CHALLENGE-1

# AML

GROUP-26

# CHALLENGE:

## DETERMINE WHETHER AN IMAGE CONTAINS A COLUMNAR CACTUS

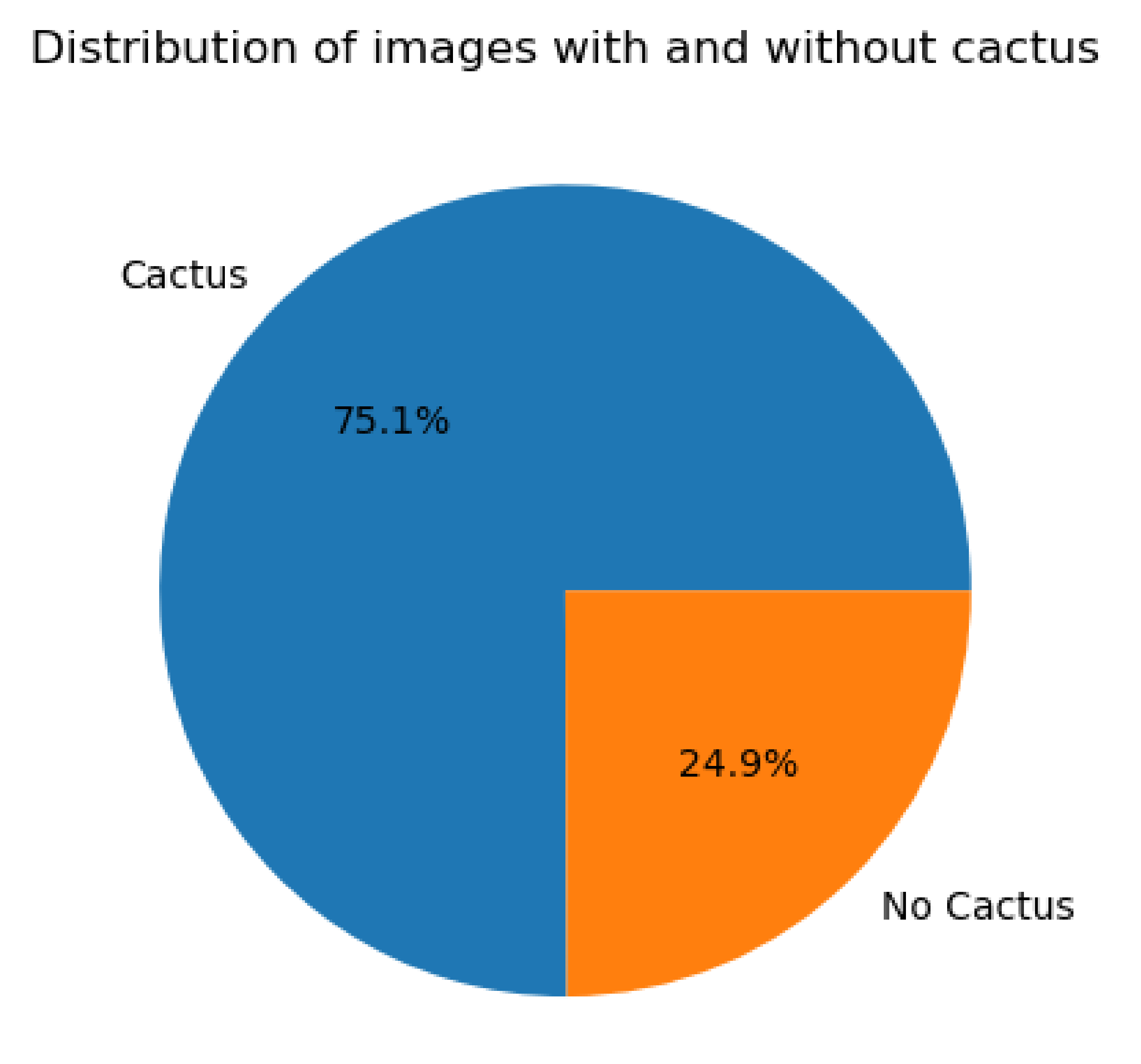
# DOES AN IMAGE CONTAIN A COLUMNAR CACTUS?

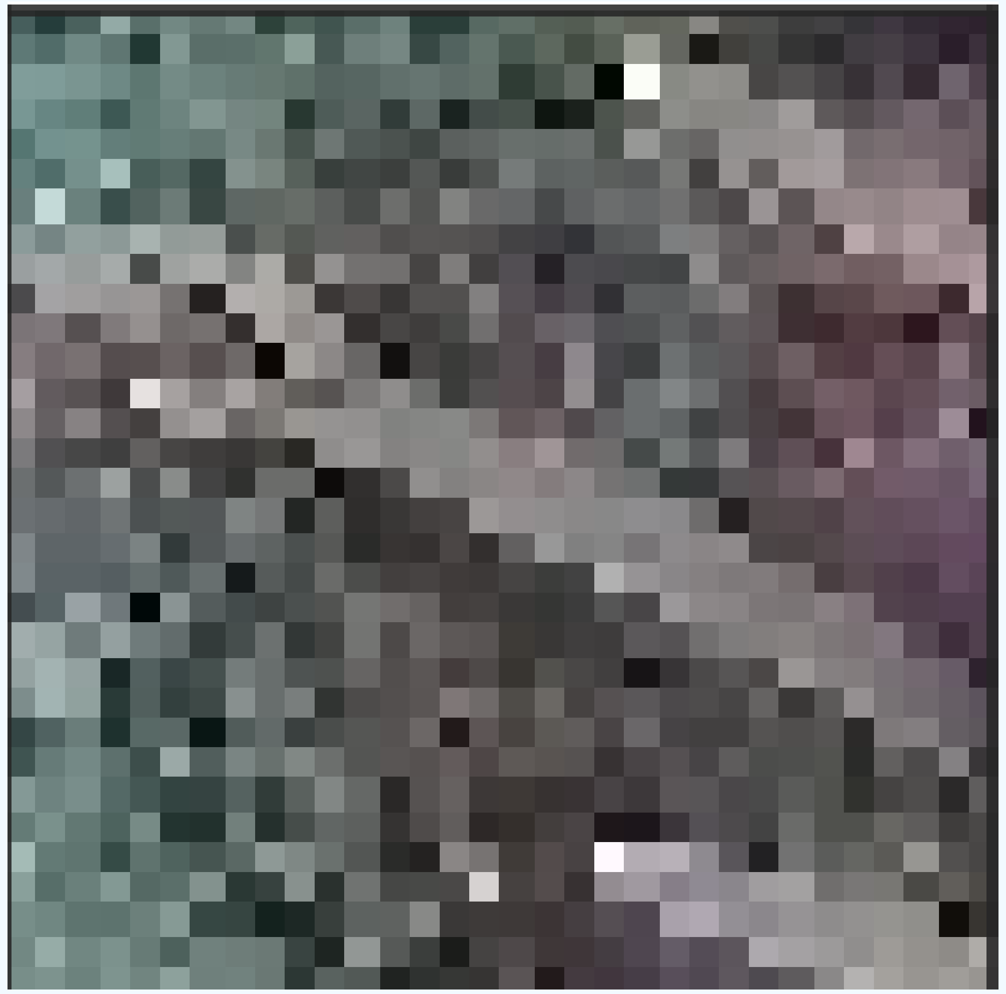
## HOW TO SOLVE IT?

- Data Exploration
- Data Preparation
- Model Selection
- Model Performance

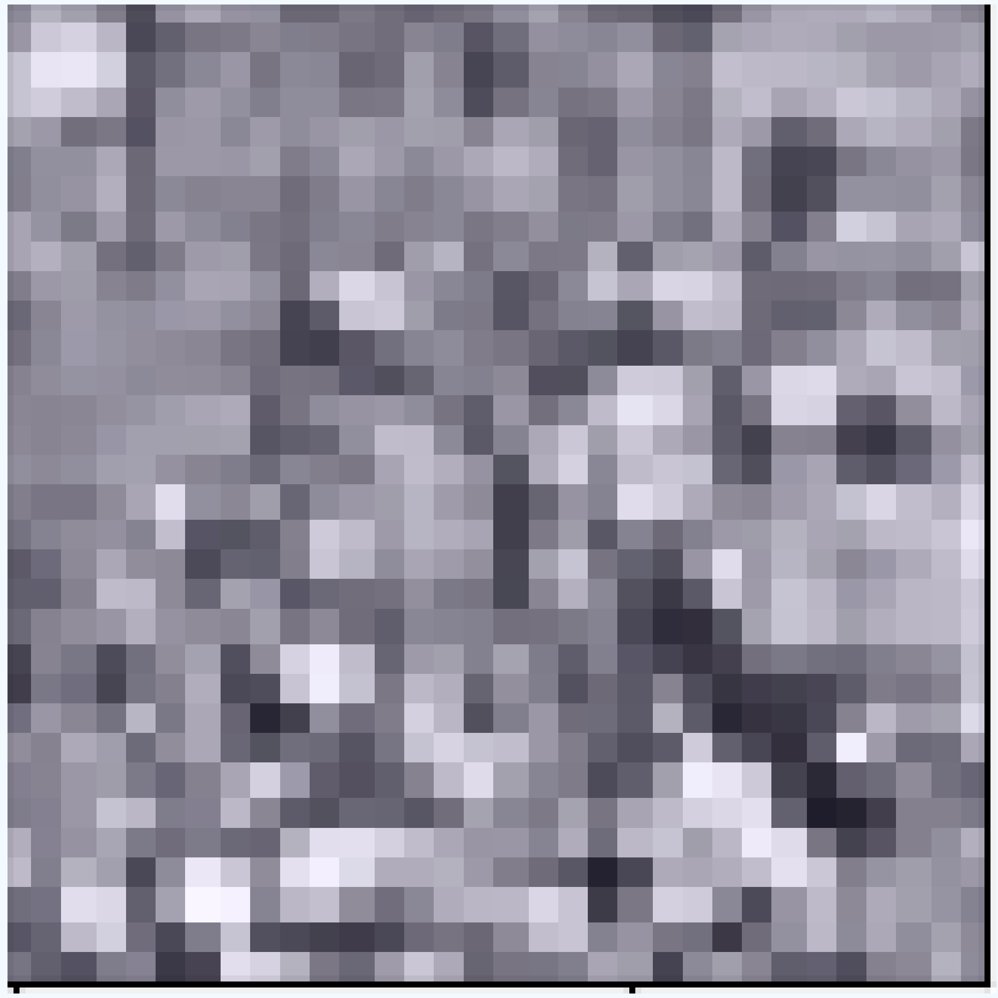
# DATA EXPLORATION

A single-label data classification problem, with 32x32 images



A cactus



Not a cactus

# DATA PREPARATION

- Applying Contrast Limited Adaptive Histogram Equalization
- Gaussian Filter
- Edge Detection
    - Laplacian filter

# CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION

## WHAT IS IT?

Contrast Limited Adaptive Histogram Equalization (CLAHE) is an image enhancement technique that improves the contrast and details in an image.

## WHY IS IT USEFUL?

CLAHE improves the contrast of images and enhances the details and textures in an image, making the cactus features more distinguishable from the background. This can aid in better differentiating cactus images from non-cactus images.

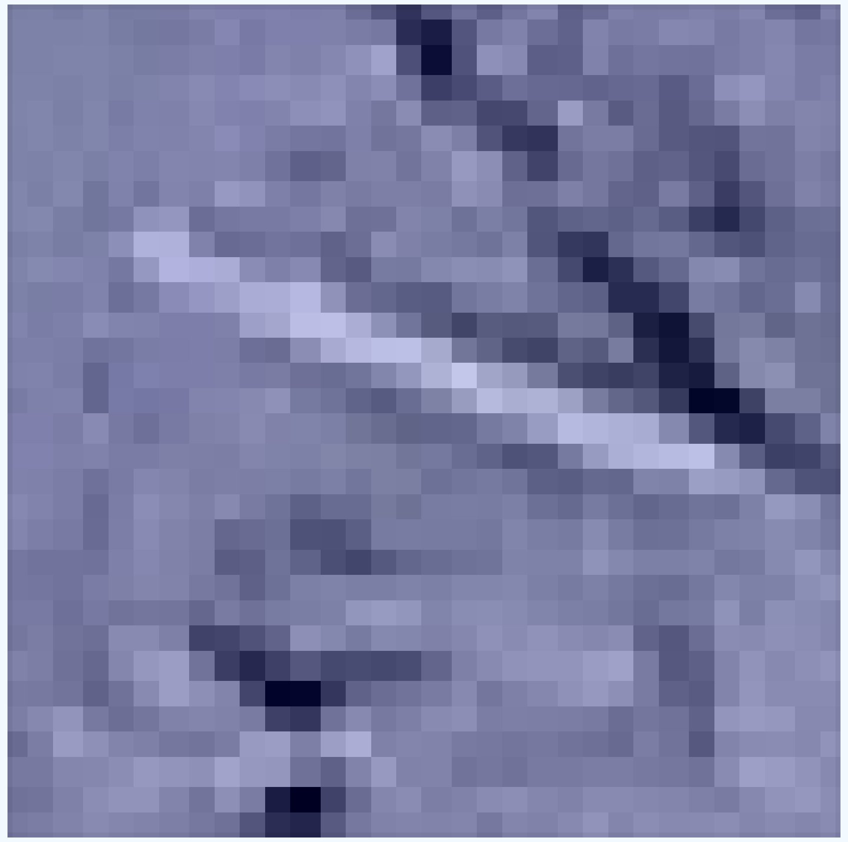
# CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION

# CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION

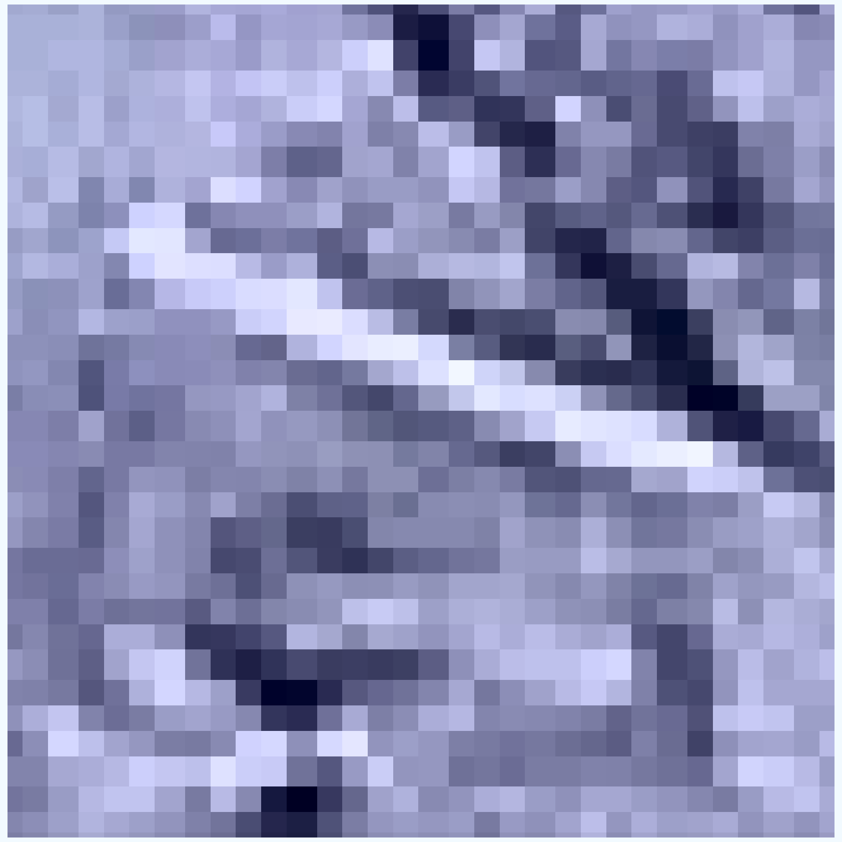
## STEPS TO APPLY CLAHE ALGORITHM

- Convert the image to the LAB color space using the cv2.cvtColor() function.
- Split the LAB image into its L, a, and b channels using the cv2.split() function.
- Apply CLAHE to the L-channel using the cv2.createCLAHE() function.
- Merge the CLAHE-enhanced L-channel with the original a and b channels using the cv2.merge() function
- Convert the enhanced LAB image back to the BGR color space using the cv2.cvtColor() function.
- Return the enhanced image

# CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION



Before CLAHE

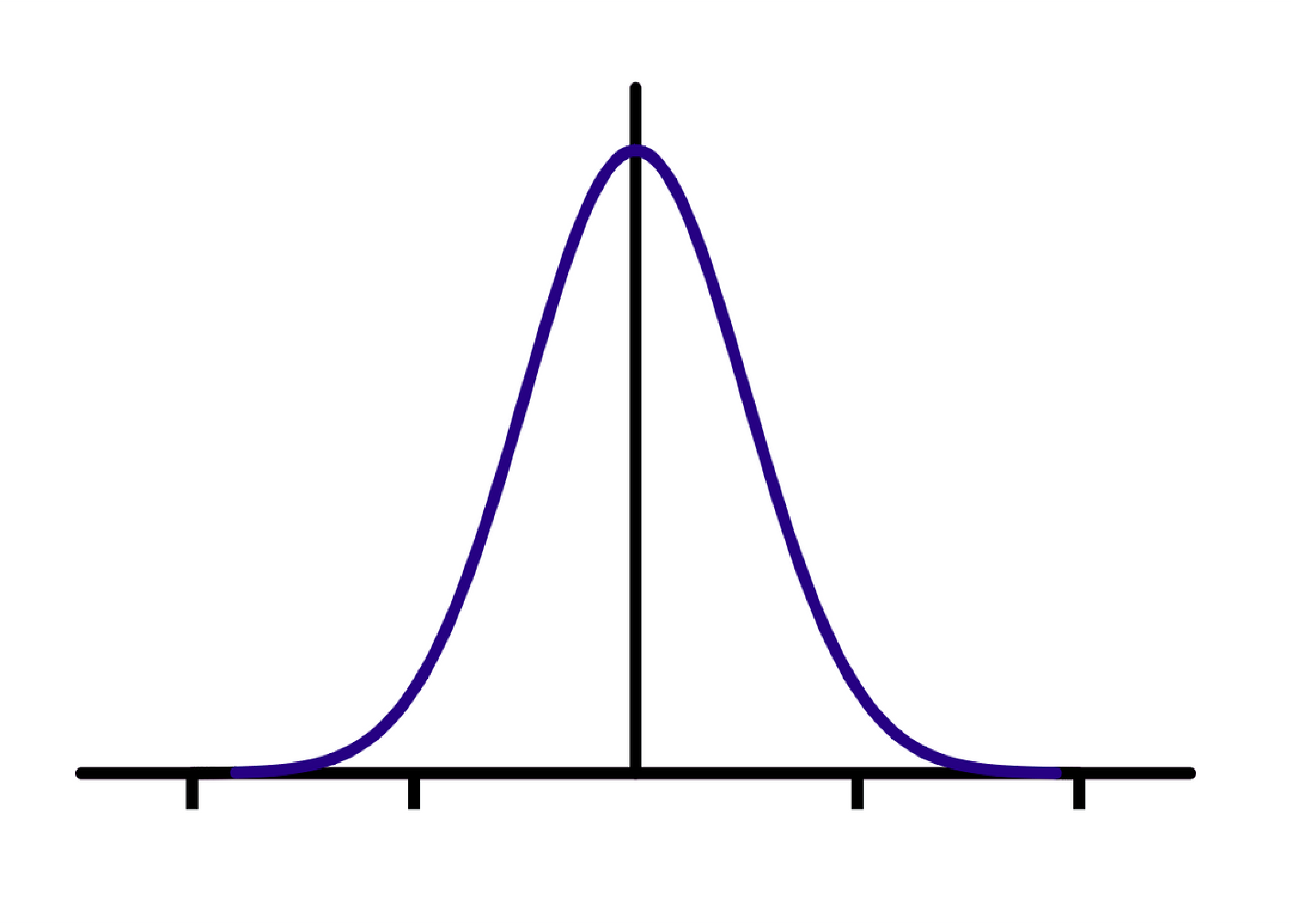


After CLAHE

# GAUSSIAN FILTER

## WHAT IS IT?

It is a low-pass filter that attenuates high-frequency components in the image while preserving the low-frequency components. This filtering operation helps in reducing noise and fine details in the image, resulting in a smoother appearance.

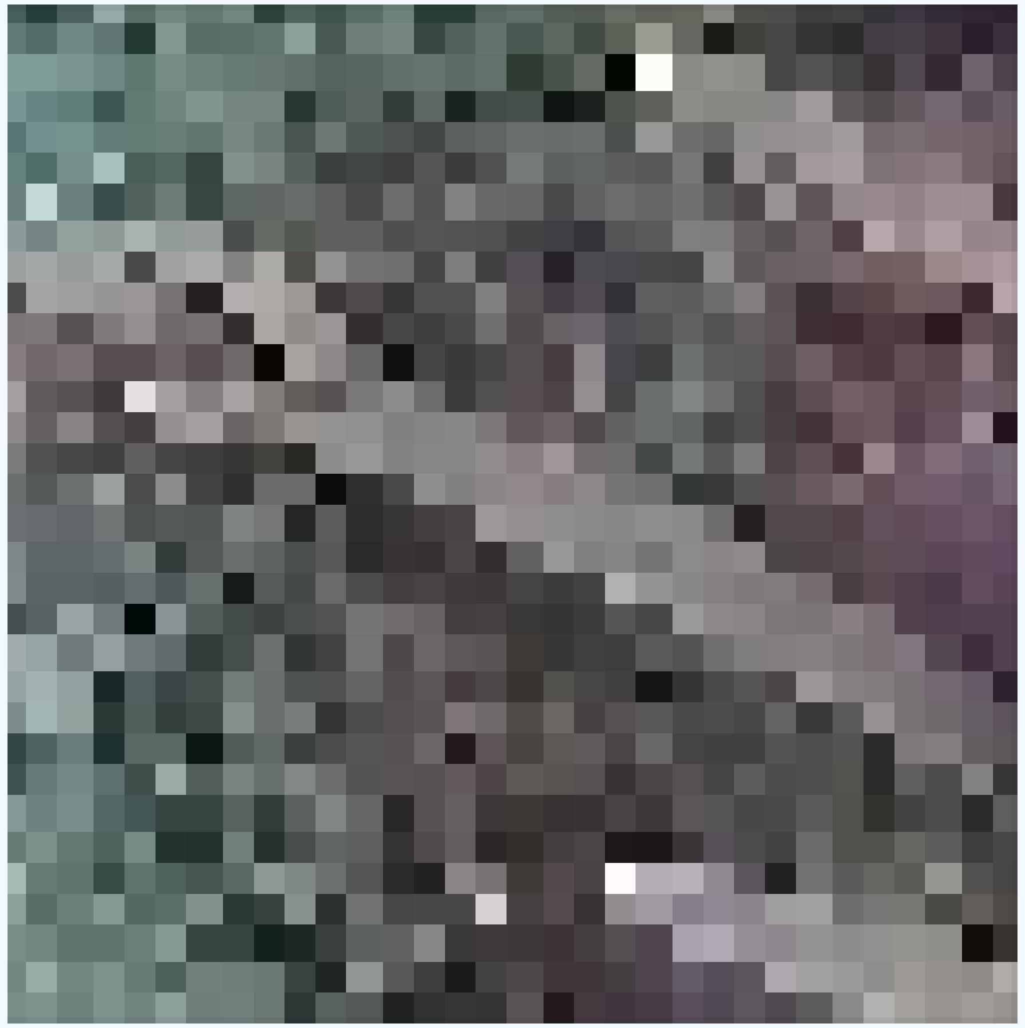


# GAUSSIAN FILTER

## HOW IT WORKS?

- A Gaussian kernel is created based on 3*3, because we were using low resolution images
- The Gaussian kernel is convolved with the input image by placing the kernel at each pixel location.
- The neighboring pixel values are multiplied by the corresponding weights in the Gaussian kernel and summed up.
- The resulting sum is divided by the total weight to obtain the smoothed pixel value.
- The Gaussian filter reduces high-frequency components and preserves low-frequency components, resulting in a blurred version of the image.

# GAUSSIAN FILTER



Image

Gaussian

Contrast

# LAPLACIAN FILTER

## WHAT IS IT?

It is an edge detection filter that highlights regions of rapid intensity changes in an image by calculating the second derivative of the image intensity.

```
Kernel For Laplacian Filter:
    0     1     0
    1    -4     1
    0     1     0
```

# LAPLACIAN FILTER

## HOW IT WORKS?

- The Laplacian filter is applied to the input image.
- The Laplacian filter calculates the second derivative of the image, which measures the rate of change of intensity.
- By taking the second derivative, the Laplacian filter enhances areas of rapid intensity change, such as edges and corners.
- The Laplacian filter highlights these areas by assigning higher values to regions with strong intensity variations.

# DATA PRE-PROCESSING
## CONCLUSIONS

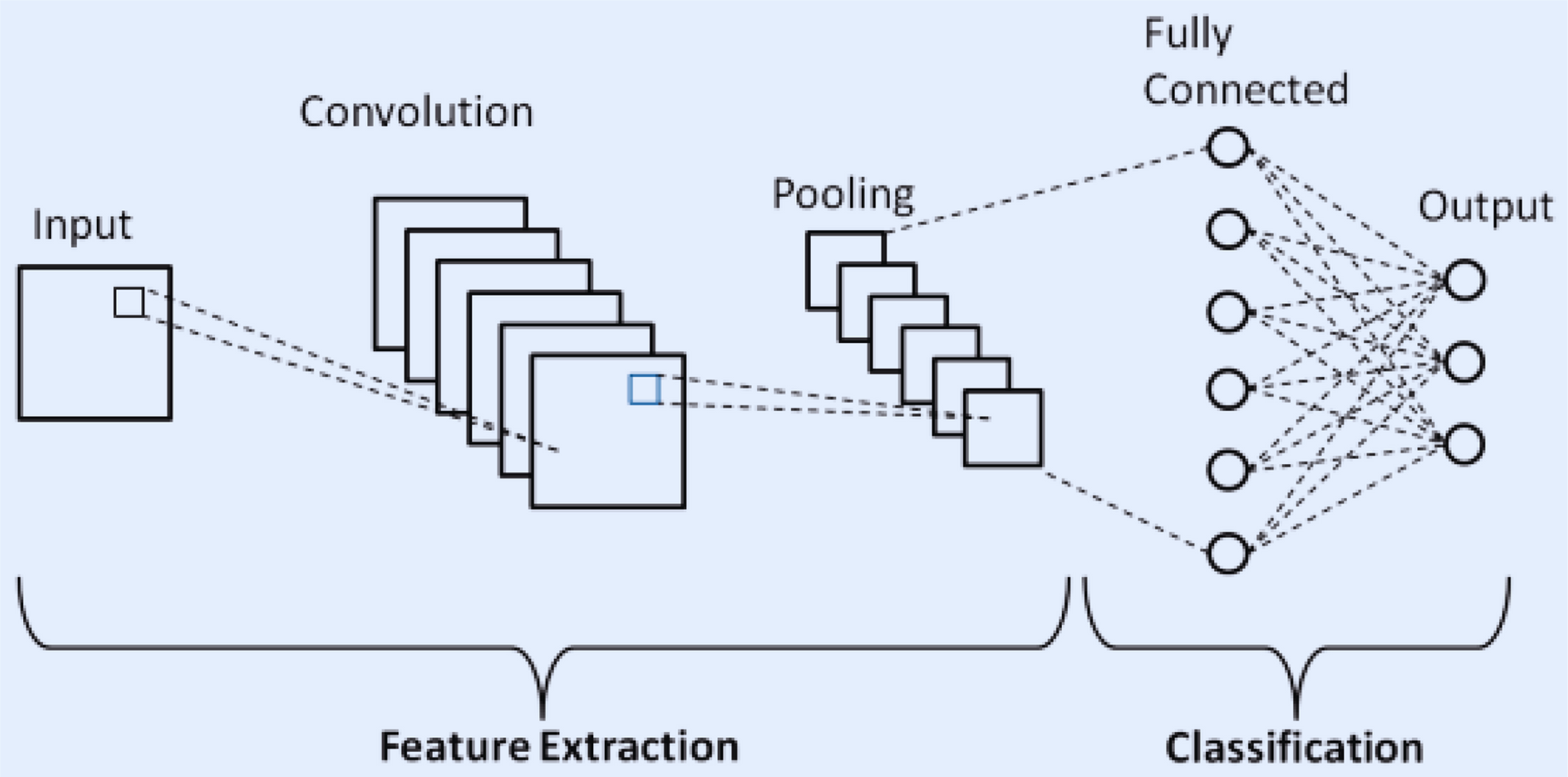
- At the beginning, to be cautious, we decided to test the models based on various image processing techniques.

- We noticed that the experiment with CLAHE, Gaussian and Laplacian Filter performed better.

- In the model selection we will see how choosing the right model will help to solve this dilemma.

# MODEL SELECTION

## CONVOLUTIONAL NEURAL NETWORKS



# MODEL SELECTION

## WHY CONVOLUTIONAL NEURAL NETWORK?

- CNNs can capture spatial hierarchies of patterns in images.
- CNNs utilize parameter sharing, where the same weights are applied across different spatial locations of the input. This reduces the risk of overfitting.
- CNNs utilize pooling layers like max pooling to downsample feature maps and retain essential information. Pooling reduces spatial dimensions while maintaining critical features, enhancing the network's resilience to input variations.

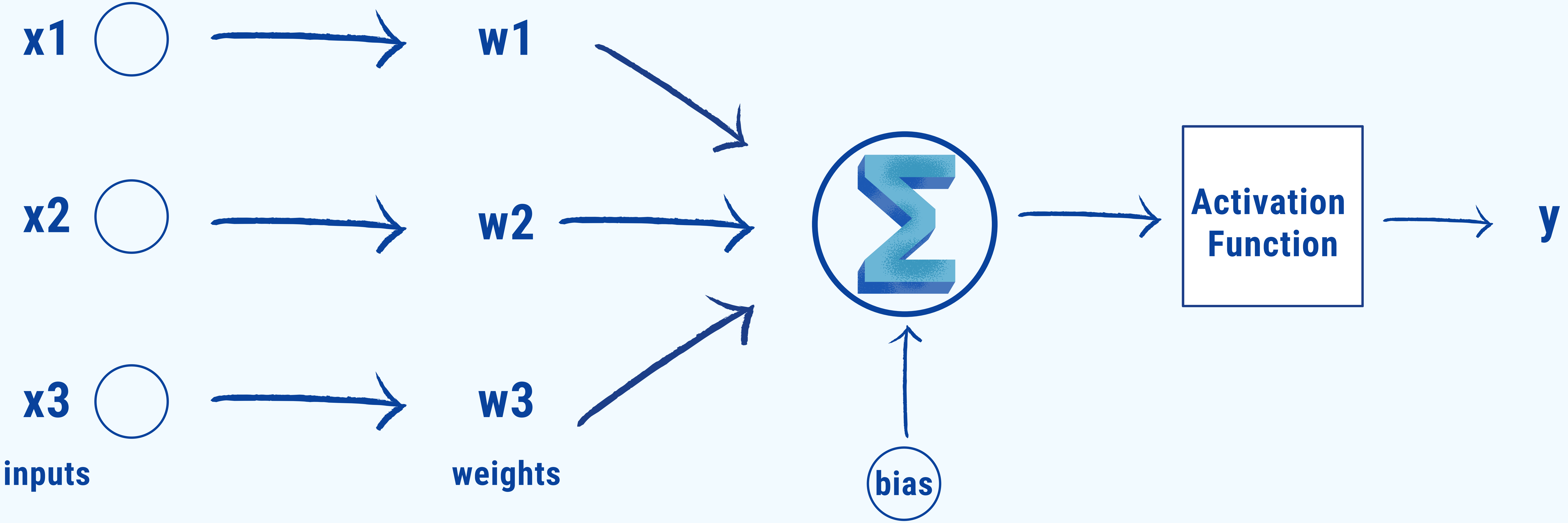
# MODEL SELECTION

## HOW CONVOLUTIONAL NEURAL NETWORK WORKS?

- Input image is convolved with filters to extract features.
- Rectified Linear Units (ReLU) activation function introduces non-linearity, capturing complex patterns.
- Max pooling reduces spatial dimensions, retaining important features.
- Dropout layers randomly deactivate neurons during training, preventing overfitting.
- Flattening converts the feature maps into a 1D vector.
- Fully connected layers perform classification based on learned features.
- Sigmoid activation function produces a binary classification prediction.

# MODEL SELECTION

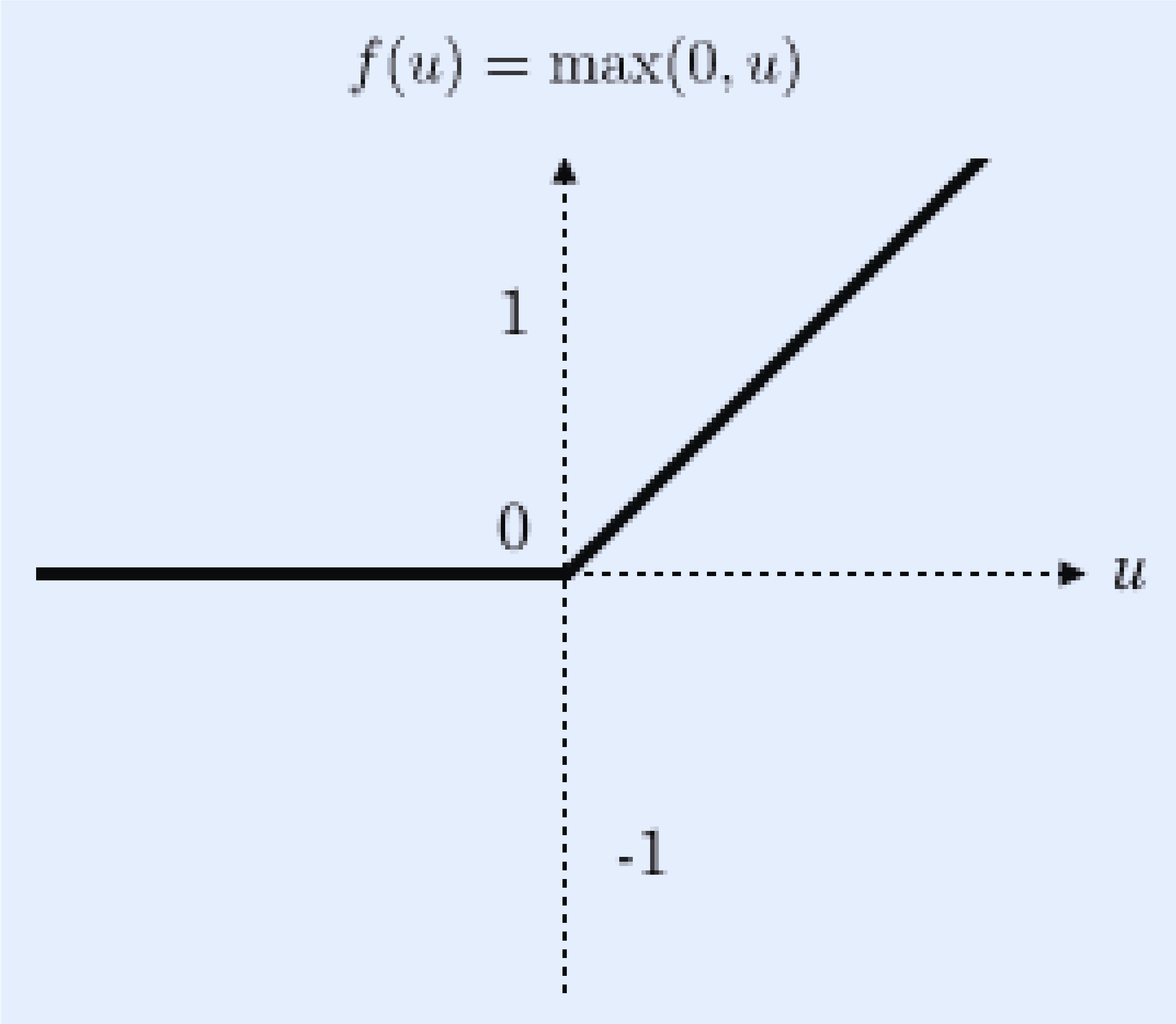
## ACTIVATION FUNCTION



# MODEL SELECTION

## WHY RELUs?

- ReLU is an activation function that introduces non-linear activation function.
- It avoids saturation problems that can occur with other activation functions.
- It promotes sparse activation by setting negative values to zero.




**$f(x) = \max(0,x)$**

# MODEL SELECTION

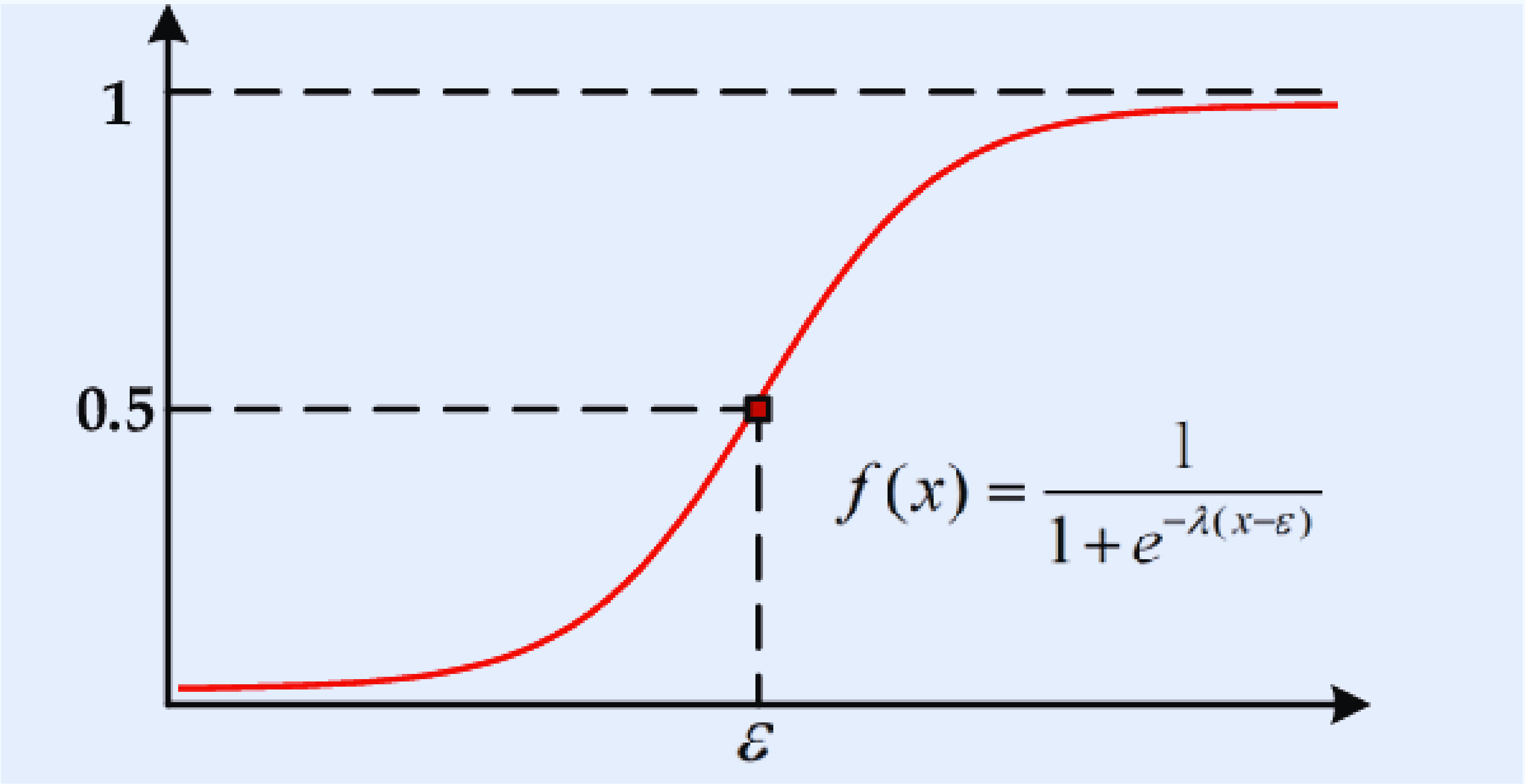
## WHY MAX POOLING?

- It reduces the spatial dimensions of the feature maps, effectively downsampling the information.
- Max pooling reduces the size of the feature maps, which can help prevent overfitting by reducing the number of parameters in subsequent layers.
- It helps to extract higher-level and more abstract representations of the input data as the receptive field increases with each pooling operation.

# MODEL SELECTION

## WHY SIGMOID ACTIVATION FUNCTION IN THE LAST LAYER?

By using the sigmoid activation function, the model can provide a clear decision boundary between the two classes, where values below 0.5 are classified as the negative class and values above 0.5 as the positive class.



# MODEL SELECTION

## WHY DROPOUT REGULARIZATION?

- It reduces the interdependence of neurons by randomly dropping out a certain percentage of neurons during training.
- By reducing co-adaptation among neurons, dropout helps to prevent overfitting.
- Dropout promotes faster convergence and reduces the chances of getting stuck in local optima.

# MODEL SELECTION

## WHY ADAM OPTIMIZER?

- It computes individual learning rates for different parameters, which helps in faster convergence and better optimization.
- It reduces the need for manual tuning of hyperparameters.
- Adam uses momentum, which accumulates the past gradients and uses it to update the parameters. This helps in faster convergence by preventing the optimizer from getting stuck in local minima.

# MODEL SELECTION

## WHAT ARE EPOCHS?

The number of complete passes that the entire training dataset undergoes during the training process.

### TOO MANY EPOCHS?

**OVERFITTING**

The model may have high accuracy on the training set but fail to generalize well.

### TOO LESS EPOCHS?

**UNDERFITTING**

It may have a high bias and low variance, resulting in poor accuracy on both the training and test/validation sets.

# HYPERPARAMETER TUNING

## WHAT ARE HYPERPARAMETERS?

Hyperparameters are the variables which determines the network structure and the variables which determine how the network is trained.

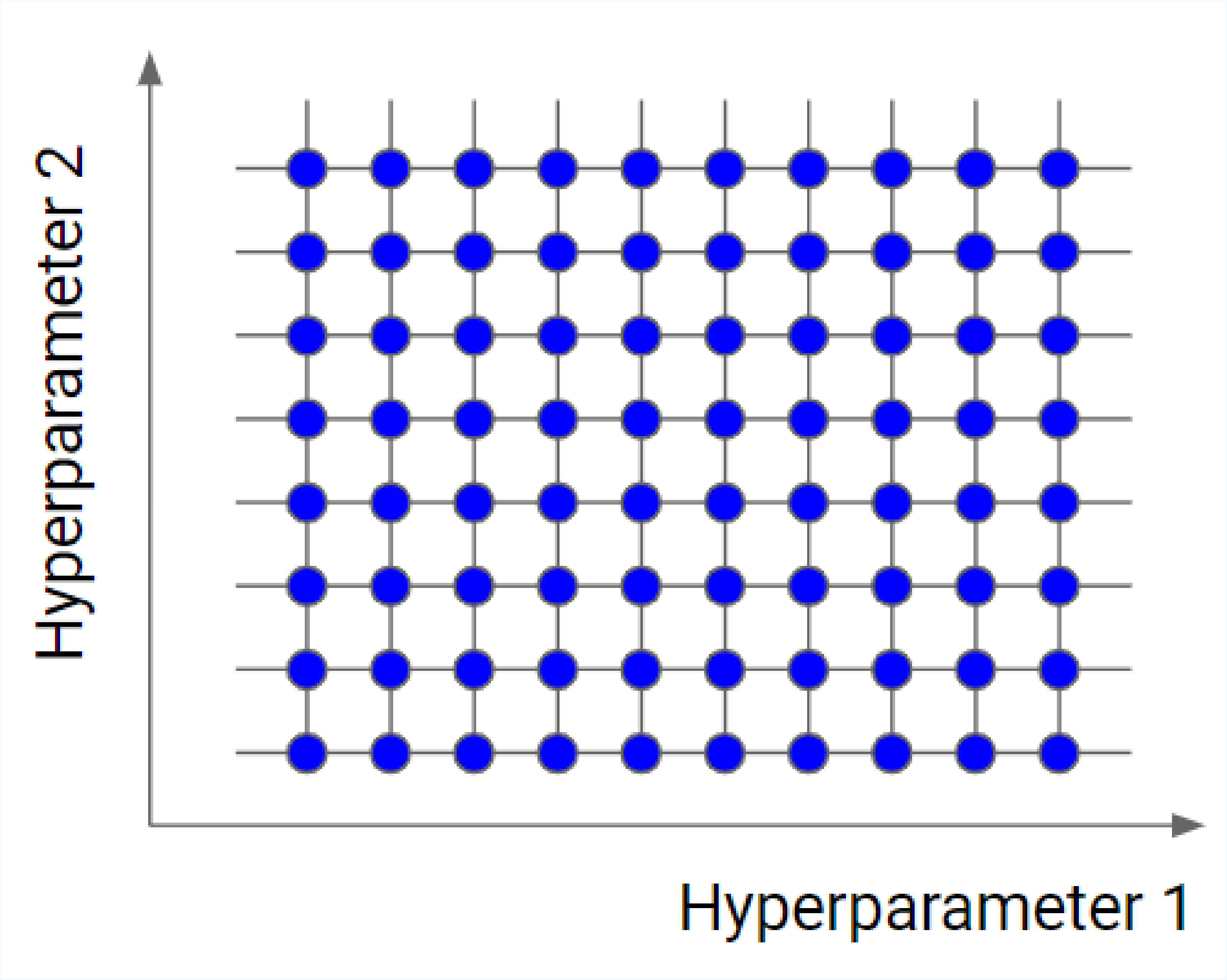
## WHY HYPERPARAMETER TUNING?

It aims to optimize model performance, enhance generalization ability, ensure training stability, efficiently utilize resources, and adapt the model to the specific characteristics of the data.

# HYPERPARAMETER TUNING

## WHAT IS GRID SEARCH?

Grid Search uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters. The dimension is the number of hyper-parameters that we are tuning.



# HYPERPARAMETER TUNING

## WHY GRID SEARCH?

- Automates the process of hyperparameter tuning by systematically exploring different combinations of hyperparameters.
- Improves the generalization of the model by finding hyperparameters that work well across multiple cross-validation folds.
- Reduces the need for manual trial and error in selecting hyperparameters.

# HYPERPARAMETER TUNING

## PARAMETER SELECTION FOR GRID SEARCH

- **batch_size:** The grid search includes two different batch sizes, namely 32 and 64. These batch sizes represent the number of training examples processed in each iteration.

- **epochs:** Two distinct numbers of epochs, 30 and 50, were selected. These values indicate the number of complete passes or iterations over the training dataset during the model training process.

# MODEL SELECTION
## CONVOLUTIONAL LAYERS

| Convolutional layer | NUMBER OF FILTERS | FILTER SIZE | NUMBER OF PARAMETERS |
|---|---|---|---|
| Conv2D_1 | 32 | 3x3x3 | (3 x 3 x 3 + 1) x 32 = 896 |
| Conv2D_2 | 64 | 3x3x32 | (3 x 3 x 32 + 1) x 64 = 18496 |
| Conv2D_3 | 128 | 3x3x64 | (3 x 3 x 64 + 1) x 128 = 73856 |

# MODEL SELECTION

## SOME EXPERIMENTS

| MODEL | PREPROCESSING STEPS | TEST ACCURACY |
| --- | --- | --- |
| CNN | Convert To Green, Contrast Enhancement | 0.9894 |
| CNN | Contrast Enhancement, Normalization (0-1) | 0.9869 |
| CNN | Convert To Green, Contrast Enhancement, Contour Detection, Normalization (0-1) | 0.9779 |
| CNN | Averaging Filter, Convert To Green, Contrast Enhancement, Contour Detection, Normalization (0-1) | 0.9599 |
| CNN | Averaging Filter, Contrast Enhancement (CLAHE), Sobel Filter, Normalization (0-1) | 0.9685 |
| CNN | Averaging Filter, Sobel Filter, Normalization (0-1) | 0.9728 |
| CNN | Contrast, Gaussian Filter, Laplacian, Normalization (0-1) | 0.9917 |

# MODEL PERFORMANCE

- The use of convolutional layers in the CNN model allows the model to capture spatial relationships and patterns in the input images effectively.
- The multiple convolutional layers in the model can learn hierarchical representations of the cactus images.
- The CNN model is able to effectively extract meaningful features from the cactus images and make accurate predictions, resulting in better model performance for the cactus classification task.

# CONCLUSION

- The use of convolutional layers in the CNN model allows the model to capture spatial relationships and patterns in the input images effectively.
- The multiple convolutional layers in the model can learn hierarchical representations of the cactus images.
- The CNN model is able to effectively extract meaningful features from the cactus images and make accurate predictions, resulting in better model performance for the cactus classification task.
- However, we did not focus our attention on a parameterization of the CNN, which could lead to better results

# THE TEAM


**SHUBHIKA GARG**

Master's in Data Science and Engineering

**YOSSI TAPIERO**

Telecom SudParis

**DORIAN LEBRU**

Telecom SudParis

**JORDAN ALLEMAND**

Telecom SudParis