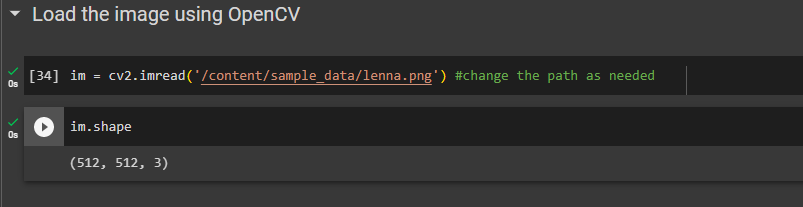
# Lab 3 – IT20650520

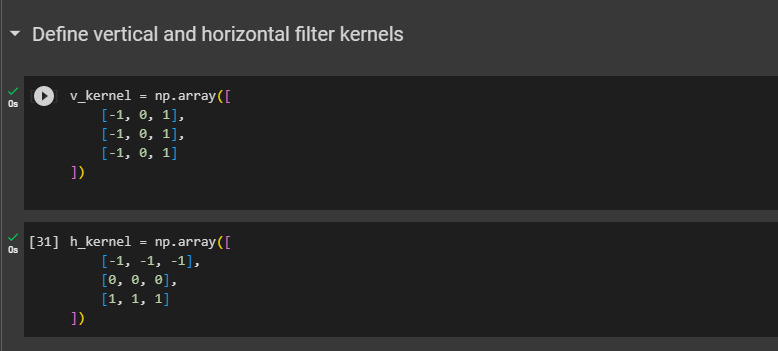
## Open Google colab. Upload the 1D\_Convolution.ipynb to colab. Run all cells. Based on the result, explain how 1D convolution can be used to identify the edges in an image.

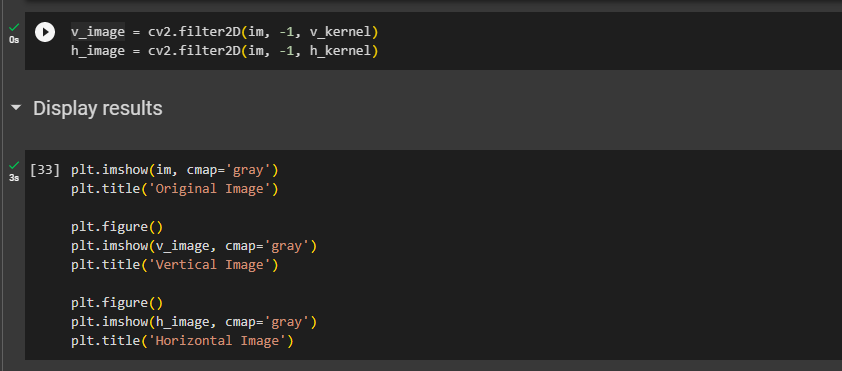
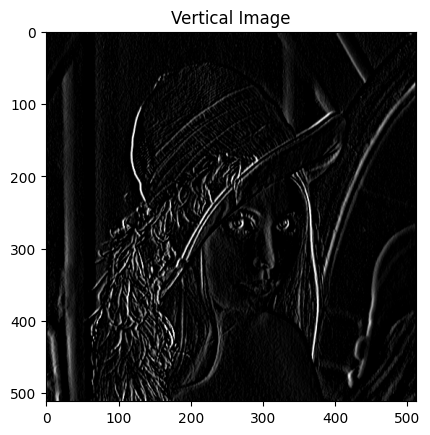
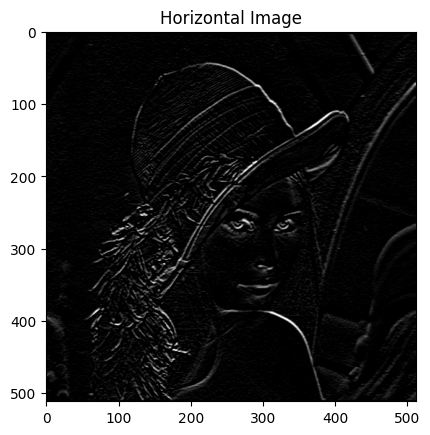
**In image processing, including convolutional neural networks (CNNs), 1D convolution can be used to identify edges in a picture by applying filters/kernels that are designed to respond strongly to intensity changes. When combined with a 1D signal (row of pixel values), a filter with coefficients like [-1, 0, 1] highlights areas where there is a transition from dark to light or light to dark, which corresponds to an edge in the image. The filter's positive and negative coefficients effectively calculate the signal's gradient and highlight edges where there are sharp changes in intensity.**

## Upload the Image\_Filtering\_(Convolution).ipynb file to colab. Change the filters and see if you can obtain different kinds of edges from the image. Download the modified ipynb file.

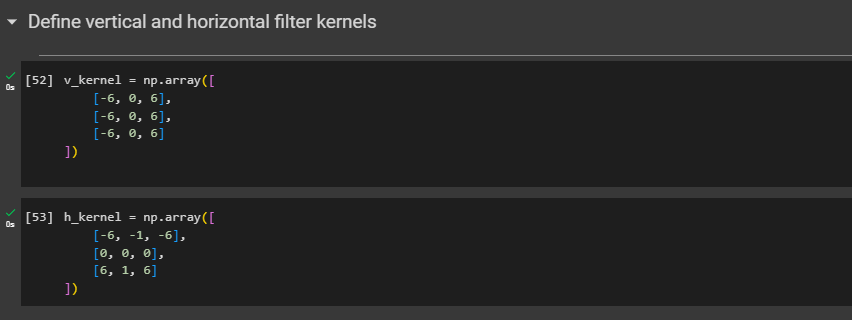


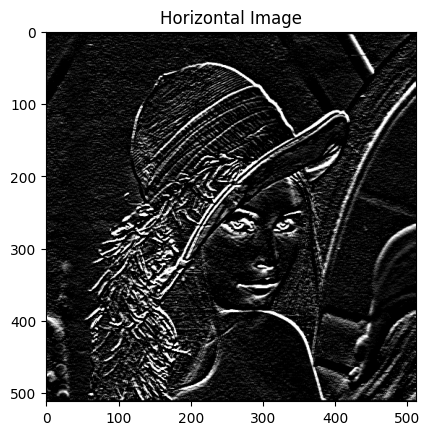
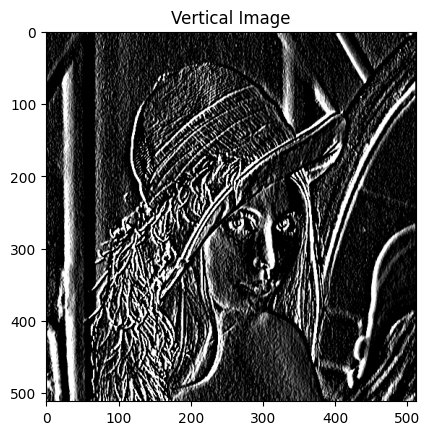
* **Change filter kernels**

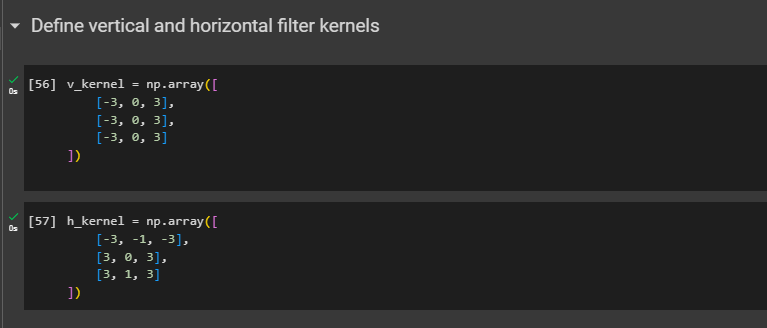


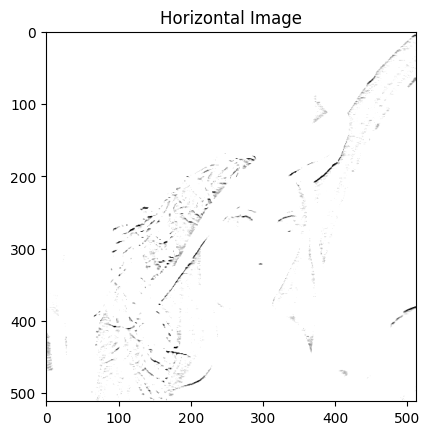
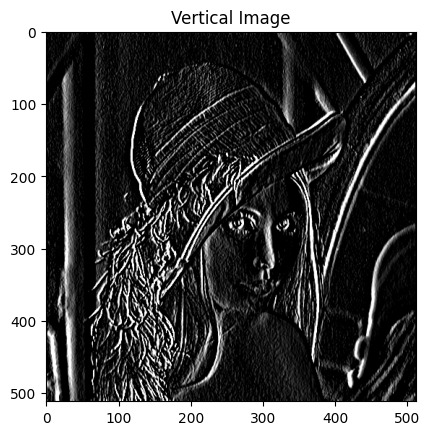


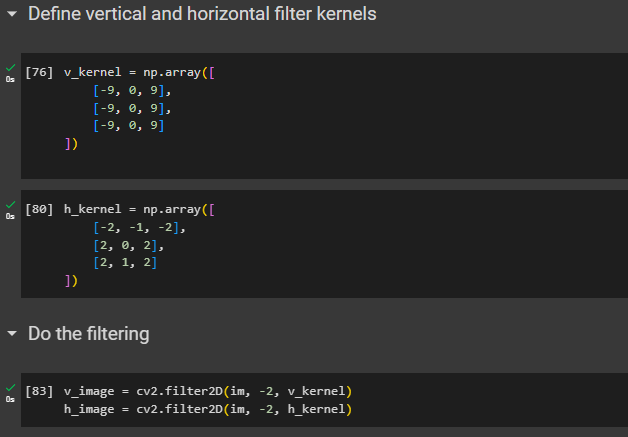


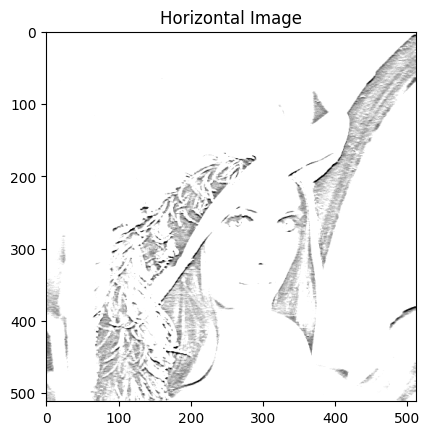
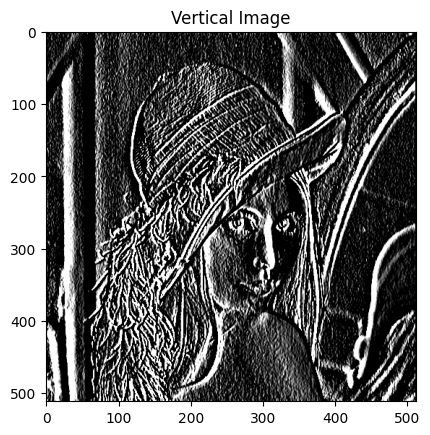






****

****

****

## Upload the CNN\_with\_keras3.ipynb file to colab. Increase the number of epochs to 50.

### Why does the validation error increase when the number of epochs are increased? Explain how you can modify the training process to stop that from happening.

**In some cases, overfitting can result from increasing the length of training periods. Overfitting arises when the model learns to fit the training data too closely, including noise and fails to generalize well to new, unseen data. This can result in an increase in validation error as the model becomes less robust.**

**Methods like early stopping or regularization can help you avoid this. Early stopping involves tracking the validation error during training and stopping when the error starts to increase. Methods of regularization, such as L1 or L2 regularization, add penalties to the loss function based on the size of the model's parameters. These punishments help the model learn more general features and stop it from making noise.**

### Explain how the mini batch SGD (Stochastic Gradient Descent) algorithm can converge faster than the batch Gradient Descent algorithm.

* **Mini Batch SGD vs. Batch GD:**

**To train machine learning models, optimization techniques like Stochastic Gradient Descent (SGD) and Batch Gradient Descent (BGD) are utilized.**

* **Batch Gradient Descent (BGD):**

**Using the whole training dataset, BGD allows you to determine the gradient of the loss function and subsequently modify the model's parameters. In spite of the fact that it can be computationally costly, especially for big datasets, this method gives a more reliable estimate of the gradient direction.**