**MODEL SUMMARY**

Convolutional neural networks were selected as the model because the job at hand is to classify the image. The primary applications of this model include anomaly detection, time series forecasting, medical image processing, and satellite image identification.

CNN has multiple layers that extract features from data.

1)Convolutional Layer:

a) Fundamental building element

b) Only needs a few items, including a feature map, a filter, and input data.

c) A feature detector, sometimes referred to as a kernel or filter, scans an image's receptive field to determine whether a feature is present. We refer to this as convolution. After applying a filter to a portion of the image and calculating the dot product, the filter moves by a step, and this procedure is repeated until the kernel has moved throughout the entire image.

2)Activation Functions:

3)Pooling Layer

Also referred to as down sampling layers, they aim to reduce the number of parameters in the input data.

Two types of pooling layers;

* Max pooling: Sends the highest value found in a set of pixels to the output array.
* Average pooling: This method transfers the average value from the receptive field to the output array.

As feedforward neural networks, CNNs process input in a form resembling a grid, which makes them excellent at interpreting visual images. Several filters that carry out convolution operations are found in the convolutional layer. Every picture is viewed as a matrix of pixel values. A convolution feature matrix is created by sliding the filter matrix over the image matrix and computing dot products.

**TRAINING PROCESS**

Image Processing

* OpenCV should be used to load every image.
* Convert picture formats to RGB.
* Resize all photos to the same scale (128, 128).
* Transfer the pictures to NumPy arrays.

Model Architecture

* ReLU activation, 32 filters, and a 3x3 kernel size comprise the convolutional layer.
* A max-pooling layer featuring a pool size of 2 by 2.
* To convert the 2D feature map into a 1D vector, apply a flatten layer.
* A dense layer with ReLU activation and 256 units.
* To avoid overfitting, a dropout layer with a 0.1 dropout rate is used.
* An additional dense layer with ReLU activation and 512 units.
* A five-unit output layer with softmax activation is used for multi-class categorization.

Model Compilation

Utilizing the Adam optimizer and sparse categorical cross-entropy loss as the metric, compile the model.

Model Training

* Using the train test split function from scikit-learn, divide the dataset into 70% for training and 30% for testing.
* Using TensorFlow, normalize the training and testing data by scaling the pixel values between 0 and 1.
* Train the model with a 0.1 validation split over about 200 epochs.

**INSIGHTS**

Although the model achieves an excellent 84% accuracy over 200 epochs, its efficacy is probably limited by the short size of the dataset (only 150 photos). The model has a propensity to overfit since it finds it challenging to identify broad trends in this small sample. Extending and broadening the training dataset and experimenting with different regularization techniques would be necessary to improve the model's performance.