# Script for SRL

## Define Data Paths

* What it's doing: Specifies the directories for training and testing data.
* Reasoning: Organises data paths for easy access and ensures the correct data is used for training and evaluation.

## Data Augmentation and Preprocessing

* What it's doing: Defines data augmentation techniques to be applied to the training data and scales pixel values.
* Reasoning: Enhances the diversity of the training set by applying transformations, helping the model generalise better.

## Create Data Generators

* What it's doing: Creates data generators for training, validation, and testing sets, using the specified augmentation and preprocessing.
* Reasoning: Efficiently loads and preprocesses images in batches, which is crucial for handling large datasets and improving model performance.
* Build the Model with Optimal Hyperparameters
* What it's doing: Uses a function to build a model with specified hyperparameters, integrating a pre-trained EfficientNetB0 as the base model.
* Reasoning: Leverages transfer learning to improve performance and speed up training, while customising the model for the specific task.

## Custom Environment for Reinforcement Learning

* What it's doing: Defines a custom environment that provides a reward mechanism for reinforcement learning.
* Reasoning: Introduces an element of reinforcement learning by incorporating custom rewards, which can enhance the training process in specific scenarios.

## Train the Model with Custom Callback

* What it's doing: Trains the model using the training data generator and a custom callback to incorporate the reinforcement learning environment.
* Reasoning: Utilises callbacks to integrate additional functionality (e.g., custom reward mechanisms) during the training process.
* Evaluate the Model on Test Data
* What it's doing: Evaluates the model’s performance on the test data and prints the test loss and accuracy.
* Reasoning: Measures the model’s generalisation ability by assessing its performance on unseen data, providing an indication of real-world effectiveness.

## Plot Training and Validation Metrics

* What it's doing: Plots the training and validation accuracy and loss over the epochs.
* Reasoning: Visualises the model’s learning process, helping to diagnose issues like overfitting or underfitting.
* Predict and Generate Confusion Matrix and Classification Report
* What it's doing: Predicts labels for the test data, generates a confusion matrix, and prints a classification report.
* Reasoning: Provides detailed performance metrics, including precision, recall, and F1-score, and helps identify where the model is performing well or poorly.

## Methodology Employed

* Overall Approach: This task employs a combination of data preprocessing, augmentation, transfer learning, reinforcement learning elements, and hyperparameter tuning.
* Data Preprocessing: Ensures that the input data is clean, normalised, and augmented to improve the model’s ability to generalise.
* Transfer Learning: Utilises a pre-trained EfficientNetB0 model to leverage existing knowledge, improving performance and reducing training time.
* Reinforcement Learning Elements: Introduces a custom reward mechanism to potentially enhance the training process.
* Hyperparameter Tuning: Uses Keras Tuner to find the optimal hyperparameters, ensuring the model is well-optimised for the task.
* Model Evaluation and Visualisation: Evaluates the model on test data and visualises training progress, providing insights into model performance and potential areas for improvement.