taingocbui / **phase5_project**

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

phase5_project

Public

1 Branch 0 Tags Go to file t Go to file + Add file About Code ...

taingocbui update README tree 3a3dd4d · 5 days ago

models	update models folder	5 days ago
photos	update README.md an...	2 weeks ago
.gitignore	Update notebook base ...	last month
README.md	update README tree	5 days ago
project5.ipynb	Created using Colab	5 days ago

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

● Jupyter Notebook 100.0%

README

Skin Cancer Detection using Convolution Neural Network, ResNet50, XGBoost

Author: Tai Ngoc Bui
Date Completion: December 7th, 2024

1.Business Understanding

This project focuses on developing a robust Convolutional Neural Network

(CNN) capable of classifying skin lesions as malignant or benign. Skin cancer is one of the most prevalent cancers worldwide, and early detection significantly improves treatment outcomes and survival rates. Despite advancements in medical technology, many cases are either detected too late or misdiagnosed due to the limitations of traditional diagnostic methods. By leveraging deep learning techniques, I aim to provide a reliable, efficient, and scalable solution to assist dermatologists and healthcare professionals in diagnosing skin cancer.

This topic is personally significant as it combines two areas of great interest: leveraging artificial intelligence to solve real-world problems and contributing to public health initiatives. On a broader level, this research has societal implications, potentially saving lives and reducing healthcare costs. The target audience for this project includes not only potential patients of skin cancer but also dermatologists and oncologists seeking diagnostic support tools, as well as the whole healthcare system. To successfully complete this project, I relied on numerous studies and projects highlighting the performance of CNNs like ResNet, InceptionNet and MobileNet for skin cancer detection. Moreover, studies on transfer learning to improve model performance with limited datasets are also utilized in this project.

This model will use recall rate as the main metric to prioritize the ability of the model to correctly identify all positive cases (e.g. malignant cancer). In medical diagnoses, a false negative (failing to detect cancer when it is present) can have severe consequences, such as delayed treatment or worsened prognosis. Recall ensures that the model minimizes false negatives, even if it occasionally produces false positives. Missing a malignant case is riskier than flagging a benign case as malignant, as false positives can often be corrected through follow-up procedures.

2.Data Understanding

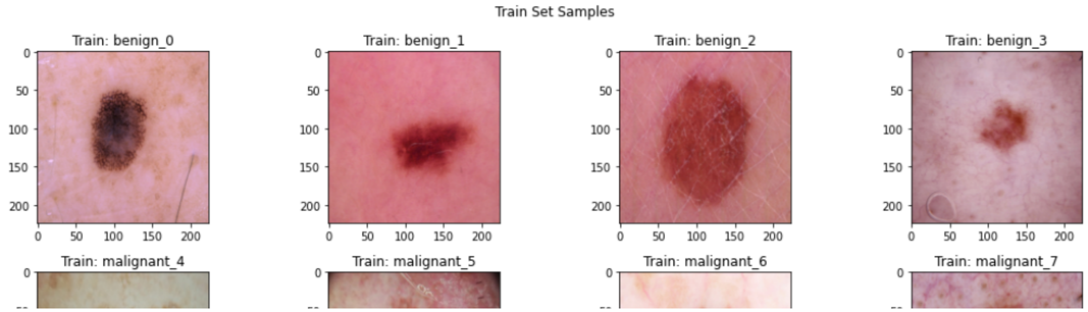
The dataset used in this project is sourced from Kaggle’s Skin Cancer: Malignant vs. Benign. It consists of two folders: train and test with images (224x244) of the two types of moles including benign skin moles and malignant skin moles. It is curated for machine learning research and adheres to ethical standards for data use. The primary features include images of skin lesions data and categorical labels indicating whether the lesion is benign or malignant. Several researchers have used this dataset to train and evaluate machine learning models for skin cancer detection.

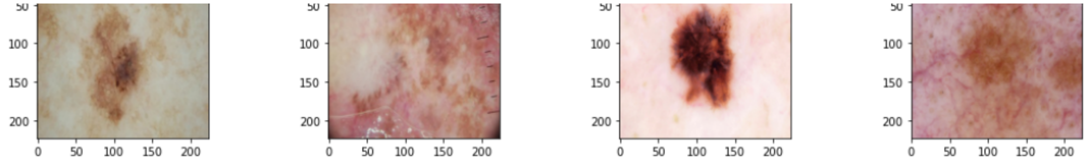
3. Objectives

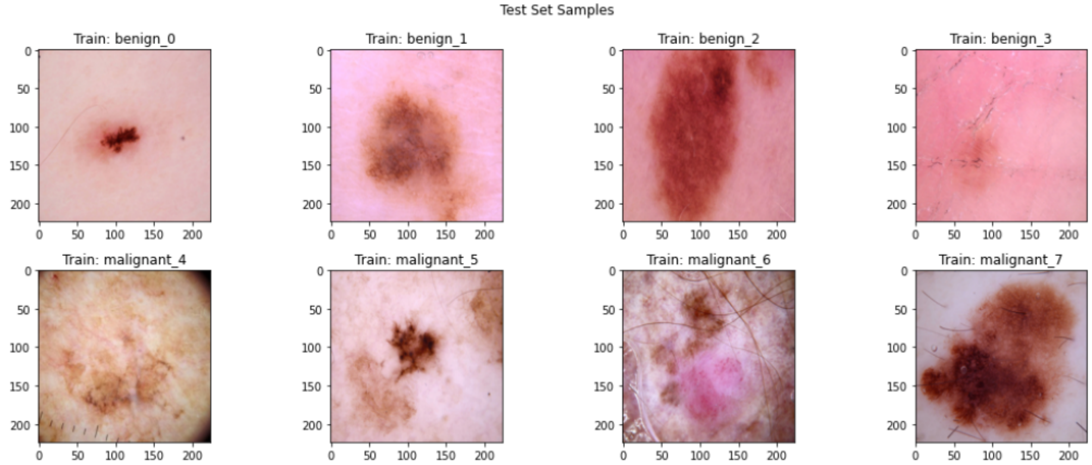
Develop a robust Convolutional Neural Network (CNN) capable of classifying skin lesions as malignant or benign using skin images. This classifying model should achieve a recall rate at least 80% to correctly identify all positive cases.

4. Exploratory Analysis

The data is stored as jpg image files and already separated into train and test folder. Each image (224x244) represents a skin lesion and is a collection of pixel values, typically stored as arrays of integers ranging from 0 to 255 (grayscale) or three-channel RGB values. The labels are categorical, indicating whether the lesion is benign (label 0) or malignant (label 1).







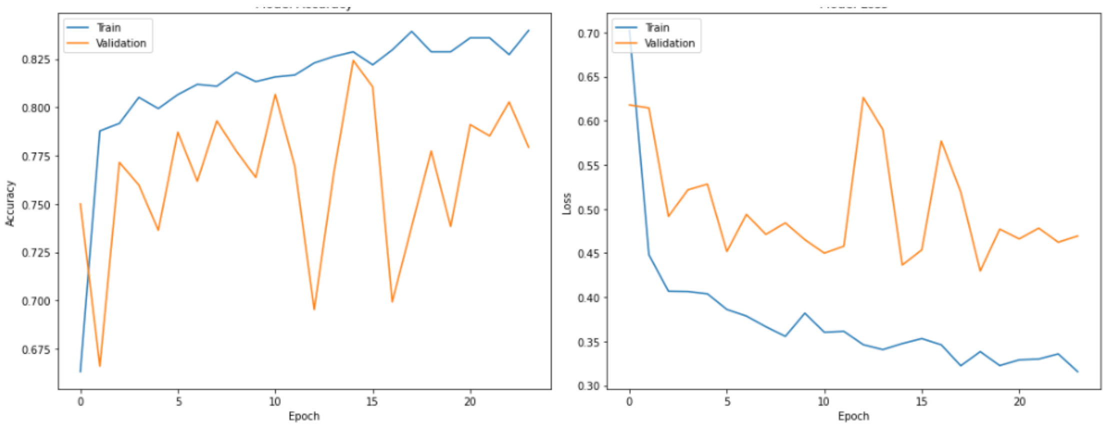
a. Base Model (CNN with shallow layers)

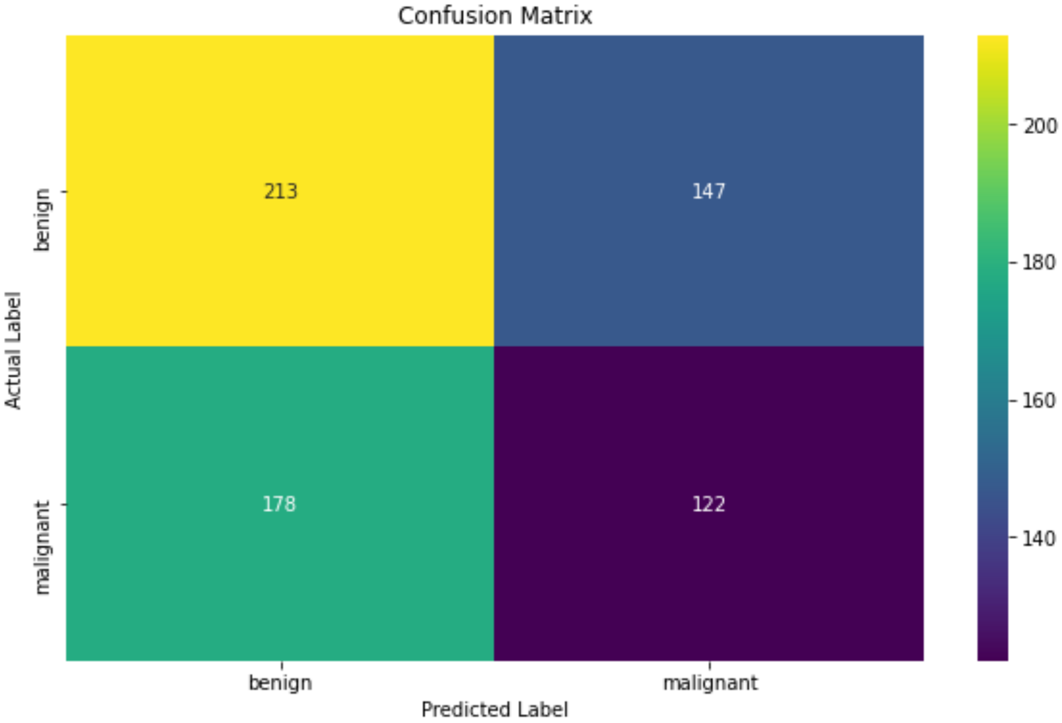
To start with, I used 3 Convolutional layers with increasing filter sizes (32, 64, 128). The "relu" activation function is used in each of the convolution layer to introduce non-linearity, enabling the network to learn complex patterns and relationships in data. Early Convolutional layers detect simple features such as edges, corners, or textures. As the network goes deeper, it needs to learn more complex patterns, such as shapes, structures, and semantic representations. Increasing the number of filters allows the network to capture more varied and complex patterns. Filters act as feature detectors. More filters mean the model can learn a richer set of features at each layer, improving its ability to generalize across diverse input data. On the other hand, starting with too many filters in early layers can lead to overfitting or unnecessary computational overhead since these layers only need to learn simple features. A gradual increase ensures a smooth transition between layers and allows the network to build on the features learned in previous layers. Followed each convolutional layer, a pooling layer is used to reduce the spatial dimensions of the feature maps.

After Convolutional and Pooling layers, a Dense layer (fully connected) combines these features into a high-dimensional representation. A layer with 256 neurons provides a balance between model capacity and computational efficiency. This size is large enough to capture meaningful patterns but small enough to avoid overfitting.

Compiling a CNN model involves specifying the loss function, optimizer, and evaluation metrics. I use binary cross entropy as the loss function, a common loss function for binary classification. With the optimizer, I used Adam optimization, a commonly used optimizer. Adam optimization not only incorporates the advantages of two popular optimization techniques, the Momentum and RMSProp, but also helps the model converge faster compared to vanilla SGD by efficiently navigating through flat or steep regions of the loss surface.

I also use 2 callbacks EarlyStopping and ModelCheckpoint to prevent overfitting, save computational resources by halting training when improvements plateau, and ensure that the best model (based on validation loss) is saved during training. The model trains for up to 50 epochs but may stop earlier if validation loss does not improve for 5 epochs due to EarlyStopping. During training, the model saves the best weights (with the lowest validation loss) to 'cnn_model_best_weights.h5' using ModelCheckpoint. The history object stores the training and validation loss and metrics for each epoch, which can be used for analysis or plotting. This CNN model shows a decent performance with the test set with 79.70% accuracy and AUC is 0.9053. The recall rate for the model is 40.66% despite accuracy achieves 81.67%.



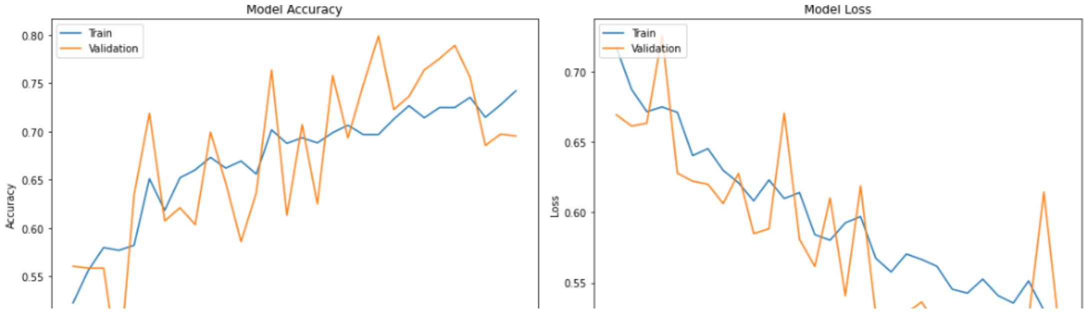


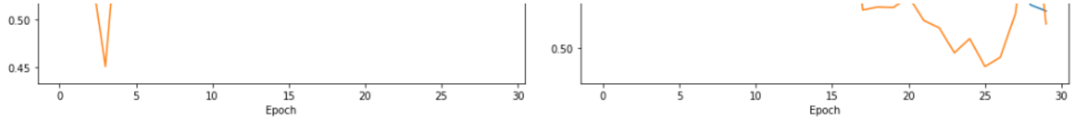
b. ResNet50

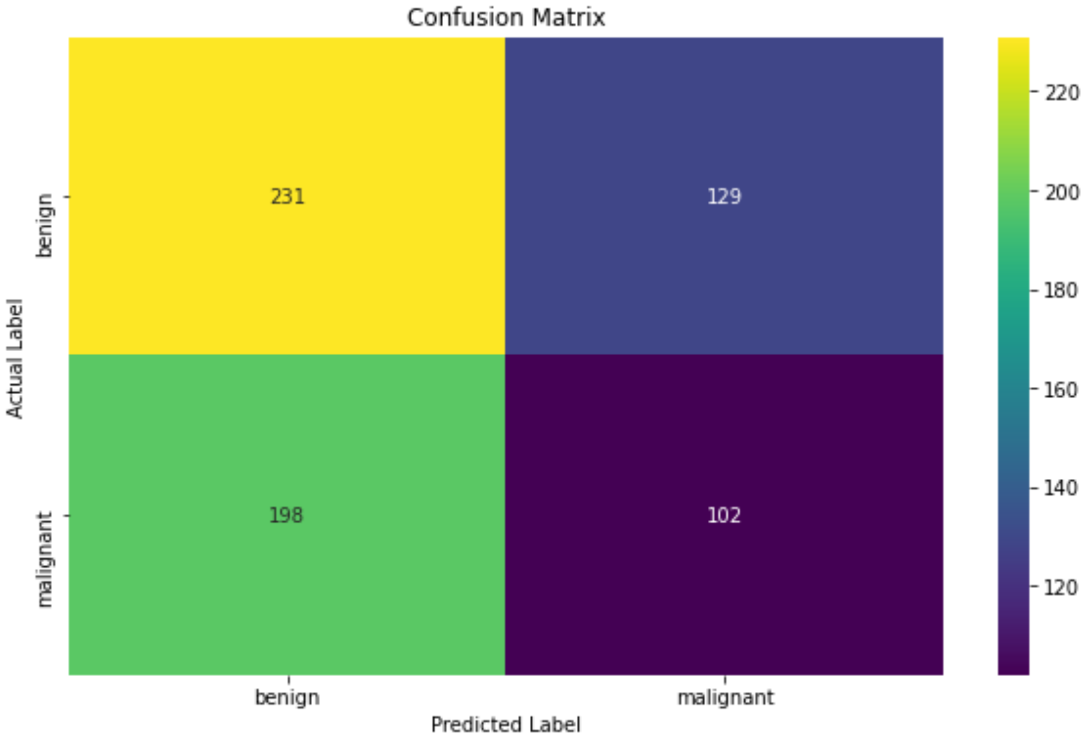
To further improve the classification model, I will apply a pre-trained model, ResNet50 for this classification problem. ResNet50 is a deep convolutional neural network (CNN) with 50 layers, part of the Residual Network (ResNet) family, designed to address the vanishing gradient problem and enable the training of very deep networks. It uses residual connections (skip connections) to allow gradients to flow through the network more effectively.

The reason I decide to use RestNet 50 is due to its proven success in various medical image classification tasks. ResNet50 is well-suited for medical image detection problems, including tasks like skin cancer classification, due to its innovative architecture and ability to handle complex features in image data. On the other hand, I also prevents the weights of the pre-trained layers from being updated during training. Freezing ensures that the base model serves as a fixed feature extractor while the custom layers learn task-specific patterns. Furthermore, I do not use the ResNet50's last layer. Here, I decide to customize the last layer by adding a dense layer with 256 nodes and a relu activation function. The output layer will use a sigmoid activation function, a common activation used in binary classification. This last customize layer will ensure all features are considered for the output.

With the use of Resnet50 model, there is no improvement in both recall rate for the malignant class and accuracy. In fact, both the recall rate and accuracy drastically dipped (34% recall rate and 72.27%). The cause of such low recall rate may result from lack of preprocessing process. Though I already did certain image augmentation process for training set, such process may not be robust enough for the ResNet50 model.



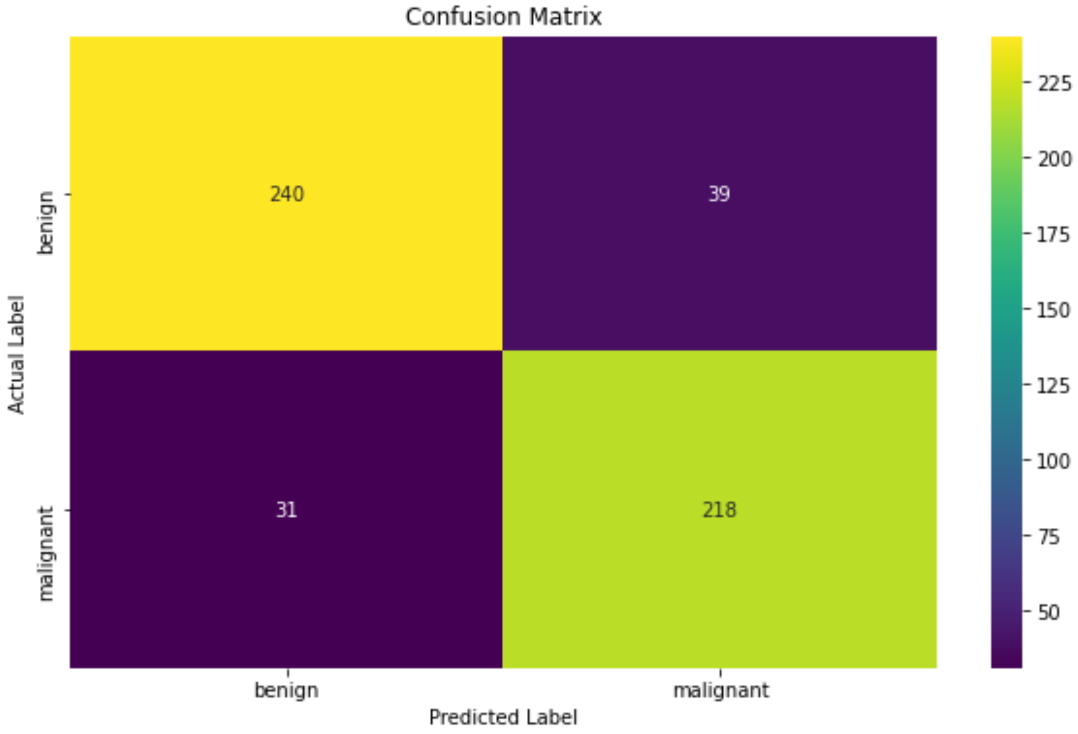




c. Hybrid Model with XGBoost and ResNet50

In the final part of this project, I will create a hybrid model, combining ResNet50's feature extraction process with the XGBoost model. XGBoost (eXtreme Gradient Boosting) is an efficient and scalable implementation of the gradient boosting algorithm. It builds an ensemble of decision trees, where each tree corrects the errors of the previous ones, optimizing a loss function. A hybrid model combining ResNet50 for feature extraction and XGBoost for classification often performs better than using ResNet50 alone for end-to-end classification. While ResNet50 specializes in extracting high-level features from images (e.g., edges, textures, shapes), XGBoost excels in handling structured data and edges, textures, shapes)), XGBoost excels in handling structured data and learning complex decision boundaries efficiently.

This hybrid model performed significantly better than both the base line CNN model and the ResNet50 model. This hybrid model achieve a 88% recall rate and 87.5% accuracy rate with the test set. Fine-tuning ResNet50 end-to-end on a small dataset can lead to overfitting due to the large number of parameters. Using ResNet50 as a frozen feature extractor reduces the risk of overfitting by focusing only on pre-trained feature extraction. XGBoost, with regularization parameters, helps generalize well even on small datasets.



5. Conclusion

Based on our analysis and testing with different models, I want to recommend a hybrid model combining ResNet50's feature extraction process with the XGBoost model for this cancer detection problem. This final hybrid model not only achieved the highest accuracy rate among the 3 models, it also maximized the recall rate, prioritize the ability of the model to correctly identify all positive cases. In medical diagnoses, a false negative (failing to detect cancer when it is present) can have severe consequences, such as delayed treatment or worsened prognosis. Recall ensures that the model minimizes false negatives, even if it occasionally produces false positives.

6. Future Works

To better improve the quality of this project, I will extend this project by investigate other popular models used in medical detection field such as VGG16, InceptionV3 or MobileNetV3. Moreover, a hybridd between these new models may potentially improve the recall rate for this skin cancer detection problem.

Project tree

- photos
  - Test_samples.png
  - Train_samples.png
  - CM_base.png
  - Validation_Loss_base.png
  - CM_RN50.png
  - Validation_Loss_RN50.png
  - CM_hybrid.png
- models

    - hybrid_xgboost_model.pkl
- project5.ipynb
- README.md