Data Collection and Preprocessing Phase

Date	11-03-2025
Team ID	740037
Project Title	Early Stage Disease Diagnosis System Using Human Nail Image Processing Using Deep Learning.
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template:

It focuses on cleaning and analyzing image and meta data for Early Stage Disease Diagnosis System assessment. It includes steps for normalizing images and extracting features like damage severity and part location. This preprocessing ensures high quality inputs for accurate estimation and validation.

Section	Description

Image Augmentation	**Image data augmentation**
	
	``python
	# Setting parameters for data augmentation
	train_datagen = ImageDataGenerator(
	    rescale=1./255,
	    shear_range=0.1,
	    zoom_range=0.1,
	    horizontal_flip=True
	)
	val_datagen = ImageDataGenerator(rescale=1./255)
	``

Body Damage	**For Body Damage**
	
	``python
	# Flow from directory
	training_set = train_datagen.flow_from_directory(
	    trainPath,
	    target_size=(224, 224),
	    batch_size=10,
	    class_mode='categorical'
	)
	test_set = val_datagen.flow_from_directory(
	    testPath,
	    target_size=(224, 224),
	    batch_size=10,
	    class_mode='categorical'
	)
	``
	
	Found 1150 images belonging to 1 classes.
	Found 1150 images belonging to 1 classes.

Level Damage	**For the level of damage**
	
	``python
	# Flow from directory
	training_set = train_datagen.flow_from_directory(
	    trainPath,
	    target_size=(224, 224),
	    batch_size=10,
	    class_mode='categorical'
	)
	test_set = val_datagen.flow_from_directory(
	    testPath,
	    target_size=(224, 224),
	    batch_size=10,
	    class_mode='categorical'
	)
	``
	
	Found 1150 images belonging to 1 classes.
	Found 1150 images belonging to 1 classes.

Data Preprocessing Code Screenshots	
Image pre-processing	```#Adding preprocessing layer to the front of vgg
vgg=VGG16(input_shape=list(imageSize) + [3], weights='imagenet',include_top=False)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 ———————— 0s 0us/step``` |
| | ```for layer in vgg.layers:
    layer.trainable=False

x=Flatten()(vgg.output)``` |
| | ```#Adding output layer
prediction = Dense(4, activation='softmax')(x)``` |
| Feature Engineering | Attached the codes in final submission. |
| Save Processed Data | - |