*Cosine growing unit ResNet*

This study examines the effectiveness of the cosine growing measure, a prominent machine learning approach for data categorization. More specifically, we assessed cosine similarity performance as where = and the vectors x and y have lengths, respectively. Two l-dimensional vectors, and . The cosine measure is often employed in data and information retrieval due to its easy interpretation and computation for sparse vectors. The angle or developing cosine of the angle between two vectors may also signify cosine growing similarity. This approach treats data with similar composition but differing totals equally, making it a common data metric.

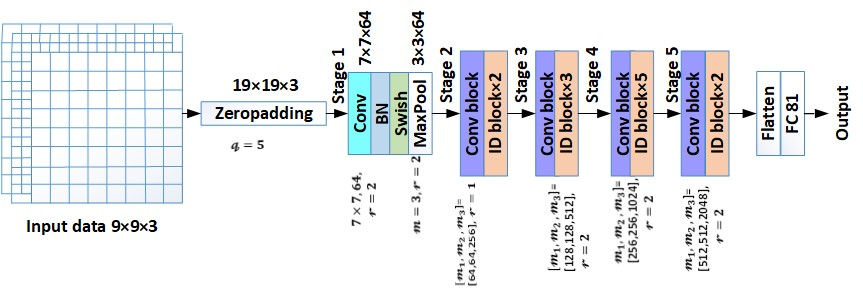
The basic building blocks may be used to generate a variety of CNN models, such as by changing the quantity and configuration of the layers, the residual network with 50 layers (ResNet-50), as seen in Fig. 5, may be achieved. Among the most sophisticated CNN models we have used for this research is this one. Remaining networks are effective in resolving the issues that remaining learning, which makes significant use of residual blocks, helps a conventional (very) deep neural network deal with issues like gradient bursting or disappearance and degradation.

First, we use Fig. 6 (top) to explain how a residual block functions. The graphic shows the two pathways that information takes to go from input to output activation the main walkway, which descends, is divided into two sections. First, the data is processed via three modules: batch normalization, non-linear activation function, and convolution layer. These modules are controlled by the standard equations shown below:

(40)

(41)

Where the first part's input is , its output is, the weight matrix is , Assuming a non-linear activation function of b[l], the bias vector is . To speed up the training, the batch normalization module is implemented.



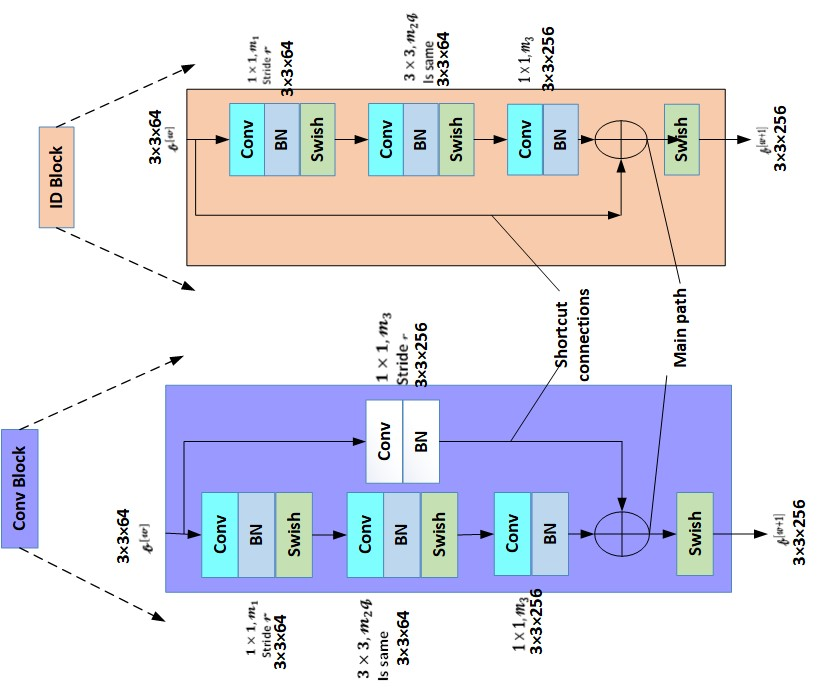


Fig. 5 ResNet-50layer architecture

Similar to this, the modules in the second section are determined by the following equations (omitting the additional operation and the other path):

(42)

(43)

A neural network's deeper hidden layer in residual networks adds with its output before implementing a function of activation that is not linear. This occurs because is fast-forwarded to that layer. The figure illustrates what is referred to as a short-cut connection. So, the following modification will be made to Eq. 43:

(44)

It becomes a residual block with . In addition, we assume the dimensions of input and (and output ) are the same. This leftover block is called the identification (ID) block. If input () and output () dimensions disagree, a convolution layer is added to meet the ultimate addition and use the shortcut connection to increase the input. Convolutional (Conv) blocks are what are known as residual blocks, as Fig. 6 (bottom) illustrates. Instead of only two hidden layers, we employ residual blocks to skip three in our work.

To enable single-layer activations can bypass subsequent layers and feed straight into deeper levels; residual blocks in ResNet-50 are layered (see Stage 2-5 in Fig. 5 (left)). Shortcut connections provide direct gradient back-propagation to prior layers during back-propagation. As shown in the picture, the input data is zero-padded with to create a output volume (Eq. 45 may be used to calculate layer output dimensions). Stage 1 uses a convolution layer with k = 7, to transform the volume to dimensions. Ultimately, the Max Pool layers with produce a output volume.

In Stage 2, the Conv block in Fig. 5 will use an input size of 3 × 3 × 64 from the preceding layer. The path has 3 segments. The first convolution layer has . The volume output matches the input dimensions. The second part convolution layer uses the "same" convolution, resulting in output with the same dimension as the input. This is achieved by setting padding to maintain output dimension. The third component has a convolution layer with , and to reduce the input dimension from. Using the shortcut connection, the convolution layer scales up the input volume from by setting, and . The outputs from both convolution layers (shortcut connection and third half of the main route) may now be joined since they have the same size.

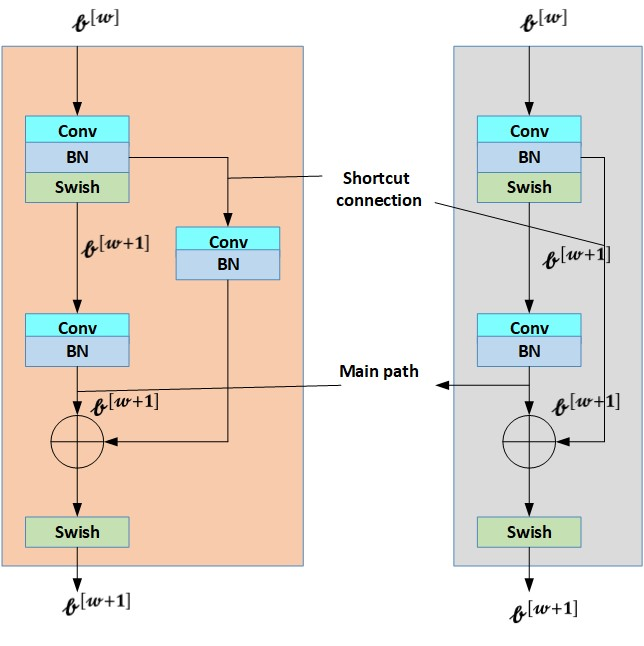


Fig. 6 Residual blocks (Top). Identity (ID) block. (Bottom) Convolutional (Conv) block.

The ID blocks in Stage 2 work similarly to the Conv block, except for the shortcut connection's layerless architecture. Since the input of ID blocks matches the output of the 256 × 3 × 256 convolution layer in its third section, the shortcut connection does not need it.

The last stage, which is a 1 x 1 × 32 volume, is comparable to stages 3-5. An 81x1 vector containing normal and under-attack cell IDs is produced by processing the data into an array by a fully linked layer (50th layer). The model's hyperparameters and the size of the input to Stage 2 layers are shown in Fig. 5 by red and blue annotations, respectively. While multi-class classification often uses a softmax function in the output layer, we employed a binary cross-entropy loss function for multi-label classification. Our approach extends a input data by padding zeros with to create a volume. This phase involves transforming the volume to dimensions using a convolution layer. A max-pooling layer follows the height or width are computed by using:

(45)

The height or width Assuming Eq. 45, the output volume dimensions are . Conv2, MaxPool2, Conv3, and MaxPool3 are the convolution and pooling layers that the volume passes through in the DRC model. After the volume is compressed and processed by two fully connected layers, the output vector, which identifies normal and under-attack cells, is 81 × 1 dimensions and is produced by the binary cross entropy loss function.