Data Collection and Preprocessing Phase

Date	22 October 2024
Team ID	SWTID1727274979
Project Title	Deep learning techniques for breast cancer risk prediction
Maximum Marks	6 Marks

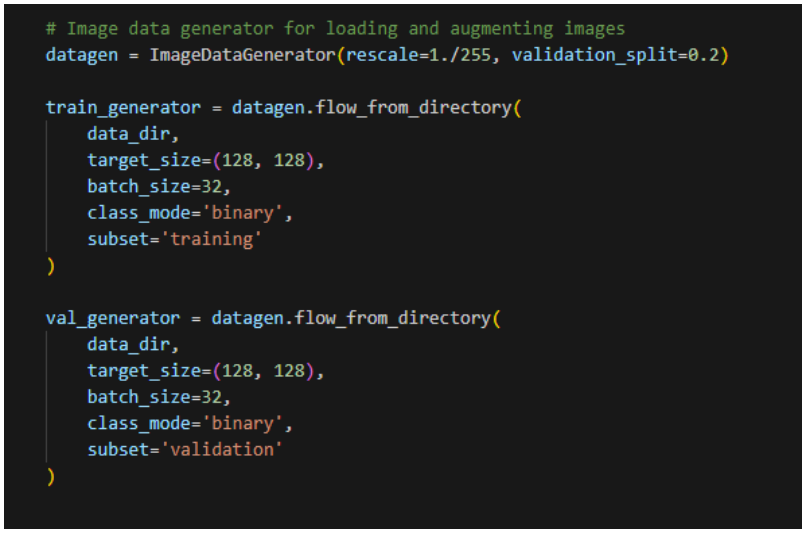
Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, and performing other relevant transformations to enhance data quality and ensure model generalization. This will improve the convergence during neural network training and lead to a robust and efficient performance across various computer vision tasks.

Section	**Description**
Data Overview	The dataset used for this project is the **Breast Histopathology Images** dataset, available on Kaggle. This dataset consists of images that represent breast cancer cells labeled as benign (class 0) or malignant (class 1). The images are of different sizes, and we preprocess them to standardize the input for the deep learning model.
Resizing	Images are resized to a target size of 128x128 pixels for consistency, ensuring uniformity in the input dimensions for the neural network.
Normalization	The pixel values of the images are normalized to the range [0, 1] by dividing by 255.0. This normalization helps the model converge faster and more effectively during training.
Data Augmentation	Apply augmentation techniques such as flipping, rotation, shifting, zooming, or shearing.
Denoising	Apply denoising filters to reduce noise in the images.

Edge Detection	Apply edge detection algorithms to highlight prominent edges in the images.
Color Space Conversion	Convert images from one color space to another.
Image Cropping	Crop images to focus on the regions containing objects of interest.
Batch Normalization	Apply batch normalization to the input of each layer in the neural network.

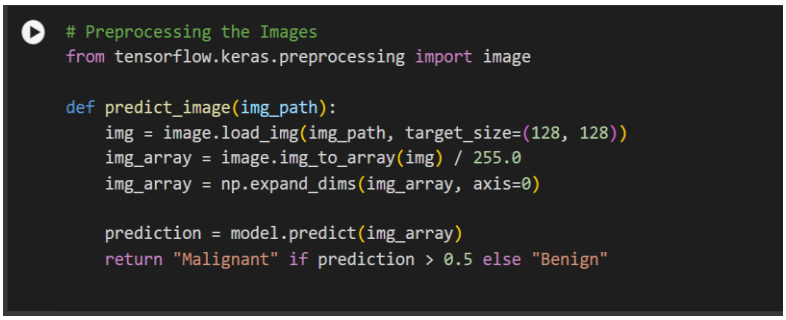
Data Preprocessing Code Screenshots

Loading Data	`datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)`
Resizing	`target_size=(128, 128),`
Normalization	`datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)`
Data Augmentation	

```python
# Image data generator for loading and augmenting images
datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)

train_generator = datagen.flow_from_directory(
    data_dir,
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary',
    subset='training'
)

val_generator = datagen.flow_from_directory(
    data_dir,
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary',
    subset='validation'
)
```

Edge Detection	--
Image Cropping	
Batch Normalization