Implementation Documentation – Deepfake vs Real Image Detection System

1. Overall Working of the Program

This program detects Real vs Fake images using EfficientNetB0, ELA analysis, and EXIF metadata extraction. It loads the dataset, creates a 45% subset, preprocesses images, trains the model in three phases, evaluates performance, and provides a Gradio interface for user testing.

2. Functions, Classes, and Components Used

A. Dataset Creation – Creates a 45% balanced subset of Real and Fake images.

B. Data Generators – Handles preprocessing and augmentation using ImageDataGenerator.

C. Model Architecture – EfficientNetB0 + global pooling + dense layers.

D. Training – Three-phase training (10 + 30 + 20 epochs).

E. Evaluation – Uses classification_report to compute precision, recall, and F1-score.

F. Forensic Functions – do_ela() and get_exif() for manipulation analysis.

G. Prediction Pipeline – predict_img() produces label, confidence, EXIF, and ELA output.

H. Gradio Interface – User uploads an image and receives results.

3. Input and Output Formats

Input: Single JPG/JPEG/PNG image uploaded through the interface.

Output: Prediction (Real/Fake), confidence percentage, EXIF metadata, and ELA image.

4. Required Libraries

TensorFlow 2.12, EfficientNet, Pillow, NumPy, scikit-learn, OpenCV (headless), Gradio, piexif.

5. Execution Instructions

1. Mount Google Drive.
2. Install required libraries using pip.
3. Set dataset path and create 45% subset.
4. Build and train the model in multiple phases.
5. Evaluate using classification_report.
6. Launch Gradio UI for testing uploaded images.