Cancer Cell Detection using the HAM10000 Dataset

By: Shreya Hegde

Problem Statement: Cancer cell detection and preparing the model based on the dataset.

Dataset: https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000

Tools used: Anaconda, Python, Numpy, Pandas, Matplotlib, Autokeras, Seaborn, Sklearn, CNN(Convolutional Neural Networks).

1. Overview of the Problem Statement:

The problem is to classify skin cancer lesions into seven different classes using the HAM10000 dataset. The goal is to develop a machine learning model that can accurately classify skin lesions based on their images.

2. About the Dataset:

The HAM10000 dataset contains 10,015 dermatoscopic images of skin lesions, which are classified into seven classes: Melanocytic nevi (nv), Melanoma (mel), Benign keratosis-like lesions (bkl), Basal cell carcinoma (bcc), Actinic keratoses (akiec), Vascular lesions (vas), and Dermatofibroma (df). Each image is associated with additional metadata such as patient information, image quality, and diagnosis.

3. Approach:

3.1 Data Preprocessing:

- Load the metadata from the CSV file provided in the dataset.

- Perform label encoding to convert the text labels into numeric values.

- Visualize the distribution of data by plotting the counts of each class, sex, localization, and age.

- Balance the data by resampling each class to have an equal number of samples.

3.2 Data Exploration and Visualization:

- Read and preprocess the images based on the image IDs from the metadata.
- Resize the images to a consistent size for model input.
- Visualize a subset of the images from each class to understand the characteristics of different skin lesions.

3.3 Model Development:

- Split the preprocessed data into training and testing sets.
- Design a convolutional neural network (CNN) model using the Keras library.
- The model consists of several convolutional layers, max pooling layers, dropout layers, and dense layers.
- The final output layer uses the softmax activation function for multi-class classification.
- Compile the model with the categorical cross-entropy loss function and the Adam optimizer.

3.4 Model Training:

- Train the model using the training data.
- Specify the batch size and the number of epochs for training.
- Monitor the training process and evaluate the model's performance on the validation set.

3.5 Evaluation and Results:

- Evaluate the trained model on the testing data.
- Calculate the accuracy of the model on the test set.
- Plot the training and validation loss and accuracy curves to analyze the model's performance.

3.6 Performance Analysis:

- Make predictions on the test data using the trained model.
- Convert the predicted classes and true labels to one-hot vectors.
- Calculate the confusion matrix to evaluate the model's performance.
- Visualize the confusion matrix and plot the fraction of incorrect predictions for each class.

4. Reasoning:

- The chosen approach involves training a CNN model on the HAM10000 dataset to classify skin cancer lesions.
- CNNs are effective for image classification tasks as they can learn hierarchical features from images.
- The dataset is imbalanced, so data balancing techniques such as resampling are used to ensure equal representation of each class.
- The model architecture includes convolutional layers for feature extraction, max pooling layers for downsampling, and dense layers for classification.
- Dropout layers are added to reduce overfitting during training.
- The softmax activation function is used in the output layer for multi-class classification.
- The Adam optimizer is chosen for efficient gradient-based optimization.
- Evaluation metrics such as accuracy, loss, and confusion matrix are used to assess the model's performance.

4.1 Automatic Model Selection with AutoKeras:

- AutoKeras is a powerful tool that automates the process of model selection and hyperparameter tuning.
- It uses a technique called neural architecture search (NAS) to explore different model architectures and find the best one for the given dataset.
- In this approach, AutoKeras was used to search for the best model architecture for skin cancer lesion classification.
- By using AutoKeras, the model architecture and hyperparameters are automatically determined based on the dataset, saving significant time and effort.
- AutoKeras explores a range of possible architectures, including different types and numbers of layers, to find the optimal combination for the given task.
- The selected architecture from AutoKeras is then trained on the dataset to obtain the final model.

The overall goal of this approach is to develop a robust and accurate model for skin cancer lesion classification using the HAM10000 dataset. By leveraging AutoKeras for automatic model selection, the process is further optimized to find the best model architecture for the task. This helps in improving the model's performance and reducing the manual effort required for architecture exploration.