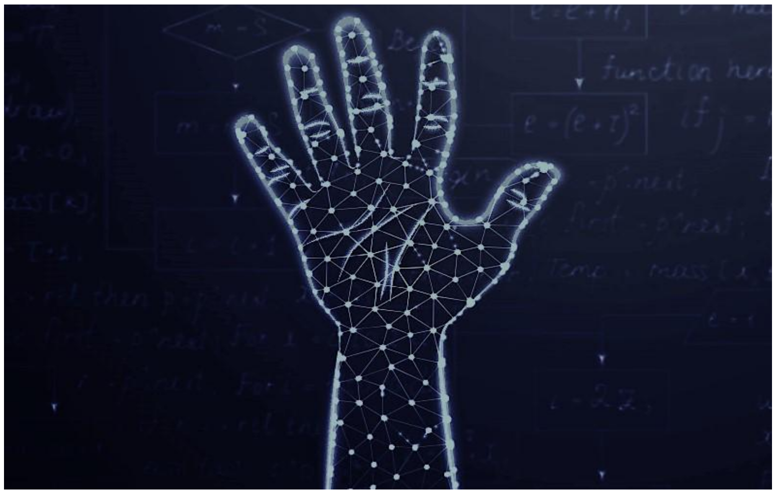
Cairo University
Faculty of Engineering

Department of Computer Engineering

Team #9

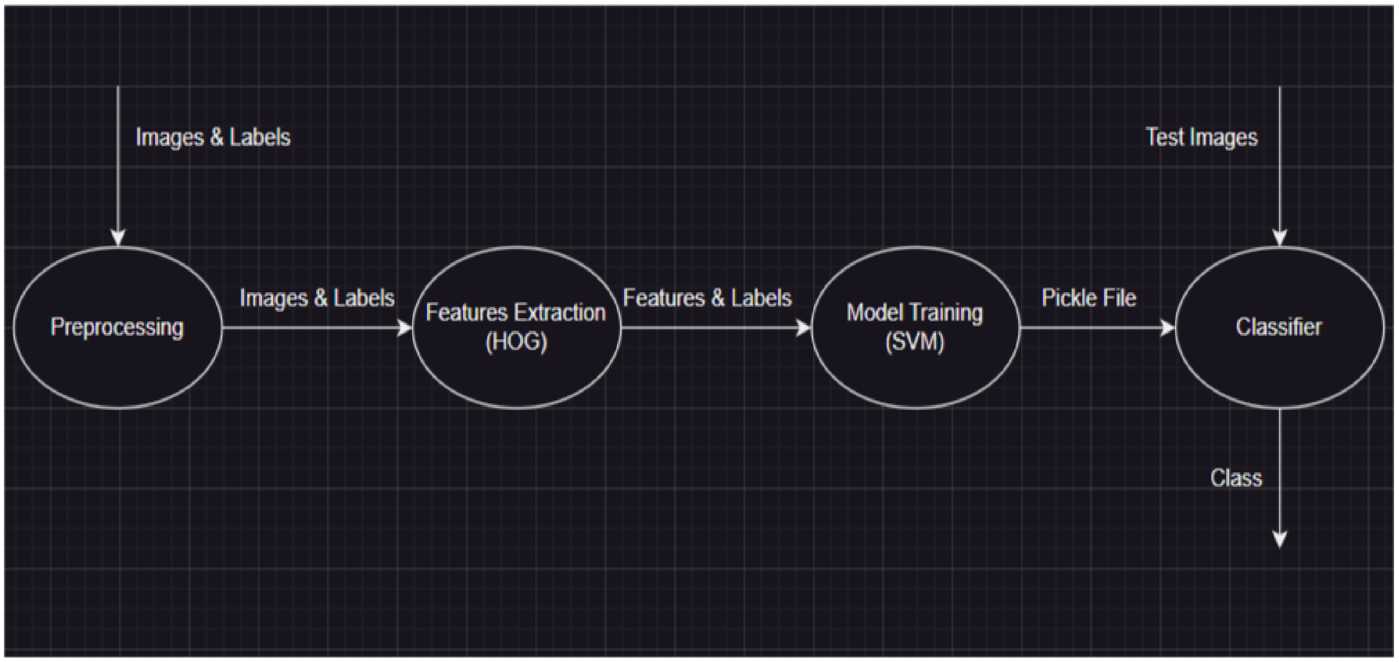
Project Report

Hand Gesture Recognition



Delivered to Eng.: Mohamed Shawky.

Project Pipeline:



Preprocessing Module:

Our main problem at this stage is the fingers shadows due to poor imaging conditions, so, we overcome over that by applying multiple steps for every input image:

1. *Convert the input image to YCrCb color space.*
2. *Get the segmented image using 'range_segmentation' method as follows:*
    i. Segment the image based on the lower and upper bounds of skin color defined in YCrCb color space.
    ii. Apply morphological operations to remove noise.
    iii. Find the contours in the binary segmented image, get the contour with the largest area.
    iv. Create a blank image to draw and fill the contours.
    v. Draw the largest contour on the blank image and fill the contour with white color.

vi. Return the image with the largest contour drawn on it.

3. *Apply thresholding on cr and cb components.*
4. *Apply the following formula bitwise (cr || cb) && range_segmented_image.*
5. *Apply morphological operations to remove noise.*
6. *Get the image with the largest contour area.*
7. *Apply histogram equalization to make the image more clear.*
8. *Cut the greyscale image around the largest contour.*
9. *Resize the image to small size to reduce extracted features array length.*

<u>Feature Extraction Module:</u>

We tried multiple algorithms: SURF, SIFT, LBP, and HOG. And we selected HOG (Histogram of Oriented Gradients) for providing high accuracy.

HOG algorithm is a computer vision technique used to extract features from images. It works by dividing an image into small cells and computing the gradient orientation and magnitude for each pixel within the cell. The gradient orientations are then binned into a histogram, which represents the distribution of edge orientations within that cell.

The HOG algorithm has high accuracy because it is able to capture important information about the edges and contours in an image. This information can be used to identify objects or patterns within the image, even if they have varying lighting conditions.

We apply HOG algorithm using hog() from skimage.feature, and with making sure that all features vectors are with the same length by padding zeros.

Model Training Module:

We tried multiple algorithms: Random Forest, Naive Bayes, Decision Tree, and SVM. And we selected SVM (Support Vector Machine) for providing high accuracy.

SVM is a powerful and popular machine learning algorithm used for classification and regression analysis. It works by finding the best hyperplane or decision boundary that separates the data into different classes. The hyperplane is chosen such that it maximizes the margin between the two classes, which helps to improve the generalization ability of the model.

SVM has several advantages that make it a good choice for classification tasks. Some of these advantages include:

1. Effective in high-dimensional spaces: SVM can effectively handle datasets with a large number of features or dimensions.
2. Robust to outliers: SVM is less sensitive to outliers in the data compared to other algorithms.
3. Versatile: SVM can be used with different types of kernels, such as linear and polynomial, which makes it versatile and applicable to different types of datasets.
4. Memory efficient: SVM uses a subset of the training data called support vectors to build the model, which makes it memory efficient and scalable to large datasets.
5. Good generalization performance: SVM aims to maximize the margin between the two classes, which helps to improve the generalization performance of the model.

Overall, SVM is a powerful and flexible algorithm that can be used for various classification tasks in machine learning.

We apply SVM algorithm using SVC() from sklearn.svm with linear kernel, and %80 of dataset images for training, and %20 of dataset images for testing.

Performance Analysis and Results:

Our result is that the trained model predicted 84% of images correctly.

You can run with your data under `data` folder under the main folder `Hand-Gesture-Recognition` the script `./FinalCode/app.py` to see output label and time taken for processing every image by using this command:

```
$ py .\FinalCode\app.py --feature 0
```

Note: You should see the output in 2 files: `results.txt` for output classes and `time.txt` for time taken to process the image after reading it and just after the prediction, every line in these 2 files is for one image.

Enhancements and Future Work:

Although we have optained an accuracy of 83% on our dataset, we believe that our work would be improved using these points.

1. *Larger dataset.*
2. *Using Deep Neural Networks.*
3. *Better imaging conditions.*

Workload Distribution:

Name	**Sec.**	**B.N.**	**Code**	**Work**
Youssef Khaled	2	37	9203760	Features Extraction and Running Model Training
Abdelrahman Mohamed Hamza	1	37	9202793	Preprocessing Module
Beshoy Morad	1	20	9202405	Model Training and Performance Analysis Module
Moaz Mohamed	2	26	9203532	Preprocessing Module and Documentation