

Design:

The model expects input images of size 100x100 pixels with three color channels (RGB).

Two convolutional layers with 32 and 64 filters of size 3x3 are used, respectively. ReLU activation is applied after each convolutional layer.

Two max-pooling layers with a pool size of 2x2 are applied after each convolutional layer. They help reduce the spatial dimensions and capture the most important features.

The output from the last convolutional layer is flattened to create a 1D vector.

A fully connected layer with 64 units and ReLU activation is added. This layer learns high-level features by combining the information from flattened feature maps.

The final dense layer consists of as many units as there are class labels (2 in this case). It uses the softmax activation function to output the predicted probabilities for each class.

Approach:

The chosen approach for image classification is a CNN. CNNs are well-suited for analyzing visual data due to their ability to automatically learn hierarchical representations of features. They leverage convolutional and pooling operations to extract local features and progressively learn more abstract representations. This makes CNNs particularly effective for tasks like image classification.

Reasoning:

By using convolutional layers, the model can capture local patterns and features in the images. These layers can learn various low-level features such as edges, corners, and textures, which are important for distinguishing between cancerous and non-cancerous cells.

Max pooling helps reduce the spatial dimensions of the feature maps while retaining the most salient information. This reduces the computational complexity of the model and enables it to focus on the most important features.

The fully connected dense layers at the end of the model allow it to learn high-level representations by combining the features extracted from the convolutional layers. These layers help in capturing global patterns and making the final classification decision.

The softmax activation function in the output layer produces a probability distribution over the classes. It allows the model to output the likelihood of the input image belonging to each class label, enabling it to make a confident classification decision.