**Dataset Preparation:**

Real Images: These images are loaded from the directory 'CASIA2/Au/'. They are converted to their Error Level Analysis (ELA) representations, resized, and normalized. These processed images are appended to the X list with the corresponding label 1 (indicating "real") in the Y list.

**Forgey Images:** Similarly, fake images are loaded from the directory 'CASIA2/Tp/', processed, and appended to the X list with the corresponding label 0 (indicating "fake") in the Y list.

**Data Reshaping and Splitting:**

The data (X) is reshaped to have the shape suitable for feeding into a CNN.

Data is split into training and validation sets using a test size of 0.2 (meaning 20% of the data will be used for validation, and 80% will be used for training).

**Building the Custom CNN Model:**

* A custom CNN model named build\_model() is defined. The architecture of this model consists of:
* Two convolutional layers, each with 32 filters of size
* 5×5 and ReLU activation. Max pooling with a pool size of 2 × 2
* Dropout layer with a rate of 0.25 for regularization.
* Flattening layer to flatten the 3D outputs to 1D.
* Dense layer with 256 units and ReLU activation.
* Another dropout layer with a rate of 0.5.
* Final dense layer with 2 units and softmax activation for binary classification.

**Early Stopping:**

* An early stopping mechanism is implemented to prevent potential overfitting and save computational resources. It's a best practice in training deep learning models.

**Loss Function: Binary Cross-Entropy**

* The loss function used is binary\_crossentropy. This is a common choice for binary classification problems, like the one being addressed.

**Evaluation Metric: Accuracy**

* Accuracy is the most straightforward metric for classification. It's defined as the number of correct predictions divided by the total number of predictions:

**Model Compilation and Training:**

* The model is compiled with the Adam optimizer, binary cross-entropy as the loss function, and accuracy as the metric.
* An early stopping callback is set up to monitor the validation accuracy. It will stop training if the validation accuracy doesn't improve for 2 epochs.
* The model is then trained on the training data with a batch size of 32 for 30 epochs.
* After training, the model is saved as 'model\_casia\_run\_hamad.h5'.

**Model Evaluation:**

* Training and validation loss curves are plotted to visualize the model's performance over epochs.
* Additionally, the code includes visualizations of the predicted class and confidence level for specific test images.

**Testing Predictions:**

* The model's predictions are tested on individual images, showing the predicted class and confidence overlaid on the images.