CS640 Project Writeup

In this project, you will work on a Kaggle competition titled ISIC 2024 - Skin Cancer Detection with 3D-TBP. This is a binary classification task in which you need to predict if the patient has skin cancer. The competition is already close, but we will explore the rich (training) dataset it provides in this class.

Data

We will be using the training dataset from the original competition for this class project. The dataset has been downloaded and preprocessed. You can find it on SCC at */projectnb/cs640grp/materials/ISIC-2024_CS640*. **You should use this downloaded dataset only, not the original one on the website.**

The directory looks like the following.

```
1 import os
2
3 project_dir = os.path.join(os.sep, 'projectnb', 'cs640grp', 'materials', 'ISIC-2024_CS640
4 os.listdir(project_dir)
```

```
['test_metadata.csv',
 'submission.csv',
 'train_metadata.csv',
 'train_image',
 'test_image']
```

The CSV files store a list of attributes for each sample, and the image folders store a JPEG image per sample. The image names are sample IDs which can be found in the correpsonding CSV files. The submission.csv file is a template of your submission.

Let's first take a peek into the CSV files and a few sample images.

Training Metadata

Note that in the metadata file, the **target** column is the label column.

```
1 import pandas
2
```

```
df_train = pandas.read_csv(os.path.join(project_dir, "train_metadata.csv"))
df_train
```

	id	target	age_approx	sex	anatom_site_general	clin_size_long_diam_mm
0	0	0	55.0	male	upper extremity	2.58
1	1	0	50.0	female	posterior torso	2.90
2	2	0	40.0	female	lower extremity	4.38
3	3	0	50.0	female	upper extremity	2.76
4	4	0	60.0	male	posterior torso	3.31
...	...	...	...	...	...	...
320842	320842	0	70.0	NaN	posterior torso	3.60
320843	320843	0	45.0	male	posterior torso	5.88
320844	320844	0	40.0	male	anterior torso	11.41
320845	320845	0	40.0	male	lower extremity	4.02
320846	320846	0	50.0	male	anterior torso	3.15

320847 rows × 41 columns

Test Metadata

The test metadata file contains the same headers as the training one. Note that in this file, the **target** column is empty by design.

```
df_test = pandas.read_csv(os.path.join(project_dir, "test_metadata.csv"))
df_test
```

	id	target	age_approx	sex	anatom_site_general	clin_size_long_diam_mm	t
0	0	NaN	30.0	male	upper extremity	2.52	
1	1	NaN	75.0	male	upper extremity	2.63	
2	2	NaN	30.0	male	lower extremity	18.31	
3	3	NaN	45.0	female	upper extremity	3.55	
4	4	NaN	55.0	male	anterior torso	7.06	
...	...	...	...	...	...	...	
80207	80207	NaN	75.0	male	posterior torso	2.88	
80208	80208	NaN	50.0	male	upper extremity	4.20	
80209	80209	NaN	40.0	female	upper extremity	2.90	
80210	80210	NaN	75.0	male	posterior torso	3.32	
80211	80211	NaN	70.0	male	posterior torso	3.14	

80212 rows × 41 columns

Submission Template

This template file simply contains the first two columns of the test metadata file. You will need to fill the **target** column and submit for evaluation.

```
df_submission = pandas.read_csv(os.path.join(project_dir, "submission.csv"))
df_submission
```

	id	target
0	0	NaN
1	1	NaN
2	2	NaN
3	3	NaN
4	4	NaN
...	...	...
80207	80207	NaN
80208	80208	NaN
80209	80209	NaN
80210	80210	NaN
80211	80211	NaN

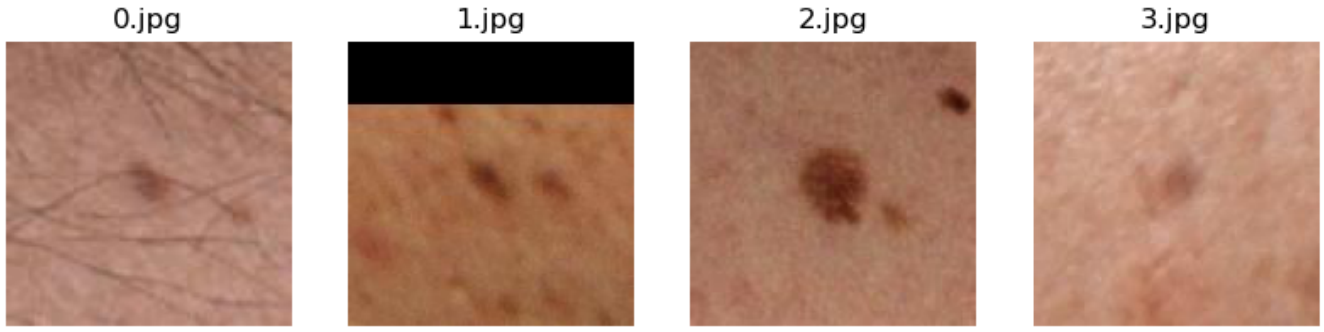
80212 rows × 2 columns

Sample Images

We will view a few images from the training set. The image files are named after the corresponding sample IDs.

```
import matplotlib.pyplot as plt
import matplotlib.image as img

fig, axes = plt.subplots(1, 4, figsize = (10, 20))
for i in range(4):
    id = str(df_train["id"][i])
    image = img.imread(os.path.join(project_dir, "train_image", id + ".jpg"))
    axes[i].imshow(image)
    axes[i].set_title(id + ".jpg")
    axes[i].set_axis_off()
plt.show()
```



Tasks

In this project, you need to work in a team of at most **four** members to build AI models to classify the cancer status (0: negative, 1: positive). The team signup sheet can be found here (you need to use your BU account to access it).

Additionally, before you start the project, you must register for a Kaggle account, join the competition on its website, and submit a screenshot of the team tab to Gradescope as a proof.

Since both tabular and image data are provided, you are expected to explore multimodal learning. To be more specific, we expect you to do the following.

- Examine and perform some statistical analysis to the data.
    - For example, you can check the distributions of the features in the metadata and find if there is a simple relation (say, linear relation) between some features and the target variable.
- Design and implement at least one model for each of the following categories:
    - Tabular data model
    - Computer vision model
    - Fusion model
        - A fusion model should combine two models from the previous two categories (one from each category, as "fusion" is defined in class).
- Perform stratified K-fold cross validation to demonstrate the performance of your models and choose your best model to predict the classes of the test samples.
    - While evaluating the performance during cross validation, consider the methods introduced at the beginning of the semester. For example, you should consider ROC

analysis and perhaps even AUC analysis (check the ROC wiki page).

- Write a report to record your methods and findings.
    - You are free to choose any tool (Word, LaTex, Notebook, etc.) to write the report. The template is available on our course website.

Challenges and Tips

We will provide some insights into the data and some ideas to get you started.

An (Extremely) Unbalance Dataset

If we compare the numbers of training samples in the negative and positive classes, we will have an astonishing finding.

```
counts = [df_train["target"].values.tolist().count(0), df_train["target"].values.tolist()
print("Number of negative training samples: " + str(counts[0]))
print("Number of positive training samples: " + str(counts[1]))
```

```
Number of negative training samples: 320533
Number of positive training samples: 314
```

How to tackle this nature of the training set is one of the major problems you need to address during the cross validation. Furthermore, note that we do **NOT** know the label distribution in the test set.

Too Many Features

There are about 40 features in the metadata. Some of these features are categorical while some are numerical. The numerical features even have different ranges. Therefore, what features to choose and how to choose the right features are two problems you will face during feature engineering.

What Models to Use

We do not put any restriction on model selection on you. In fact, you are encouraged to explore pretrained models (e.g., from PyTorch online library) or even models shared by other competitors (check the code tab, models tab, discussion tab and the leaderboard tab on the competition website).

If you use any of the existing approaches, Do

- cite the original approach by including links to its post;
- read and understand the approach (you need to explain the approach in your own words while writing the report); and
- verify the author's claims by applying their approach and try improving their method.
    - direct improvemnt of a single approach
    - fusion of different approaches
    - demonstrate how unreliable the approach is by showing, as an example, that their approach is sensitive to fluctuation or outliers in the data

But don't

- load the trained parameters;
    - the parameters here mean the weights trained by the authors
    - you can load model parameters that are not trained on this skin cancer detection task
- copy and paste the entire post to the report.
    - intead, summarize the approach in your own words (this is part of the study process)

Evaluation

You will be graded mainly based on how much effort you put into this project. We do not ask you to find the solution that yields the perfect predictions (but your solution should at least beat random guessing). Instead, we expect you to explore the data and appcoaches, which should be reflected both in your presentation and your report.

Here is a general breakdown of the grading:

- Performance (10%)
    - Need to at least beat the random guessing
    - Otherwise you will receive a low project grade
- Presentation (20%)
    - The presentation is like a short version of your report:
        - Briefly talk about your approaches
        - Summarize your results and observations
        - Your goal is to let us understand the main parts of your work with little confusion
- Report (70%)
    - Your report doesn't have to be long, but must be complete (30%)
        - You need to complete the sections listed in the template

        - You need to demonstrate your effort into the project
    - Your description should be clear (20%)
        - Do not build a wall of text
        - Do not build a wall of numbers/figures
            - Only choose what matters to the point you are trying to make
            - Always describe the numbers/figures, don't let the readers guess
        - When explaining your approaches and results, using a combination of words and figures can be very helpful
    - Statements should be backed by evidence and reasoning (20%)
        - For example, if you find some interesting relation between A and B, you should show some evidence (say, a correlation plot)