### Deep Neural Network - Implementing CNN

Convolutional neural network (CNN) model for binary classification, where the task is to classify images into two categories (normal and covid). Here's a breakdown of what the code does:

* Sequential() creates an empty model object.
* BatchNormalization() is a preprocessing layer that normalizes the input data.
* Conv2D() adds a convolutional layer with 32 filters, a kernel size of (3,3), and 'same' padding. The activation function used is 'relu'.
* MaxPool2D() adds a max pooling layer with a pool size of (2,2).
* Dropout() adds a regularization layer that randomly drops out 25% of the nodes in the previous layer during training.
* Flatten() flattens the output of the previous layer so that it can be fed into a dense layer.
* Dense() adds a dense layer with 128 units and the activation function 'relu'.
* Another Dropout() layer is added, this time with a rate of 0.1.
* Dense() adds the output layer with 2 units and the activation function 'sigmoid', which is suitable for binary classification problems.
* compile() compiles the model with the Adam optimizer, binary crossentropy loss function, and accuracy metric for evaluation.

**Summary Explanation**

This is a sequential neural network model with several layers, including convolutional, pooling, dropout, and dense layers. It is designed for image classification with 2 classes. The model has a total of 23,041,294 parameters, and all of them are trainable.

The first layer is a batch normalization layer that normalizes the input data to improve the stability and performance of the model. The second layer is a 2D convolutional layer with 32 filters and a kernel size of (3, 3). This layer applies 32 different filters to the input image to extract different features from it. The third layer is a max pooling layer with a pool size of (2, 2) that downsamples the feature maps obtained from the previous convolutional layer.

The fourth layer is a dropout layer that randomly drops out some of the nodes to prevent overfitting. The fifth layer is a flatten layer that flattens the output from the previous layer into a 1D array. The sixth layer is a dense layer with 128 units that applies a linear transformation to the input data. The seventh layer is another dropout layer to further prevent overfitting. Finally, the last layer is a dense layer with 2 units that applies the softmax activation function to classify the input image into one of the two classes.