TEAM – 03

LUNG DISEASE DETECTION

1. Frequency domain filters for smoothing and sharpening:

Smoothing operation:

(Low Pass Domain Filters)

Gaussian Low Pass:

```
% Gaussian Low Pass
image = imread('COVID_19.jpeg');

image = im2double(image);

sigma = 200.0;
D = 2.5 * sigma;

[M, N] = size(image);

u = 0:(M - 1);
v = 0:(N - 1);
u(u > M/2) = u(u > M/2) - M;
v(v > N/2) = v(v > N/2) - N;
[V, U] = meshgrid(v, u);

H = exp(-(U.^2 + V.^2) / (2 * sigma^2));

H = ifftshift(H);

filtered_image = ifft2(fft2(image) .* H);
subplot(1, 2, 1);
imshow(image);
title('Input Image');

subplot(1, 2, 2);
imshow(abs(filtered_image), []);
title('Output Image');
```

Ideal Low Pass:

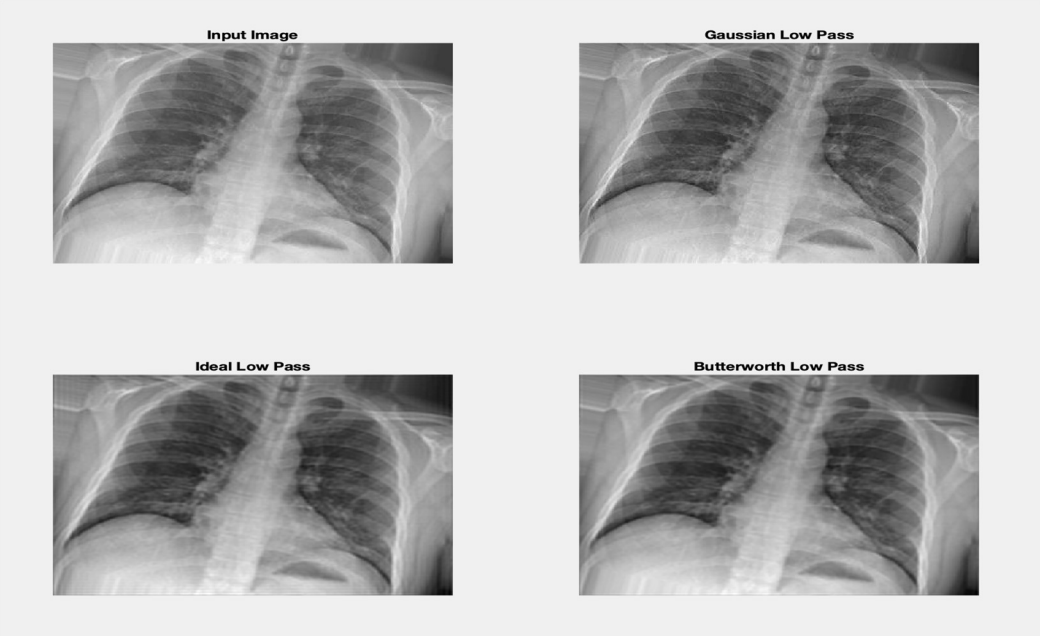
```
%Ideal Low Pass Filter
input_image = imread('COVID_19.jpeg');

[M, N] = size(input_image);

FT_img = fft2(double(input_image));

D0 = 50;
u = 0:(M-1);
idx = find(u>M/2);
u(idx) = u(idx)-M;
v = 0:(N-1);
idy = find(v>N/2);
v(idy) = v(idy)-N;

[V, U] = meshgrid(v, u);

D = sqrt(U.^2+V.^2);

H = double(D <= D0);

G = H.*FT_img;
output_image = real(ifft2(double(G)));

subplot(1, 2, 1), imshow(input_image),
title('Input Image')

subplot(1, 2, 2), imshow(output_image, [ ]);
title('Output Image')
```

Butterworth Low Pass:

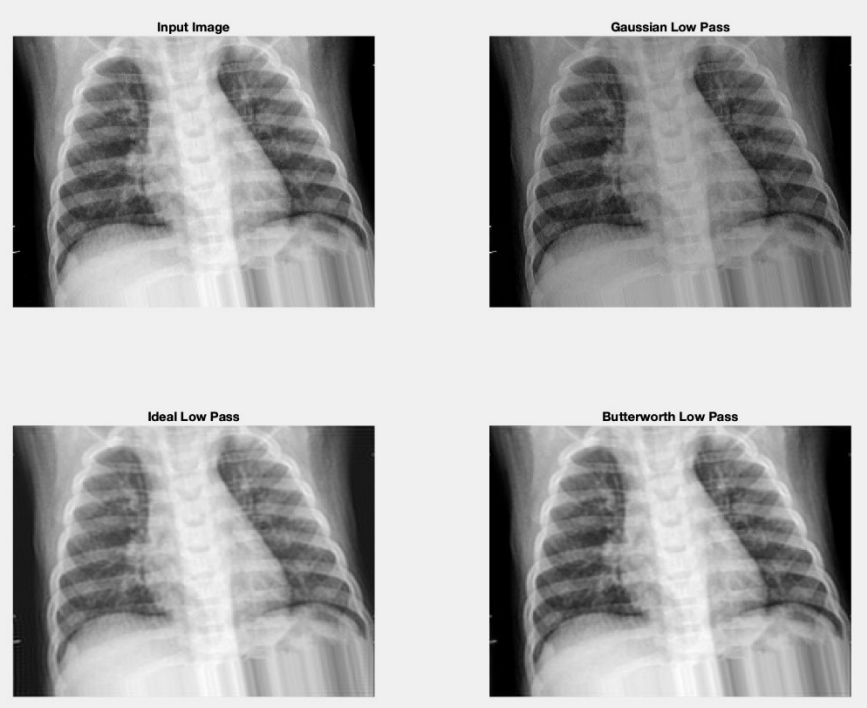
```
%Butterworth Low Pass
image = imread('COVID_19.jpeg');
if size(image, 3) == 3
    image = rgb2gray(image);
end
n = 4;
fc = 75.0;

[M, N] = size(image);

u = 0:(M - 1);
v = 0:(N - 1);
u(u > M/2) = u(u > M/2) - M;
v(v > N/2) = v(v > N/2) - N;
[V, U] = meshgrid(v, u);

D = sqrt(U.^2 + V.^2);
H = 1 ./ (1 + (D / fc).^(2 * n));

filtered_spectrum = fft2(image) .* H;
filtered_image = ifft2(filtered_spectrum);
figure;
subplot(1, 2, 1);
imshow(image);
title('Input Image');

subplot(1, 2, 2);
imshow(abs(filtered_image), []);
title('Output Image');
```

OUTPUTS:

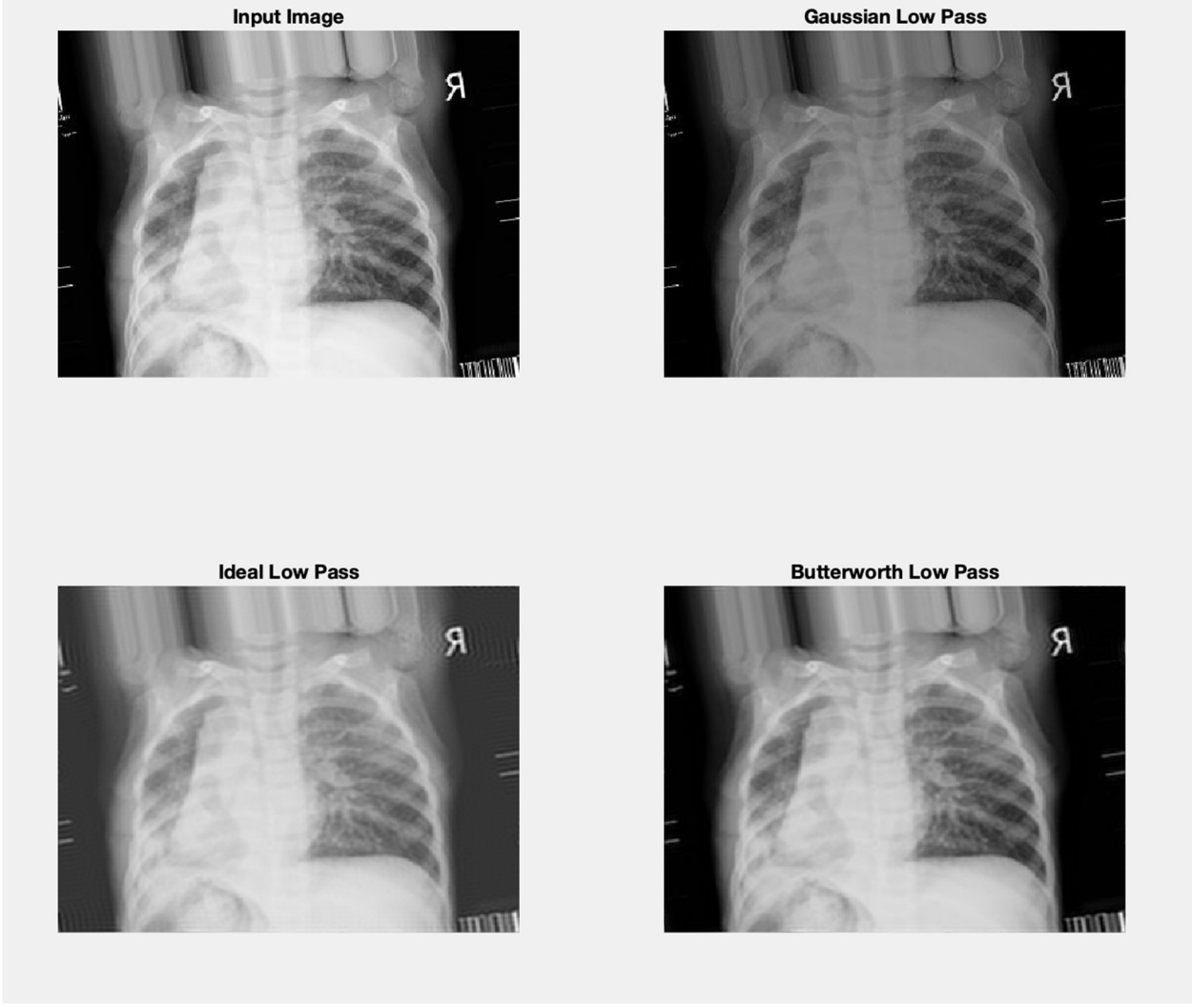
COVID 19:



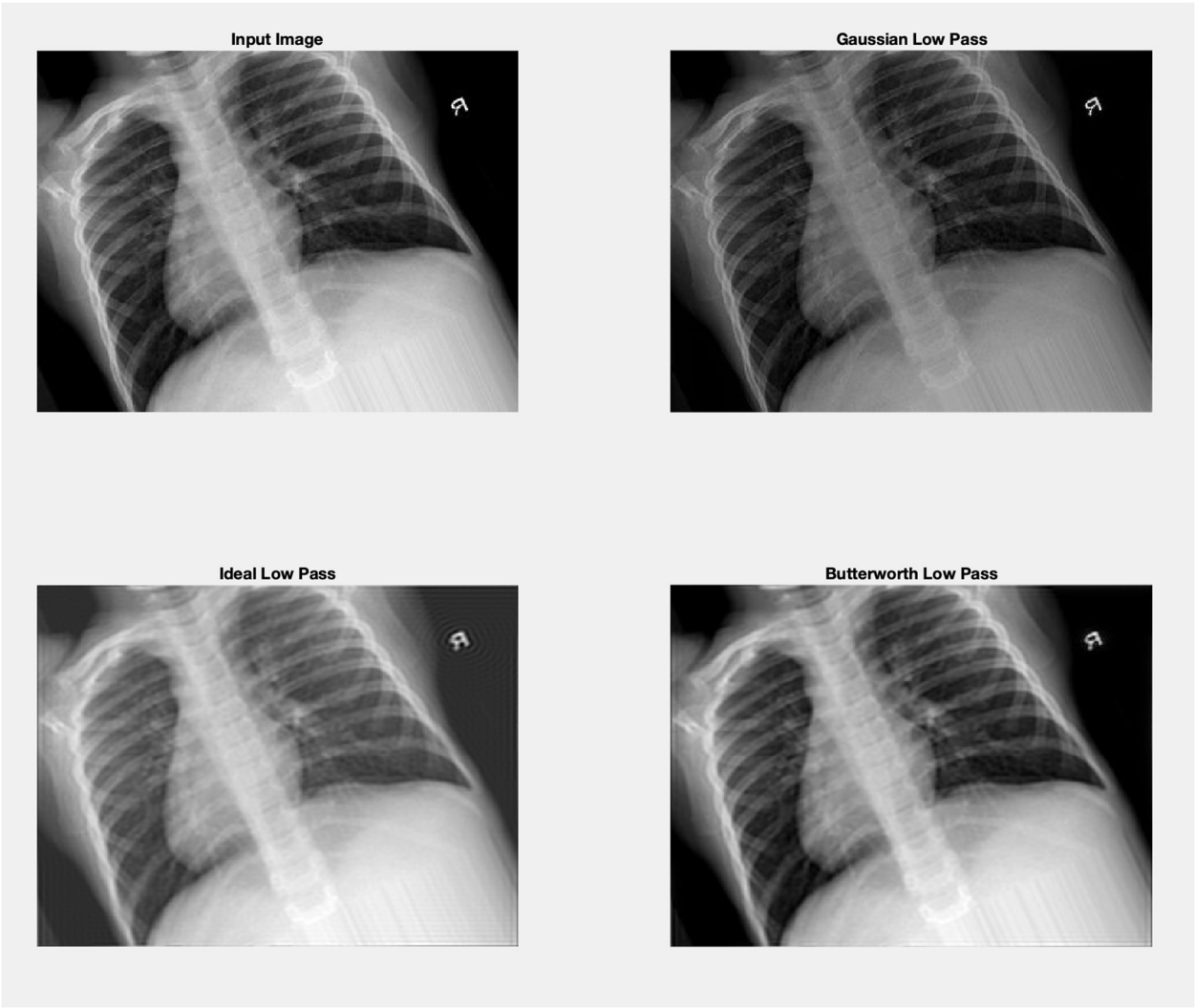
ViralPneumonia:



BacterialPneumonia:



Normal:



Inference:

COVID-19:

In case of Images from patients affected with covid-19 gaussian low pass filter works the best as it darkens the unaffected regions of the lungs and such that the diseased regions become clearly visible.

ViralPneumonia, BacterialPneumonia,Normal:

In case of Images from patients affected with ViralPneumonia ,BacterialPneumonia and Normal lung images butterworth low pass filter works the best as it reduces the noise so the image is clearly visible.

To be precise it reduces the whitish shade in the Xrays which might give use significant error when these images are used to train the model.

Sharpening operation:

(High Pass Domain Filters)

Ideal High Pass:

```
input_image = imread('covid.jpg');

[M, N] = size(input_image);
FT_img = fft2(double(input_image));

D0 = 50;
u = 0:(M-1);
idx = find(u>M/2);
u(idx) = u(idx)-M;
v = 0:(N-1);
idy = find(v>N/2);
v(idy) = v(idy)-N;

[V, U] = meshgrid(v, u);
D = sqrt(U.^2 + V.^2);
H = double(D > D0);

G = H.*FT_img;
output_image = real(ifft2(double(G));

subplot(1, 2, 1), imshow(input_image),
title('Input Image')
subplot(1, 2, 2), imshow(output_image, [ ]);
title('Ideal High-Pass Filter)')
```

ButterWorth High Pass:

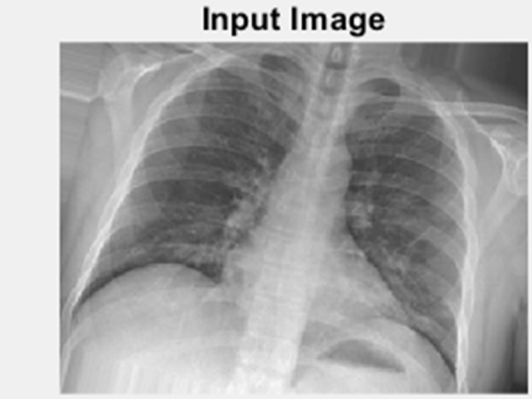
```
image = imread('covid.jpg');
if size(image, 3) == 3
    image = rgb2gray(image);
end
n = 4;
fc = 75.0;
[M, N] = size(image);

u = 0:(M - 1);
v = 0:(N - 1);
u(u > M/2) = u(u > M/2) - M;
v(v > N/2) = v(v > N/2) - N;
[V, U] = meshgrid(v, u);

D = sqrt(U.^2 + V.^2);
H = 1 - 1 ./ (1 + (D / fc).^(2 * n));

filtered_spectrum = fft2(image) .* H;
filtered_image = ifft2(filtered_spectrum);
figure;
subplot(1, 2, 1);
imshow(image);
title('Input Image');

subplot(1, 2, 2);
imshow(abs(filtered_image), []);
title('Output Image (High-Pass Filter)');
```

Gaussian High Pass:

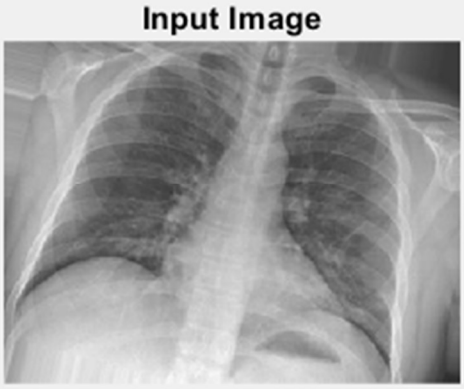
```
% Gaussian High Pass
image = imread('covid.jpg');

image = im2double(image);

sigma = 200.0;
D = 2.5 * sigma;

[M, N] = size(image);

u = 0:(M - 1);
v = 0:(N - 1);
u(u > M/2) = u(u > M/2) - M;
v(v > N/2) = v(v > N/2) - N;
[V, U] = meshgrid(v, u);

H = 1-exp(-(U.^2 + V.^2) / (2 * sigma^2));

H = ifftshift(H);

filtered_image = ifft2(fft2(image) .* H);
subplot(1, 2, 1);
imshow(image);
title('Input Image');

subplot(1, 2, 2);
imshow(abs(filtered_image), []);
title('Gaussian High Pass');
```

OUTPUTS:

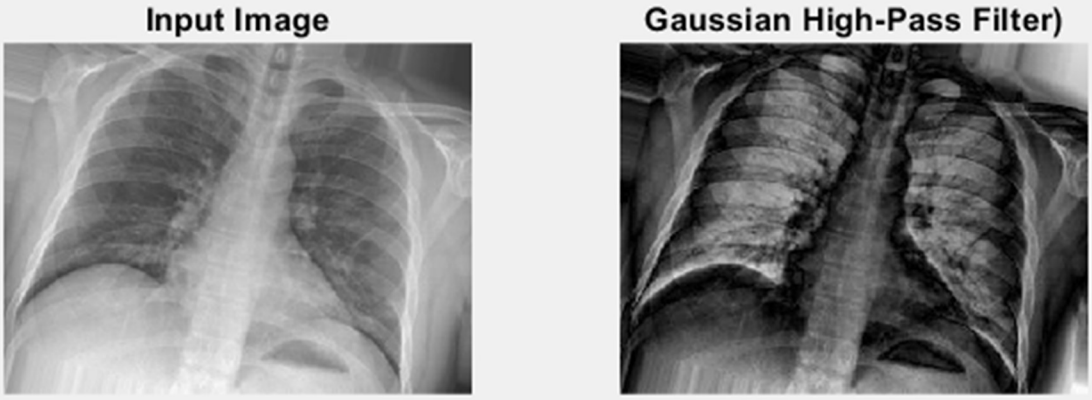
Ideal High Pass:



Butterworth High Pass:



Gaussian High Pass Filter:



Inference:

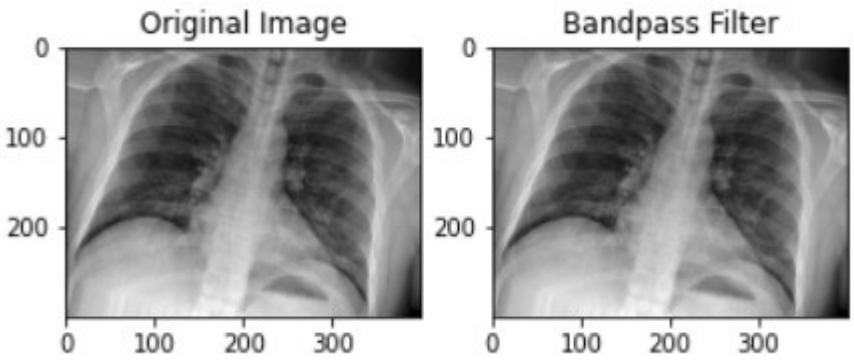
Ideal mask filter here helps in enhancing the diseased regions which will help in getting good model score (accuracy) while training the model.

Bandpass filter operation:

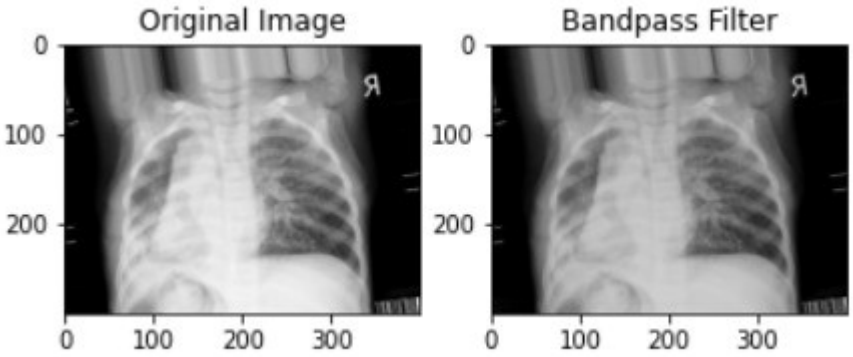
Ideal Filter:

In [116]:
```
import numpy as np
import cv2
from matplotlib import pyplot as plt

image = cv2.imread('covid_19.jpeg', cv2.IMREAD_GRAYSCALE)

lower_cutoff =  1# Adjust this value according to your requirements
upper_cutoff = 220  # Adjust this value according to your requirements

dft = np.fft.fft2(image)
dft_shift = np.fft.fftshift(dft)

rows, cols = image.shape
center_row = rows // 2
center_col = cols // 2

# Create a filter mask
mask = np.zeros((rows, cols), dtype=np.uint8)

for i in range(rows):
    for j in range(cols):
        distance = np.sqrt((i - center_row) ** 2 + (j - center_col) ** 2)
        if lower_cutoff <= distance <= upper_cutoff:
            mask[i, j] = 1


filtered_dft = dft_shift * mask

filtered_image = np.abs(np.fft.ifft2(np.fft.ifftshift(filtered_dft)))

plt.subplot(121), plt.imshow(image, cmap='gray'), plt.title('Original Image')
plt.subplot(122), plt.imshow(filtered_image, cmap='gray'), plt.title('Bandpass Filter')
plt.show()
```

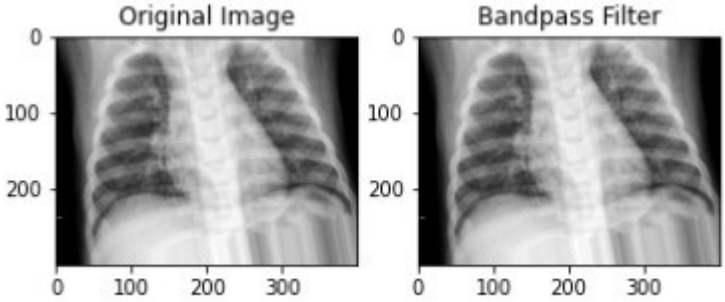
Covid:



Bacteria:


Original Image
Bandpass Filter
0
100
200
300

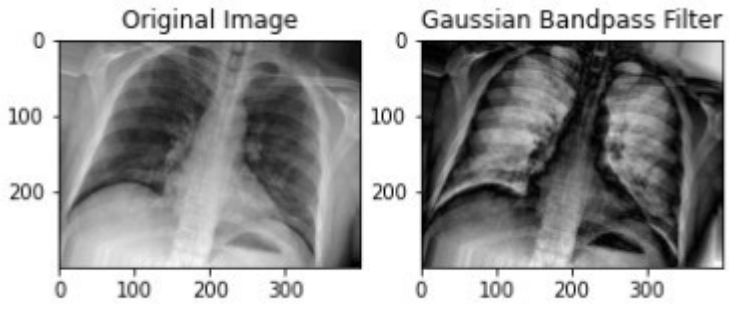
Viral:


Original Image
Bandpass Filter
0
100
200
300

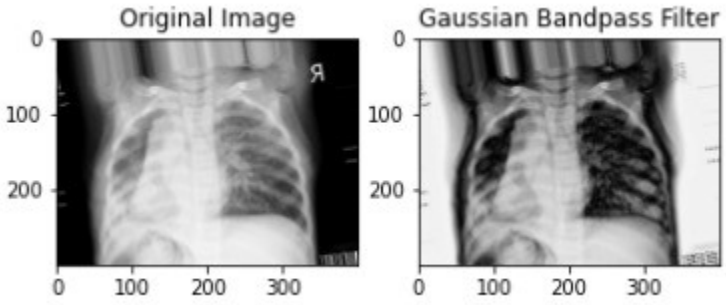
Gaussian Filter:

```
: import numpy as np
  import cv2
  from matplotlib import pyplot as plt

  image = cv2.imread('covid_19.jpeg', cv2.IMREAD_GRAYSCALE)

  lower_cutoff =  100# Adjust this value according to your requirements
  upper_cutoff = 250  # Adjust this value according to your requirements

  dft = np.fft.fft2(image)
  dft_shift = np.fft.fftshift(dft)

  rows, cols = image.shape
  center_row = rows // 2
  center_col = cols // 2

  sigma = 10  # Adjust this value according to your requirements
  mask = np.zeros((rows, cols), dtype=np.uint8)

  for i in range(rows):
      for j in range(cols):
          distance = np.sqrt((i - center_row) ** 2 + (j - center_col) ** 2)
          mask[i, j] = np.exp(-(distance ** 2) / (2 * sigma ** 2))

  mask = 1 - mask  # Invert the filter to create a bandpass effect

  filtered_dft = dft_shift * mask

  filtered_image = np.abs(np.fft.ifft2(np.fft.ifftshift(filtered_dft)))

  plt.subplot(121), plt.imshow(image, cmap='gray'), plt.title('Original Image')
  plt.subplot(122), plt.imshow(filtered_image, cmap='gray'), plt.title('Gaussian Bandpass Filter')
  plt.show()
```

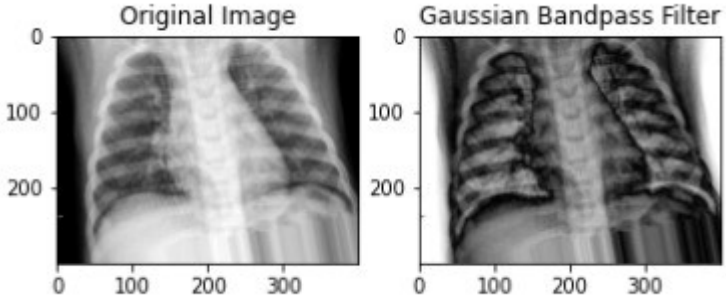
Covid:



Bacteria:



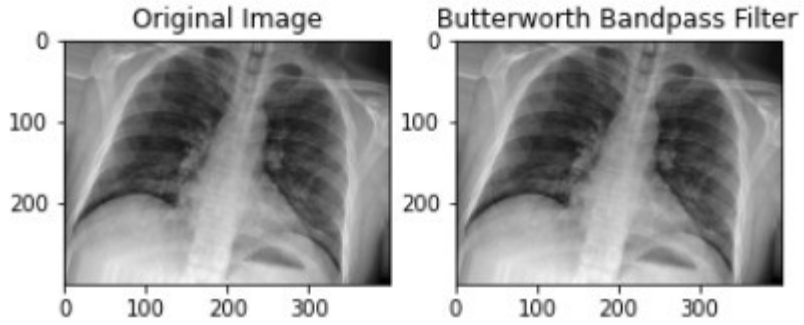
Viral:



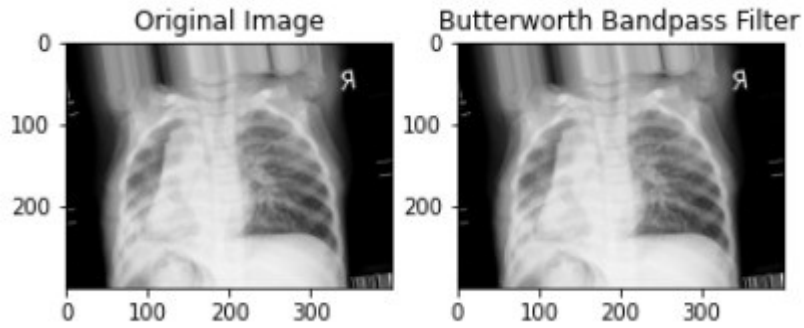
ButterWorth Filter:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Read the input image
image = cv2.imread('viral_pnemonia.jpeg', cv2.IMREAD_GRAYSCALE)

# Set the lower and upper cutoff frequencies and the filter order
lower_cutoff = 50 # Adjust this value according to your requirements
upper_cutoff = 150 # Adjust this value according to your requirements
order = 2  # Adjust this value according to your requirements

# Calculate the 2D Fourier Transform of the image
dft = np.fft.fft2(image)
dft_shift = np.fft.fftshift(dft)

# Create the Butterworth bandpass filter in the frequency domain
rows, cols = image.shape
center_row = rows // 2
center_col = cols // 2

mask = np.zeros((rows, cols), dtype=np.uint8)

for i in range(rows):
    for j in range(cols):
        distance = np.sqrt((i - center_row) ** 2 + (j - center_col) ** 2)
        mask[i, j] = 1 / (1 + (distance / upper_cutoff) ** (2 * order))
        mask[i, j] *= 1 - 1 / (1 + (distance / lower_cutoff) ** (2 * order))

# Apply the filter to the Fourier-transformed image
filtered_dft = dft_shift * (1-mask)

# Calculate the inverse Fourier Transform to obtain the filtered image
filtered_image = np.abs(np.fft.ifft2(np.fft.ifftshift(filtered_dft)))

# Display the original and filtered images
plt.subplot(121), plt.imshow(image, cmap='gray'), plt.title('Original Image')
plt.subplot(122), plt.imshow(filtered_image, cmap='gray'), plt.title('Butterworth Bandpass Filter')
plt.show()
```

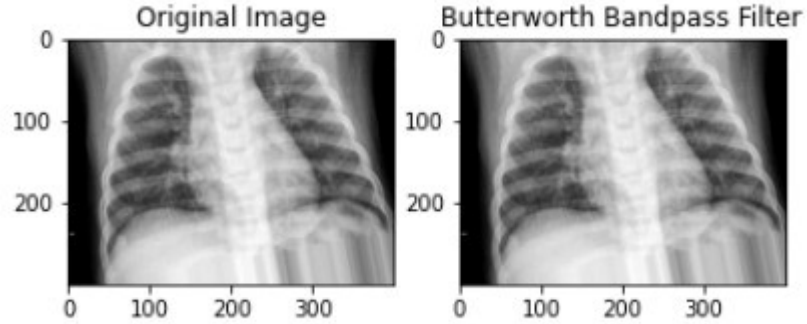
Covid:



Bacteria:



Viral:



Inference:

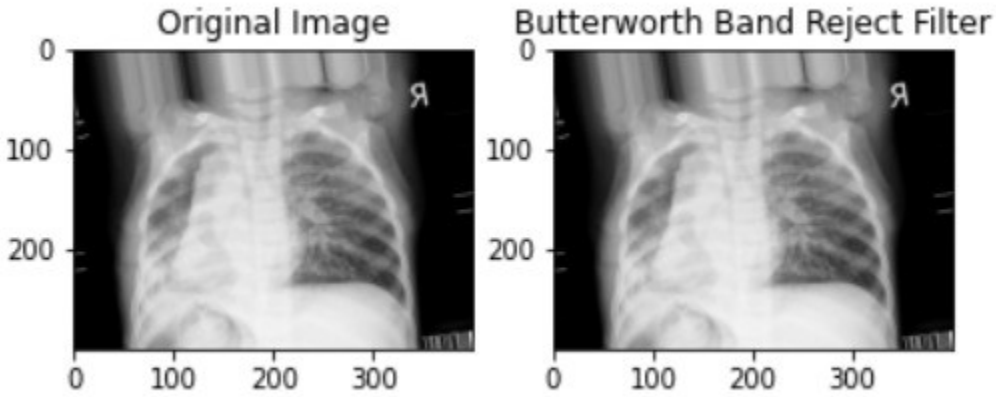
For bacterial pneumonia gaussian bandpass works best
For viral and covid we can use original images itself. Filtering does not help

Band Reject Filter:

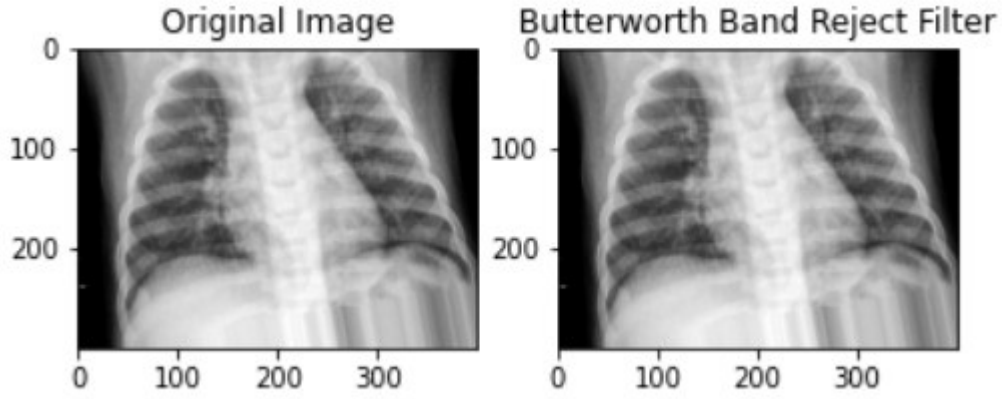
```
plt.subplot(121), plt.imshow(image, cmap='gray'), plt.title('Original Image')
plt.subplot(122), plt.imshow(image - 0.5*filtered_image, cmap='gray'), plt.title('Reject Filter')
plt.show()
```

Butterworth Filter:

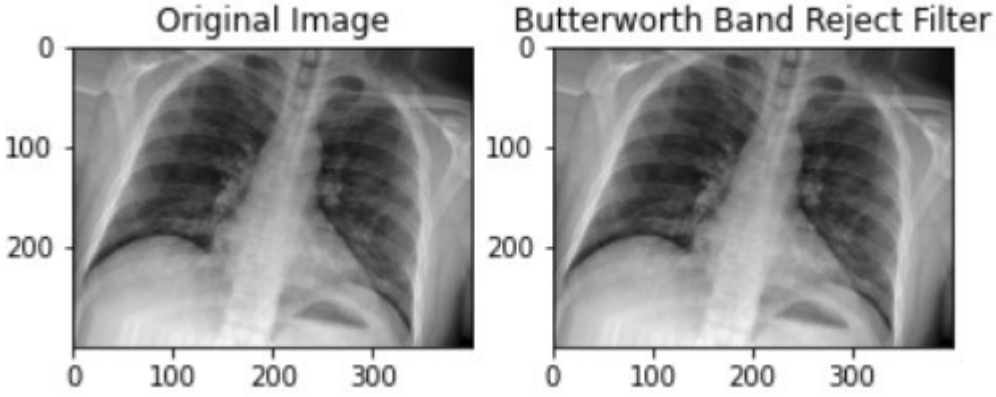
Bacterial:



Viral:

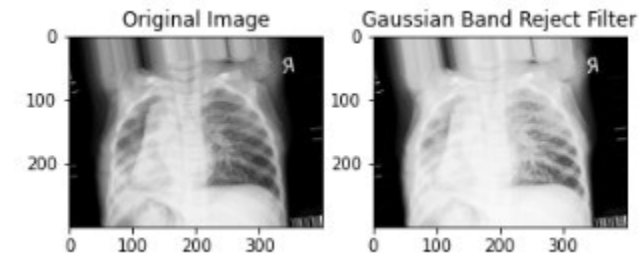


Covid:

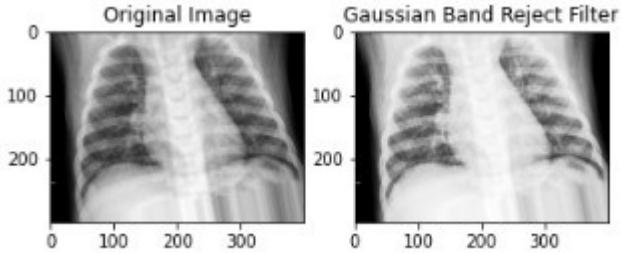


Gaussian Filter:

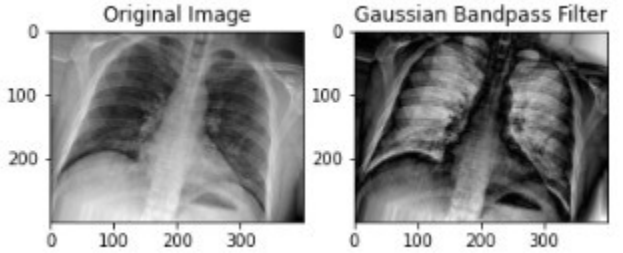
Bacterial:



Viral:

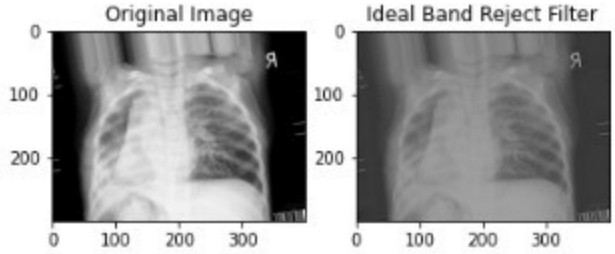


Covid:

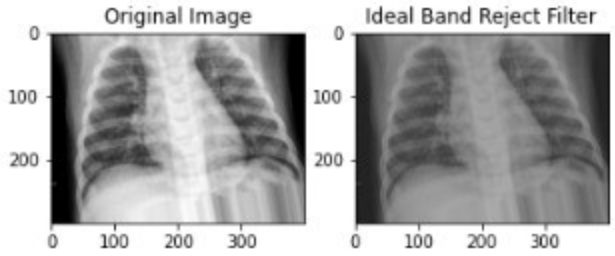


Ideal Filter:

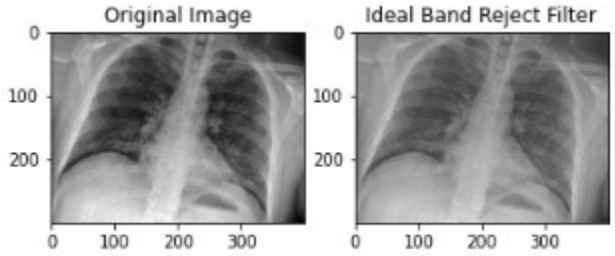
Bacterial:



Viral:


Original Image
Ideal Band Reject Filter
0
100
200
0
100
200
300

Covid:


Original Image
Ideal Band Reject Filter
0
100
200
0
100
200
300

2. Inference based on the results:

Low Pass Filter:

COVID-19:

In case of Images from patients affected with covid-19 gaussian low pass filter works the best as it darkens the unaffected regions of the lungs and such that the diseased regions become clearly visible.

ViralPneumonia, BacterialPneumonia,Normal:

In case of Images from patients affected with ViralPneumonia ,BacterialPneumonia and Normal lung images butterworth low pass filter works the best as it reduces the noise so the image is clearly visible.
To be precise it reduces the whitish shade in the Xrays which might give use significant error when these images are used to train the model.

High Pass Filter:

Ideal mask filter here helps in enhancing the diseased regions which will help in getting good model score (accuracy) while training the model.

Band Pass Filter:

For bacterial pneumonia gaussian bandpass works best
For viral and covid we can use original images itself. Filtering does not help