Objective : Detection of Covid19 using patient's CT-Scan images (How Model Works)

Model Name :

1. **ResNet50 Pretrained CNN Model**

Overview :

Basically in this model I have used **ResNet50** pretrained model which belongs to ImageNet.

In this pdf you will see how the model works.

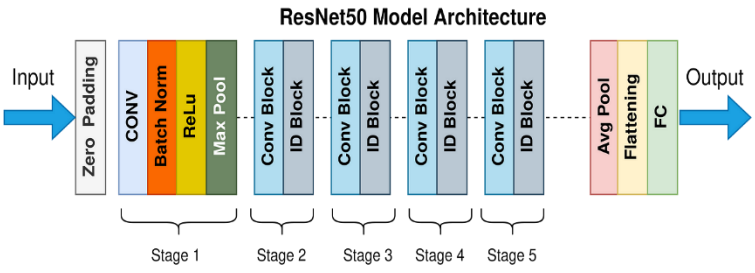
ResNet-50 is a deep convolutional neural network architecture that was introduced as part of the ResNet(Residual Network) family by researchers from **Microsoft** in **2015**. It is specifically desined for image classification tasks.

The key idea behind ResNet-50 is the introduction of residual connections, also known as skip connections or shortcut connections. These connections. These Connections allow information to bypass a few layers and be directly forward to deeper layers in the network. This helps to alleviate the problem of vanishing gradient and enables the network to learn more easily and effectively.

Here's a brief overview of the ResNet-50 architecture:

2. **Input Layer**: Takes the input image of size 224x224x3 (RGB channels).
3. **Convolutional Layers:** The initial convolutional layer performs a 7x7 convolution with 64 filters and a stride of 2. It is followed by a max pooling layer.
4. **Residual Blocks:** ResNet-50 consists of several residual blocks, each containing multiple convolutional layers. The basic building block is the identity block, which has three convolutional layers. The first two layers have a filter size of 64, while the third layer has a filter size of 256. Identity blocks are used multiple times throughout the network.
5. **Projection Blocks:** In addition to identity blocks, ResNet-50 also includes projection blocks, which are used when the input and output dimensions of a residual block are different. These blocks include a 1x1 convolutional layer to adjust the dimensions.
6. **Global Average Pooling:** After the residual blocks, a global average pooling layer is applied to reduce the spatial dimensions of the feature maps to a vector.
7. **Fully Connected Layers:** The global average pooled features are connected to a fully connected layer with 1000 units, followed by a softmax activation function to output the class probabilities.

Overall, ResNet-50 has 50 convolutional layers, including convolutional, pooling, and fully connected layers. The skip connections in ResNet-50 allow for very deep networks to be trained successfully and have led to significant improvements in performance on various image classification tasks, including the ImageNet dataset.



Explanation :

We have seen in ResNet-50 Network there are 2 block are repeated 1. Conv Block and 2. ID block

Lets see brief about them

1 . Conv Block :

- Conv block stand for convolutional block.
- Conv block is the first block of the every layer.
- In this block stride is used as 2 and other block (ID Block) strides used as 1.
- Each convolutional layer in ResNet-50 consists of multiple convolutional filters, also known as kernels. These filters are small square matrices that are applied to the input data in a sliding window manner. Each filter performs a dot product operation between its weights and a local receptive field of the input data, producing a single value known as the activation or feature map.

In code conv block is written as :

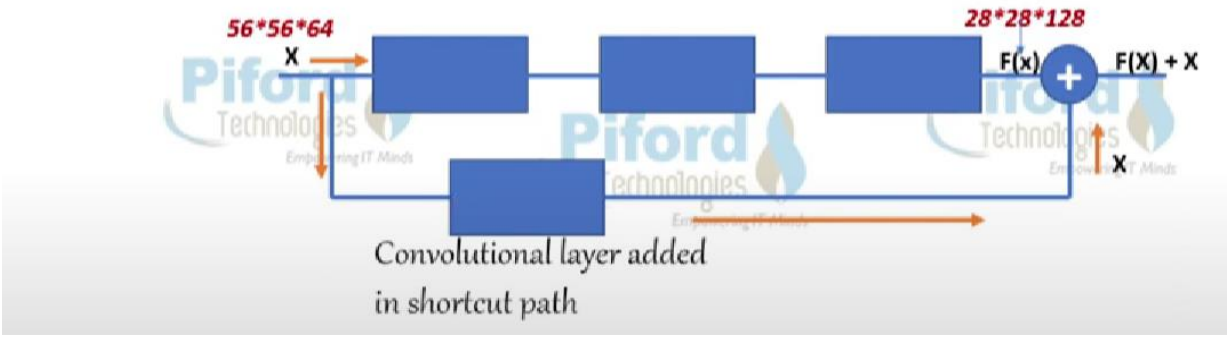
```
# Retrive Filters
F1,F2,F3 = filters

# save the input value
X_actual = X

# First Layer
X = Conv2D(F1, (1,1),strides=(s,s), activation='relu')(X)
X = BatchNormalization(axis =3)(X)

# Second Layer (filter = (3,3) by default)
X = Conv2D(filters=F2, kernel_size=(f,f),strides=(1,1), activation='relu', padding='same')(X)
X = BatchNormalization(axis =3)(X)

# Third Layer
X = Conv2D(filters = F3, kernel_size=(1,1),strides=(1,1), activation='relu', padding='valid')(X)
X = BatchNormalization(axis =3)(X)

# Shortcut path
X_actual = Conv2D(filters = F3,kernel_size=(1,1), strides=(s,s), padding='valid')(X_actual)
X_actual = BatchNormalization(axis=3)(X_actual)

# Final step: Add actual value in calculation and pass it through relu activation function
X = Add()([X,X_actual])
X = Activation('relu')(X)
```

ResNet-50, there are multiple convolutional layers stacked together, forming the backbone of the network. The depth and number of filters in each convolutional layer may vary depending on the specific layer and block within the network.

Mathematical Calculation :

Convolutional Block (input size != Output size)

(56*56*64 != 28*28*128)



If we make F(x) = 0 then it is easy for us to make input equal to output

Y=X+F(X)

Y=X+0

Y=X

Identity Block

Identity Block :

Code :

```
#retrive filters
    F1,F2,F3 = filters
    X_actual = X

    # First Layer
    X = Conv2D(filters = F1, kernel_size=(1,1),strides=(1,1), padding='valid',
activation='relu')(X)
    X = BatchNormalization(axis=3)(X)

    # Second Layer
    X = Conv2D(filters = F2, kernel_size=(f,f),strides=(1,1), padding='same',
activation='relu')(X)
    X = BatchNormalization(axis=3)(X)

    # Third Layer
    X = Conv2D(filters = F3, kernel_size=(1,1),strides=(1,1),
padding='valid')(X)
    X = BatchNormalization(axis=3)(X)

    # Final Step : adding a actual X value as present in the formula
    X = Add()([X,X_actual])
    X = Activation('relu')(X)
```

an identity block is a basic building block that preserves the dimensions of the input feature map. It is used to create deeper networks while maintaining information flow and preventing the degradation of performance.
The identity block consists of three main components: a series of convolutional layers, batch normalization, and element-wise addition.

Here's a step-by-step explanation of the identity block:

Input: The input feature map is passed into the identity block.

Convolutional Layers:

The first convolutional layer applies a 1x1 filter to reduce the number of input channels (if necessary) and generate a lower-dimensional representation.

The batch normalization layer normalizes the output of the previous convolutional layer by adjusting the mean and variance.

The ReLU activation function is applied to introduce non-linearity.

Convolutional Layers:

The second convolutional layer performs a 3x3 convolution on the output of the previous layer.

The batch normalization layer normalizes the output.

The ReLU activation function is applied.

Convolutional Layers:

The third convolutional layer applies a 1x1 filter to expand the number of channels (if necessary) and match the original number of input channels.

The batch normalization layer normalizes the output.

Element-wise Addition: The output of the third convolutional layer is added element-wise to the original input feature map. This shortcut connection allows the information to bypass the convolutional layers and preserves the original signal.

Activation: The resulting sum of the shortcut connection and the output from the last batch normalization layer is passed through the ReLU activation function.

The identity block is repeated multiple times in ResNet-50 to create deeper networks while preserving the identity mapping. By introducing these skip connections, ResNet-50 alleviates the problem of vanishing gradients and allows the network to be trained more effectively, leading to improved performance on image classification tasks.