Computer Vision (COL 780)
Assignment-3: Object Detection and Recognition

Pulkit Singal (2023AIB2064)

Tools and Technologies Used

Libraries used:

1. NumPy: For matrix algebra
2. Pandas: For writing prediction results to an excel sheet
3. Matplotlib: For creating and saving plots and histograms
4. OpenCV: For image reading, saving, resizing, etc.
5. os: For working with files and directories
6. MediaPipe: For hand detection in images
7. Scikit-Learn: For creating, training and evaluating SVM model
8. Joblib: For saving and loading SVM model
9. ast, sys: For taking command-line input arguments
10. pygame: For controlling (play/pause) music

Computer Vision Concepts used:

1. Convolution
2. Image Derivatives
3. Image Gradients (Magnitude and Direction)
4. Object Detection using HOGs

Machine Learning Concepts used:

1. Classifier: Support Vector Machines (SVMs)
2. Evaluation Metrics: Accuracy, Precision, Recall, F1 Score, Confusion Matrix
3. True Positive Rate, False Positive Rate, ROC-AUC

Technical Requirements:

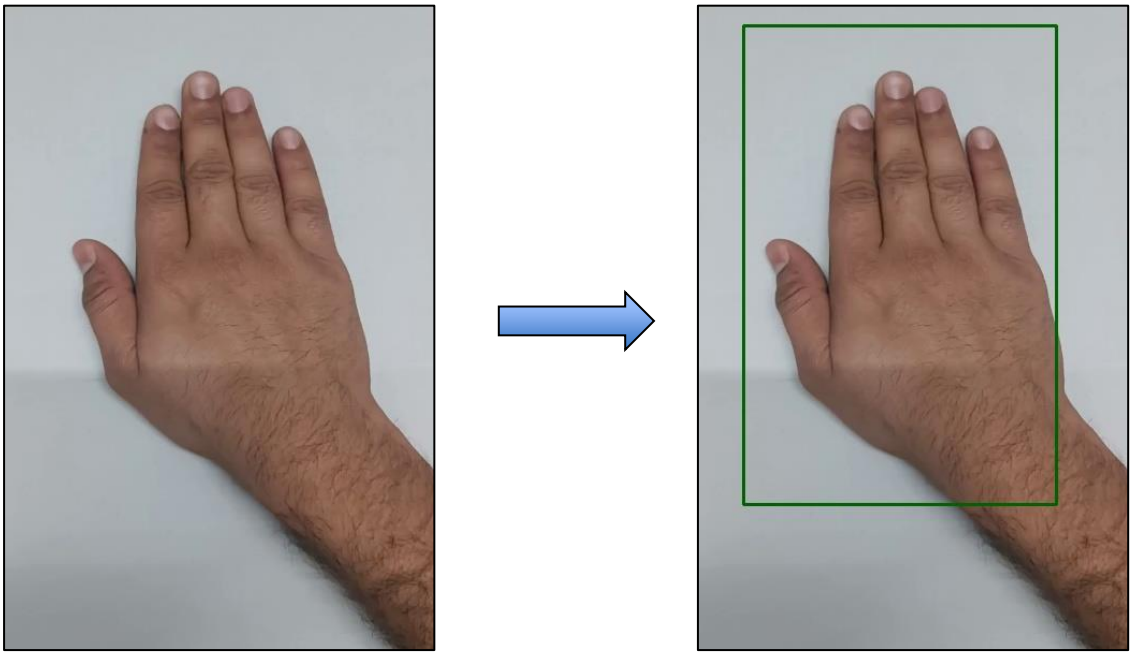
Laptop/Computer with

1. Python
2. Good RAM
3. Webcam (For Bonus Task)

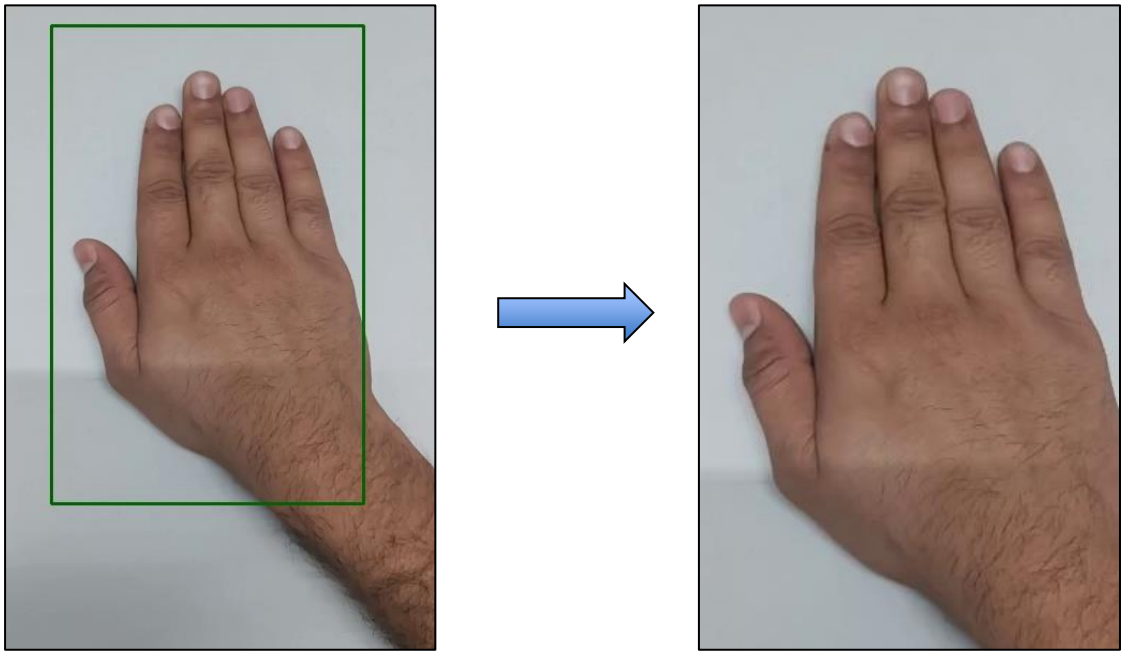
Brief Algorithm of the Code, Results and Conclusions

Training of the Model:

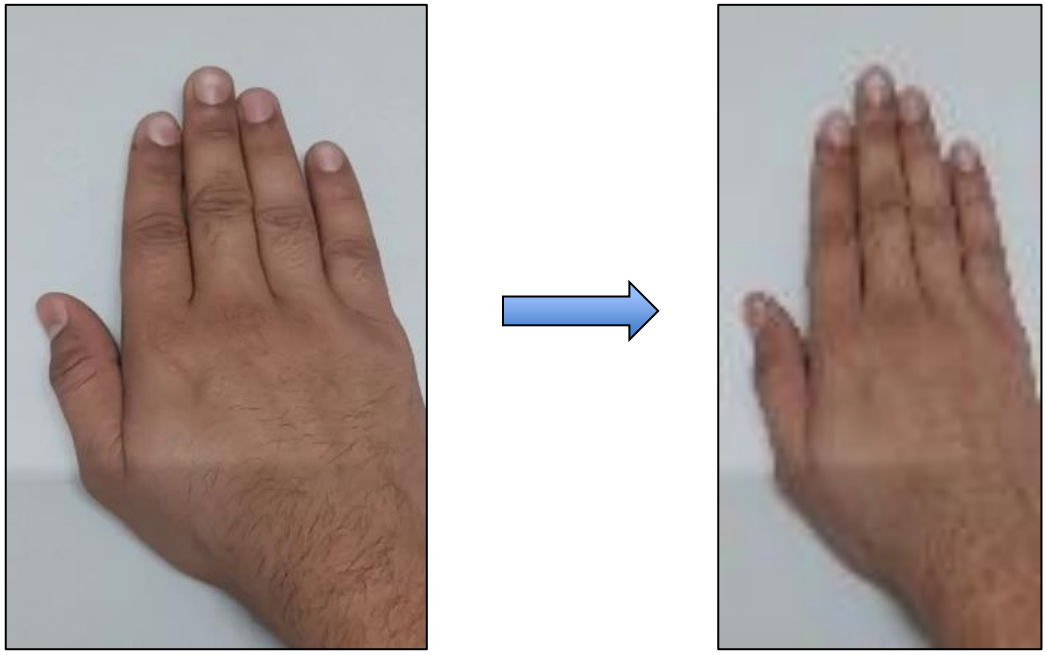
1. Detection of hand region (for each hand) from the image of training dataset using the libraries and logic given in ROI_Coordinates.py python code file.



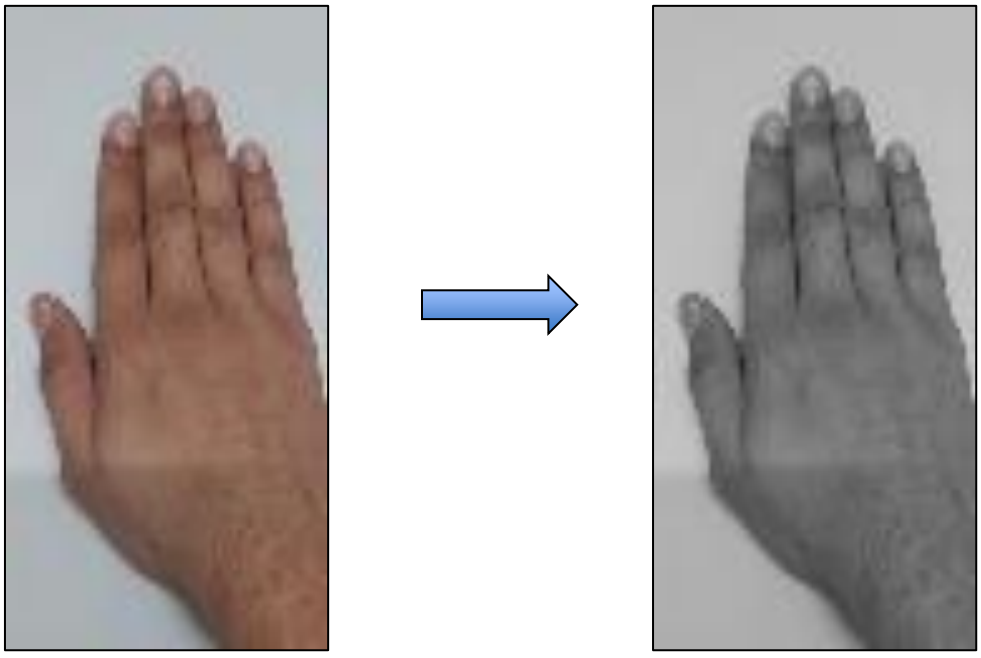
2. Cropping of the hand region from the image.



3. Resizing of the cropped image of the hand to 128 pixels (height) X 64 pixels (width) standard size.



4. Conversion of the cropped image of hand from RGB to grayscale image patch.



5. Conversion of the grayscale image patch into Histogram of Oriented Gradients (HOGs) using the below procedure:

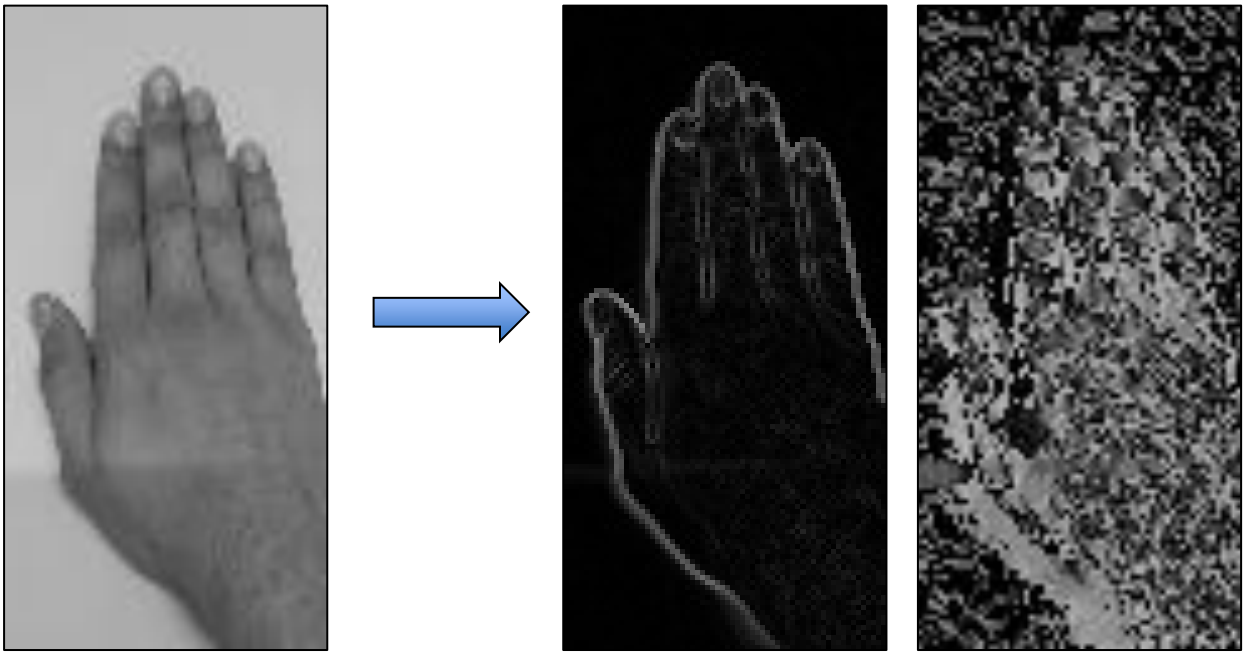
a) Finding the gradient magnitude and orientation of the grayscale image patch by convolution with the following kernels:

For Gradient in X-Direction (Grad_X): $[1 \quad 0 \quad -1]$

For Gradient in Y-Direction (Grad_Y): $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$

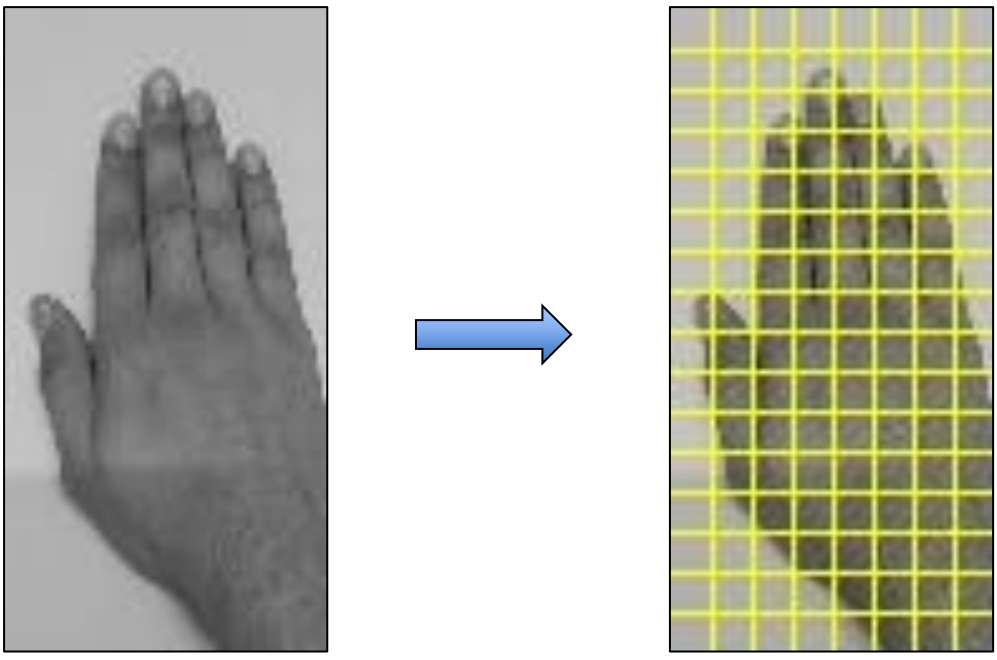
Gradient Magnitude = $\sqrt{\text{Grad\_X}^2 + \text{Grad\_Y}^2}$

Gradient Orientation = $\tan^{-1}\left(\frac{\text{Grad\_Y}}{\text{Grad\_X}}\right)$

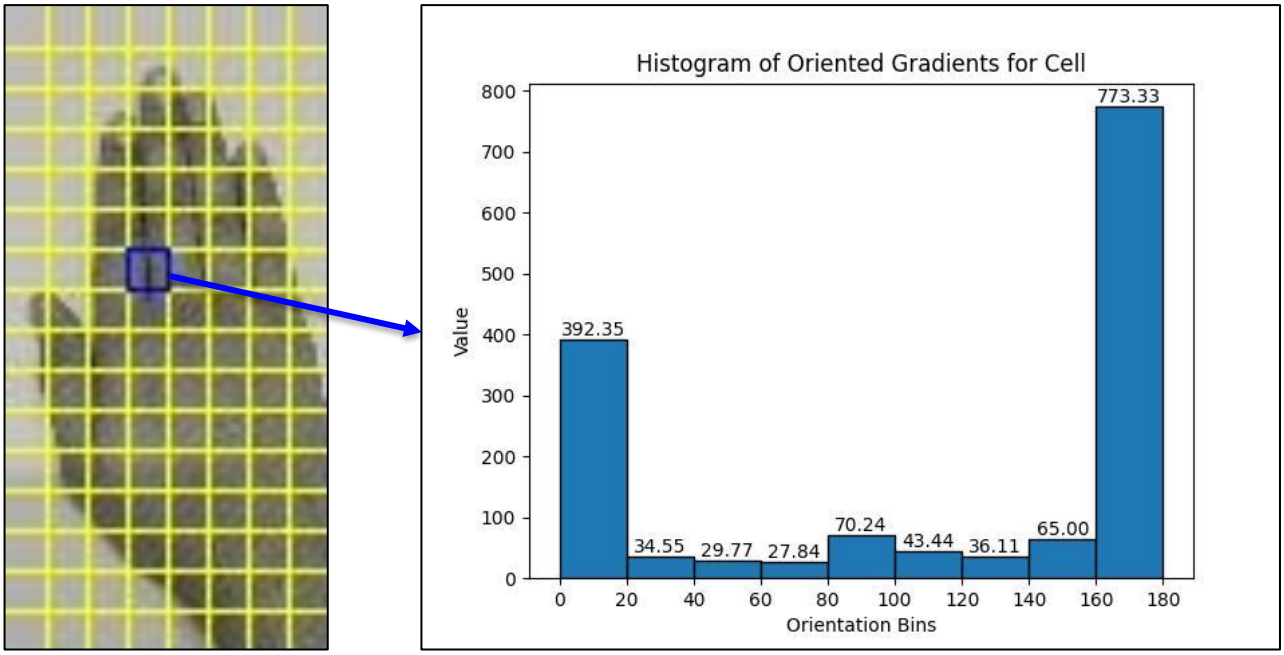


Gradient Magnitude Gradient Orientation

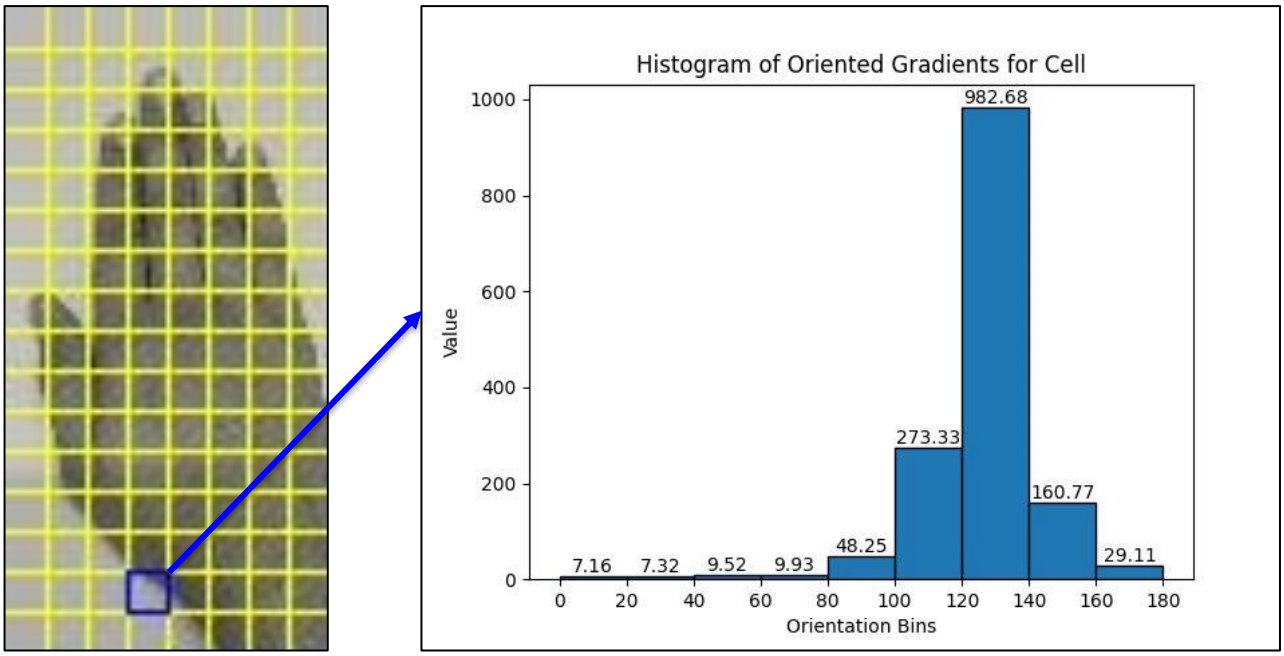
b) Dividing the grayscale image patch into cells of size 8 pixels X 8 pixels.



c) Quantizing the unsigned gradient orientations (0°- 180°) into 9 bins of 20° each (0°- 20°, 20°- 40°, …. 160°- 180°). For each cell (of size 8 pixels X 8 pixels), the bin corresponding to a pixel gradient orientation is voted by the corresponding pixel gradient magnitude, which is interpolated bi-linearly between the neighboring bin centers. This gives a feature vector of dimensions 9 X 1 for each cell.

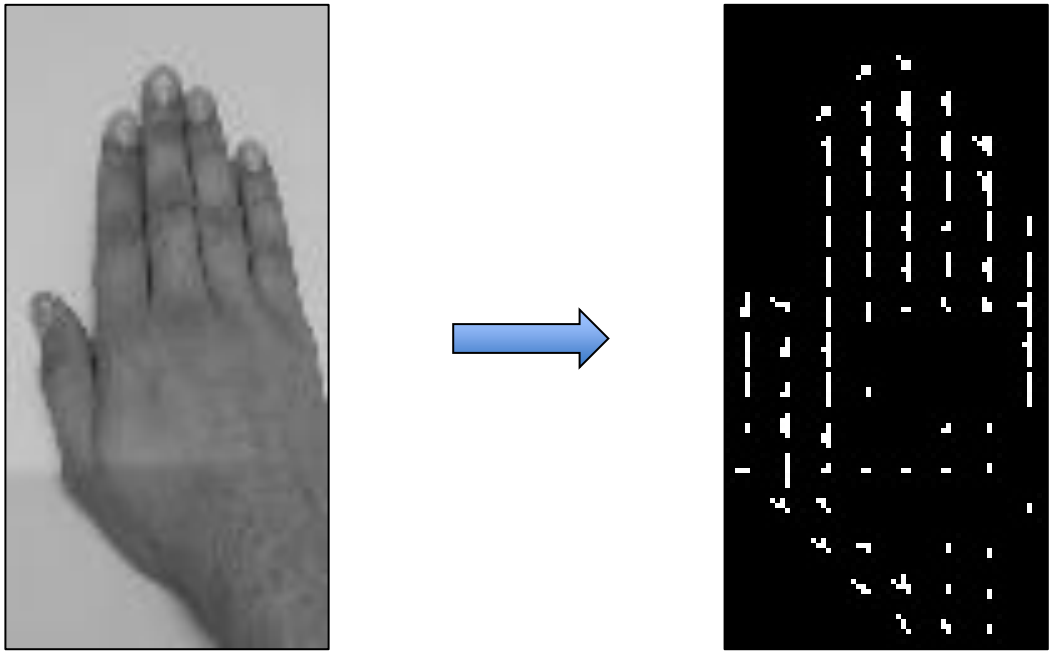


Cell histogram for an edge at ~ 90° (hence gradient orientation ~ 0° or 180°)
Cell HOG Vector: [392.35, 34.55, 29.77, 27.84, 70.24, 43.44, 36.11, 65.0, 773.33]

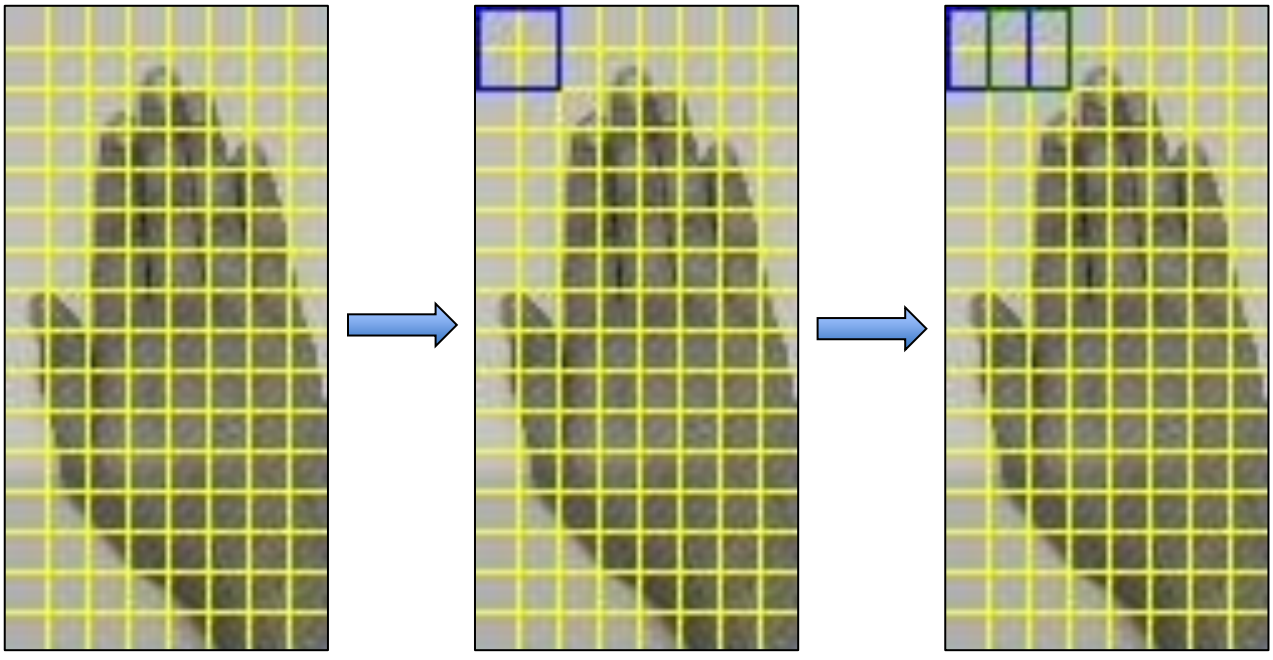


Cell histogram for an edge at ~ 45° (hence gradient orientation ~ 135°)
Cell HOG Vector: [7.16, 7.32, 9.52, 9.93, 48.25, 273.33, 982.68, 160.77, 29.11]

d) Creating and displaying the HOG Image for the grayscale image patch, thus showing the edge orientations for each cell.



e) Dividing the grayscale image patch into blocks of size 16 pixels X 16 pixels of 50% overlap (7 X 15 = 105 blocks in total). Each block consists of 2 X 2 cells. The feature vectors of all four cells in each block are concatenated and block normalized (L_2 Norm) to get the feature vector of dimensions 36 X 1 for each block.

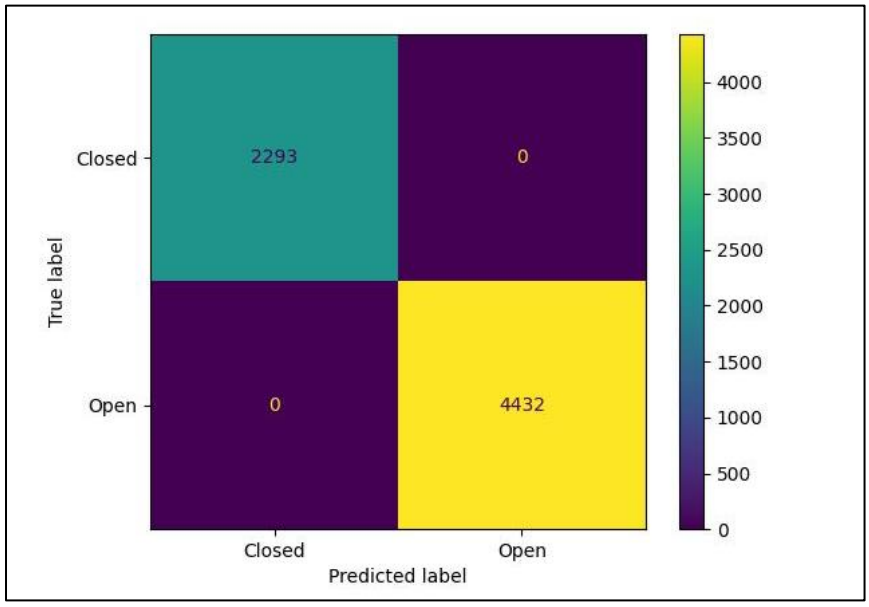


f) Concatenating the block normalized feature vectors of all 105 blocks to get the 1-D feature vector of dimensions 105 X 36 = 3780 for the grayscale image patch.

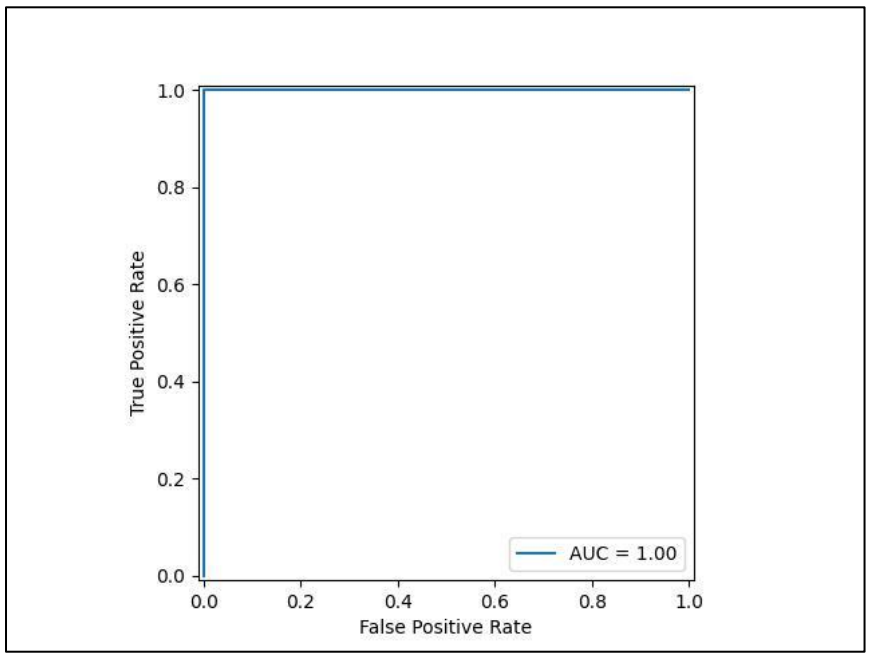
6. Training and saving of SVM model using the obtained feature vectors and given labels (Closed/Open)

7. Evaluation of the trained SVM model on Training Dataset:
   Accuracy: 1.0

Class	Precision	Recall / True Positive Rate (TPR)	False Positive Rate (FPR)	F1-Score	ROC-AUC
Closed	1.0	1.0	0.0	1.0	1.0
Open	1.0	1.0	0.0	1.0	1.0



Confusion Matrix for Training Dataset



ROC Curve for Training Dataset

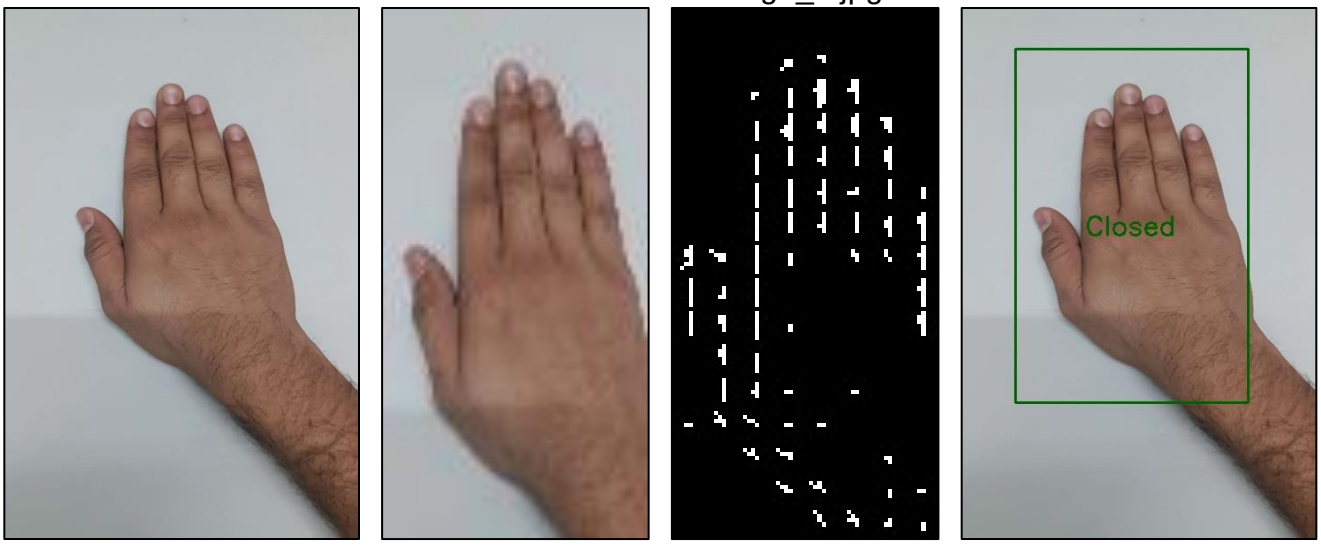
Validation of the Model:

1. Loading of the pre-saved SVM Model.

2. Creation of HOG feature vectors for each hand in the image of the validation dataset using the same procedure as done during training.

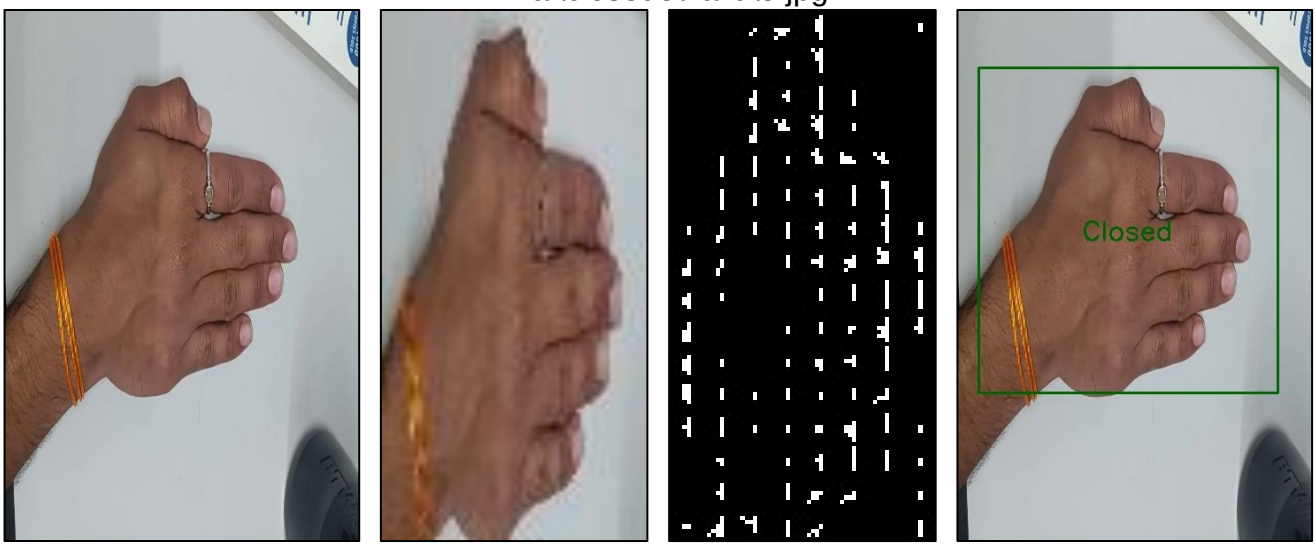
3. Prediction of the class (Closed/Open) using the SVM Model and saving of the results in an excel sheet.

Final\closed1\valid\image_6.jpg

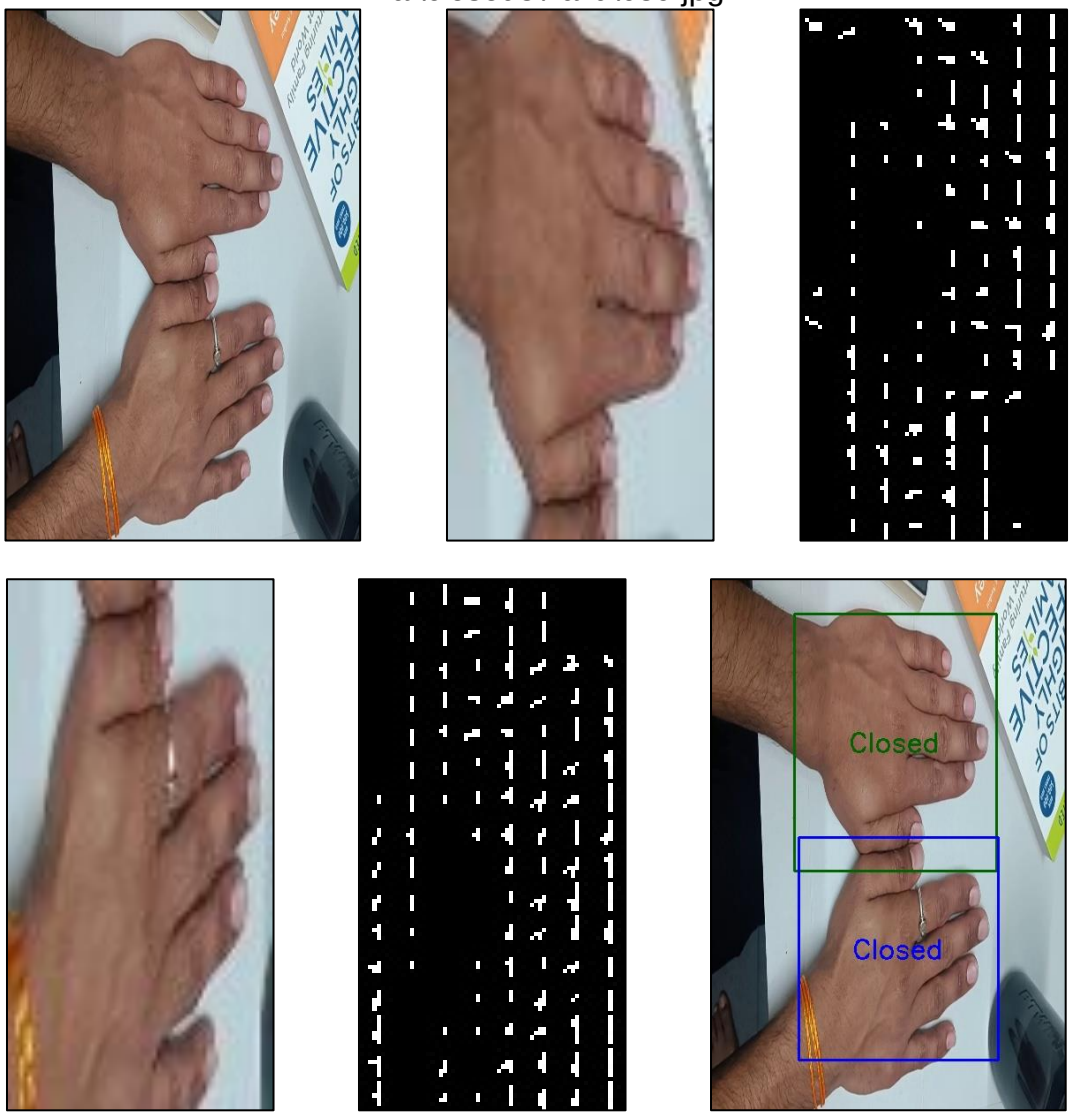


Final\closed2\valid\image_17.jpg

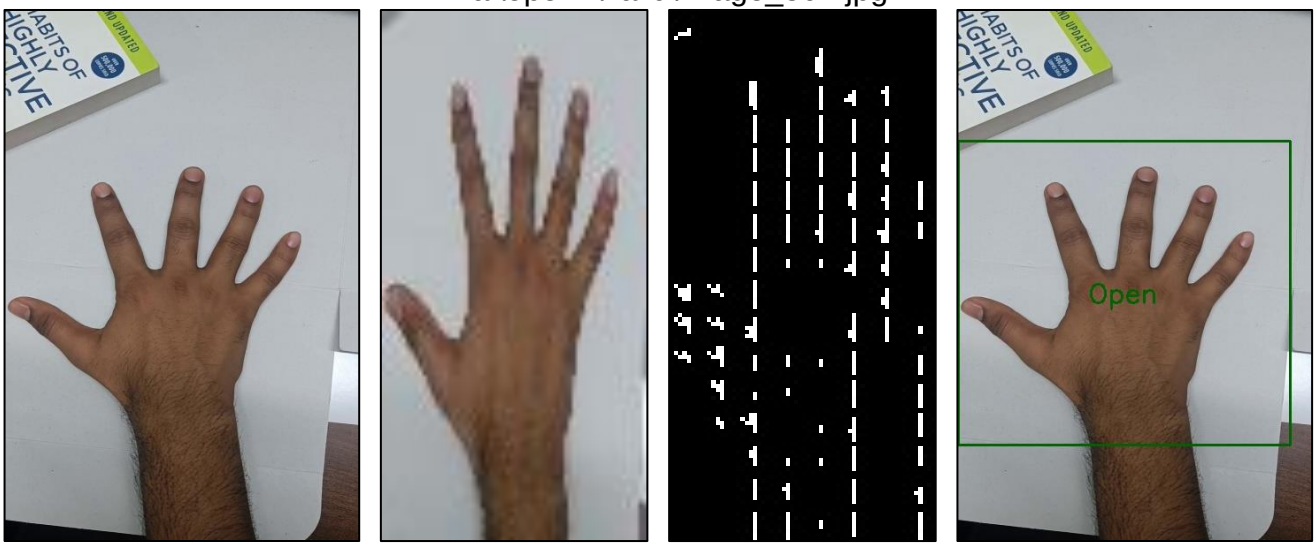
Final\closed3\valid\5.jpg



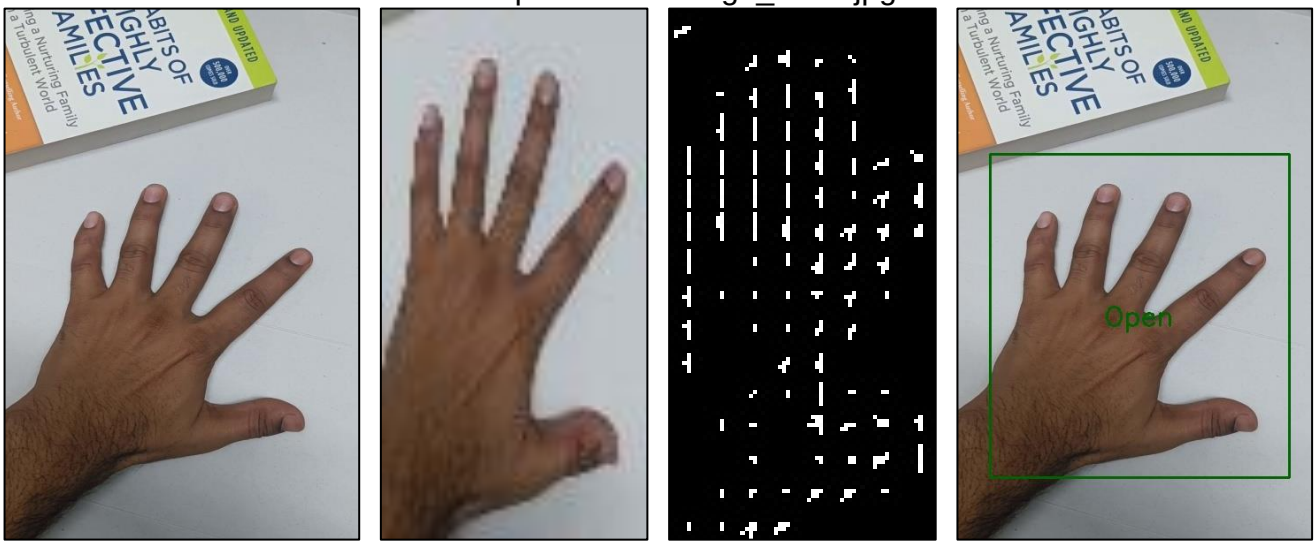
Final\closed3\valid\689.jpg



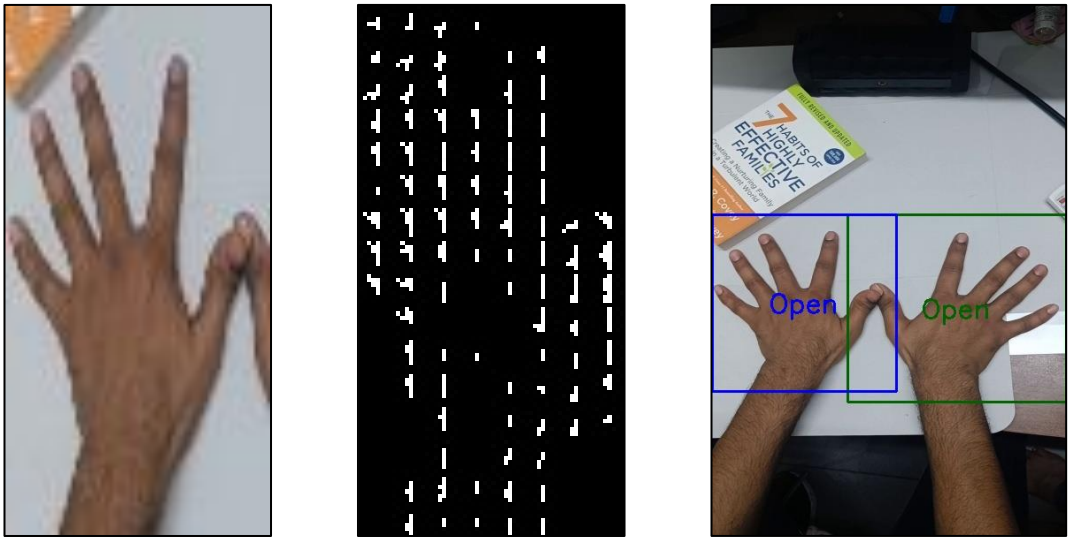
Final\open1\valid\image_502.jpg



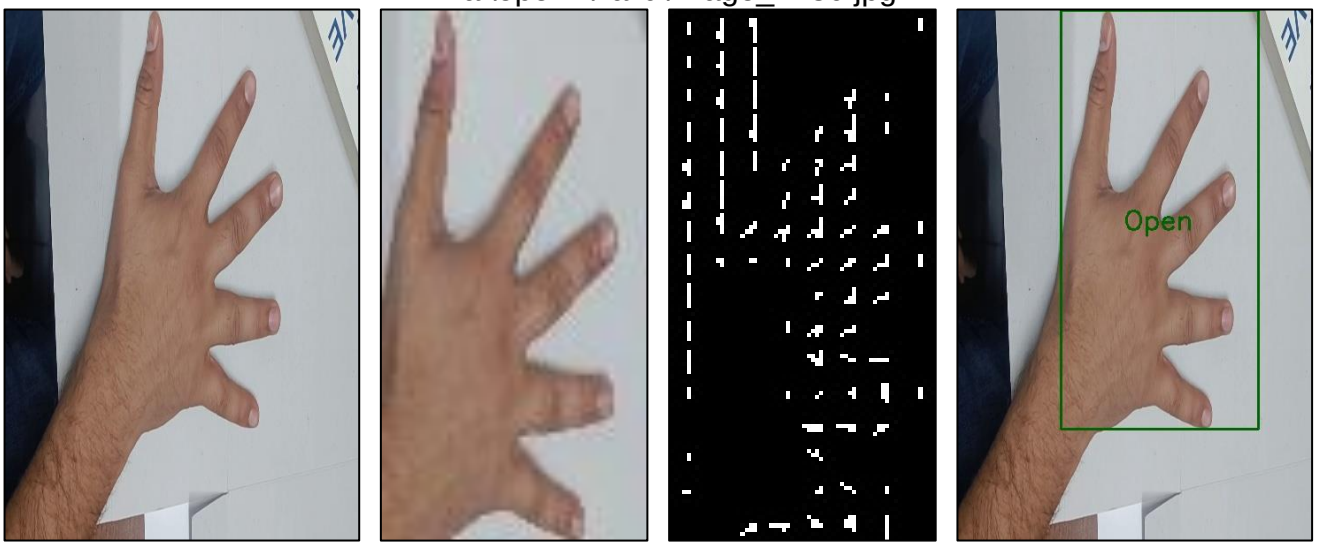
Final\open1\valid\image_1619.jpg



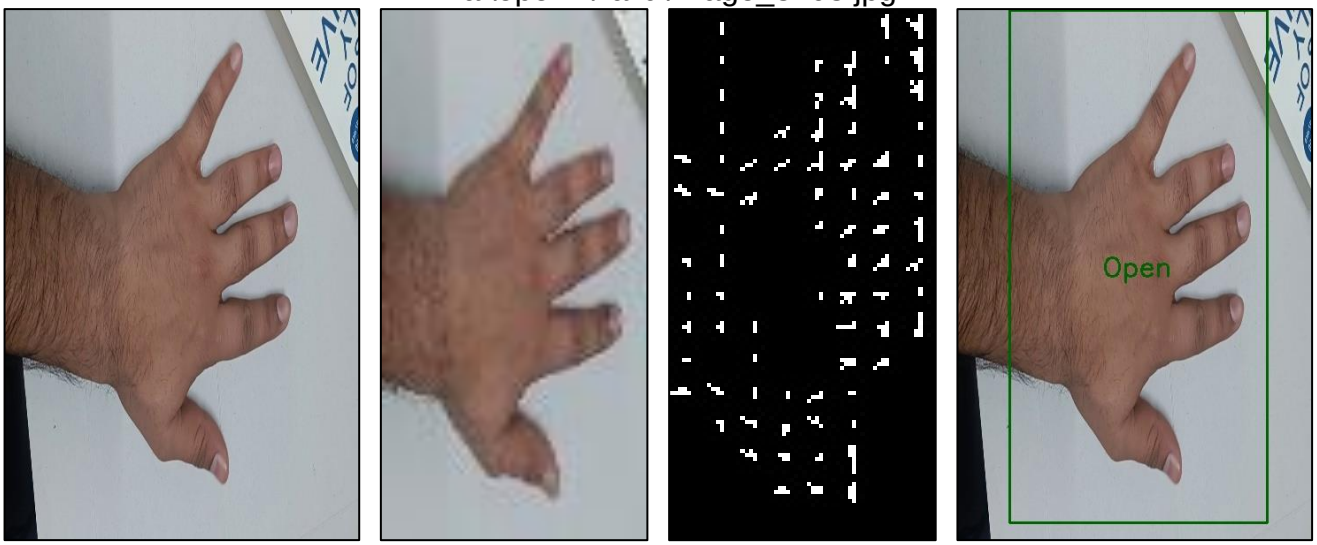
Final\open1\valid\image_2163.jpg



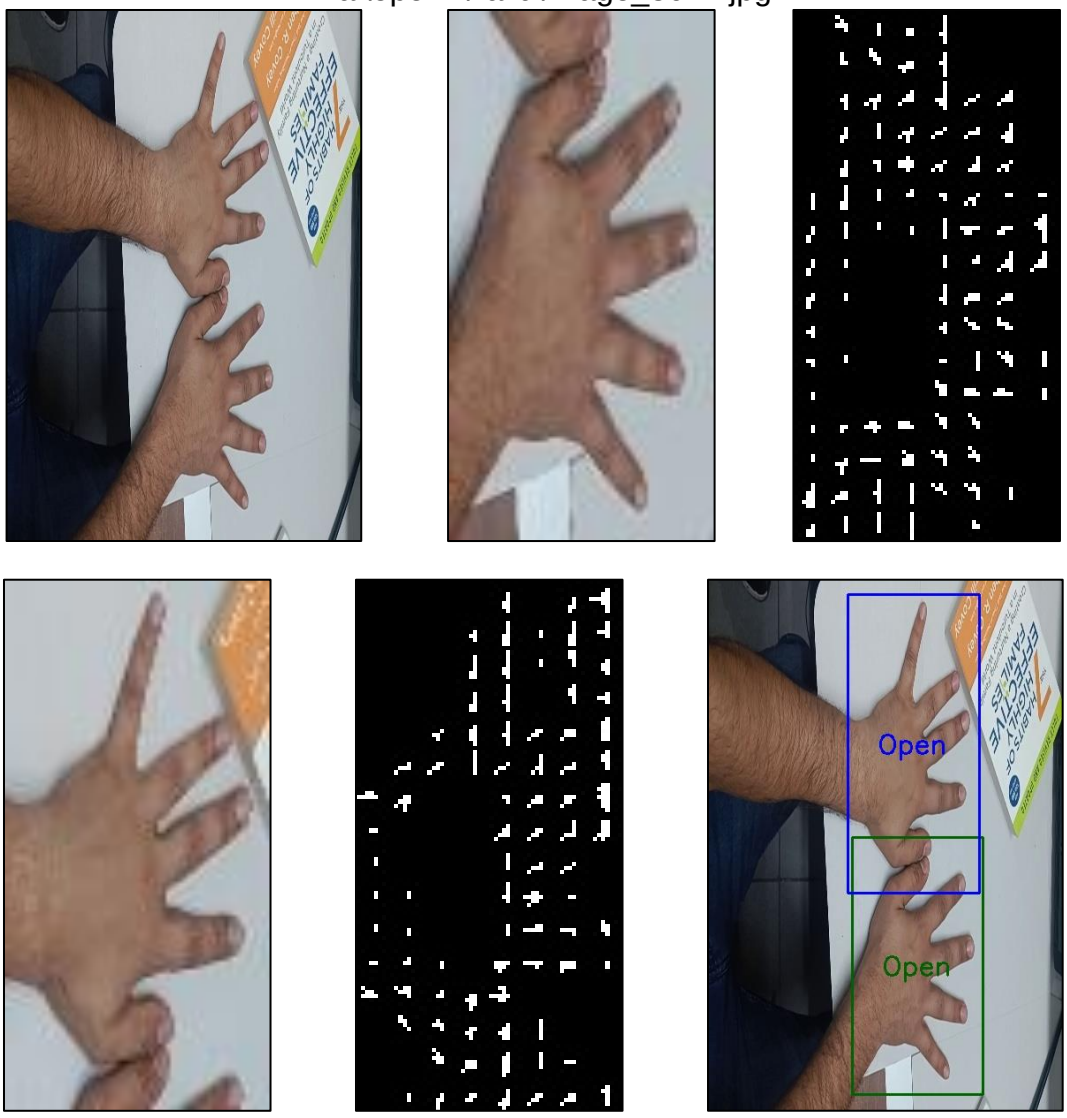
Final\open2\valid\image_2130.jpg



Final\open2\valid\image_3403.jpg



Final\open2\valid\image_3624.jpg

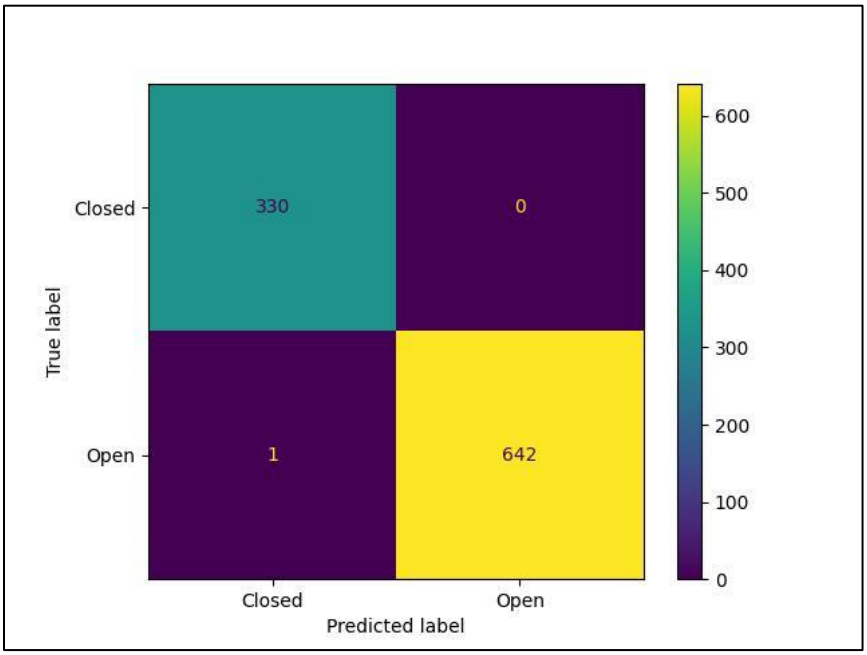


Directory	Image Name	Hand No.	Prediction (Open/Closed)
Final/closed1/valid	image_6.jpg	1	Closed
Final/closed2/valid	image_17.jpg	1	Closed
Final/closed3/valid	5.jpg	1	Closed
Final/closed3/valid	689.jpg	1	Closed
Final/closed3/valid	689.jpg	2	Closed
Final/open1/valid	image_1619.jpg	1	Open
Final/open1/valid	image_2163.jpg	1	Open
Final/open1/valid	image_2163.jpg	2	Open
Final/open1/valid	image_502.jpg	1	Open
Final/open2/valid	2130.jpg	1	Open
Final/open2/valid	3403.jpg	1	Open
Final/open2/valid	3624.jpg	1	Open
Final/open2/valid	3624.jpg	2	Open

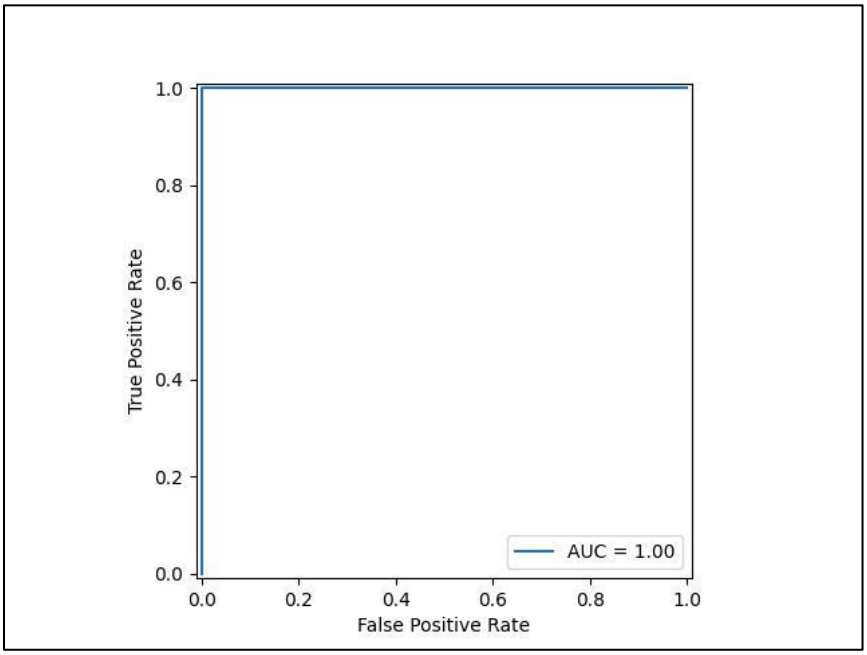
4. Evaluation of the trained SVM model on Validation Dataset:

Accuracy: 0.999

Class	Precision	Recall / True Positive Rate (TPR)	False Positive Rate (FPR)	F1-Score	ROC-AUC
Closed	0.997	1.000	0.002	0.998	1.0
Open	1.000	0.998	0.000	0.999	1.0



Confusion Matrix for Validation Dataset



ROC Curve for Validation Dataset

Conclusions from the Evaluation Results:

1. The model has very high accuracy (99.9%), suggesting it performs exceptionally well on the validation data as well and hence did not overfit the training dataset.
2. The values of precision, recall, F1-score, and ROC-AUC for both classes indicate robust performance in classifying both "Closed" and "Open" instances.
3. The false positive rates are extremely low, indicating minimal misclassification errors.
4. The high ROC-AUC score suggests the model's ability to distinguish between the classes is optimal.

Bonus Task: Controlling Music Player using Hand Gestures

Technical Requirements:

Laptop/Computer with

5. Python
6. Good RAM
7. Webcam

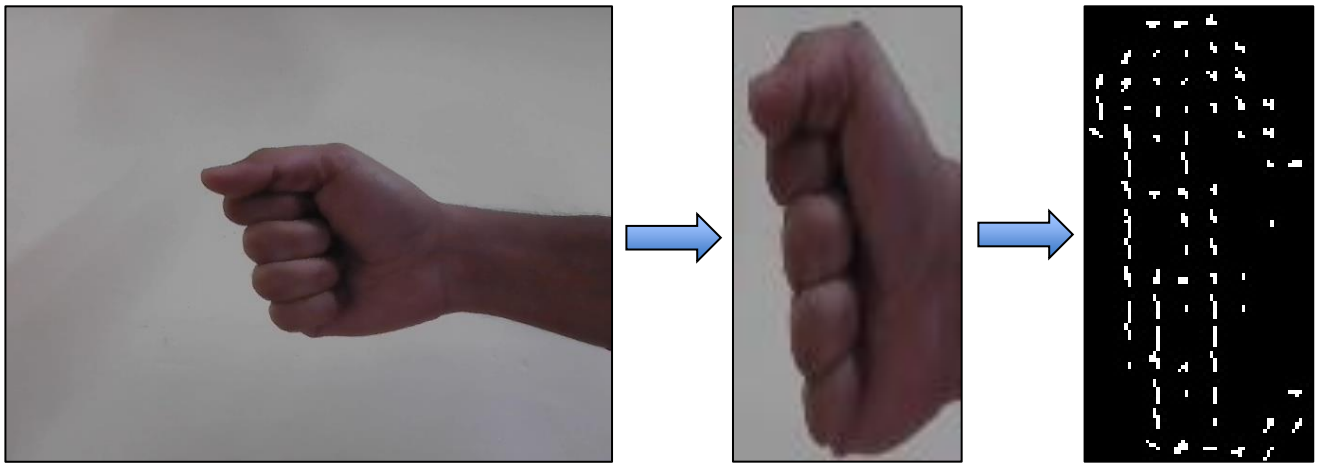
Algorithm:

1. Creation of training dataset consisting of 100 images of Closed Hand and 100 images of Open Hand, clicked using the camera of laptop.

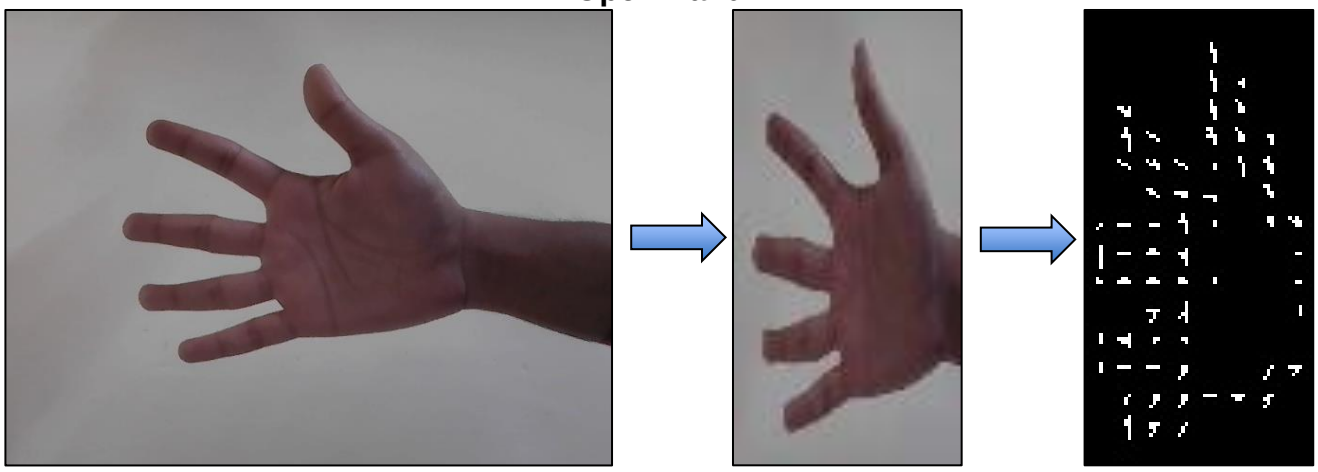
2. Creation of HOG feature vectors for each hand in the image of the training dataset.

3. Training of the SVM Model on training dataset.

Closed Hand (Fist)

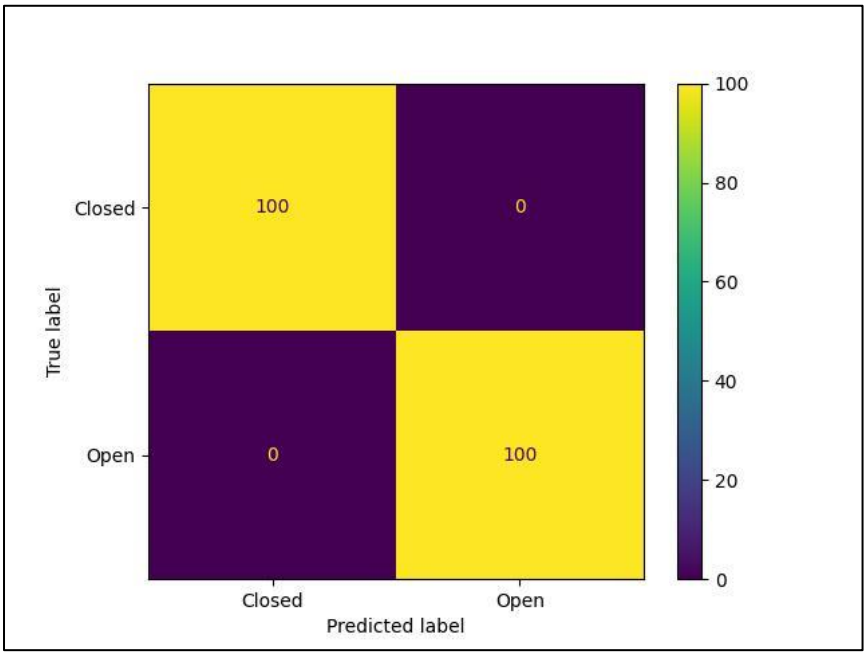


Open Hand

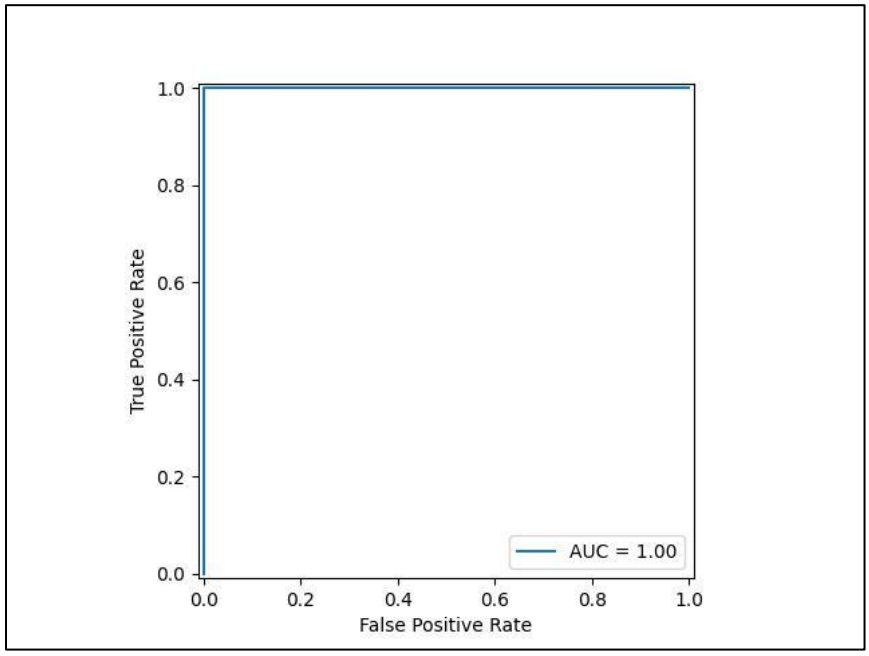


4. Evaluation of the trained SVM model on Training Dataset (created):
   Accuracy: 1.0

Class	Precision	Recall / True Positive Rate (TPR)	False Positive Rate (FPR)	F1-Score	ROC-AUC
Closed	1.0	1.0	0.0	1.0	1.0
Open	1.0	1.0	0.0	1.0	1.0



Confusion Matrix for Training Dataset (created)



ROC Curve for Training Dataset (created)

5. Continuous clicking of pictures of hand using laptop camera, converting it into HOG feature vector, use of SVM model to predict the class (Closed/Open), and use of the class label to control music playing in laptop (Closed = Pause, Open = Play).

Potential real-world applications of real-time hand detection and gesture recognition:

1. Gesture-controlled music players
2. Accessibility (For individuals with disabilities or mobility impairments)
3. Virtual reality (VR) and augmented reality (AR) applications
4. Interactive classroom teaching, advertisements, and marketing campaigns
5. Entertainment and games like musical chairs, etc.

Problems, Issues and Challenges Faced

1. **Correction in Bounding Box of Hand returned by the ROI code**: The original ROI code given for hand detection returned negative coordinates. The code was corrected to return coordinates that were within the width and height of the image.

2. **Visualization of the HOG Image:** Plotted the HOG vector representation for each cell as white lines on a black image. The length of each line was proportional to the bin value, and each line was plotted at an angle orthogonal to the bin-centered angle value (to represent edges).

3. **Improvement in Bonus Task:** Instead of using a pre-trained SVM model for controlling (Pause/Play) music on a laptop, added code for live training of the SVM model on two different hand gestures (Closed/Open) for robust and efficient control of music.

References

1. Slides on Conventional Object Detection by Prof. Chetan Arora, IIT Delhi
2. Videos on Classical CV - HOG and SIFT intuitions, Cogneethi
3. Lecture on Image Features - HOG and SIFT by Prof. M. K. Bhuyan, IIT Guwahati
4. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177