Procedure-I Used keras tuner for tuning the hyperparameter of the CNN model by that in 10 trials i got 73.8% (the whole procedure took 1.5 hrs for 10 trials) **.**By implementing `RandomSearch` in Keras Tuner, we explored 10 different hyperparameter combinations for the following parameters:

- Number of Conv2D layers (2 to 4)
- Number of filters (32 to 128)
- Kernel sizes (3 or 5)
- Dropout rates (0.2 to 0.5 for Conv layers, 0.3 to 0.6 for Dense layer)
- Optimizer choice between Adam and RMSprop

Failed Attempts-

1.Initially, resizing images to (128,128,3) yielded an accuracy of 45%, likely due to overfitting on the validation dataset.

2.Reducing the size to (64,64,3) and applying standard CNN hyperparameters improved accuracy to 55%

3.Converting images to grayscale at (64,64,1) with the same hyperparameters led to a drop in accuracy to 48%.

Data Leakage Prevention-

1.Separate Directories: Training, validation, and test datasets were stored in separate folders, ensuring isolated processing.

2.Label Consistency Check: Each label was matched carefully with its image to avoid any misalignment that could inadvertently provide data leakage cues to the model.

Software Stack:

1.TensorFlow and Keras: Used for model building, training, and evaluation.

2.Keras Tuner: Employed for hyperparameter optimization.

3.Pandas and NumPy: Managed label processing and data manipulation.