Cairo University

Faculty of Engineering
Cairo University

Hand Gesture Recognition System

Pattern Recognition and Neural Networks

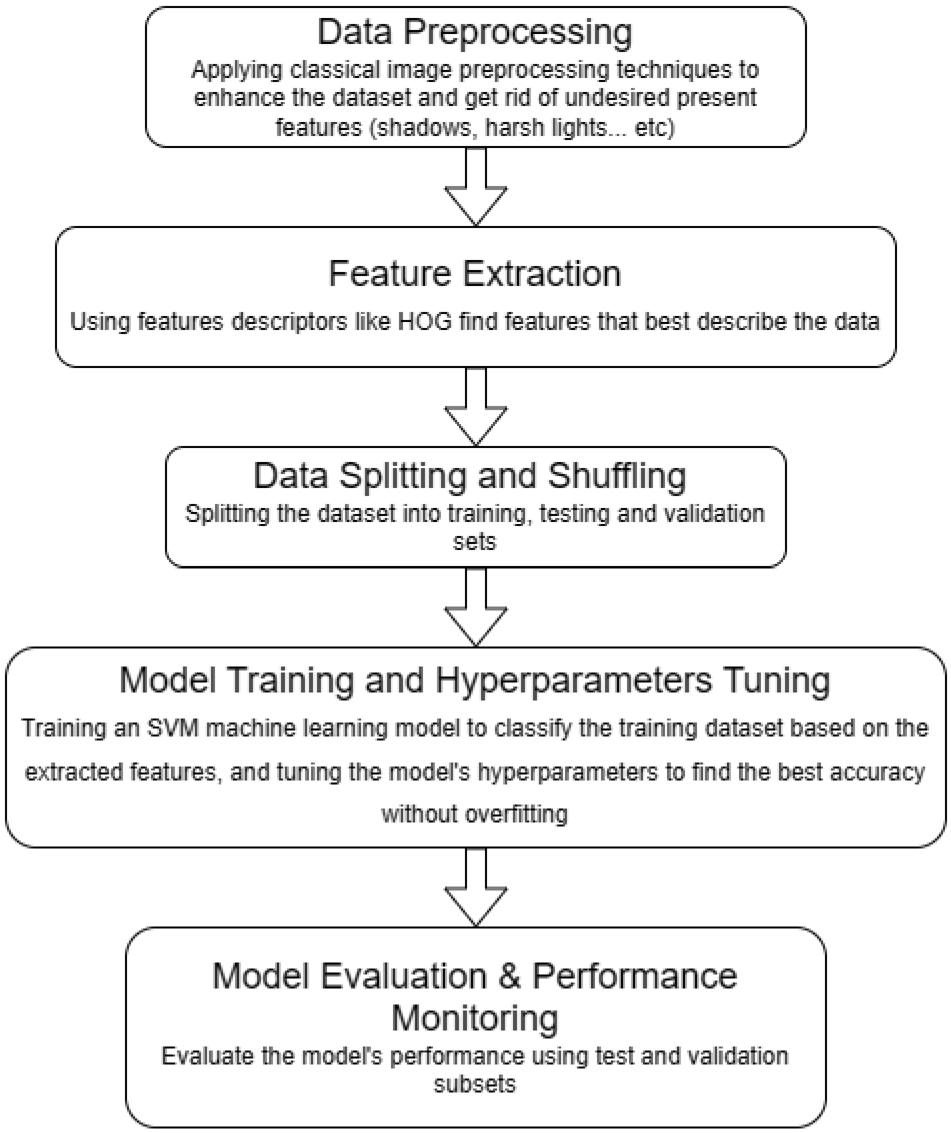
CMP3006

Name	Section	BN	Code
Sarah Mohamed Hossam	1	30	9202618
Ahmed Ata Abdallah	1	6	9202139
Rufaida Kassem	1	26	9202550
Doaa Magdy	1	24	9202521

Supervised by: Dr. AbdelMonem Bayoumi

Submitted to: Eng/ Mohamed Shawky

- **Project Pipeline**


Data Preprocessing
Applying classical image preprocessing techniques to enhance the dataset and get rid of undesired present features (shadows, harsh lights... etc)
Feature Extraction
Using features descriptors like HOG find features that best describe the data
Data Splitting and Shuffling
Splitting the dataset into training, testing and validation sets
Model Training and Hyperparameters Tuning
Training an SVM machine learning model to classify the training dataset based on the extracted features, and tuning the model's hyperparameters to find the best accuracy without overfitting
Model Evaluation & Performance Monitoring
Evaluate the model's performance using test and validation subsets

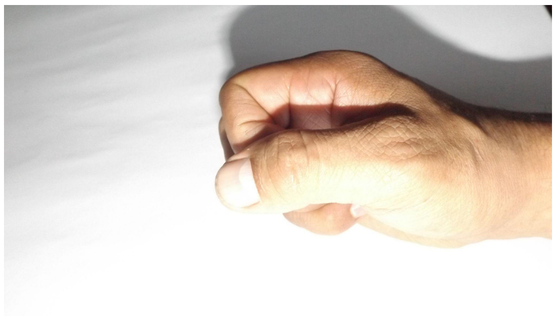
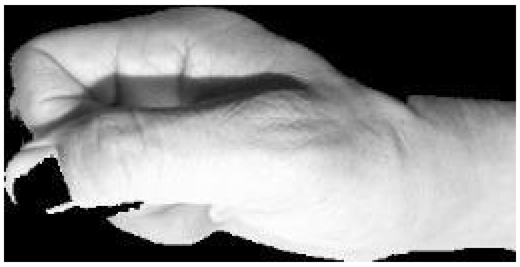
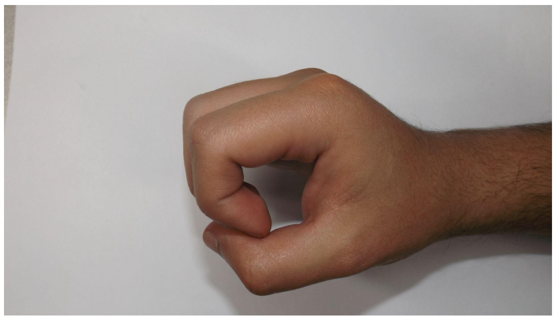
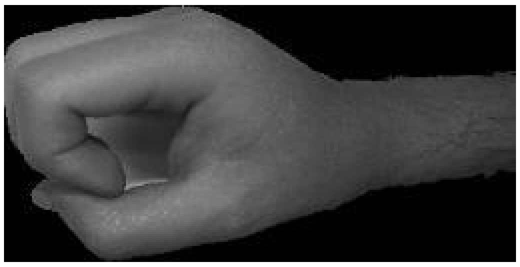
After several research attempts and multiple trials in terms of models and features choice, the final project proposed pipeline is as follows:

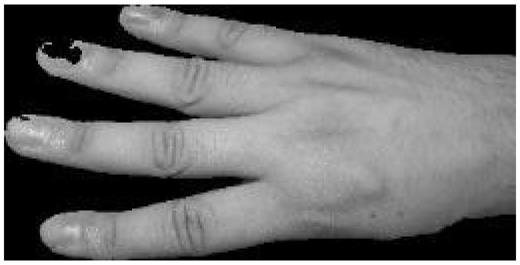
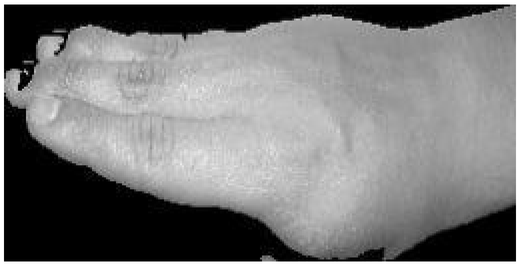
1. Data preprocessing: by applying classical image processing techniques to preprocess the data for removing shadows, enhancing colors, clipping the area surrounding the hand, rotating so all hands would be pointing in the same direction and resizing for efficiency.

2. Extract the HOG features.
3. Splitting the features into training, validation and testing sets.
4. Pass the training set to the support vector machine (SVM) model, with a poly kernel.
5. Using the confusion matrix, f score, and cross validation to evaluate the model's efficiency.

- **Preprocessing**

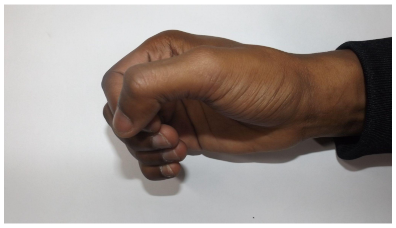
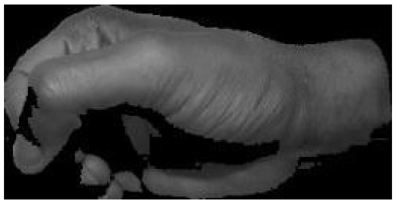
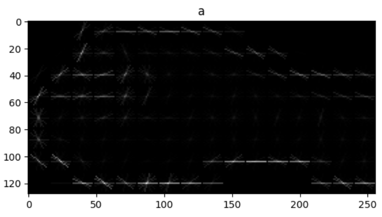
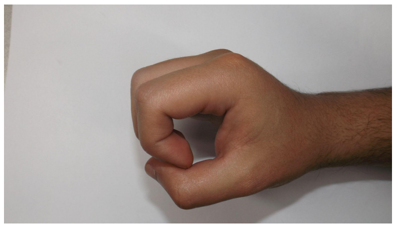
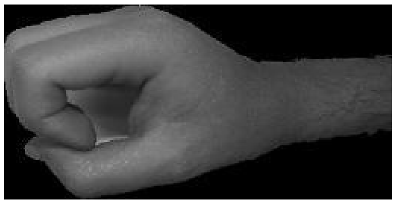
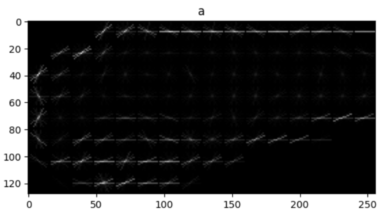
Segmentation is done using the YCRB channels using a basic thresholding and ignoring the Y(illumination channel)
Then Clipping the photo to only contain the hand.
Lastly, to unify the orientations of hands, the preprocessed image is passed to a function to flip it so that the fingers are pointing to the left.

Raw Image	Preprocessed Image
	
	

- **Feature Extraction**
  - HOG
    Using the Histogram of Gradients feature descriptor was our best choice, since it’s robust to variations in appearance, computationally efficient, its discriminative power as it’s very efficient in capturing the distinguishing features of an object, and lastly its compatibility with machine learning algorithms.

- Visualized HOG examples

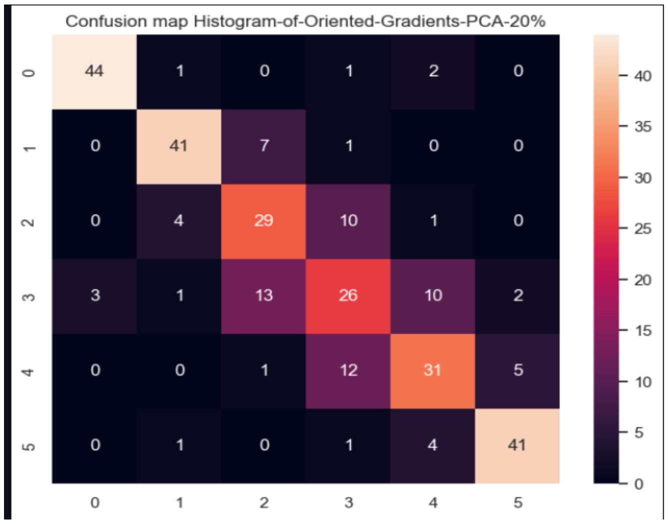
Raw	Preprocessed	HOG
		
		

- Trials, and they were all followed by the same classifier:

Feature Extraction Algorithm	HOG + PCA	Geo-metric	ORB + K-means clustering	SIFT + K-means clustering	PCA	Elliptic Fourier	LBPH	HOG + LBPH	HOG
Accuracy	88%	20%	30%	63%	56%	74%	30%	91%	91.97 %

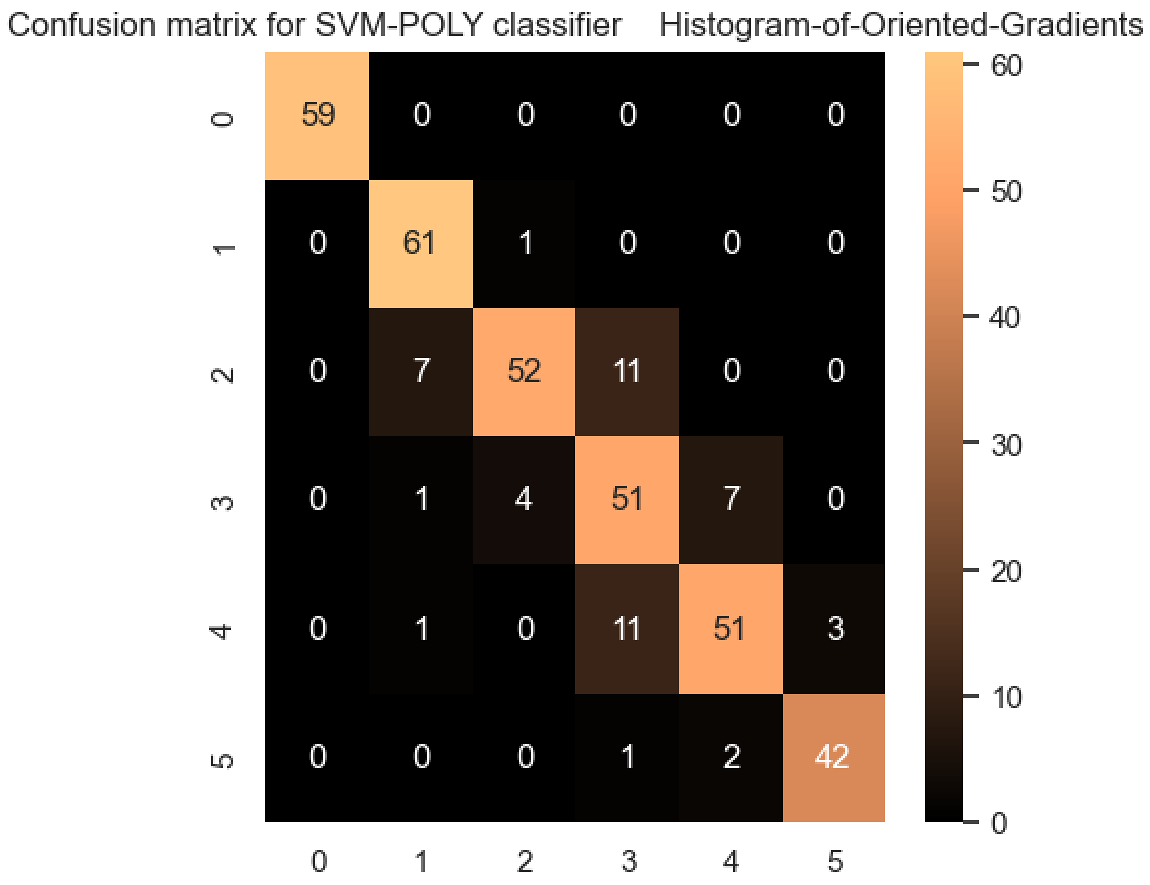
- HOG with PCA
    - Combining HOG & PCA decreased the accuracy with 2%, but enhanced the training time. However, since the training time is not our biggest concern, we decided on using raw HOG features.
- HOG with LBPH
    - Same accuracy as HOG only with larger feature vectors.
- Geometric (maximum contour area, convex hull, ..) features
- ORB followed by K-means clustering
- SIFT followed by K-means clustering

        - PCA
        - Elliptic Fourier
        - LBPH
- All the above feature extraction methods are used from OpenCV and sci-kit learn libraries.

- **Model Selection and Training**
    - SVM with rbf kernel, and all other default parameters
        - We use SVM on the HOG extracted feature
        - We try to run SVM on the whole dataset and get the following confusion matrix
- 

	0	1	2	3	4	5
0	44	1	0	1	2	0
1	0	41	7	1	0	0
2	0	4	29	10	1	0
3	3	1	13	26	10	2
4	0	0	1	12	31	5
5	0	1	0	1	4	41

        - Then we notice that the max confusion occur between 4 & 3 and 2 & 3
        - So, we train one model on the 2 & 3 dataset only
        - Then train another on the 3 & 4 dataset only
        - Then train one on the rest of the data
        - Hence, we use a multi-layer model by combining all models, when getting an output that's more likely to get confused with another, then it's sent to the another classifier that's trained to differentiate between that output and the other ones it's confused with…etc to get the following confusion matrix:



- We can note that the confusion become less than in case of splitting models based on the classes that get confused together the most than training one model on the whole dataset
- And the accuracy becomes ….. **91.97**!!!

- We tried tuning the model's hyperparameters using Grid Search and randomized search, but the default ones were good enough.

  - Less accurate trials:
    All the following trials used the models' default hyperparameters.

Model	**Accuracy**

SVM with poly kernel	91.78%
Random Forest	88.49%
KNN	87.67%
SGD	86.02%
Naive Bayes	85.47%
Sklearn voting using few combinations	85%

- **Performance Analysis**
    - Analyzing the models' performance using confusion matrices, scores and accuracies to find the best feature-model combination.

We used multiple Models and computed their fitting time and the time they take to predict a label and the result are as follows
Fitting Times:

```
Random Forest: [2.686070680618286]
Naive Bayes: [0.11193251609802246]
SVM-RBF: [2.0852882862091064]
SVM-POLY: [1.9109680652618408]
KNN: [0.006542205810546875]
SGDClassifier: [0.981544017791748]

And the prediction times of wholetrain data
Random Forest: [0.015572786331176758]
Naive Bayes: [0.06784844398498535]
SVM-RBF: [1.3060433864593506]
SVM-POLY: [0.6664779186248779]
KNN: [0.05323219299316406]
SGDClassifier: [0.007521867752075195]
```

- **Future Work and Enhancement**
  A possible enhancement would be on the preprocessing module to eliminate the bad lighting effects, shadows, different hand orientations and positions.

Another possible enhancement would be finding a better set of features that describe the data better, and give intuitive, meaningful features of the data.
Lastely, tuning the classifier’s hyperparameters and searching for a classifier that best fits the problem.

- **Workload distribution**

Student	**Workload**
Ahmed Atta	Data preprocessing
Rufaida Kassem	Feature extraction
Doaa Magdy	Feature extraction
Sarah Mohamed Hossam	Model training and performance analysis