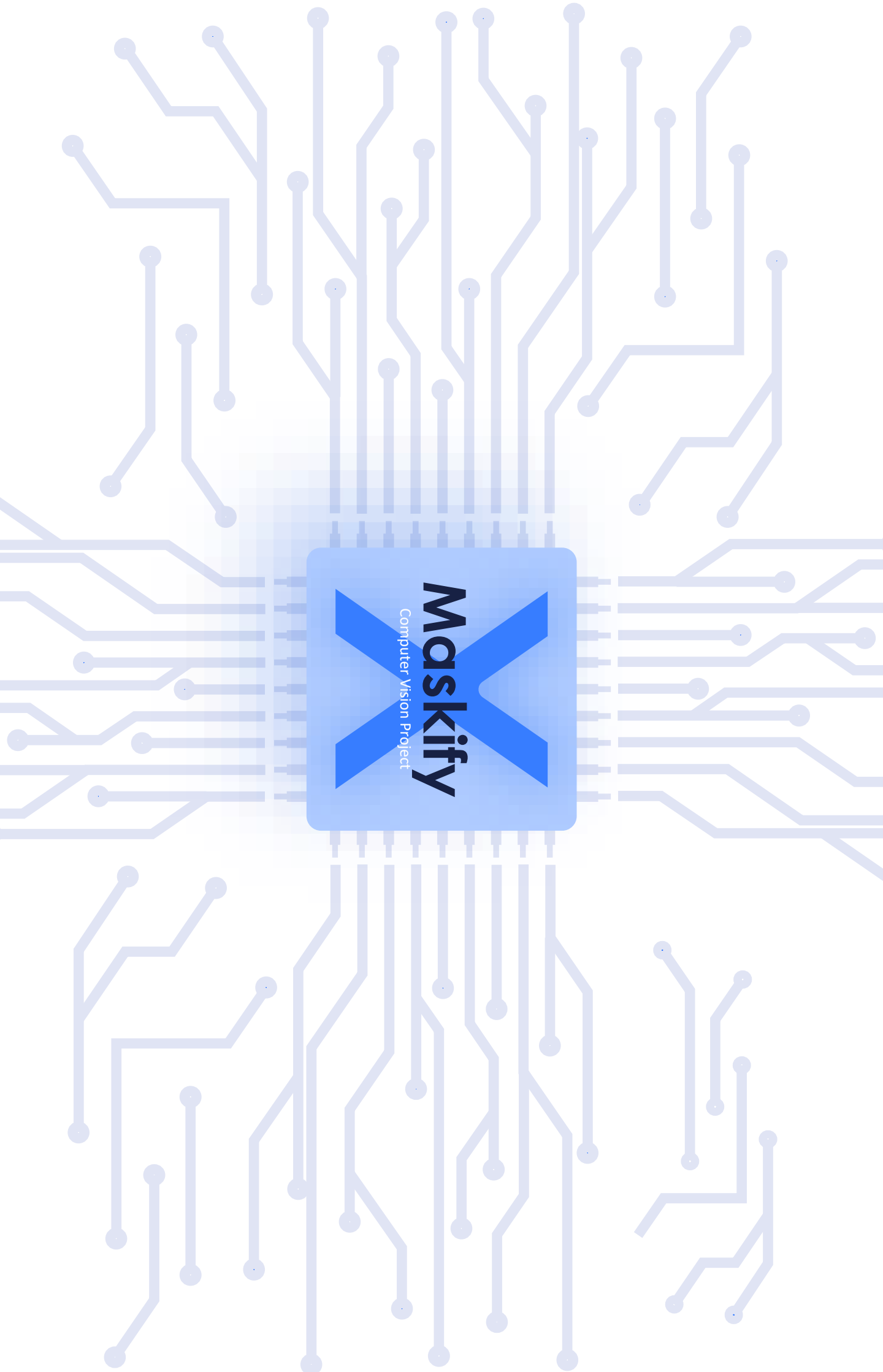

Maskify
Computer Vision Project

Our Team

UW-Bothell Masters of Computer Science and Software Engineering Students

Harpreet Kaur

- SQL 95%
- Machine Learning 90%
- C++ 85%

Saurav Jayakumar

- Machine Learning 95%
- Python 90%
- C++ 85%

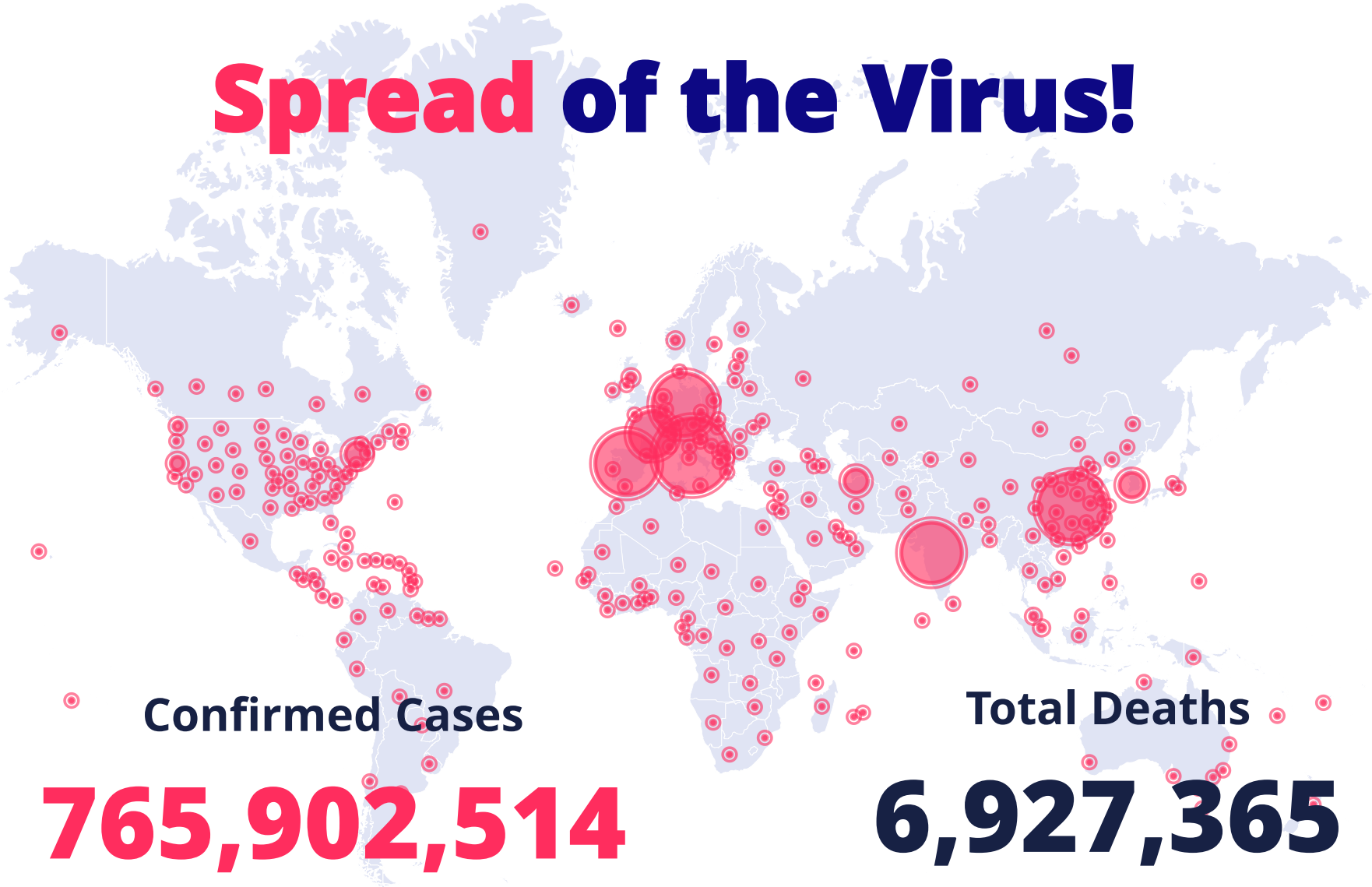
Utkarsh Darbari

- Python 95%
- PowerPoint 90%
- C++ 85%

Problem Statement

Wear a face mask.

Spread of the Virus!



Confirmed Cases

765,902,514

Total Deaths

6,927,365

Solution

Mask Detection System

Pre-Processing
Greyscaling, Histogram Equalization, Gaussian Filtering.

Face Detection
Haar Cascade Detection

Mask Detection
YCrCb + Otsu, Oronasal region selection, Comparison of skin area.

Preprocessing

Gaussian filtering
Image smoothening technique to reduce noise & detail of the image

Histogram equalization
Enhances the contrast by redistributing the intensity

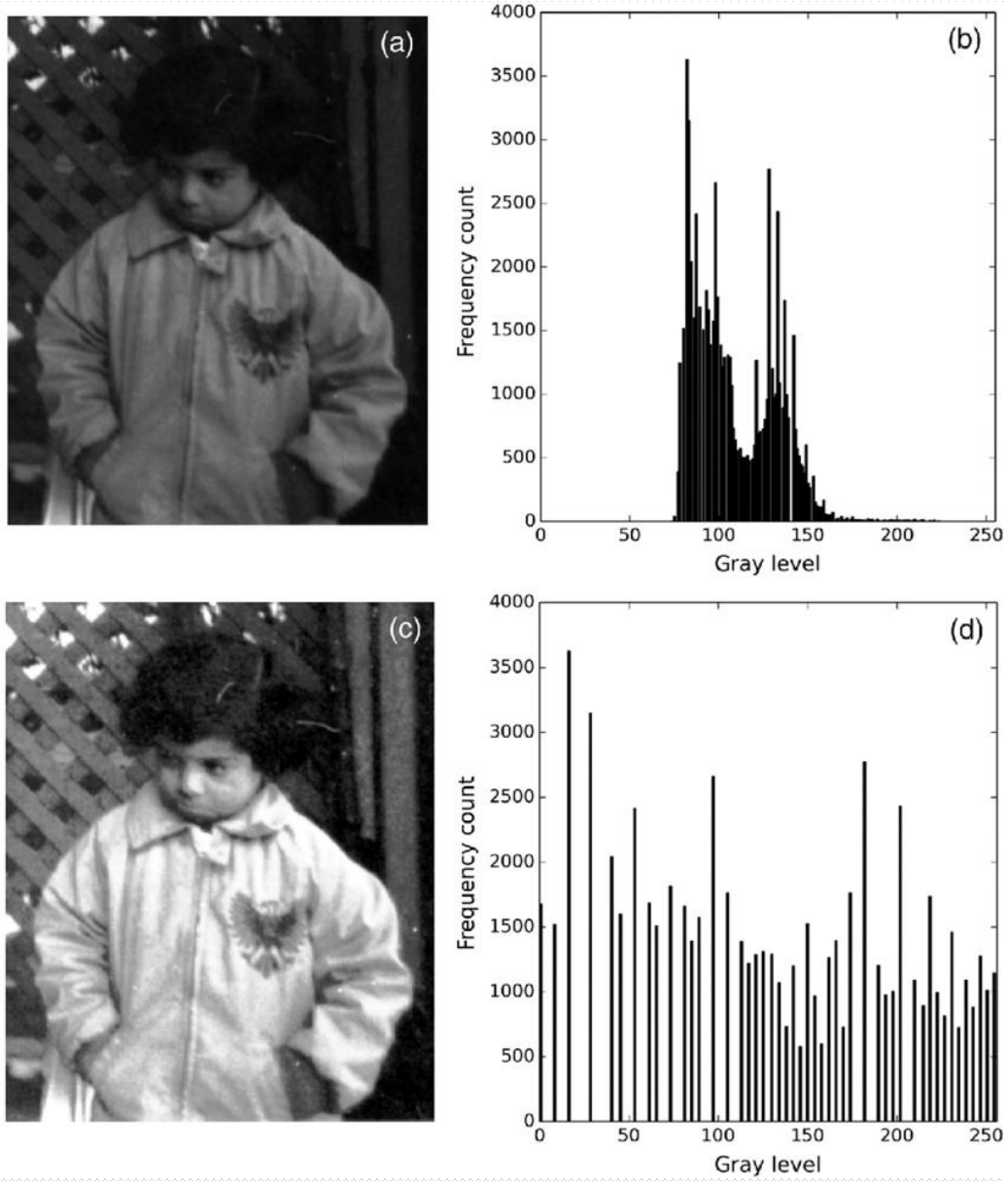
Grey-scale Image
Convert colored image to shades of grey

Preprocessing

Grey Scaling

Simplifies the image to a single channel of information, making it easier to detect features or patterns in the image.

It improves the performance of many computer vision tasks, such as edge detection, object detection, and image classification.



Preprocessing

Histogram Equalization

Improves the contrast by redistributing the intensities

Useful for removing shadows and other lighting effects from an image.

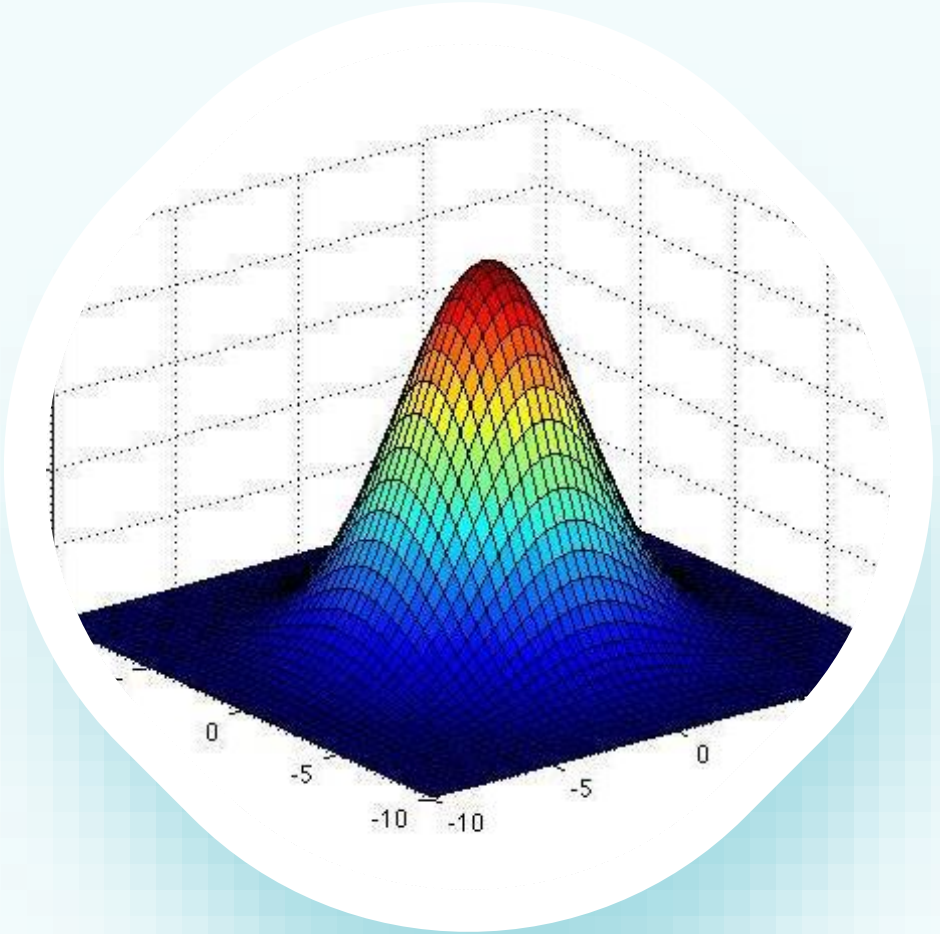
Preprocessing

Gaussian Filtering

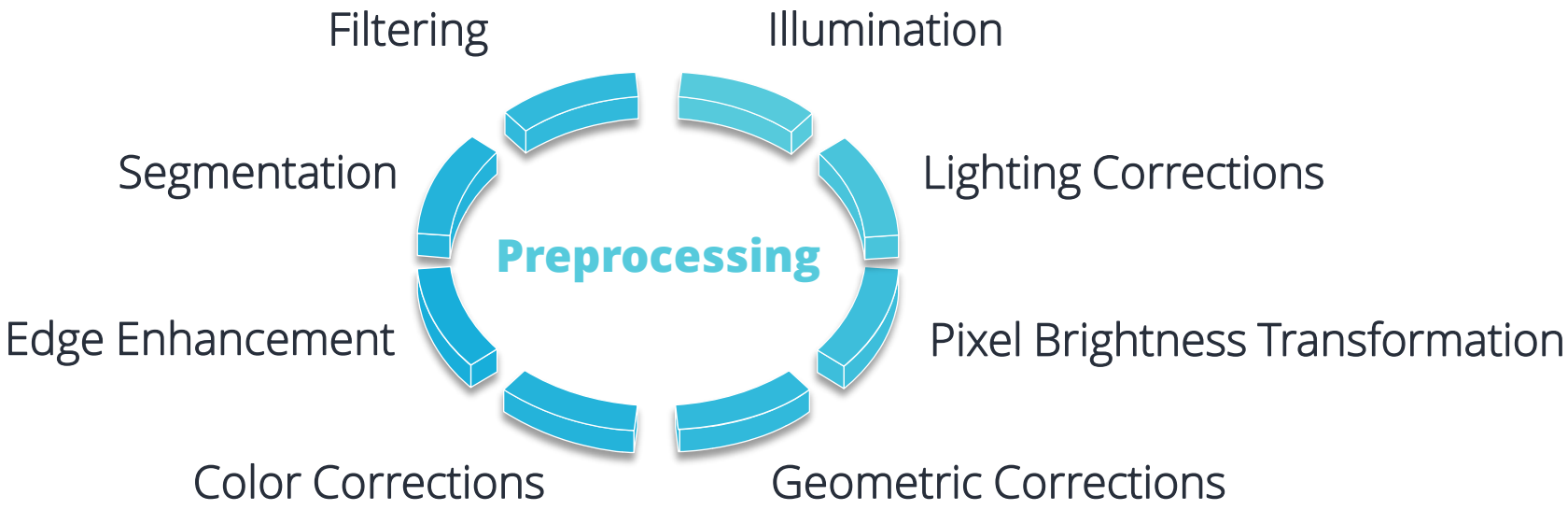
Used to reduce noise in an image while preserving the overall structure.

The standard deviation of the filter affects the amount of blurring

The size of the filter affects the amount of smoothing applied



Other Preprocessing Methods



Face Detection System

Haar Cascade Classifier

A Haar cascade classifier can be used to detect objects in images.

It creates a series of Haar features. (A mathematical expression that compares brightness of 2 regions of an image.)

It then trains a classifier to identify these features in images.

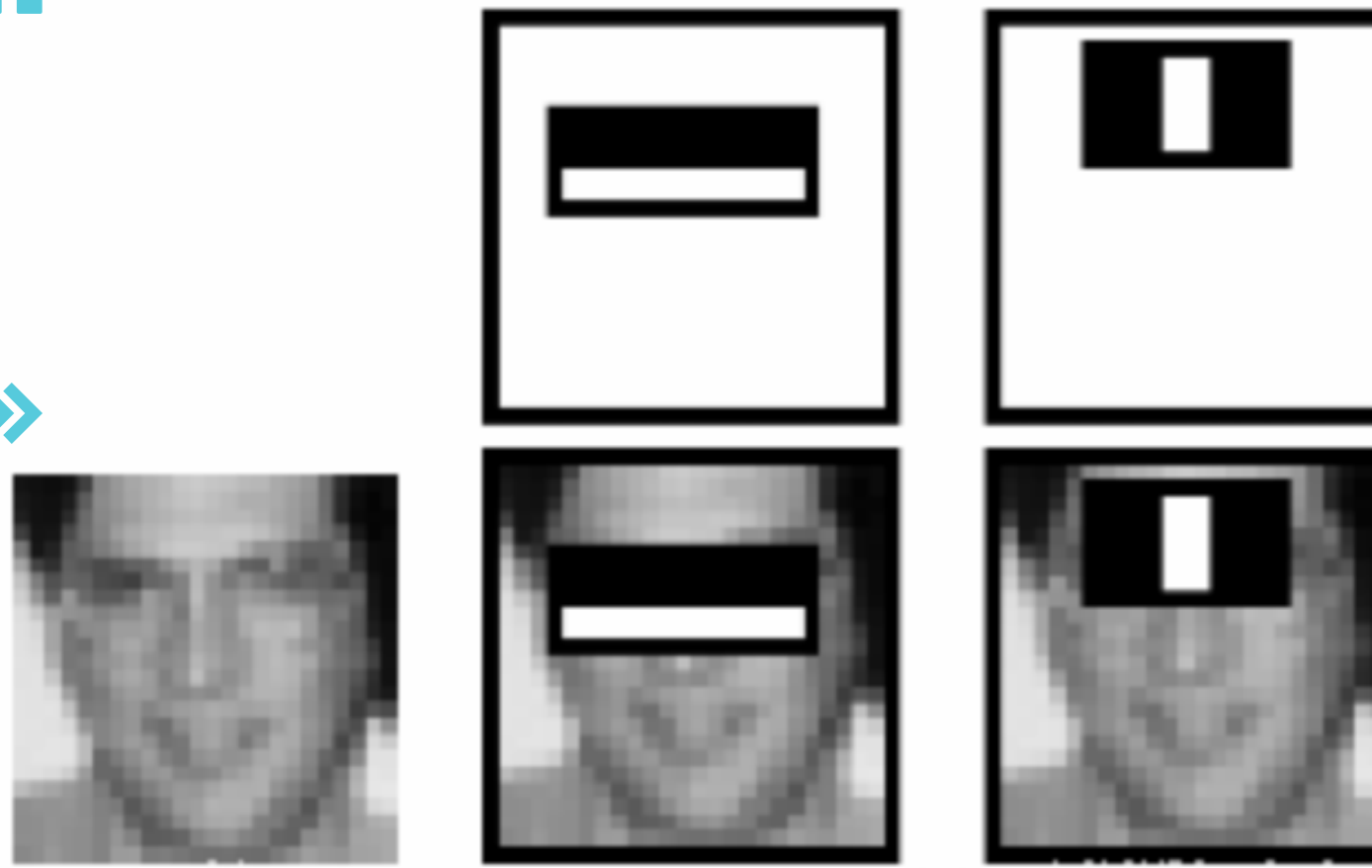
Once the classifier is trained, it can be used to detect faces in new images.

Face Detection

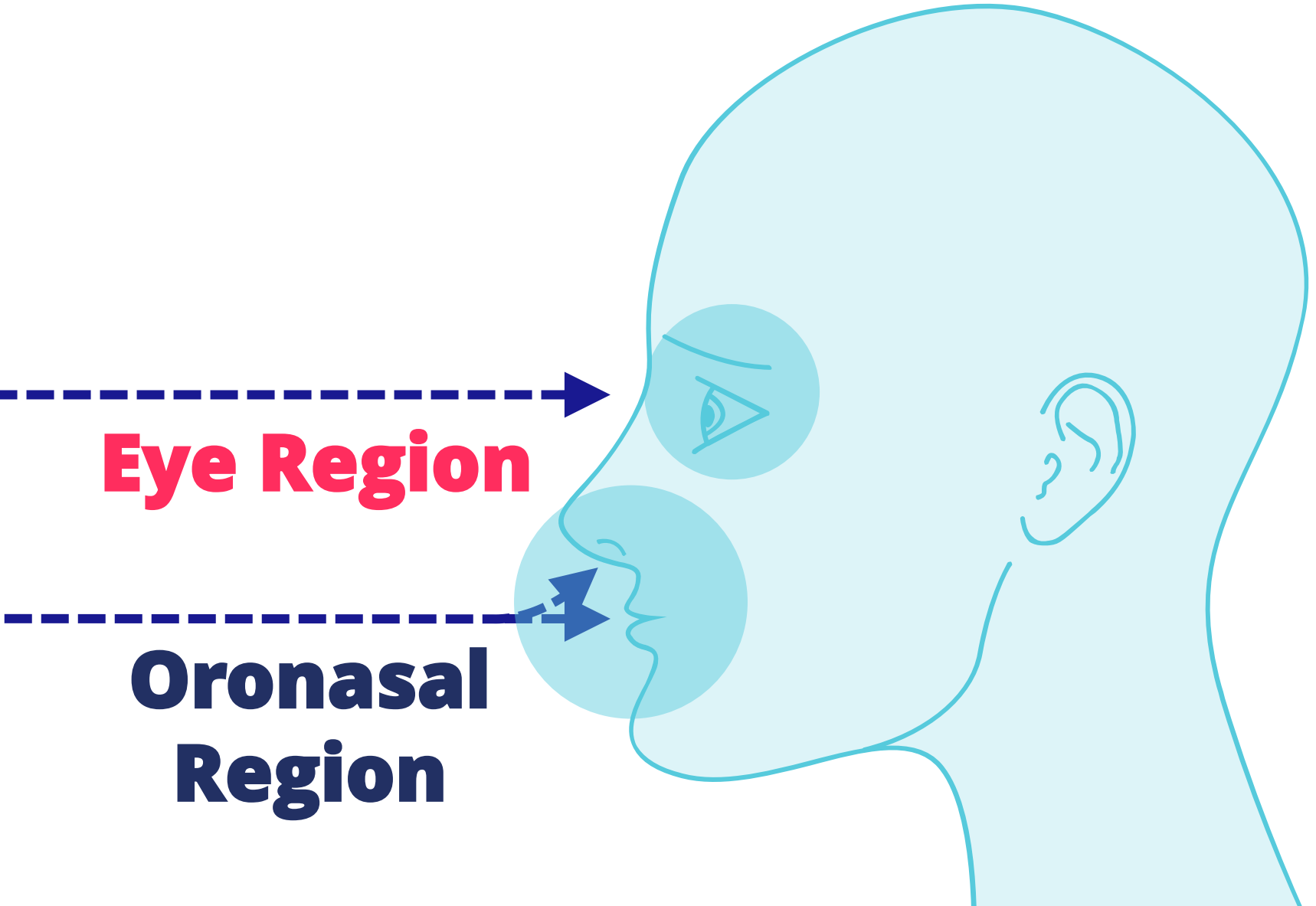
Haar Cascade Classifier Algorithm

Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

Methods used:
1. Haar Cascade Frontal Face Alt
2. Haar Cascade Eye Tree Eyeglasses



Mask Detection



1. YCrCb skin color model (YCrCb + Otsu)

2. Oronasal region selection

3. Comparison of skin area between eye region and oral and nasal region

Mask Detection Process

YCBCR

Y is the light intensity of the color, Cb and Cr is the blue component and red component related to the green component. These components are less sensitive
to the human eyes.

Otsu Thresholding

Algorithm returns a single intensity threshold that separate pixels into two classes, foreground and bac kground

Oronasal Area

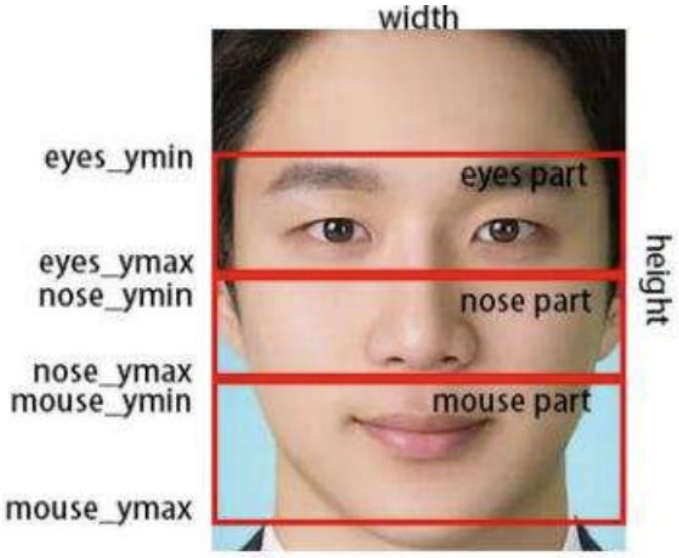
Area of the nose and mouth, calculated using area around eyes

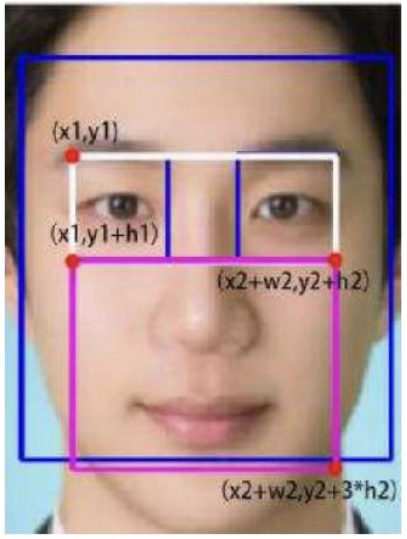
Comparison of skin area

if skin area around eye is 1.2 times more than skin area around nose and mouth then mask, else no mask

Example of Mask Detection

CR + Otsu





Oronasal Region Selection & Comparison of Skin Area.

Maskify X

Dataset

[Face Mask Detection Dataset | Kaggle]

contains 7553 images with 3 color channels (RGB). Images of faces with mask are 3725 and images of faces without mask are 3828.

Data Structures

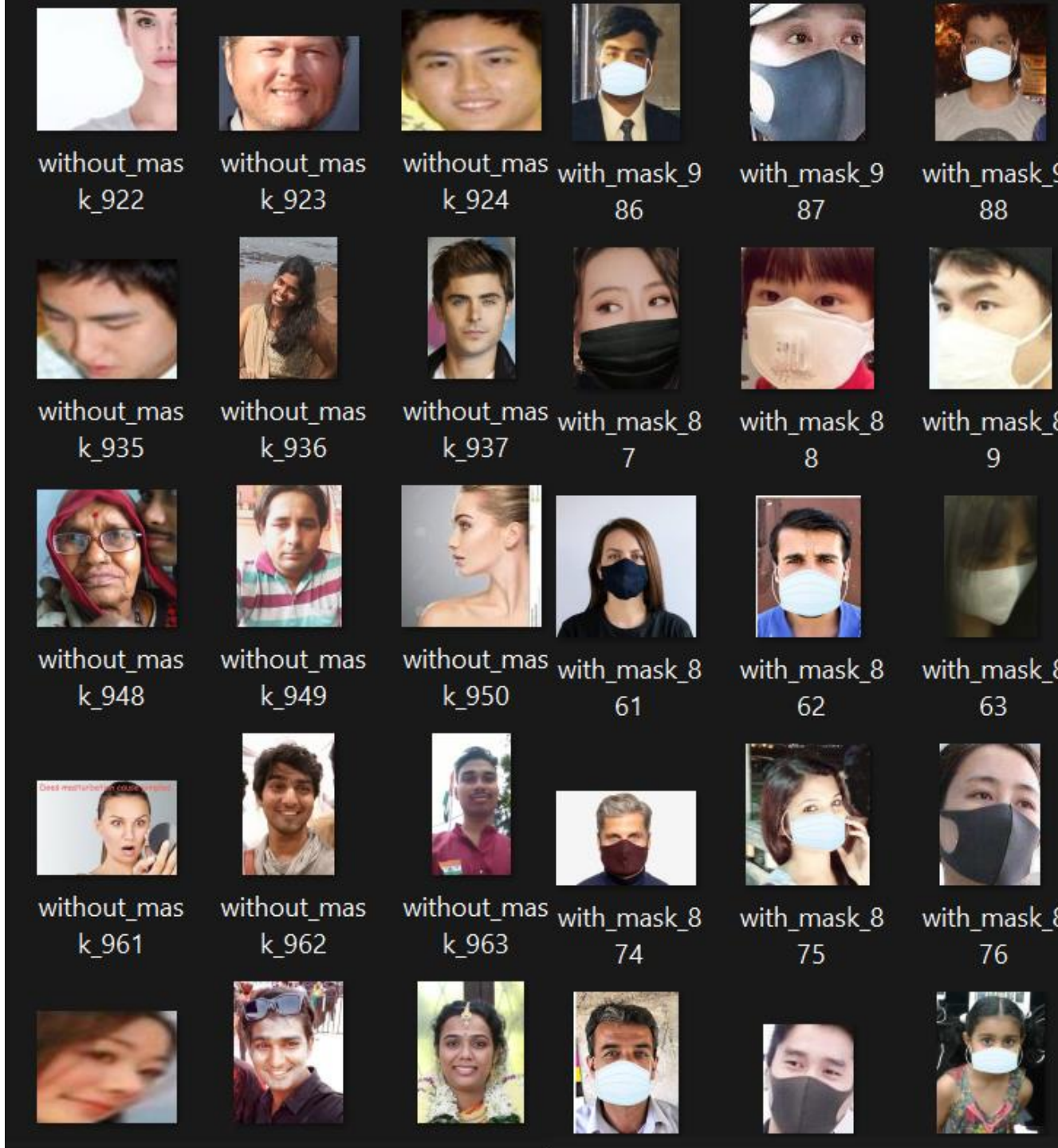
Map, Vector of Matrices, Vector of Rectangles

Classes

objDetect

Testing Methodology

Precision, recall, and accuracy for mask detection



Watch Out...

Mask Detection

Demo Time !!!

NEXT STEPS


Questions?