Draft

Introduction, objectives, motivation

Skin cancer can be life-threatening if not detected early, yet many communities lack access to specialized dermatologic care. To bridge this gap, algorithms designed for use in primary care or non-clinical settings must excel at analyzing lower-quality images[5].

This project tries to address those problems and goals mentioned above. In our work, we aim to create an algorithm that differentiates histologically confirmed malignant skin lesions from benign lesions on a patient. The quality of the images used for the algorithm resembles close-up smartphone photos, which are regularly submitted for telehealth purposes. Installing an algorithm like this on their cell phones could help people directly who do not have easy access to medical professionals. It could also reduce the cost of healthcare, helping those, who cannot afford expensive examinations.

Previous solutions, their advantages, disadvantages

One solution was presented in Nature, in 2017[2]. The researchers used a deep CNN (Google's Inception v3) and 129,450 clinical images of varying quality. Here are the advantages and disadvantages of the approach:

Advantages:

- The algorithm achieved diagnostic performance comparable to 21 board-certified dermatologists.
- The model demonstrates robustness against variations in photographic quality, including differences in zoom, angle, and lighting, which are common in real-world images. It also does not require image preprocessing.

Disadvantages:

- The algorithm does not take patient history into account.
- Heavily reliant on the available training data (true for all deep learning algorithms), therefore regions with limited data or diverse skin types might see reduced performance.

‘‘‘*A few more previous examples are needed here*‘‘‘

Besides the algorithms mentioned above, there are systematic reviews about deep CNN-s used for skin cancer detection[3][4].

System design

We used transfer learning for our project. In the beginning, we import the image data and the metadata and match every image with its corresponding label. After that, we balance the dataset using techniques like data augmentation and oversampling and convert it into a format so the CNN can work with it well. The network consists of Google's Inception v3 CNN and a fully connected neural network. In the end, we evaluate the model and plot the accuracy, and the loss, and also visualize the confusion matrix.

Database

We have used a dataset provided for the ISIC 2024 - Skin Cancer Detection with 3D-TBP challenge. The dataset includes diagnostically labeled images accompanied by metadata. The images are in JPEG format, while the corresponding .csv file provides a binary diagnostic label (target), potential input variables, and supplementary attributes such as the image source and detailed diagnosis[5].

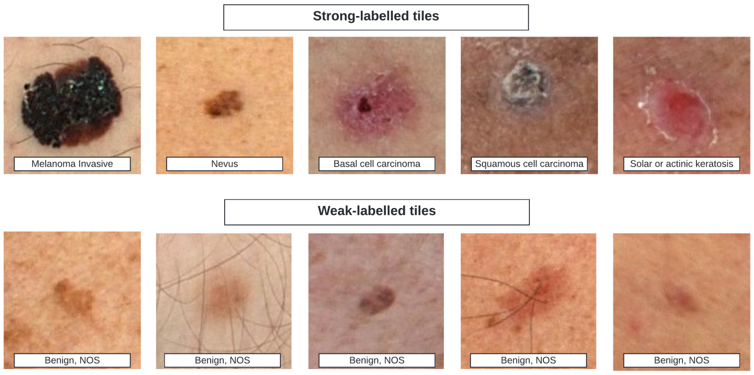

Figure 1: Example of dataset images. Replace with actual image.

The goal is to differentiate benign from malignant cases. We classify each image based on which class it belongs to. To replicate non-dermoscopic images, the dataset utilizes standardized, cropped lesion images derived from 3D Total Body Photography[5].

Architecture, training, challenges, and their solutions

We used Google's Inception v3 as it has been used previously for really similar projects. It was pretrained on roughly 1.28 million images spanning 1,000 object categories from the 2014 ImageNet Large Scale Visual Recognition Challenge and then fine-tuned on our dataset using transfer learning[2]. This was followed by a fully connected network made up of dense layers of varying numbers of neurons for each layer (exact neuron numbers to be added). We also used dropout to avoid overfitting.

We encountered a substantial problem while visualizing the data. The number of malignant cases was much lower than the number of benign cases (99.9% were benign and only 0.01% of cases were malignant). This would result in the model overfitting, and classifying each case as benign, while being right 99.9% of the time.

Most of our efforts were concentrated on solving this problem. At first, we only used class weights to balance the dataset, however, this did not yield good training accuracy, and it was overfitting, based on the validation data.

Later we tried oversampling the minority class, which produced slightly different results, however, it was still overfitting based on the validation data.

Afterward, we tried data augmentation, which had the same problems as the previous approaches. We also experimented with undersampling the majority class, which resulted in the best results up to that point, and the model did not overfit as much as previously. Despite that, it lacked accuracy, and the loss was also quite high.

Therefore, we decided to combine these preprocessing techniques. We oversampled the minority class and undersampled the majority class, used class weights, and data augmentation as well.

Figure 2: Final preprocessing results. Replace with actual image.

We still have our work cut out for us to perfect the model, optimize the hyperparameters, and work on the preprocessing of the dataset.

Results and their evaluation

Here are the initial loss and accuracy diagrams, and also the confusion matrix:

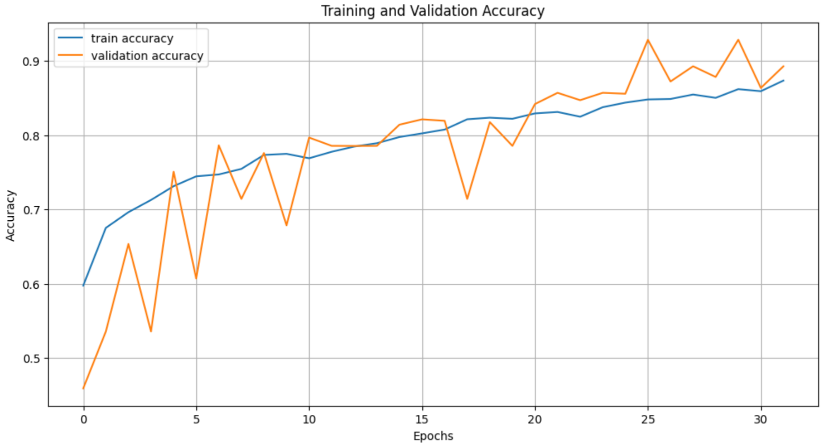

Figure 3: Final preprocessing results. Replace with actual image.

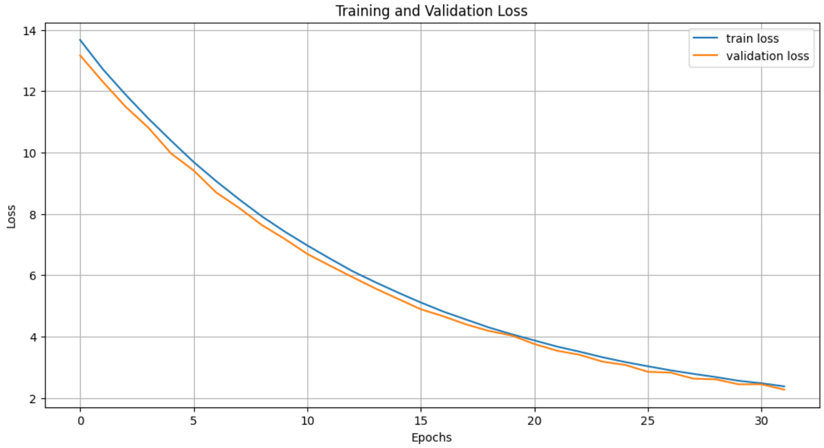

Figure 4: Final preprocessing results. Replace with actual image.

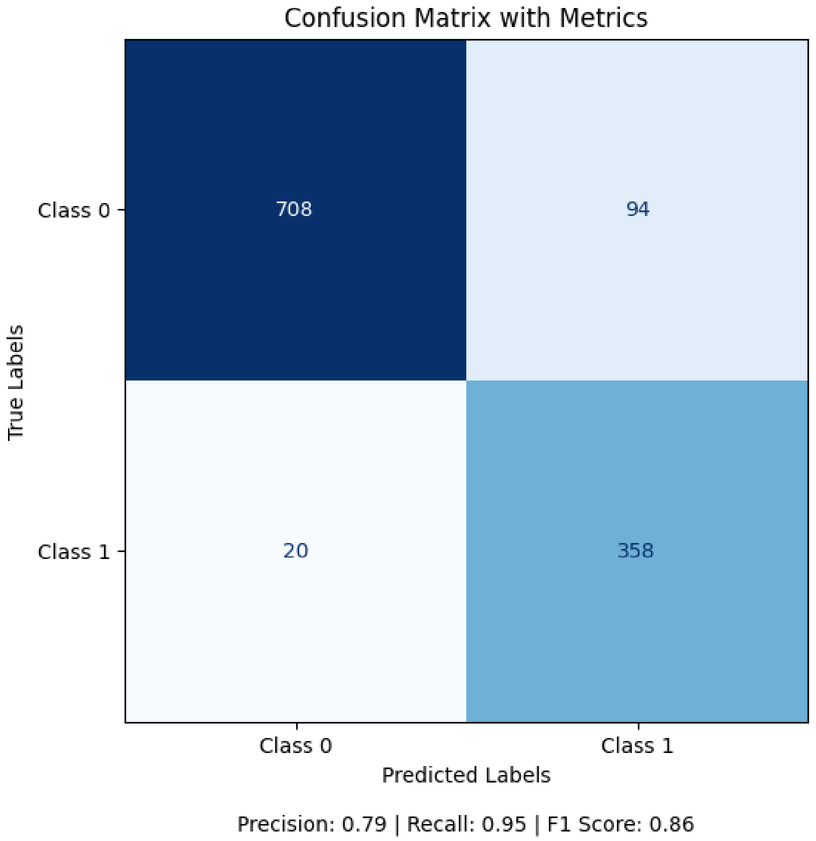

Figure 5: Final preprocessing results. Replace with actual image.

DEMO

pass

Summary

pass

References

1. `https://www.frontiersin.org/journals/medicine/articles/10.3389/fmed.2023.1305954/full`

2. https://www.nature.com/articles/nature21056

3. https://www.frontiersin.org/journals/medicine/articles/10.3389/fmed.2023.1305954/full

4. https://www.mdpi.com/2072-6694/16/3/629

5. https://www.kaggle.com/competitions/isic-2024-challenge/data