MODEL SUMMARY

As the task is to classify the image, the model chosen is Convolutional Neural network. This model is mainly used to identify satellite images, process medical images, forecast time series and detect anomalies.

* CNN has multiple layers that extract features from data.

1. Convolution Layers –
   1. Core building block
   2. Requires few components like input data, filter and a feature map
   3. Feature Detector also known as kernel or filter moves across the receptive field of images to check if the feature is present. This is called convolution. Filter is applied to an area of the image and dot product is calculated and afterwards the filter shifts by a stride and repeat the process until the kernel has moved all over the entire image
2. Activation Functions
3. Pooling Layers

Also known as down sampling, reduces the no of parameters in the input. Similar to convolution layer, this sweeps a filter across the entire input and the difference is that it doesn’t have any weights.

Mainly two kinds of pooling layers

1. Maxpooling : It selects the maximum value in the pixel to send to the output array.
2. Average pooling: It calculates the average value within the receptive field to send to the output array.

* It is a feed forward neural network generally used to analyse visual images by processing data with grid technology.
* 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

**Image Preprocessing:**

* Load each image using OpenCV.
* Convert the image to RGB colour space.
* Resize the image to (128, 128).
* Convert the image to a NumPy array.

**Model Architecture:**

A sequential CNN model is used with the following layers,

* + Convolutional layer with 32 filters, 3x3 kernel size, and ReLU activation.
  + Max pooling layer with 2x2 pool size.
  + Flatten layer to convert the 2D feature map to a 1D vector.
  + Dense layer with 256 units and ReLU activation.
  + Dropout layer with 0.1 dropout rate to prevent overfitting.
  + Dense layer with 512 units and ReLU activation.
  + Output layer with 5 units and softmax activation for multi-class classification.

**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

Despite achieving an impressive accuracy of 84% after 200 epochs, the model's performance is likely hindered by the limited size of the dataset, which consists of only 150 images. This small dataset makes it challenging for the model to learn generalizable features, leading to a tendency towards overfitting. To improve the model's performance, it would be helpful to increase the size and diversity of the training dataset, and to experiment with different regularization techniques.