Implementation of HOG Feature Extraction with Tuned Parameters for Human Face Detection

Mohammed G. Mohammed and Amera I. Melhum

Abstract—**Extracting and tracking face in image sequences is a required first step in many applications such as face recognition facial expression classification and face tracking, it is a challenging problem in computer vision field because of many factors that effects on the image, some of these factors are luminosity, different face colors, background patterns, face orientation and variability in size, shape, and expression. The objective of this paper is to Experiment wide range of parameters for HOG face detector and setting up the most suitable kernel for Support Vector Machine (SVM) and then, comparing this method with some well-known methods for face detection and identifying the most reliable one. The aim of this study is not providing the best face detector method rather than a try to find out the performance of HOG feature for detecting a face, experimenting different kernels and eventually finding the tuned parameters for HOG descriptors for detecting a face, in this study based on experimental results as shown in Table IV. The HOG + SVM scores the highest value of precision, accuracy, and sensitivity. As 0.8824, 0.9986 and 0.75 respectively compared to Viola-Jones method which scores 0.6512, 0.9973 and 0.7 finally skin color method which scores 0.3968, 0.9947 and 0.625.**

Index Terms—**Face detection, histogram of oriented gradient, machine learning, support vector machine, viola-jones, skin colour.**

I. INTRODUCTION

Human face detection is one of the most important first step action in many computer vision applications such as face tracking, emotion recognition, facial feature extraction, document control, access control, clustering, gender recognition and classification. Human computer interaction (HCI) and lately threat detection from facial expressions. Early face detection methods used anthropometric and heuristic techniques as in the 1970s [1]. These methods tend to be failed because they were very sensitive to changes in terms of lighting, scaling and orientation conditions. Furthermore, they were only able to detect faces in simple environments like pictures with only white background. While in the meantime, there are many approaches and well-developed algorithms are available for face detection such as knowledge-based methods, feature based and template matching techniques [2]–[4]. In the last 20 years, various algorithms and methods have been introduced for face detection such as Viola-Jones 2001 [5], which is based on haar like features, skin based methods and machine learning methods such as Convolutional Neural Network (CNN) or ConvNet, a support vector machine (SVM) combined with a feature extraction technique such as Histogram of oriented gradients (HOG), speeded-up robust features (SURF) and Local binary patterns (LBP). In this paper our main attention would be on geometrical shape-based method (HOG) with tuned parameters and experimental results of using different kernels such as Gaussian and Laplacian. Then, comparing our results with a Haar-Like feature based method (Viola-Jones), in order to make a fair comparison the same database which is MIT CBCL face database has been used for training both methods.

II. RELATED WORKS

A. Viola-Jones Method

One of the most successful attempts was by Paul Viola and Michael Jones [5]. Their approach was based on a cascade of simple features available on a human face. Resulting in one of the most successful achievement at that time in terms of accuracy and speed. In this method Viola and Jones used Haar-Like features which is a scalar product between some Haar-Like feature templates as shown in Fig. 1 and an image. Mathematical representation is shown in equation 1 and resulted image of the applied equation shown in Fig. 2. In the following equation *I* and *P* indicate a picture and a pattern respectively, both of the similar size N × N the feature associated with pattern *P* of image *I* is defined by:

$$\sum_{1\le i\le N}\sum_{1\le j\le N} I(i,j)1_{P(i,j)\text{ is white}} - \sum_{1\le i\le N}\sum_{1\le j\le N} I(i,j)1_{P(i,j)\text{ is black}} \qquad (1)$$

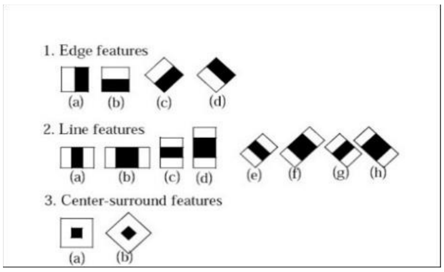


Fig. 1. Set of Haar-like features (Yi-Qing, 2014).

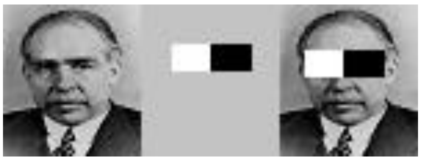


Fig. 2. Niels Bohr face combined with Haar-feature template.

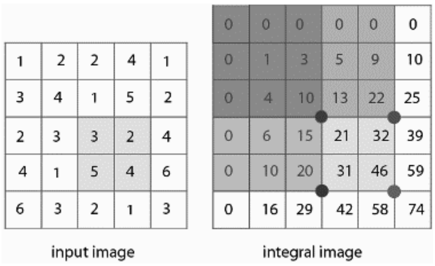
Only the pixels in black and white are used for feature calculation. Then, adaboost learning algorithm used to give a meaning to the calculated features after a scalar product

Manuscript received August 2, 2019; revised June 24, 2020.

The authors are with the Dept. of Computer Science, College of Science, University of Duhok, Duhok, Kurdistan Region, Iraq (e-mail: mohammed.guhdar@uod.ac, amera_melhum@uod.ac).

doi: 10.18178/ijmlc.2020.10.5.987

between the image and Haar-Like component after that, it decides whether the certain area (window) is a face or not a face, one of the most interesting point on the work that is presented by Viola and Jones. In contrast, to most of the face detection systems which focusing on pixel intensities, Viola-Jones used a new representation of image called the "integral image" or summed area table to increase the speed of features calculation as shown in Fig. 3.



1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

input image

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

integral image

Fig. 3. Pixel values of image (Left) integral image representation (Right).

A linear combination of weak classifiers constructed a strong classifier in order to increase accuracy and reducing computation time. Fig. 4 shows the structure of cascade classifier. Adaboost requires positive and negative data for training in order to be able to correctly classify face candidates from non-face candidates.

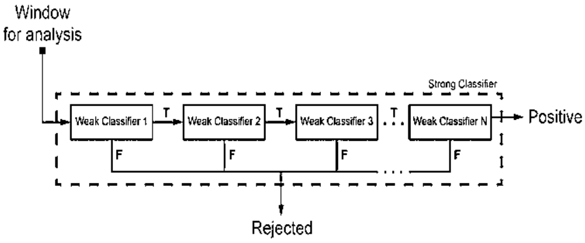


Fig. 4. Cascade classifier concept (Viola and Jones, 2001).

One serious problem with Viola-Jones method is that having different orientation of face, However, it can be solved by adding more cascades in different orientation.

B. *Skin Color Detection*

One of the fastest methods for face detection would be human skin color detection [6]-[10]. Skin color is utilized to decide if the specific pixel is a skin color pixel or non-skin shading pixel. For this method in order to get accurate results for detecting different face colors, most of the researchers attend to use YCbCr color model space, because unlike RGB color space the YCbCr is independent in terms of luminance. Therefore, it achieves better accuracy. This algorithm is fast because it eliminates many unwanted pixels mostly about 60% to 70% of image area that requires to be processed by eliminating all non-skin color pixels.

For the components of YCbCr Y stands for luma component which represent the luminance and computed from nonlinear RGB [11]. It is acquired as weighted sum of RGB values. Cb is the dimension of contrast among blue and luma segment and Cr is the dimension of distinction among red and luma part. The Y stands for the luminance, while, Cb and Cr are the chrominance segment.

$$Y = 0.299R+0.587G+0.114B$$
$$Cr = R-Y$$
$$Cb = B-Y$$

The threshold used in this specific algorithm for detecting skin colour is given to be 76< Cb <127 and 132 < Cr < 173 [12], Unlike RGB, it has separate luminance and chrominance parts which make this shading space alluring for skin color division [13].

III. Support Vector Machines (SVMs)

Support vector machines (SVMs) are a set of related supervised learning methods in machine learning used for regression and classification [14]. The simple term for a given set of training data each of which pointed to one of two classes, SVM algorithm builds a model for predicting feature data into which category it belongs. Intuitively, an SVM model is a representation of the data as mapped points in space, so the data points in space becomes distinguishable. And then new entered data are mapped into that same space and predicted in to which classes it belongs based on the side of the gap it falls on

A linear support vector machine is composed of a set of given support vectors *z* and a set of weights *w*. The calculation for the output of a given SVM with *N* support vectors *z*1, *z*2, … , *zN* and weights *w*1, *w*2, … , *wN* is then given by equation 2 [15].

$$F(x) = \sum_{i=1}^{N} w_i \langle z_i, x \rangle + b \qquad \dots (2)$$

A. *Kernel Trick*

One of the simplest definitions for kernel trick would be transforming linearly inseparable data like shown in Fig. 5 to linearly separable ones like shown in Fig. 6 by using a plane.

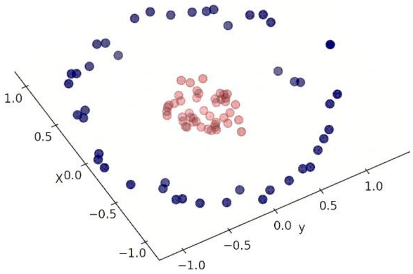


Fig. 5. Viewing data in lower dimension.

The kernel function is what is applied on each data instance to map the original non-linear observations into a higher-dimensional space in which they become separable.

In 1992, Bernhard Boser, Isabelle Guyon and Vapnik suggested a tricky way to create non-linear classifiers from a linear classifier by applying the kernel trick (originally proposed by Aizerman *et al.*) to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be non-linear and the transformed space high dimensional, thus the classifier is a hyperplane in the high-dimensional feature space. It might be non-linear in the original input space.

Using kernels, the original formula for the SVM given SVM with support vectors *z*1, *z*2, … , *zN* and weights *w*1, *w*2, … , *wN* is now given by:

$$F(x) = \sum_{i=1}^{N} w_i\, k(z_i, x) + b \qquad \dots (3)$$

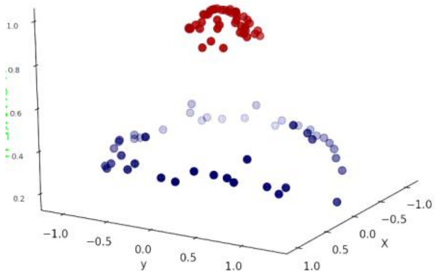


Fig. 6. Viewing data in higher dimensions (kernel Effect).

1) *Gaussian kernel*

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \qquad (4)$$

Alternatively, it could also be implemented using

$$k(x,y) = \exp(-\gamma\|x-y\|^2) \qquad (5)$$

The sigma (σ) is an adjustable parameter in which it plays an important role in the accuracy of the kernel. It should be tuned carefully depending on the problem. If it is overestimated, the exponential will behave linearly causing the higher-dimensional projection to lose its non-linear power. In the other side, by underestimating it, the function will lack regularization and the decision boundary will become sensitive to noise. *x* and y are $N\times N$ size convolution matrices.

2) *Laplacian kernel*

The Laplace kernel is closely related to the Gaussian kernel, with only the square of the norm left out. It is also a radial basis function kernel. But it is less sensitive to changes in the sigma parameter.

$$k(x,y) = \exp\left(-\frac{\|x-y\|}{\sigma}\right) \qquad (6)$$

By using Kernel function (Laplacian) the data can be transformed into higher dimensions in which it has more freedom to separate them, *x* and *y* are $N\times N$ size convolution matrices.

IV. Histogram of Oriented Gradient and Support Vector Machine

Dalal and Triggs at the CVPR conference in 2005 [17], by using HOG descriptor + linear SVM and a detection window that is 64 pixels wide by 128 pixels tall successfully detected a person in a given image with a great accuracy. In 2018 Chee, Kok Wei, and Soo Siang, implemented a combination of HOG with HOM (histogram of Magnitude) for the same purpose and achieved even greater accuracy which is 99.0% accuracy compared to HOM (95.5%) or HOG (98.6%) features when they are used independently [18] and same concept has been applied for human tracking [19] with great accuracy.

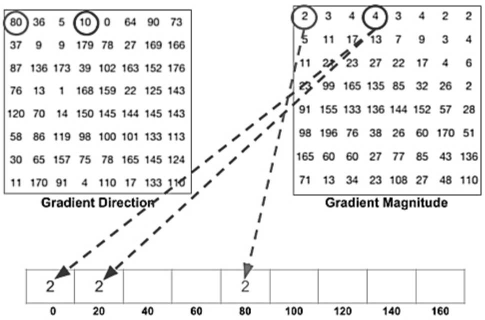
HOG features are coded version of the image created by using cells and block structures. They provide information about shape orientations based on equation 7, and luminosity densities based on equation 8 for each pixel. Most importantly we can observe changes in shape with each direction.

1) A HOG features vector are generated by combining the gradient calculations of each pixel as shown in Fig. 7.
2) Generating a histogram for each block by using gradients value.
3) Calculating the normalization of the histograms as shown in Fig. 8.

Collecting the normalization vectors together for each block.

$$\text{Angel} = \theta = \arctan\frac{g_y}{g_x} \qquad (7)$$

$$\text{Magnitude} = \sqrt{g_x^2 + g_y^2} \qquad (8)$$



80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

0	20	40	60	80	100	120	140	160
2	2			2				

Fig. 7. 8×8 Gradient Directions (left), 8x8 Gradient Magnitude (right).

Calculating an 8×8 block histogram of gradients based on the above data a histogram of 9 bins is calculated as shown in Fig. 8. The, magnitudes of less than 10 are added to degree 0 values and magnitudes between 10 and less than 30 are added to degree 20 and so on.

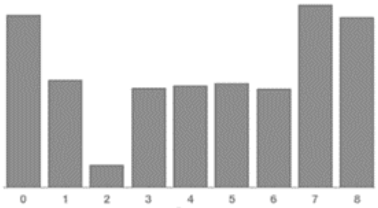


Fig. 8. Histogram of 9 bins.

Based on the above steps and after pre-processing steps, all the images in MIT CBCL face databases are converted in to their HOG representation. Which is a feature vector that represents face image as visualized in right side of Fig. 9.

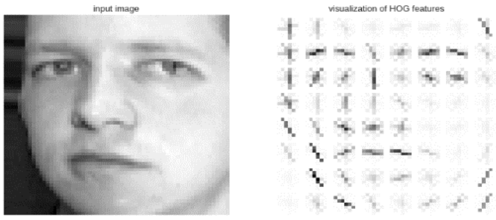


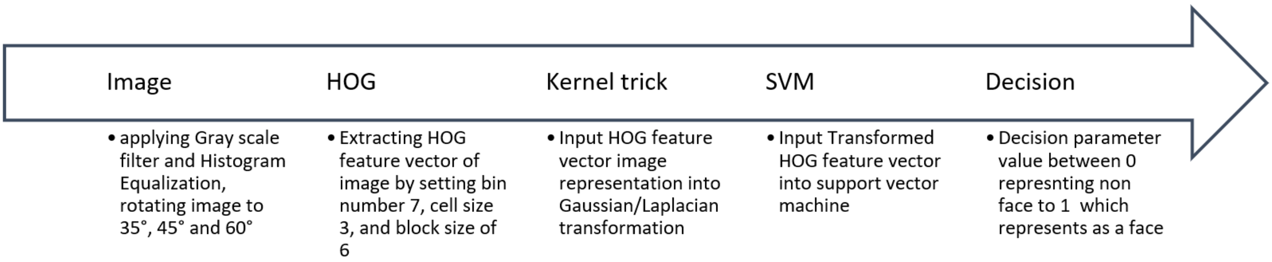
Fig. 9. A person image from ORL database (Left), HOG visualization of the same image (Right).

V. The Proposed Method

The proposed method uses support vector machine with

HOG features in either Laplacian or Gaussian kernel method for transforming data. Furthermore, tuned parameters for HOG features which are a number of bins, Cell size and Block size has been identified for face detection based on experimental results. In which for Bin size=7, Cell size 3 and Block size 6 achieved the most successful results in detecting faces in a crowded and complex environment in an image.

Viola-Jones suffers from detecting faces in different orientations, to solve this issue Viola-Jones method adds more Haar-like features in rotated order. Many available methods tend to use rotating sliding windows to 35 °, 45 °and 60 ° degree in both directions in order to detect faces in different orientation. All of these additional steps increase the complexity and decreases the performance.



- applying Gray scale filter and Histogram Equalization, rotating image to 35°, 45° and 60°
- Extracting HOG feature vector of image by setting bin number 7, cell size 3, and block size of 6
- Input HOG feature vector image representation into Gaussian/Laplacian transformation
- Input Transformed HOG feature vector into support vector machine
- Decision parameter value between 0 represnting non face to 1 which represents as a face

Fig. 10. Configuration of the proposed method.

Another problem with these methods is using different size of sliding window for detecting an object in multi scales and for the Viola-Jones just adds more Haar-Like features but this time larger and smaller window sizes used to detect a face in multi scales

In this proposed method the pyramid images have been used to detect objects in multi scale and instead of rotating sliding window a rotated image has been used. This increased the speed for almost 90%.

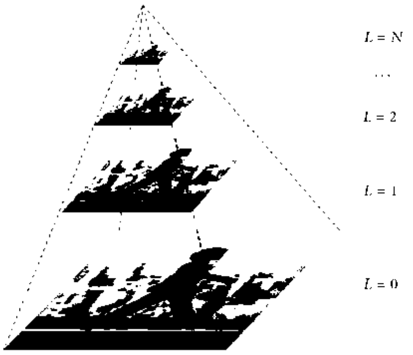


Fig. 11. pyramid image.

For example, if we have a 500px by 350px image and if we use a sliding window 1px slide in X and one px in Y direction each time. Eventually we will end up of having 175,000 windows if we rotate each window 3 times for each degree it will become 525,000 operations and let's say a 4 degrees of scaling we will end up of having 2,100,000 operations which is a huge number of operations, especially when we want to implement a method for real time.

While in our case we will still have 175,000 windows in the first step but with only 3 operations because we will rotate the whole image 3 times instead of rotating window by window. For scaling level by using pyramid image by dividing image dimensions by two each time for 3 times, we will have 175,000, 43,750 and 10,875 = 229,625 windows, multiplying this number by 3 for each rotation degree process, the total number of operations will be 688,875 operations.

The rotation operation on an image performs a geometric transform which maps the position ($x1$, $y1$) of a picture element in an input image onto a position ($x2$, $y2$) in an output image by rotating it through a user-specified angle θ about an origin O through an iterative process, which means each pixel [$x1(i)$, $y1(i)$] is rotated one by one by using the equation 9 and 10, where i represents a pixel location. In most implementations, output locations ($x2$, $y2$) which are outside the boundary of the image are ignored. Rotation is mostly used to improve the visual appearance of an image, although it can be useful as a pre-processor in applications where the directional operators are involved.

The rotation operator performs a transformation of the form:

$$x2=\cos(\theta)*(x1-x0) - \sin(\theta)*(y1-y0) + x0 \qquad (9)$$

$$y2=\sin(\theta) * (x1-x0) - \cos(\theta) * (y1-y0) + y0 \qquad (10)$$

where ($x0$, $y0$) are the coordinates at the center of rotation (in the input image) and θ is the angle of rotation with clockwise rotations having positive angels, Fig. 12 shows an image rotation in 90 °degrees.

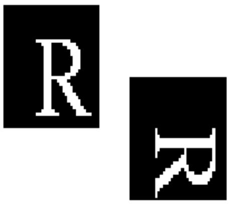
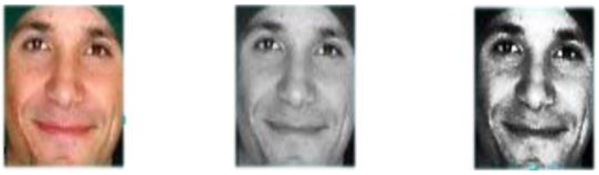


Fig. 12. Rotated Image after performing transformation.

VI. Pre-processing

In the pre-processing steps, first grey scale filter is applied on the image by using accord.net open source library which is a predefined function to achieve this task, then histogram equalization applied as shown in Fig. 13. While Fig. 14 shows that how low performance is achieved before applying histogram equalization on the image, and in Fig. 15 appears how the performance is improved after applying the histogram equalization.



Normal Image Graye Image Histogram equalization

Fig. 13. Pre-processing steps.

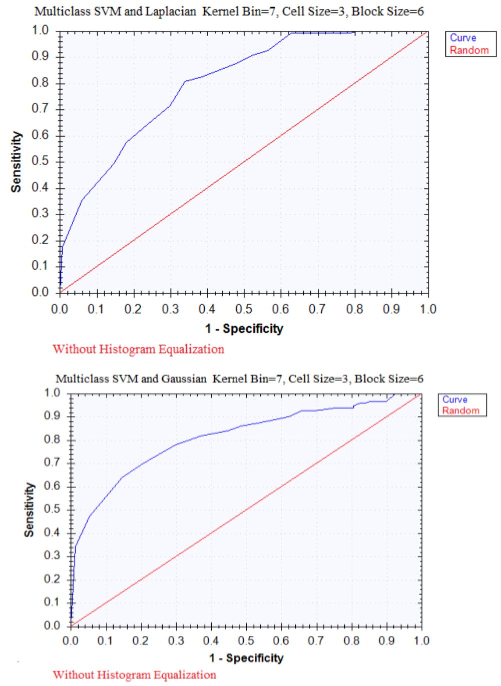


Fig. 14. Performance before applying histogram equalization.

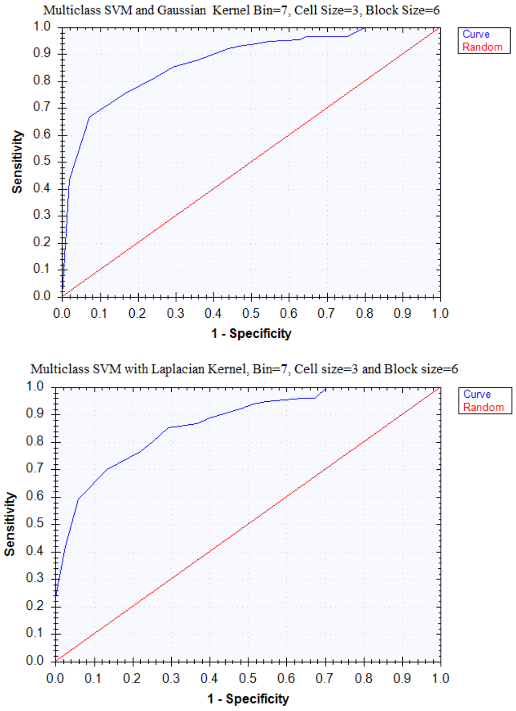


Fig. 15. Performance after applying histogram equalization.

VII. Experimental Results and Discussion

A confusion matrix and measurements below are the average test results; a confusion matrix is a popular representation of the performance of classification models. The matrix (table) demonstrates to us the quantity of correctly and mistakenly classified data points, contrasted with the real results (target esteem) in the test examples. One of the benefits of utilizing this matrix as assessment tool is that, it permits increasingly investigation (for example, if the model is befuddling two classes) than straightforward extent of effectively classified data points (i.e. accuracy) which can give deceiving results if the dataset is unequal (for example at the point when there are tremendous contrasts in number of between distinction classes).

Abbreviations for confusion matrix:

True Positive (TP): Correctly predicting a label

True Negative (TN): Correctly predicting the other label

False Positive (FP): Falsely Predicting a label

False Negative (FN): Missing Positives: test result that indicates that a condition does not hold, while in fact it does.

Derivations

Sensitivity	**TPR=TP/(TP+FN)**
Specificity	SPC=TN/(FP+TN)
Precision	PPV=TP/(TP+FP)
Negative Predictive Value	NPV=TN/(TN+FN)
False Positive Rate	FPR=FP/(FP+TN)
False Discovery Rate	FDR=/(FP+TP)
False Negative Rate	FNR=FN/(FN+TP)
Accuracy	F1=2TP/(2TP+FP+FN)
F1 Score	ACC=(TP+TN)/(P+N)
Matthews Correlation Coefficient	TP*TN-FP*FN/sqrt((TP+FP)*(TP+FN)*(TN*FP)*(TN*FN))

TABLE I: Confusion Matrix and Measures for HOG + SVM

	True Positive	True Negative
Predicted Positive	30	4
Predicted Negative	10	9996

Measure	Values
Sensitivity	0.7500
Specificity	0.9996
Precision	0.8824
Negative Predictive Value	0.9990
False Positive Rate	0.0004
False Discovery Rate	0.1176
False Negative Rate	0.2500
Accuracy	0.8108
F1 Score	0.9986
Matthews Correlation Coefficient	0.8128

A. *Test Results for HOG + SVM*

In this research we attempted to use HOG descriptors + SVM with Gaussian and Laplacian kernels for face detection, For training MIT CBCL face database has been used in which contains about 4500 negative samples and 2,500 positive samples for training all in gray PGM formats. The HOG transformation is applied before training. Same databases have been used in Viola-Jones for training adaboost, after tuning the HOG hyper parameters, the highest precision value achieved as shown in Table I, that happened to be between 86%-97% detection rate and 1%-4.5% false positive rate. That is, after applying a lot of experiments and adjustments on HOG cell size, block size and number of bins. The experiment shows that the best result is achieved by setting cell size to 6, block size to 3 and bin number to 7.

1) Experimenting different kernel effect on HOG + SVM performance

Based on the experiments the Gaussian kernel achieved the best performance among all the other experimented kernels and Laplacian kernel scored the second best performance.

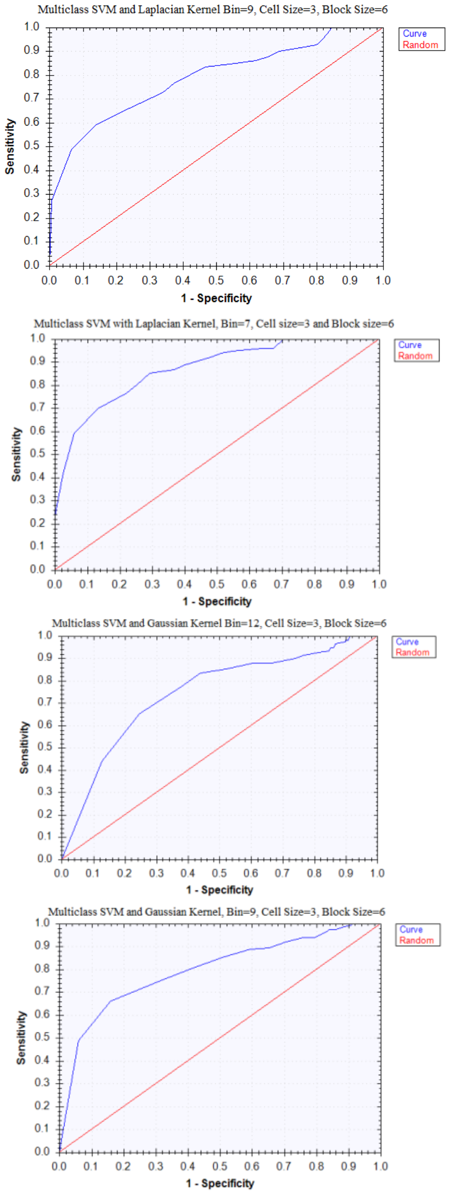
B. *Test Results for Viola-Jones*

After training the Viola-Jones on MIT CBCL database for face detection, it achieves good results with a frontal face of MIT database, however when testing it by CMU test set the accuracy falls down significantly, a possible reason for that could be that the training set in MIT which mainly contains frontal faces as shown in the left side of Fig. 17, while in other hand the CMU training set mainly contains different illumination and multi-view variation conditions as shown in the right side of Fig. 17. Furthermore, according to [5] and [20] the first Haar-like features has limitations for multi-view face detection and lack of robustness in detecting faces under various lighting conditions, as shown in Table II. The experiment shows that the Viola-Jones algorithm scores low precision value and high false positive rate.

TABLE II: CONFUSION MATRIX AND MEASURES FOR VIOLA-JONES METHOD

	True Positive	True Negative
Predicted Positive	28	15
Predicted Negative	12	9985

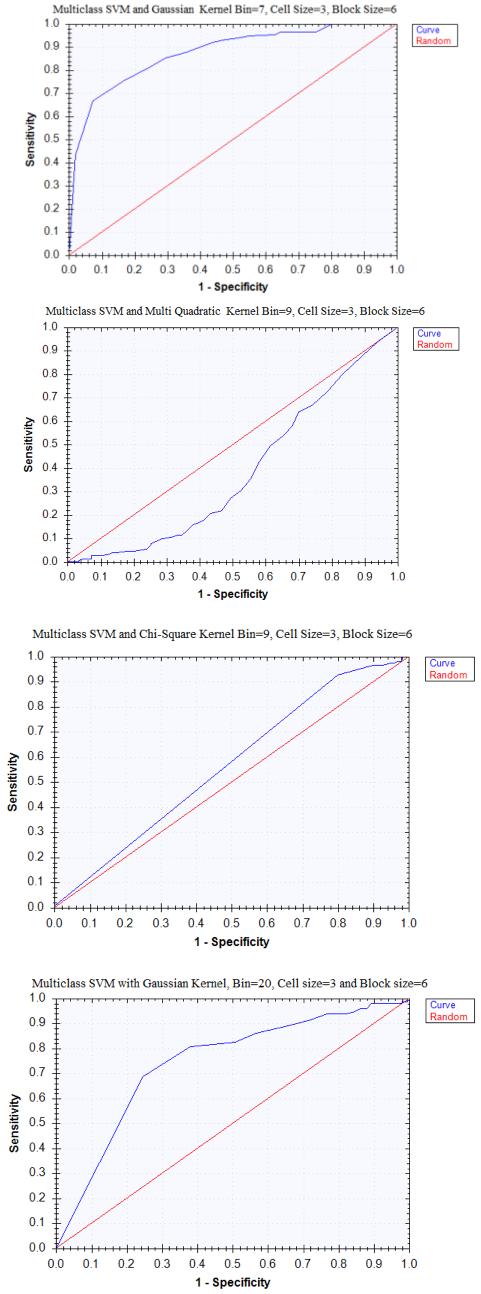
Measure	Values
Sensitivity	0.7000
Specificity	0.9985
Precision	0.6512
Negative Predictive Value	0.9988
False Positive Rate	0.0015
False Discovery Rate	0.3488
False Negative Rate	0.3000
Accuracy	0.6747
F1 Score	0.9973
Matthews Correlation Coefficient	0.6738


Multiclass SVM and Laplacian Kernel Bin=9, Cell Size=3, Block Size=6

Multiclass SVM with Laplacian Kernel, Bin=7, Cell size=3 and Block size=6

Multiclass SVM and Gaussian Kernel Bin=12, Cell Size=3, Block Size=6

Multiclass SVM and Gaussian Kernel, Bin=9, Cell Size=3, Block Size=6


Multiclass SVM and Gaussian Kernel Bin=7, Cell Size=3, Block Size=6

Multiclass SVM and Multi Quadratic Kernel Bin=9, Cell Size=3, Block Size=6

Multiclass SVM and Chi-Square Kernel Bin=9, Cell Size=3, Block Size=6

Multiclass SVM with Gaussian Kernel, Bin=20, Cell size=3 and Block size=6

Fig. 16. Different Kernel effect on face detection performance by using HOG + SVM.

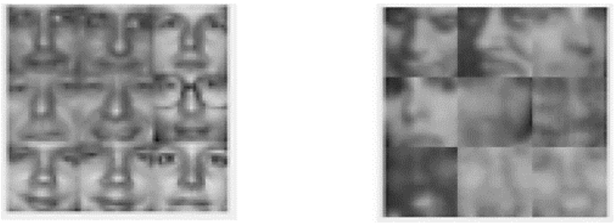


Fig. 17. Training images (left) and testing images (right).

C. *Test Results for Skin-Based Face Detection*

The test results for detecting faces based on skin color are shown in Table III proving that by depending only on skin color we get very low precision value and high rate of false positives.

Based on images color tone we can either get a better or a worse results, because of diversity in skin color. That would be one reason for skin-based face detection to fail at some point because there is no limit for color tones. In Fig. 18 and

Fig. 19 same method with the same configurations has been applied but totally different results have been achieved.

TABLE III: CONFUSION MATRIX AND MEASURES FOR SKIN COLOR DETECTION METHOD

	True Positive	True Negative
Predicted Positive	25	38
Predicted Negative	15	9965

Measure	Values
Sensitivity	0.6250
Specificity	0.9962
Precision	0.3968
Negative Predictive Value	0.9985
False Positive Rate	0.0038
False Discovery Rate	0.6032
False Negative Rate	0.3750
Accuracy	0.9947
F1 Score	0.4854
Matthews Correlation Coefficient	0.4956

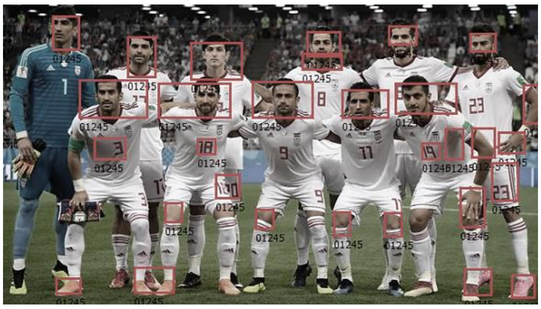


Fig. 18. Football team face detection based on skin.

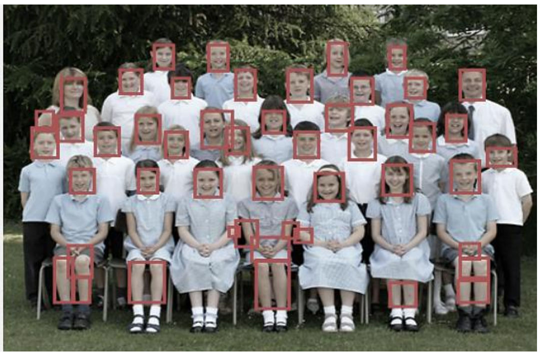


Fig. 19. School class face detection based on skin.

TABLE IV: OVERALLTEST RESULTS BETWEEN HOG + SVM, VIOLA-JONES AND SKIN COLOR

Measure	HOG + SVM	Viola Jones	Skin Color
Sensitivity	0.7500	0.7000	0.6250
Specificity	0.9996	0.9985	0.9962
Precision	0.8824	0.6512	0.3968
Negative Predictive Value	0.9990	0.9988	0.9985
False Positive Rate	0.0004	0.0015	0.0038
False Discovery Rate	0.1176	0.3488	0.6032
False Negative Rate	0.2500	0.3000	0.3750
Accuracy	0.9986	0.9973	0.9947
F1 Score	0.8108	0.6747	0.4854
Matthews Correlation Coefficient	0.8128	0.6738	0.4956

The above images are the test results of skin pixel detection. Test shows that there are many false positives since at this stage we only depend on skin color. Most of the false positives can be eliminated by using and additional step. One example for that purposes would be template matching or any machine learning techniques such as support vector machines or neural networks. But the accuracy of face locations is the real issue here and the solution for that problem is a hot topic of researching area.

The experiments show that one cannot depend only on skin detection for detecting a face accurately. There should always be an additional method alongside the skin detection in order to get rid of all these false positives. For example, if we use a template matching method or any classification methods as an additional method with skin detection the false detection rate could be reduced between 30% to 60% percent based on training data that will be provided to this additional methods for training purpose.

D. Running Speed

The performance of three methods is tested on 64-bit 1.2GHz quad-core ARM Cortex-A53 CPU. And, there is a huge difference in the speed of the three methods as shown in Table V.

TABLE V: SPEED DIFFERENCE OF THE THREE METHODS

Method	Time(s)
HOG	0.16
Viola-Jones	0.01
Skin	0.008

Because HOG method is more precise and covers most of the cases as rotation, scale and even when people wearing glasses or showing face to either side. In most of the frames it can detect faces, which results in slowing down the whole process.

VIII. CONCLUSION

It's concluded that, for face detection based on skin color we struggled to filter out many false positives even though after using template matching methods. Variant illuminations and different skin tone were the main problems for face detection based on skin color. In other hand; Viola-Jones algorithm takes a lot more time for training than SVM based algorithm but in detection for real time it achieves promising results in terms of speed. It is good for detecting frontal faces and can overcome on different illumination problems by using histogram equalization. But yet it struggles for detecting faces in different orientations, while Haar-like features are more robust to illumination changes than color. It becomes inefficient in terms of memory and processing cost while trying to solve different orientation problem by adding more haar like features. For Histogram of Orientated Gradient (HOG) turns into an incredible elective element choice as it is generally invariant to various enlightenment changes and is fit for catching geometric properties of countenances that are hard to catch with direct edge channels, for example, Haar-like highlights, it is fast, accurate and achieves promising results. Finally based on experimental results of this research recommends HOG + SVM over Viola-Jones and skin color for face detection.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Mohammed came up with idea, programming and designing the system

Ameera did data collection, support and advice for tuning the parameters.

REFERENCES

[1] E. Hjelmås and B. K. Low, "Face detection: A survey," *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236-274, 2001.

[2] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon*, September 2003, vol. 3, pp. 85-92.

[3] A. Albiol, L. Torres, and E. J. Delp, "Optimum color spaces for skin detection," in *Proc. 2001 International Conference on Image Processing*, October 2001, vol. 1, pp. 122-124.

[4] M. H. Yang, D. J. Kriegman, and N. Ahuja, 2002, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58.

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, vol. 1, no. 3, pp. 511-518.

[6] Y. Rao and R. Zhang, "A method of face detection based on skin color model in fixed scene," in *Proc. International Symposium on Intelligence Computation and Applications*, Springer, Berlin, Heidelberg, October 2012, pp. 152-158.

[7] S. Zhao, L. Shen, and W. Tan, "A method of dynamic hand gesture detection based on local background updates and skin color model," in *Proc. 2010 International Conference on Computer Application and System Modeling*, October 2010, vol. 5, pp. V5-657, IEEE.

[8] F. Işikdoğan and S. Albayrak, "Automatic recognition of Turkish fingerspelling," in *Proc. 2011 International Symposium on Innovations in Intelligent Systems and Applications*, June 2011, pp. 264-267, IEEE.

[9] J. H. An and K. S. Hong, "Finger gesture-based mobile user interface using a rear-facing camera," in *Proc. 2011 IEEE International Conference on Consumer Electronics*, January 2011, pp. 303-304. IEEE.

[10] K. Kano, T. Goto, S. Hirano, and S. L. Phung, March 2019, "Improvement of face image super-resolution by high-precision skin color detection," in *Proc. International Forum on Medical Imaging in Asia 2019*, International Society for Optics and Photonics, vol. 11050, p. 110500F.

[11] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon*, September 2003, vol. 3, pp. 85-92.

[12] A. Kaur and B. V. Kranthi, 2012, "Comparison between YCbCr color space and CIELab color space for skin color segmentation," *International Journal of Applied Information Systems*, vol. 3, no. 4, pp. 30-33.

[13] S. L. Phung, A. Bouzerdoum, and D. Chai, 2005, "Skin segmentation using color pixel classification: analysis and comparison," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 1, pp. 148-154.

[14] J. A. Suykens and J. Vandewalle, 1999, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293-300.

[15] Y. Tang, 2013, *Deep Learning Using Linear Support Vector Machines*, arXiv preprint arXiv: 1306.0239.

[16] Y. Q. Wang, 2014, "An analysis of the Viola-Jones face detection algorithm," *Image Processing On Line*, vol. 4, pp. 128-148.

[17] N. Dalal, 2005, "Histogram of oriented gradients for human detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886-893.

[18] K. W. Chee and S. S. Teoh, "Pedestrian detection in visual images using combination of HOG and HOM features," in *Proc. 10th International Conference on Robotics, Vision, Signal Processing and Power Applications*, Springer, Singapore, 2019, pp. 591-597.

[19] D. Prasanna and M. Prabhakar, 2018, "An effiecient human tracking system using Haar-like and hog feature extraction," *Cluster Computing*, pp. 1-8.

[20] C. Zhang and Z. Zhang, *A Survey of Recent Advances in Face Detection*, 2010.

Mohammed Guhdar received his bachelor degree in computer science from Duhok University in 2013. He is currently a MSc. Student. His area of interests are in mathematic, artificial intelligent and computer vision.

Amera Ismail Melhum received her B.Sc. in 1987 and M.Sc. in 1992 in computer science from Mosul University. Currently she is an assistant prof. at Duhok University. Her research interests include neural network, machine learning, compiler design and image processing.