MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics

ISIC 2024 - Skin Cancer Detection

Solution of Team "42"

Team members:

Dávik Frink (HU7WN7)

Gergő László Gyulai (JJY2VO)

Ágoston Horváth (PGBDTC)

Budapest, January 5, 2025

Contents

1 Introduction

Skin cancer can be deadly if not caught early, but many populations lack specialized dermatologic care. Over the past several years, dermoscopy-based AI algorithms have been shown to benefit clinicians in diagnosing melanoma, basal cell, and squamous cell carcinoma. However, determining which individuals should see a physician in the first place has a great potential impact. Triaging applications have a significant potential to benefit underserved populations and improve early detection of skin cancer, the key factor in long-term patient outcomes.

Dermatoscope images reveal morphological features that are not visible to the naked eye, but these images are typically only captured in dermatology clinics. Algorithms that benefit people in primary care or nonclinical settings must be adept at evaluating lower quality images. This competition leverages 3D TBP to present a novel dataset of every single lesion from thousands of patients across three continents with images that look like cell phone photos.

Our team will develop AI algorithms that differentiate histologically confirmed malignant skin lesions from benign lesions on a patient. Our work will help to improve early diagnosis and disease prognosis by extending the benefits of automated skin cancer detection to a broader population and settings.

2 Previous Solutions

The top solutions presented at the competition involved advanced gradient boost techniques such as LGBM, XGBoost, and CatBoost. These are technologies that we have never heard of. Looking at these solutions gave us little to no help in pursuing the goal of creating an image classifier that is understandable to our knowledge.

Also, these solutions heavily utilized the metadata in the competition, while our stubborn team wanted to create a solution only based on images, to later create a mobile app.

After all, we had to rely on the practices we learned in class. We looked deeper into transfer learning, autoencoders and parameter optimization. Our solution might not break world records, but helped us understand some of the concepts related to image processing.

3 System Design

Our team explored various methods to develop an accurate model for the given challenge. However, only a small portion of them proved effective in achieving the desired performance.

At the start of the project, we experimented with anomaly detection. We trained a model using the benign dataset and then applied anomaly detection to identify unseen malignant samples. Unfortunately, while it was theoretically sound, this approach did not yield the best results in practice.

The next approach was slightly different. We leveraged transfer learning using EfficientNetB0 and employed a modified binary-crossentropy loss function, where the loss value was weighted according to the respective class weights. Regrettably, this method, which involved hyperparameter optimization, did not yield the expected results, as the vast majority of predictions were false positives. This was primarily caused by the disproportionate weight of the malignant dataset.

The next solution, which failed, was derived from the previous one. It was trained on a balanced dataset. We made the dataset balanced, by adding new data to ours. The main reason for its failure was it's ability to only detect the differences between the datasets, and not the underlying characteristics.

Finally, we decided to take a different approach. In the final solution, we trained an autoencoder on a bunch of benign images. Thanks to this, we got an encoder that has a really good "idea" about the dataset. Then we used this autoencoder and a fully connected layer to train a classifier model that can now more or less detect malignant images.

4 Databases

The implementation of our system is based on the dataset provided by the Kaggle competition, ensuring consistency and alignment with the competition's parameters. To process the data effectively, we utilized metadata from the train-metadata.csv, which allowed us to extract the unique identifiers for each image. These identifiers were then used to locate and retrieve the corresponding image data stored in files with the .hdf5 extension. This structured approach ensured a seamless workflow for accessing and utilizing the dataset for training and testing our model.

Our goal was to enhance visibility in terms of precision. The scarcity of malignant samples posed a significant challenge for us.

5 Architecture, Training, Difficulties and Their Solutions

Training the model proved to be very difficult, as the number of malicious cases was incredibly low. That is why we used the large number of benign images to train the autoencoder. In the end, we managed to overcome the problem somewhat, but increasing the number of malicious data could clearly help the solution.

However, the difficulty is that the data is extremely specific. The images were taken from a similar distance, have similar resolution and are mostly cleaned, so they are almost incompatible with other databases.

6 Evaluation results

Kaggle challange submissions are evaluated on partial area under the ROC curve (pAUC) above 80% true positive rate (TPR) for binary classification of malignant examples. The receiver operating characteristic (ROC) curve illustrates the diagnostic ability of a given binary classifier system as its discrimination threshold is varied. However, there are regions in the ROC space where the values of TPR are unacceptable in clinical practice. Systems that aid in diagnosing cancers are required to be highly-sensitive, so 4 this metric focuses on the area under the ROC curve AND above 80% TRP. Hence, scores range from [0.0, 0.2].

7 Encapsulation application

If you wish to try the Android demo application:

1. Run the `deep-learning-42milestone2.ipynb` notebook and download the `model.tflite` file.

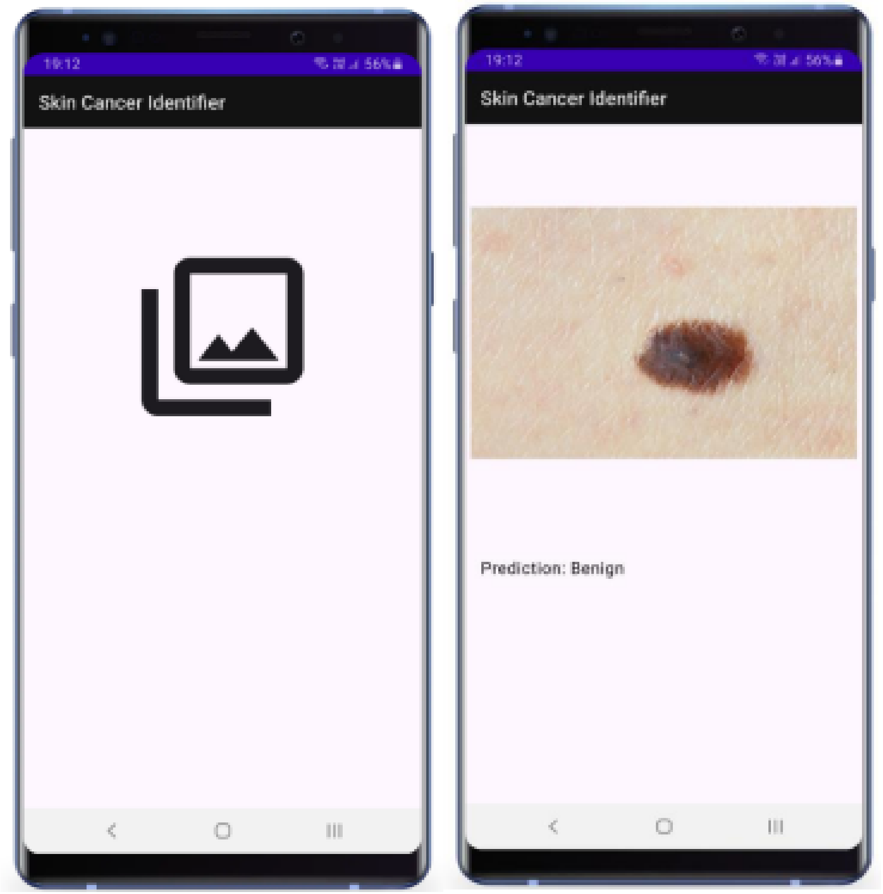
2. Execute `Metadata.ipynb` with the `model.tflite` to add metadata to your model.

3. Replace the existing `model.tflite` in the `SkinCancerIdentifier` application with the updated version.

4. Open the `SkinCancerIdentifier` application in Android Studio, and run it on either an emulator or a physical device.

5. On the home screen, tap the image, then choose the appropriate image file from your device.

6. The application will display its prediction using the model.



Prediction: Benign

Figure 1: Cancer Identifier application

8 Conclusion

Our project demonstrates the potential of leveraging advanced deep learning techniques, in the critical field of medical image processing. By using an autoencoder architecture and integrating it into a compact, accurate model, we aim to create a tool that can assist in skin cancer detection. The foundation of our work lies in developing a solution that is not only effective but also practical for deployment on mobile devices, thereby making preventive healthcare more accessible. Although our system is still under development, it embodies the spirit of innovation and dedication to solving real-world problems.

References

[1] Kaggle - ISIC 2024 - Skin Cancer Detection with 3D-TBP
`https://www.kaggle.com/competitions/isic-2024-challenge`
Last accessed: 2025-01-05

[2] The ISIC 2024 Challenge Dataset
`https://challenge2024.isic-archive.com`

Last accessed: 2025-01-05

[3] Datacamp - Introduction to Autoencoders
`https://www.datacamp.com/tutorial/introduction-to-autoencoders`
Last accessed: 2025-01-05

[4] Medium - Applied Deep Learning
`https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798`
Last accessed: 2025-01-05

[5] Pyimagesearch - Autoencoders
`https://pyimagesearch.com/2020/02/17/autoencoders-with-keras-tensorflow-and-deep-learning`
Last accessed: 2025-01-05

[6] Google - Adding metadata to TFLite models
`https://ai.google.dev/edge/litert/models/metadata`
Last accessed: 2025-01-05

[7] Medium - TensorFlow Lite model implementation in Android
`https://medium.com/walmartglobaltech/custom-tensorflow-lite-model-implementation-in-android-5c1c65bd9f97`
Last accessed: 2025-01-05

[8] NIH - Skin Lesion Classification Using Densely Connected Convolutional Networks with Attention Residual Learning
`https://pmc.ncbi.nlm.nih.gov/articles/PMC7764313`
Last accessed: 2025-01-05

[9] NIH - Skin Lesion Classification Using Additional Patient Information
`https://ai.google.dev/edge/litert/models/metadata`
Last accessed: 2025-01-05