

Optimizer = 'Adam' , loss function = 'binary cross entropy'

The sequential () model is used from keras. The input is given to a 2D convolution layer followed by a maxpooling operation on the resultant features. The feature vector is then an input to a 2nd 2D convolution layer and again pooled using maxpooling. The pooled images are then converted into a continuous vector through flattening. The 2D array is converted into a one-dimensional single vector. A dense function is added which creates a fully connected layer and to this layer we are going to connect the set of nodes obtained after the flattening step. These nodes will act as an input layer to these fully connected Layers. The final layer is the output layer which contains only one node as it is binary classification. Sigmoid activation function is used for the final layer. The model was trained for 10 epochs.

First Convolutional layer:

filters =32, shape of the filter = (5,5), input= (64,64,1), activation=relu

Output shape = (None, 60, 60, 32)

Max Pooling:

pool size= (2,2), strides= (1,1)

Output = (None, 30, 30, 32)

Second convolutional Layer:

Conv2D Layer filters =32, shape= (5,5), input= (64,64,1), activation=relu

Output shape = (None, 60, 60, 32)

Max pooling:

pool size= (2,2), strides= (1,1)

Output = (None, 30, 30, 32)

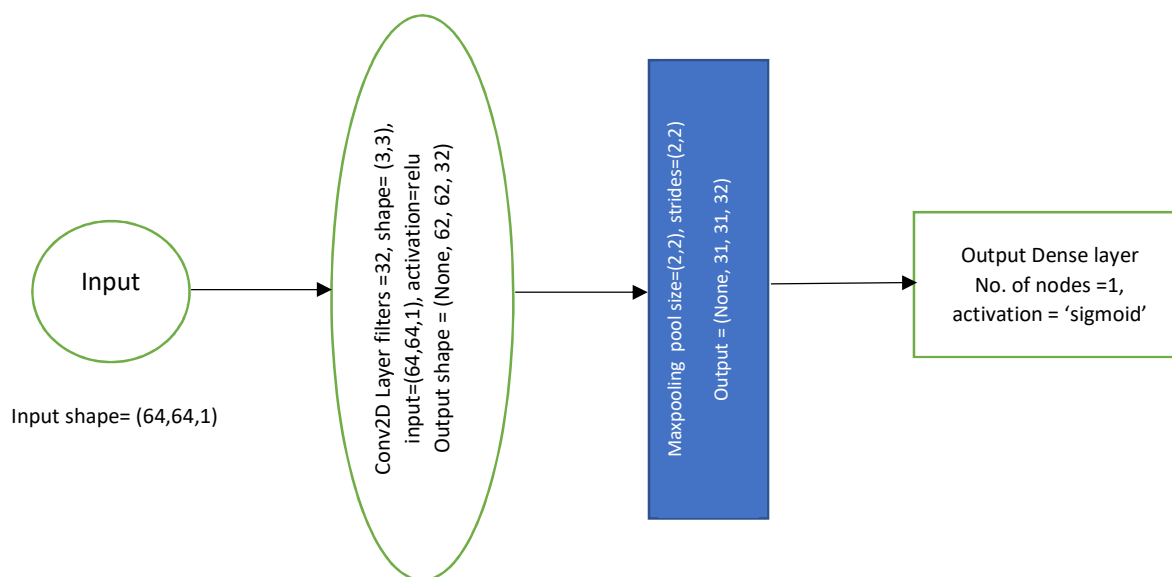
Dense Layer:

Units =128 activation = 'relu'

Output Dense Layer:

Number of output nodes = 1, activation = 'sigmoid'

Williams CNN model:



The input is given to a 2D convolution layer with number of filters = 32 and shape of each filter = (3,3). The activation used is 'relu' rectifier function. It is then followed by a maxpooling operation with pool size = (2,2) and strides = (2,2) on the resultant features. The output layer is a dense layer with one output node and sigmoid activation function as it is binary classification. The model was trained for 5 epochs. Training for more than 5 epochs resulted in overfitting and having very less to almost zero F1 on the test set.

First Convolutional layer:

filters = 32, shape of the filter = (3,3), input = (64,64,1), activation = relu
Output shape = (None, 61, 61, 32)

Max Pooling:

pool size = (2,2), strides = (1,1)
Output = (None, 31, 31, 32)

Output Dense Layer:

Number of output nodes = 1, activation = 'sigmoid'

To compile:

Optimizer = 'Adam', loss function = 'binary cross entropy'

Results:

F1 Table

Dataset	Train F1	Test f1
Bush	0.98778833	0.759878
Williams	0.88888888	0.625