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.

a.The many layers of CNN are used to extract features from data.

First, Convolution Layers

Fundamental building block

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.

Functions of Activation

1. Using Pooling Layers

Also referred to as down sampling, this technique lowers the number of input parameters. This works similarly to a convolution layer, only instead of using weights to sweep a filter across the entire input.

Two primary types of pooling layers

1. Maxpooling: This method chooses which pixel's maximum value to deliver to the output array.

2. Average pooling: To send to the output array, it determines the average value in the receptive field.

It is a feed forward neural network that is typically used to analyze visual pictures through grid technology data processing.

Convolution layer has several filters that perform the convolution operation. Every image is considered as a matrix of pixel values. Slide the filter matrix over the image matrix to compute the dot product and get the convolution feature matrix.

TRAINING PROCESS

* **Preprocessing Images: OpenCV should be loaded for each image.**
* **Change the image's color space to RGB.**
* **Adjust image sizing to (128, 128).**
* **Convert the picture to an array in NumPy.**

**Model Architecture:**

The following layers make up a sequential CNN model: o A convolutional layer with 32 filters, a 3x3 kernel size, and ReLU activation.

o A maximum pooling layer with a 2 by 2 pool.

o Flatten the layer to create a 1D vector from the 2D feature map.

o Dense layer with ReLU activation and 256 units.

o A dropout layer with a rate of 0.1 to avoid overfitting.

o Dense layer with ReLU activation and 512 units.

o A multi-class classification output layer with five units and softmax activation.

**Model Compilation:**

**Model is compiled using the adam** sparse categorical crossentropy using accuracy as it parameter

Model Training:

* + Dataset is split for training using the train\_test\_split method from scikit learn in the ratio of 70% training and 30% testing data.
  + Normalize the training and testing data using tensorflow to scale the pixel values between 0 and 1.
  + Train the model for about 200 epochs with a validation split 0.1

CRITICAL FINDINGS

After 200 epochs, the model's astounding accuracy of 86% was attained. Its performance is probably limited, though, because the dataset is small—only 150 photos total. The small size of the dataset makes it difficult for the model to understand features that are applicable to a wider context, which leads to an overfitting problem. It is advisable to expand the amount and variety of the training dataset in order to improve the model's performance. It could also be helpful to experiment with other regularization strategies.