand, struggle to extract relevant features from raw image data and lack effective dimensionality reduction. We will explore both neural network architectures, using FCNs as a baseline.

Baseline

We employed MLPClassifier from sklearn as a baseline, which is a simple fully connected neural network. It serves as an ideal point of comparison against an image classification specific architecture.We tuned the hyperparameters to a batch size of 64, 10 epochs, and learning size of 3e-4.

Deep-Learning Model

Our group chose to use LeNet-5, a convolutional neural network to classify the cell images by the attribute particle type. The model contains two layers each with pooling, and it uses ReLU as the activation function and an Adam optimizer. Through Weights and Biases, we ran sweeps to find the hyperparameters of batch size, epochs, and learning rate to get the

highest training accuracy. This resulted in an optimal setting of a batch size of 64, 10 epochs, and learning size of 3e-4 . Then, we modified the same model and dataset so the class label would be a combination of particle type and radiation dosage, for example (“Fe”, 0.82) as a class. For this multi-label classification, the sweep resulted in an optimal setting of a batch size of 128, 10 epochs, and learning size of 1.5e-4.

Performance Functions

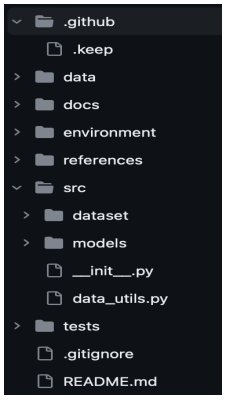
The functions we used to measure success of the different models we created and tuned were the training loss, validation loss, validation accuracy, Jaccard similarity, and Hamming loss. The accuracy score captures if the model accurately predicts the exact label. We decided to calculate the Jaccard similarity and Hamming loss since they are more appropriate than exact match accuracy scores for multi-label classification evaluations. Both these scores take into account partial matches in the multi-labeling, which gives more insight into the performance of this model.

Tests

While developing the scripts to download and prepare datasets of the cell images from the AWS S3 bucket, we utilized written tests to check functions in src/data_utils.py, src/dataset/bps_dataset.py, and src/dataset/augmentations.py. These tests ran each function from the scripts and verified that it performed its expected purpose through the GitHub classroom.

Replicability

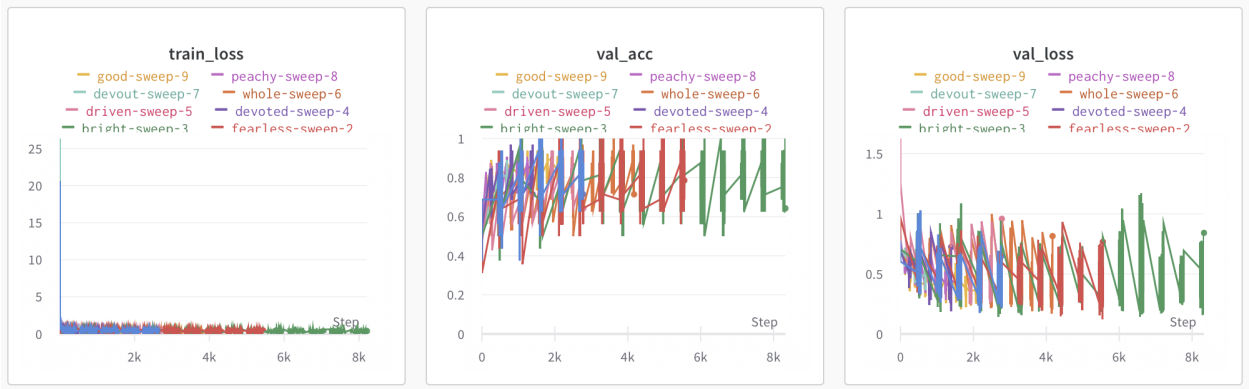
The GitHub project repository has been organized to facilitate the replicability of our research outcomes. The repository is structured effectively to separate different files in different locations. For example, the data folder contains space for the cell images. Additionally, the src folder encompasses a collection of scripts to download data and run the Machine Learning models.



Specifically, src/dataset contains scripts that employ the AWS boto3 API to retrieve the data employed in training our model. To obtain this data, one can execute the Python script located at src/dataset/bps_datamodule.py within our repository. Upon execution, the images necessary for training the model will be automatically populated in the data/processed folder. Subsequently, the model can be trained by accessing src/models/lenet_scratch.py and …/mlp_model.py, and then configuring the relevant options specific to your dataset prior to executing the file.

If the specific hyperparameters for the models are to be replicated, the reference for each sweep and model logging is displayed in our team Weights and Biases page. There is a project section for the LeNet model called SAP-lnet-from-scratch. It includes numerous runs and sweeps that train the cell data and log the training loss, validation loss, and validation accuracy of each run. Only the multi-label sweeps include Hamming loss and Jaccard similarity for

logging. For sweeps, it shows at which hyperparameters produce the values for the different types of scores.



Scores for Sweep of Single-Label LeNet Model

Scores for Sweep of Multi-Label LeNet Model

By adhering to these sequential steps, the research findings can be accurately reproduced.

Tools, Compute and Software Equipment

In this research project, we employed a range of tools, software, and frameworks to facilitate our investigations. This process can utilize either GPU or

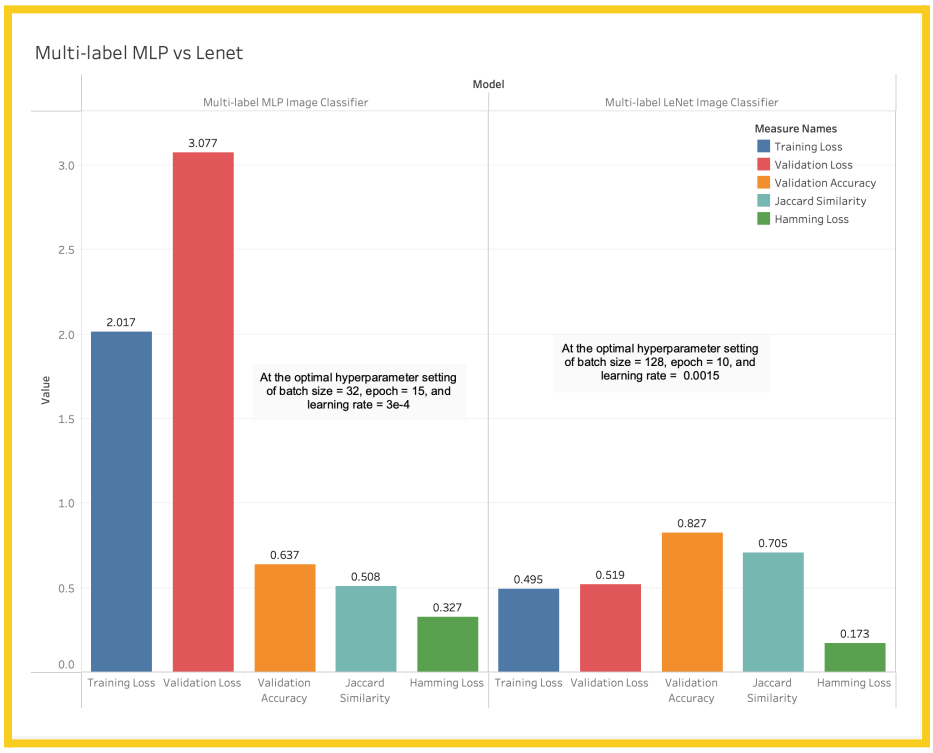
CPU. There are respective yaml files for the softwares to install for each respective hardware under environment/setup. To retrieve the required data, we utilized the AWS boto3 API, which offered the capability to specify a specific subset of the dataset for download.

For the training and development of our model, we leveraged the PyTorch/PyTorch Lightning framework. This framework played a pivotal role in simplifying the model-building process by automating numerous repetitive tasks associated with training. It handled crucial aspects like distributed training, automatic batching, and checkpointing, enabling us to focus more on the model architecture and experiment design.

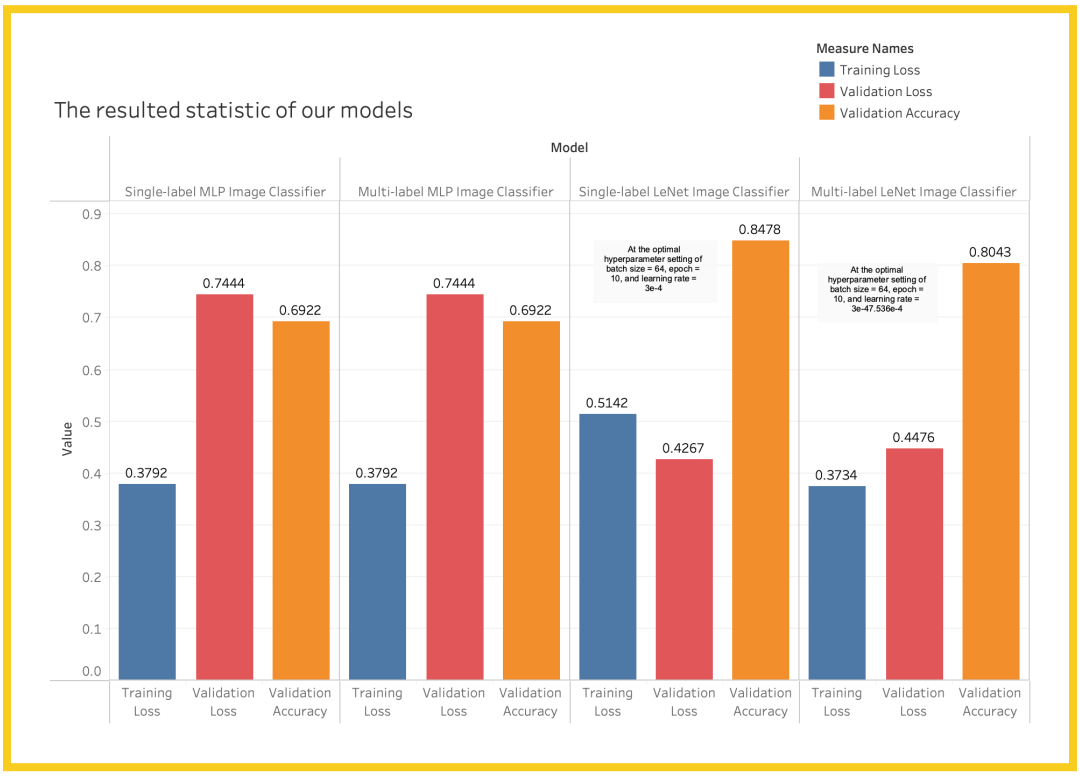
To capture and analyze our results, as well as conduct parameter sweeps to identify the optimal combination of hyperparameters for maximizing model accuracy, we relied on Weights and Biases (WandB). This tool facilitated the recording and tracking of our research outcomes, helping us to explore various configurations and evaluate their impact on the model's performance. The hyperparameters we tested are the batch size, epochs, and learning rate. For the runs, training loss, validation loss, validation accuracy, Hamming loss, and Jaccard similarity were logged on Weights and Biases.

By integrating these tools, software, and frameworks into our research workflow, we enhanced the efficiency, reproducibility, and accuracy of our experiments, ultimately contributing to the comprehensive exploration and analysis of our research objectives.

Results



Multi-label MLP Model Scores vs. Multi-label LeNet Model Scores



Scores to Measure Success of Each Model

In our comparison, we observed that the MLP-Classifier showed identical losses and validation accuracy for both single and multi-label predictions. When compared to our LeNet architecture, the single-label classifier exhibited superior performance in terms of validation accuracy and loss, although its training loss was slightly worse. Similarly, the multi-label classifier demonstrated improved validation accuracy and loss, while also outperforming its baseline in terms of training loss, albeit to a small degree.
The enhanced performance achieved by our LeNet architecture can be attributed to several contributing factors, including overfitting in our baseline model and the inherent suitability of LeNet for image classification tasks when compared to MLP.

Here is the source code for our final project:
https://github.com/UC-Irvine-CS175/final-project-supervise-me
Here is the link to the sweep for the Single-label LeNet Model:
https://wandb.ai/supervise-me/SAP-lnet-from-scratch/sweeps/xdvnnuuc?workspace=user-dahoang1
Here is the link to the sweep for the Multi-label LeNet Model:
https://wandb.ai/supervise-me/SAP-lnet-from-scratch/sweeps/qv6zpwd3?workspace=user-dahoang1

Future Work

In the future, our next steps are to add complexity to our Deep Learning Model. This includes adding more convolutional layers with different kernel sizes and trying more complex convolutional neural networks or a Transfer model. Furthermore, as Lenet-5 is a relatively simple CNN architecture, we could upgrade to more advanced CNN architectures such as AlexNet and VGGNet.

Additionally, we would like to develop an API capable of accepting new cell images and generating predictions on radiation particle type, predicted exposure levels, and other relevant factors. The API would be able to preprocess the images submitted by users in order to prepare them to contribute to the dataset. By providing this service, we seek to encourage the utilization of our prediction model and foster the expansion of the dataset, potentially improving the model's accuracy.

Conclusion

During spaceflight, astronauts are exposed to ionizing radiation, which can cause DNA damage, central nervous system effects, and immune system effects. The DNA damage can be apparent through fluorescent imaging to show radiation-induced foci and a linear pattern of them caused by the heavy-ion tracks breaking through DNA strands.

In order to obtain more data not involving humans, NASA’s laboratory has created a simulation of the space radiation on mice. The mice undergo two different particle types of radiation, and each particle type has a low, medium, and high radiation dosage. Then, an image is captured of the mice’s cells at four hours, one day, and two days after the radiation exposure.

Finally, the fluorescent cell images of these mice are collected into the BPS Mice Microscopy Dataset.

At the beginning of this project, a design sprint was conducted to produce an answer to how this problem could be solved. After consulting domain experts, we developed a better understanding of the subject, especially how we could apply artificial intelligence to solve the problem. Additionally, we were encouraged to create an end-to-end solution and a ML pipeline for review amongst peers and collaborators. At the end of the design sprint, we generated safe, stretch, and bold goals for our project.

Now using the Microscopy Dataset, we could employ Machine Learning models to predict DNA damage on cells depending on the radiation particle type or radiation particle type and dosage by performing image classification with Convolutional Neural Networks.

The development of the models started with writing scripts to download the data from the AWS S3 bucket and generate subsets of augmented cell images for training, validation, and testing. Then, we were able to train the data on a baseline model of the MLPClassifier from sklearn, which is a simply fully connected neural network.

To add more complexity and gain better results, we chose to use Convolutional Neural Networks as the project framework for its advantages in processing and analyzing visual data. The specific model employed was the LeNet-5 where it would predict the particle type affecting the cell data. The model contained two layers each with pooling alongside the ReLU activation function and Adam optimizer. During fine tuning of the model, we utilized Weights and Biases to run sweeps to find the optimal hyperparameters for the batch size, epochs, and learning rate and log the scores of loss and accuracy of the model. Then, we modified the model into a multi-label model for predicting the particle type and radiation dosage. We used Weights and Biases as well to fine tune the modifications and add Jaccard similarity and Hamming loss as logged scores.

Comparing the results of the MLPClassifier and LeNet-5, we found that the LeNet Image Classifier produced the highest validation accuracy. Between Single-Label and Multi-Label, the Single-Label LeNet Image Classifier had a higher validation accuracy and lower validation loss than that of the Multi-Label LeNet Image Classifier.

References and Bibliography

Quickstart Guide to Weights and Biases Sweeps:
https://docs.wandb.ai/guides/sweeps/quickstart
Multi-Label Image Classification with PyTorch and Deep Learning:
https://debuggercafe.com/multi-label-image-classification-with-pytorch-and-deep-learning/
Biological and Physical Sciences (BPS) Microscopy Benchmark Training Dataset:
https://registry.opendata.aws/bps_microscopy.

Acknowledgements: This outline on technical memorandum preparation is adapted from student materials from The Frontier Development Lab summer research program

A special thanks to the collaborators that helped make this project possible:

- Lauren Sanders, Ph.D. - Provider of dataset and domain expert
- Sylvain Costes, Ph.D. - NASA domain expert
- Kevin Li - Guest lecturer on Transfer Learning