**TASK-2**

**Introduction:**

This project aims to develop a deep learning-based image classification model for the Common Fever Causing Disease dataset, which includes images of four different disease objects. The model will use transfer learning architectures like ResNet50 and VGG19 model techniques to maximize accuracy. The project will evaluate the model's performance using metrics and key insights.

**Data Preparation:**

1. Based on the raw image data, created a dataset by resizing the image into (150,150) into the X variable and appending the virus name into the y variable as the list.
2. Checked the shape of the images in the X variable as (3878, 150, 150, 3) and the y variable as (3878,).
3. Plotted 10 random images from the data.
4. Additionally, looked at the distribution of the disease image according to the virus name, where:
   1. Adeno has 1142
   2. Rhino has 781
   3. Entero has 1000
   4. Influenza has 955
5. Created a mapping dictionary for the target variable (virus name) by mapping the virus names to numerical values.
6. Converted the target variable to a vector using the to\_categorical function.

A graph of a number of red rectangular bars

Description automatically generated

**Feature Engineering:**

* Implemented the split of the data into train, test, and validation using the train\_test\_split function from sklearn.model\_selection.
* Used a random state of 42 to ensure the random numbers generated during the training process remain consistent every time the code is executed.
* Divided the data into 80% training and 20% testing for training and testing purposes.
* Took the above 80% training data and split it further into an 80:20 ratio to create the Validation dataset.
* Performed scaling and reshaping for the input images to keep the Input size consistent.
* Converted the target variable to an array for the training, testing, and validation datasets.

**Modeling:**

* For Implementing the Transfer learning, Pick the ResNet50 and VGG19
* Utilised a pre-trained model with weights from the ImageNet.
* By setting include\_top=False, you are instructing the model to exclude the final classification layer and instead return the output of the last convolutional layer to benefit the model during Transfer learning, Flexibility, and Performance.
* Updated the input shape to 150x150 pixels with 3 color channels.
* Freezing the layers to prevent the pre-trained weights from being updated during the training process.
* Created a new sequential model and added the pre-trained model as a layer.
* Flattened the output into a 1D tensor, this is typically necessary before passing the data to fully connected (dense) layers, which expect one-dimensional input.
* Added a dense layer with the SoftMax activation function, which is mostly used for classification tasks as an output layer.
* For both models, the Adam optimizer is used, which potentially improves the model's convergence and performance compared to other optimizers.
* Categorical Cross-Entropy is used as the loss function.
* The learning rate is set to 0.001.
* The batch size is set to 64.
* Epochs is set to 25
* The early stopping callback monitors the validation loss during training.
* Training is stopped early if the validation loss doesn't improve for 8 consecutive epochs.
* This prevents overfitting and saves computational resources.

**Results:**

The resultant tables give a better understanding of how the model worked for evaluation and validation based on the data.

|  |  |  |
| --- | --- | --- |
| Model | Evaluation(Accuracy) | Validation(Accuracy) |
| ResNet50 | 0.99 | 0.99 |
| VGG19 | 1.00 | 0.99 |

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

The ResNet50 model effectively learns training data patterns, achieving high accuracy and low loss on the training set, as indicated by the overall trend in both plots.

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

The VGG19 model effectively learns training data patterns, achieving high accuracy and low loss on the training set, as indicated by the overall trend in both plots.

**Visualization:**

A collage of images of heat

Description automatically generated

This visualization shows how the ResNet50 model predicted the disease with the actual value of the disease.

A collage of images of heat

Description automatically generated

This visualization shows how the VGG19 model predicted the disease with the actual value of the disease.

**Models Architecture:**

**VGG19:**

* Consists of 19 layers (16 convolutional layers, 3 fully connected layers).
* Uses very small (3x3) convolution filters throughout the network.
* Has a straightforward and uniform architecture with a series of convolutional and max-pooling layers followed by fully connected layers.
* Known for its simplicity and depth, which helps in learning rich feature representations.

**ResNet50:**

* Consists of 50 layers.
* Incorporates residual connections (skip connections) to combat the vanishing gradient problem, allowing for much deeper networks.
* Utilizes bottleneck blocks (3 layers: 1x1, 3x3, 1x1 convolutions) to reduce the number of parameters while maintaining depth.
* Known for its ability to train very deep networks without suffering from degradation problems.
* Why VGG19 Outperformed ResNet50

**Feature Extraction:**

* VGG19’s architecture, with its uniform convolutional layers and pooling layers, might be better suited for extracting relevant features from the fever-causing disease dataset.
* The consistent use of small 3x3 filters in VGG19 could provide more fine-grained feature extraction, beneficial for the dataset.

**Dataset Characteristics:**

* The specific characteristics of the fever-causing disease dataset might favor the simpler, more uniform architecture of VGG19 over the more complex residual connections in ResNet50.
* The diseases’ visual features might be more effectively captured by the depth and design of VGG19's **layers.**

**Conclusion:**

**VGG19:**

* Simpler, more uniform architecture.
* Effective for the given dataset, possibly due to better feature extraction capabilities with small filters.

**ResNet50:**

* More complex, deeper architecture with residual connections.
* Generally, performs well across many tasks but may not be as efficient for this specific dataset compared to VGG19.

Overall, both models are powerful, but the choice between them can depend on the specific dataset and task.

For the common fever-causing disease dataset, VGG19 slightly outperformed ResNet50, likely due to better feature extraction suited to the dataset's characteristics.