Here’s a concise summary of the architecture you provided for the deepfake detection model:

1. Input Layer:

- **Shape:** (128, 128, 3)
    - The model expects images of size 128x128 with 3 color channels (RGB).

2. Xception Base Model (Pre-trained on ImageNet):

- **Purpose:** The model uses the Xception model as a base for feature extraction, excluding the top classification layers (which are not needed for this task).
- **Frozen Layers:** The initial layers are frozen, and only the last 10 layers of the Xception model are made trainable to fine-tune the model for the deepfake detection task.

3. Texture Extraction Layers:

- **Conv2D Layers:** Two sets of convolutional layers (32 filters in the first set, 64 filters in the second set) are added for texture feature extraction, each followed by a **MaxPooling2D** layer to reduce spatial dimensions.
    - **Conv2D (32 filters) → MaxPooling2D**
    - **Conv2D (64 filters) → MaxPooling2D**
    - These layers help in learning detailed texture patterns from the images that can be useful for detecting deepfakes.

4. Global Pooling:

- **GlobalAveragePooling2D:** Reduces the dimensionality of the feature maps from the previous layers by averaging the spatial dimensions, which helps in preventing overfitting and reduces the number of parameters.

5. Dense Layers:

- **Dense Layer (256 units):** Fully connected layer with ReLU activation to learn complex patterns from the extracted features.
- **Dropout Layer (50%):** Dropout for regularization to prevent overfitting by randomly setting half of the neurons to zero during training.

6. Output Layer:

- **Dense Layer (1 unit with sigmoid activation):** The final layer for binary classification (either "REAL" or "FAKE"), using sigmoid activation to output a probability between 0 and 1.

7. Model Compilation:

- **Optimizer:** Adam optimizer is used for efficient training.
- **Loss Function:** Binary cross-entropy loss is used for binary classification.
- **Metrics:** Accuracy is used to evaluate the model performance.

Model Summary:

- This architecture combines a powerful pre-trained base model (Xception) with custom texture extraction layers and dense layers, fine-tuning it to detect deepfakes. The model is designed to leverage both high-level features learned by Xception and low-level texture features to improve its classification accuracy. The final output is a binary prediction indicating whether the image is "REAL" or "FAKE."