

Name : Khan Mohammad Tahir

Email id : tahir khan93228@gmail.com

LinkedIn : <https://www.linkedin.com/in/khan-tahir-4aa8aa191/>

Contact : 9326374034

Assignment

Skin Cancer Detection

I performed skin cancer detection using deep learning , it is a technique that utilizes deep neural networks to classify skin lesions and detect the presence of skin cancer. The given dataset, HAM10000_metadata.csv, is likely a metadata file containing information about skin lesion images such as their labels, clinical attributes, and other relevant information.

Reason behind taking this project :

The reason why I have taken this project can be multifaceted, but it likely revolves around my desire to make a positive impact in the field of healthcare.

By undertaking this project, I aim to leverage the power of deep learning and data analysis techniques to develop an effective skin cancer detection system.

This project presents an opportunity to contribute towards early detection and diagnosis of skin cancer, which can significantly improve patient outcomes and save lives.

Additionally, by working on this project, I am driven by a curiosity to explore the potential of artificial intelligence in the medical domain and advance the understanding and application of machine learning algorithms in real-world scenarios.

Overall, the motivation behind taking this project stems from a combination of a passion for healthcare, the desire to make a meaningful difference, and the excitement of pushing the boundaries of technology and innovation.

Following are the steps that is used for the development of a project :

Step 1 : Importing the required libraries , modules and dataset for the skin cancer detection project

Library used :

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Tensorflow
- Keras
- Sklearn

Link of dataset used :

- <https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000>
- https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000?select=hmnist_28_28_RGB.csv
- https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000?select=HAM10000_images_part_1 (For images)

Step 2 : Data Preprocessing

In our skin cancer detection project I performed several data preprocessing steps to prepare the dataset for analysis and model training. I handled missing data by imputing missing values, and preprocessed the images by resizing them to a consistent resolution and normalizing pixel values. I also encoded categorical variables, such as the type of cells and disease localization, into numerical representations.

Step 3 : Visualizing the dataset for analysis

In the step of visualizing the dataset for analysis in our project, Various data visualization techniques has been utilized to gain insights and understand the characteristics of the dataset.

I also visualized the distribution of different cell types using count plots, providing an overview of the class distribution , also examined the gender distribution of patients using a pie chart and analyzed the distribution of age using a histogram. Furthermore, I explored the localization of the disease over gender using a bar plot, highlighting the frequency of different disease locations.

These visualizations allowed us to identify patterns, trends, and imbalances in the dataset, aiding in the subsequent analysis and decision-making processes for skin cancer detection.

Step 4 : Importing new data for visualizing the diseases

In the given code, I performed several operations on the image data. First, we loaded the image data from a CSV file using Pandas and examined its content. Then, I also rescaled the image data by dividing it by 255 to normalize the pixel values between 0 and 1. I checked the shape of the rescaled image data to verify its dimensions.

Next, i accessed a specific row of the rescaled image data and retrieved its values. I reshaped the values into a 28x28x3 array representing an RGB image , then visualized a grid of 49 images from the rescaled image data using a subplot and imshow function from Matplotlib.

After that, First inspected the first few rows of the cancer data to ensure its structure. Finally, i applied a lambda function to each path in the cancer data to open the corresponding image, resize it to 100x100 pixels, and convert it to a NumPy array. I stored the resulting images in the 'image' column of the cancer data. Lastly, i used another lambda function to map the shape of each image in the 'image' column and obtained the value counts of the different image shapes in the dataset.

Step 5 : Splitting data into Independent(X) and Dependent(y)

In the step of splitting data into independent (X) and dependent (y) variables, the dataset is divided into two separate entities that are crucial for machine learning tasks.

The independent variables (X) represent the input features or attributes that we use to make predictions or classify data, while the dependent variable (y) represents the target variable we want to predict or classify.

Step 6 : Model Building

In Step 6, Model Building, I develop the machine learning model for skin cancer detection using the preprocessed data. This involves selecting an appropriate algorithm or architecture for the task, such as a convolutional neural network (CNN) which is commonly used for image classification tasks.

Step 7 : Model Evaluation

In the model evaluation step of the project, I assess the performance and effectiveness of the skin cancer detection model and also apply the trained model to the testing dataset and evaluate its predictions against the actual labels. Various evaluation metrics such as accuracy, precision, recall and F1-score are used to measure the model's performance.

I got accuracy about 76% on testing dataset.

Step 8 : Model Prediction

In the model prediction step of the project, i utilize the trained skin cancer detection model to make predictions on new, unseen data. I input the independent variables (features) of the new data into the model and obtain the predicted output or class labels.

These predictions help us determine whether a given instance of skin cancer is malignant or benign based on the learned patterns and features from the training data. The model uses its learned parameters and weights to make these predictions, providing valuable insights and information for medical professionals and decision-making processes related to skin cancer diagnosis and treatment.

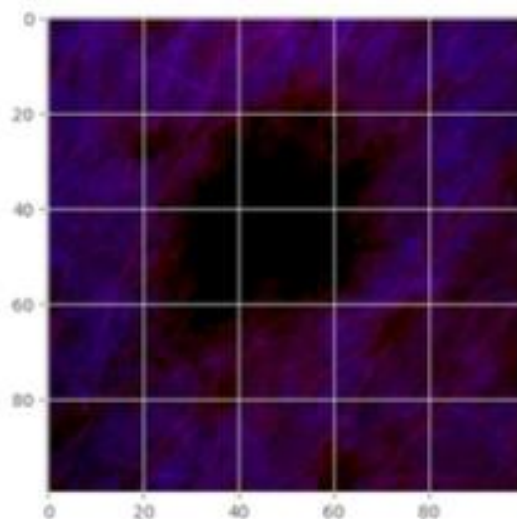
Step 9 : Visualizing Output of the project

Step 9 : Visualizing Our Output

```
plt.figure(figsize = (5,5))
plt.imshow(X_test[60] , "gray")
print(f"Actual Label : {y_test[60]}")
print(f"Predicted Label : {model_prediction[0]}")
if model_prediction[10] == 0:
    print("Cancer Disease Name : Actinic keratoses ")
elif model_prediction[10] == 1:
    print("Cancer Disease Name : Basal cell carcinoma")
elif model_prediction[10] == 2:
    print("Cancer Disease Name : Benign keratosis-like lesions")
elif model_prediction[10] == 3:
    print("Cancer Disease Name : Dermatofibroma")
elif model_prediction[10] == 4:
    print("Cancer Disease Name : dermatofibroma ")
elif model_prediction[10] == 5:
    print("Cancer Disease Name : Melanocytic nevi ")
else:
    print("Cancer Disease Name : Vascular lesions ")
```

WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).

Actual Label : 2
Predicted Label : 2
Cancer Disease Name : Melanocytic nevi



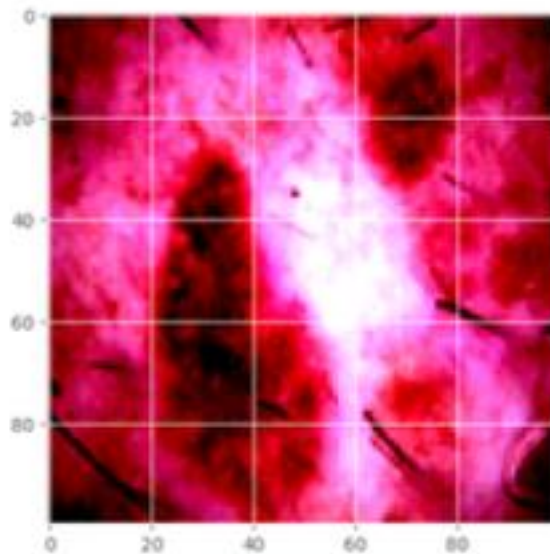

```

plt.figure(figsize = (5,5))
plt.imshow(X_test[100] , "gray")
print(f"Actual Label : {y_test[100]}")
print(f"Predicted Label : {model_prediction[100]}")
if model_prediction[10] == 0:
    print("Cancer Disease Name : Actinic keratoses ")
elif model_prediction[10] == 1:
    print("Cancer Disease Name : Basal cell carcinoma")
elif model_prediction[10] == 2:
    print("Cancer Disease Name : Benign keratosis-like lesions")
elif model_prediction[10] == 3:
    print("Cancer Disease Name : Dermatofibroma")
elif model_prediction[10] == 4:
    print("Cancer Disease Name : dermatofibroma ")
elif model_prediction[10] == 5:
    print("Cancer Disease Name : Melanocytic nevi ")
else:
    print("Cancer Disease Name : Vascular lesions ")

```

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

Actual Label : 0
Predicted Label : 0
Cancer Disease Name : Melanocytic nevi



```

plt.figure(figsize = (5,5))
plt.imshow(X_test[1000] , "gray")
print(f"Actual Label : {y_test[1000]}")
print(f"Predicted Label : {model_prediction[1000]}")
if model_prediction[10] == 0:
    print("Cancer Disease Name : Actinic keratoses ")
elif model_prediction[10] == 1:
    print("Cancer Disease Name : Basal cell carcinoma")
elif model_prediction[10] == 2:
    print("Cancer Disease Name : Benign keratosis-like lesions")
elif model_prediction[10] == 3:
    print("Cancer Disease Name : Dermatofibroma")
elif model_prediction[10] == 4:
    print("Cancer Disease Name : dermatofibroma ")
elif model_prediction[10] == 5:
    print("Cancer Disease Name : Melanocytic nevi ")
else:
    print("Cancer Disease Name : Vascular lesions ")

WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).

Actual Label : 1
Predicted Label : 1
Cancer Disease Name : Melanocytic nevi

```

