Pain Detection/Classification Framework including Face Recognition based on the Analysis of Facial Expressions for E-Health Systems

Fatma Elgendy[1], Mahmoud Alshewimy[2], and Amany Sarhan[2]
[1]Kafrelshiekh Higher Institute for Engineering and Technology, Egypt
[2]Computer and Control Engineering Department, Tanta University, Egypt

Abstract: *Facial expressions can demonstrate the presence and degree of pain of humans, which is a vital topic in E-healthcare domain specially for elderly people or patients with special needs. This paper presents a framework for pain detection, pain classification, and face recognition using feature extraction, feature selection, and classification techniques. Pain intensity is measured by Prkachin and Solomon pain intensity scale. Experimental results showed that the proposed framework is a promising one compared with previously works. It achieves 91% accuracy in pain detection, 99.89% accuracy in face recognition, and 78%, 92%, 88% accuracy, respectively, for three levels of pain classification.*

1. Introduction

Some patients have difficulties in reporting their level of pain, either due to language difficulties, impairment or disability. Automatic pain detection can assist medical staff in accessing the pain level of these individuals. The experience of pain is often represented by changes in facial expression. Evidence of pain that is available from facial expression has been the subject of scientific research. It is found that face is a powerful biometric that conveys information about a person's feeling, age and gender classification. Therefore, analysis of face features and expressions has been studied extensively, see for example [6, 22]. Nowadays, it is used in automatic pain estimation systems in the E-health domain.

Automatic pain detection is based on the analysis of spontaneous face expressions by coding different movements of the face muscles with different intensity levels, forming a Facial Action Coding System (FACS) [22]. Prkachin and Solomon developed a pain metric to measure the intensity of pain, which is known as Prkachin and Solomon Pain Intensity (PSPI) [16]. The PSPI metric sums the intensities of four Action Units (AU). They are orbital tightening, levator contraction, lip raising, and eye closure to compute the amount of pain according to the following Equation [16]:

$$Pain = AU4 + \max(AU6, AU7) + \max(AU9, AU10) + AU43 \qquad (1)$$

AU4, AU6, AU7, AU9, and AU10 is rated from 0-5, where 0 is not present and 5 is more intense. AU43 is 0 or 1,where 0 is present and 1 is not. Therefore, the pain scale range from 0 to 16. Each image in the UNBC-Master Shoulder database is hand labeled by its PSPI.

Automatic pain detection consists of three stages: preprocessing, feature extraction and classification. Preprocessing includes face detection, image alignment and image enhancement by removing of noise and filtering. Feature extraction includes extracting the feature from the detected face by one of the two categories: appearance-based or geometric-based techniques [1, 22]. Appearance-based category is based on texture extraction from the face image and training a set of classifiers for the proper expression [1]. This category includes Gabor filter [13] and Log-Normal filters [10]. Geometric-based category measures the deformation of key face points such as eyebrows, lips, and chin from the natural state. Active Appearance Model (AAM) is an example of geometric category [1].

Ashraf *et al*. [1] used AAMs to explore various face features for recognizing facial expressions of pain and Support Vector Machine (SVM) for classifying pain versus no pain enabling 82% accuracy for pain detection. Lucey *et al*. [16] used AAM and SVM and reached accuracy of 78% for AU detection and 84.7% for pain detection. Zafar and Khan [30] used geometric features and K-Nearest Neighbor (KNN) as a classifier with a subset of AUs, which resulted in 87.4% accuracy for pain detection and 84.02% accuracy for classification of AUs. In [10, 22], pain recognition algorithms were designed based on appearance models, such as Gabor filters, Gaussian filters, and Log-Normal filters. Shier and Yanushkevich [22] used Gabor features and SVM to classify pain versus no pain

instances and no pain, weak pain, and strong pain, and reached 74% accuracy for pain detection and 74%, 30%, and 78% accuracy for no pain, weak pain, and strong pain, respectively.

Irani *et al*. [11] separated steering filters into spatial and temporal domains in feature extraction stage, and measured the energies released by the facial muscles that are active during the pain process to classify the pain, and reached an accuracy of 77%, 62%, and 70% for no pain, weak pain, and strong pain, respectively. Hammal and Cohn [10] used Log-Normal filters and SVM and introduced 73% accuracy for pain detection. Singh *et al*. [24] used Speedup Robust Features (SURF) and Scale Invariant Feature Transform (SIFT) for feature extraction and SVM for classification to attain 87% accuracy for classification of four-level pain intensity.

Khan *et al*. [12] extracted shape information using Pyramid Histograms Of Orientation Gradients (PHOG) and appearance features by Pyramid Local Binary Pattern (PLBP), to obtain an accuracy of 96.4% by SVM and 96.9% by 2NN classifiers for pain detection.

Rathee and Ganotra [19] proposed the use of Thin Plate Spline (TPS) mapping for modeling the deformation of facial features and Distance Metric Learning (DML) method to measure the distance between features belonging to different levels of pain; they obtained 96% accuracy for pain intensity detection using SVM.

Pedersen [18] proposed a discriminative feature extractor resembles traditional auto-encoder for automatic pain detection by training with a combined loss function that balances the reconstruction error and the classification error, SVM as a classifier, and had 86.1% accuracy for pain detection. Rodriguez *et al*. [20] trained a VGG-16 Convolutional Neural Network (VGG-16 CNN) for a pain level estimation and Long Short Term Memory (LSTM) as Recurrent Neural Network (RNN) to exploit the temporal relation between frames to have 83.8% accuracy for normalized score and 90.3% for unbalanced score.

Bargshady *et al*. [3] fine-tuned VGG face in feature extraction and Joint RNN network in classification to get 75.2% accuracy for four pain levels. They also applied the fine-tuned pre-trained CNN to extract the features from face images followed by Principal Component Analysis (PCA) to reduce these features and developed Enhanced Joint Hybrid CNN-BiLSTM (EJHCNN-BiLSTM) as a classifier to obtain 90% accuracy by using 10-fold CV and 85% by using Leave one subject out [2]. A 3D spatiotemporal convolutional network is fine-tuned to train the target database and get 98.5% Area Under Curve (AUC) in [28]. A summary of the most relevant work is listed in Table 1.

Pain detection and classification using face images or videos still face many challenges which include the accuracy of detection and classification, the similarity between pain and no pain images, interference between pain expression with other expressions like smiling, gender and age differences.

Table 1. Summary of relevant work.

Author	Feature descriptor	Classifier	Performance Measure	Pain levels	Accuracy
Ashraf *et al*. [1] (2009)	AAM	SVM	Leave One Out CV	2	82%
Lucey *et al*. [16] (2011)	Modified AAM	SVM	Leave One Out CV	2	84.7%
Zafar and Khan [30] (2014)	Geometric features	KNN	Leave One Out CV	2	87.4%
Shier and Yanushkevich [22] (2016)	Gabor features	SVM	Leave One Out CV	2	74%
				3	74% no pain 30% weak pain 78% strong pain
Irani *et al*. [11] (2015)	Separable steerable filters	Facial muscles energies	Leave One Out CV	3	77% no pain 62% weak pain 70% strong pain
Hammal *et al*. [10] (2012)	Log-Normal filters	SVM	Leave One Out CV	2	73%
Singh *et al*. [24] (2013)	SURF+SIFT	SVM	Leave One Out CV	4	87%
Khan *et al*. [12] (2013)	PHOG+PLPB	SVM	10 folds CV all the database	2	96.4%
		2NN		2	96.9%
Rathee and Ganotra [19] (2015)	TPS+DML	SVM	Leave One Out CV	16	96%
Pedersen [18] (2015)	Discriminative feature extractor	SVM	Leave One Out CV	2	86.1%
Rodriguez [20] (2017)	VGG-16 CNN	LSTM	Leave One Out CV	2	83.8%
Bargshady *et al*. [3] (2019)	Fine-tuned VGG face	Joint RNN network	Leave One Out CV	4	75.2%
Bargshady *et al*. [2] (2020)	fine-tuned pre-trained CNN	(EJHCNN-BiLSTM	Leave One Out CV	4	85%
			10 folds CV		90%

So, our main motivation is to develop a framework for pain detection/classification with high accuracy compared to the previously mentioned work and utilize the ideas and algorithm presented in many of them.

The framework consists of four stages. Firstly, face detection that is established using the well-known Viola-Jones algorithms [27]. Secondly, Gabor filter [13] is used in features extraction because it can take care of different shapes, sizes and smoothness levels in the image. Thirdly, the extracted features are then optimized using Relieff filter [14] followed by the Self-Adaptive Differential Evolution technique [7] to select the most important features. Finally, face recognition, pain detection and classification are accomplished via two powerful classifiers and suited for our application; Adaboost methods [8, 15, 25] and KNN [23]. The rest of the paper is organized as follows. Section 2 presents the proposed framework including its details. The experimental results are given in section 3. The conclusion of the work and some of the identified future work are presented in section 4.

2. Proposed Framework

In this work, we introduce a framework for pain detection/classification that uses a combination of different techniques in face detection, features extraction, and features selection in addition to KNN [23] and Adaboost classifiers [8, 15, 25]. This combination improves the accuracy of pain detection and classification in comparison with the previously published work according to the experimental results which will be discussed in section 3. Another contribution is the usage of optimal technique to determine the most appropriate features for the classification process. This selection of features speeds up the system performance with less computation time and high accuracy for their distinguished ability. Finally, the usage of UNBC database in this paper for the first time in face recognition which was used for different purposes in the previously published work [1, 10, 16,17, 22, 24, 30].
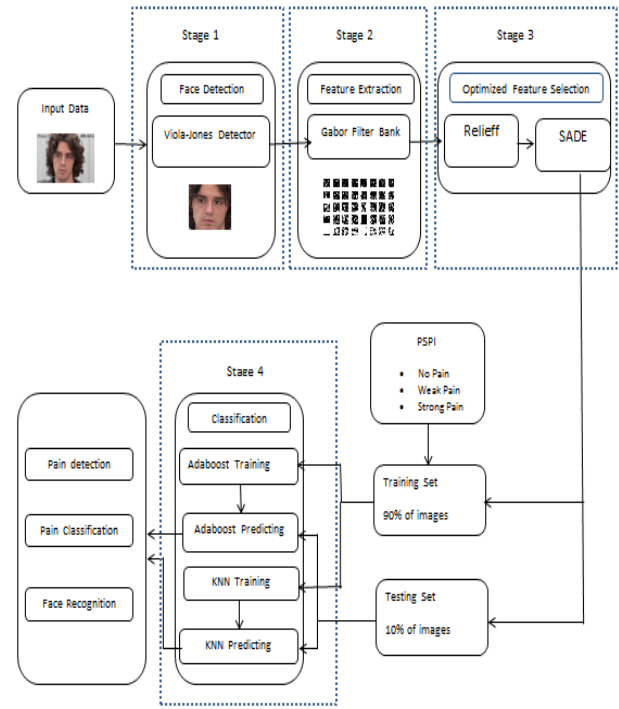

Figure 1. The proposed framework.

The proposed framework consists of:

1. Face detection by Viola Jones algorithm [27].
2. Feature extraction by Gabor filter [13].
3. Feature selection using Relieff-SADE [31].
4. Recognition and classification by Adaboost [8, 15, 25] and KNN [23] classifiers, individually, which include face recognition, pain detection and pain classification, as depicted in Figure 1.

Face detection determines the face location within an image. In this work, we rely on a pattern detection technique and associated statistical analysis and machine learning, specifically the Viola-Jones algorithm [27]. This is a widely used algorithm in real-time applications because it is simple, accurate, fast, its training rate is very high, and the result is more accurate than PCA and ANN [4, 27]. Face detection was implemented according to the steps mentioned in [27].

For feature extraction, Gabor filter was used because it has a solid mathematical base, high face recognition results in addition to being and invariant to rotation, illumination, translation, and scale [21, 27]. It is a special band pass filter that relies on a Gaussian envelope carried by a complex sinusoid [9, 13] represented as [1]:

$$g(x,y,\lambda,\theta,\Psi,\sigma,\gamma) = \frac{1}{2\pi\sigma^2} exp(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}) exp(i2\pi \frac{x'}{\lambda} + \Psi) \quad (2)$$

Where;
x and y denote the spatial domain coordinates
λ is the frequency
θ is the orientation
Ψ is the phase offset of sinusoid wave

σ is the standard deviation of the round Gaussian function

γ represents the spatial aspect ratio

$x' = x \cos\theta + y \sin\theta$

$y' = -x \sin\theta + y \cos\theta$

$exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2})$ is the Gaussian envelope

In this work, Ψ was set to 0 for simplifying Equation (1) by setting the offset of the sinusoid at 0 and set σ to 1 [22]. Five frequencies and 8 orientations were used to create a bank of 40 filters because it achieved the highest recognition rate among the different combinations of orientations and frequencies used through the experiments as in [21]. These filters were applied separately as in [22] and the output filtered images were concatenated to produce the final image. Thereby, each pixel in the image has 40 features and the total features number is equal to the number of pixels multiplied by 40.

The extracted features are entered in the optimized features extraction algorithm which is presented as Algorithm 1:

Algorithm 1: optimized features selection algorithm.

Set weight of all features W[A]=0

For all A from 1 to m

Calculate weight using Eq. (2)

End for

Remove all features weighted below 0.02

List ranked features in (M×N)

Initialize SADE parameters (population size=N, no. of features=1000, no. of generation=N, fitness function=KNN)

Apply SADE algorithm

End

The features extracted from Gabor filter were ranked using the Relieff filter [31]. Relieff filter has many characteristics that made it suitable to our work; it is fast, not limited by data types, fairly noise-tolerant, and unaffected by feature interaction. After applying the filter, the features below a threshold level of 0.02, which is the weight of unimportant features greater than 30% of overall features as in [31], were removed. The remaining features were subjected to a Self-Adaptive Differential Evolution algorithm (SADE), which searches for the best 1000 features giving the highest accuracy when used in the classification step individually. Only 1000 features were chosen to decrease the computation time required for classification without losing the accuracy which surpassed the results in previous work.

The Relieff filter [14, 31] weights the features using a distance criterion that measures how instances from the same class and others from other classes are distinguished. The weighted features are rearranged based on their weights and those below a certain threshold are removed. Features with the highest importance (i.e., weight) are selected to be introduced to the classifier.

The process starts with randomly choosing instances (i.e., features), then finding the nearest hit (data point from the same class) and the nearest misses (data point from the different class), and finally calculating the weight based on relevance using Equation (3) [5]:

$$w_i = \sum_{j=1}^{N}(x_i^j - nearmmiss(x^i)_j)^2 - \left((x_i^j - nearhit(x^i)_j)^2\right) \quad (3)$$

Where;

w_i is the weight of the ith features, x_i^j is the value of the ith feature for point x^j, N is the total numbers of data points, nearhit x^j and near miss x^j are the nearest data point to x^j in the same and different class, respectively.

The Differential Evolution algorithm (DE) [7, 26] was selected to choose the best features in this work because it is simple, easy to use, able to handle any cost function, has good convergence properties, and parallelizable to cope with computation intensive cost functions. For more details on its implementation, see ref [7].

In the proposed work, two distinct powerful algorithms; Adaboost methods and KNN, were selected for classification for their accuracy introduced in previous work and to compare their results with each other. Adaboost, short for "Adaptive Boosting", is a meta-algorithm that forms a strong classifier from many weak learners [8, 29]. It weights weak learners by computing the error resulted from each in classification to specify the correct classifiers with less error [4, 8, 22].

The classification stage includes three phases: face recognition, pain detection, and pain classification. In Face detection and pain classification, Adaboost M2 [8], Bag [25], Subspace [25], and Total boost were used. However, pain detection is a binary classification problem (pain/no pain); thus, Adaboost M1 [8], Logitboost [15], Totalboost and Gentleboost [15] were used as binary classifiers.

KNN creates a model based on training dataset and predicts new data by searching training data for the k-most similar cases [23]. It strongly retains all observations selected at the time of training. This prediction data case of k-most similar cases is recapitulated and returned as the forecast for a new case. The selection of distance metric functions for finding similarity measure depends on structure of data. Available functions in the literature are Euclidean, Manhattan, Chessboard, and Cosine and we found that it is appropriate to use *ED* as *ED* between two pixels, features elements, *p* (*x,y*) and *q(s,t)* in image is computed by:

$$ED(p,q) = \sqrt{(x-s)^2 + (y-t)^2} \quad (4)$$

Where *ED* represents the straight-line distance between these two pixels.

3. Experimental Results

Experiments were performed to analyze the performance of the proposed framework. It was performed using Matlab R2016a on Core i7 laptop with 8 GB RAM and Windows 10.1. There are three sets of conducted experiments: pain detection, pain classification, and face recognition to compute the accuracy of each one.

The proposed framework performance was evaluated using the UNBC-Master Shoulder database [1, 10, 16, 17, 22, 30]. This database is composed of 200 video sequences for 25 subjects who were suffering from shoulder pain. The database contains the PSPI values for each frame which were computed manually [17]. Two from the 25 subjects did not have any pain images; therefore, their images were removed from the database before using it. Each video frame was normalized to 96×116 pixels size from 240×320 pixels size for more less computation without losing accuracy. Figure 2 shows some of the images found in the UNBC-Master Shoulder database.

a) No pain with PSPI=0.   b) Weak pain with PSPI=1.   c) Strong pain with PSPI=6.

Figure 2. Facial expressions from UNBC-Master database.

For pain detection, the database was divided into 2 groups: pain and no pain images; 500 images with no pain and 500 images with pain were chosen from the database. Each subject had equal distribution images collected from different videos. The chosen images from the same frame were taken across video time-equally distribution. The number of features extracted from the Gabor filter was 407,040, and were fed to the Relieff filter which reduced them to 42,384 features. The SADE algorithm searches the top 1000 features in the pain classification stage. Training and test data were selected randomly: for each 10-instances, 9 instances were assigned for training and one for testing.

The obtained accuracy of pain detection using KNN classifier was 91% with k=1 and Euclidean distance as the distance criterion. For the binary Adaboost methods, the obtained accuracies are depicted in Figure 3. The figure shows that the highest accuracy was 89% using Logitboost with 150 classifiers.

For the pain classification phase, 500 images with no pain (PSPI=0), 500 images with weak pain (0<PSPI<=3), and 500 images with strong pain (PSPI>3) were chosen from the database with fair distribution on all subjects. The features extracted from the Gabor filter (407,040 features) were fed to the Relieff filter, which reduced them to 63,178 features. Then, the SADE algorithm chooses the best 1000 features to use in the training and classification processes. KNN and multiple Adaboost classifiers were used in the classification stage.
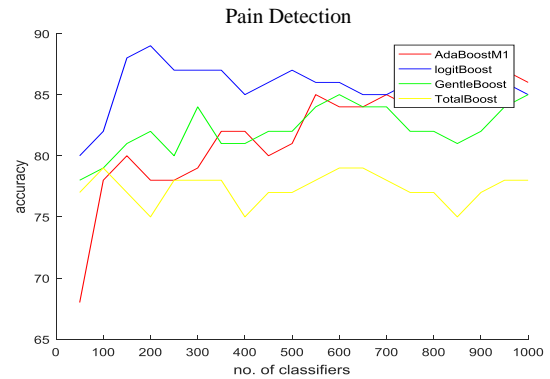

Figure 3. Pain detection accuracy.

The classification process includes: Bag, Subspace with KNN and Subspace with Discriminant, and Totalboost. The best obtained accuracies were 76%, 77%, and 92% for no pain, weak pain, and strong pain, respectively when using Bag algorithm with 200 classifiers as depicted in Figure 4. However, when KNN was used as a classifier, the resulted accuracies were 78%, 92%, and 88% for no pain, weak pain, and strong pain, respectively.

It is obvious that Bag and KNN algorithms results exceeded those of Subspace with KNN, Subspace with Discriminant, and Totalboost, where Subspace with KNN has obtained an accuracy of 60%, 56%, and 62%, Subspace with Discriminant has obtained 78%, 6%, and 74% accuracy results, while Totalboost had 48%, 80%, and 42% accuracy for no pain, weak pain, and strong pain, respectively with 200 classifiers.

Forty images of each subject were taken from the database for face recognition. The features extracted using the Relieff filter were 274,619 features chosen from the overall 407,040 features extracted from the Gabor filter. The SADE algorithm specified the 1000 best features which were used in the classification phase.

We have tested several Adaboost algorithms that classify multiple classes namely; Subspace with KNN, Subspace with Discriminant, and Bag in addition to KNN algorithm to choose the best of them. It is obvious from Figure 5 that the highest face recognition rate among them was 98.91% obtained by Bag algorithm. However, the recognition rate resulted from using KNN alone was 99.89% using k=1 and Euclidean distance search method which exceeding the accuracy of Bag.

As pain detection and pain classification were evaluated using this database by various previous work, we can thus compare our results with theirs.

Table 2 shows the accuracy resulted from pain detection using the proposed framework and some of the previously published ones.

From the results, it is obvious that our framework achieves the best accuracy compared to other works

and it is competitive enough to all of them. However, it is important to demonstrate that using KNN for pain detection introduces better results than using Logit boost algorithm since KNN is suited for large training data and robust for noisy data.

a) Bag.

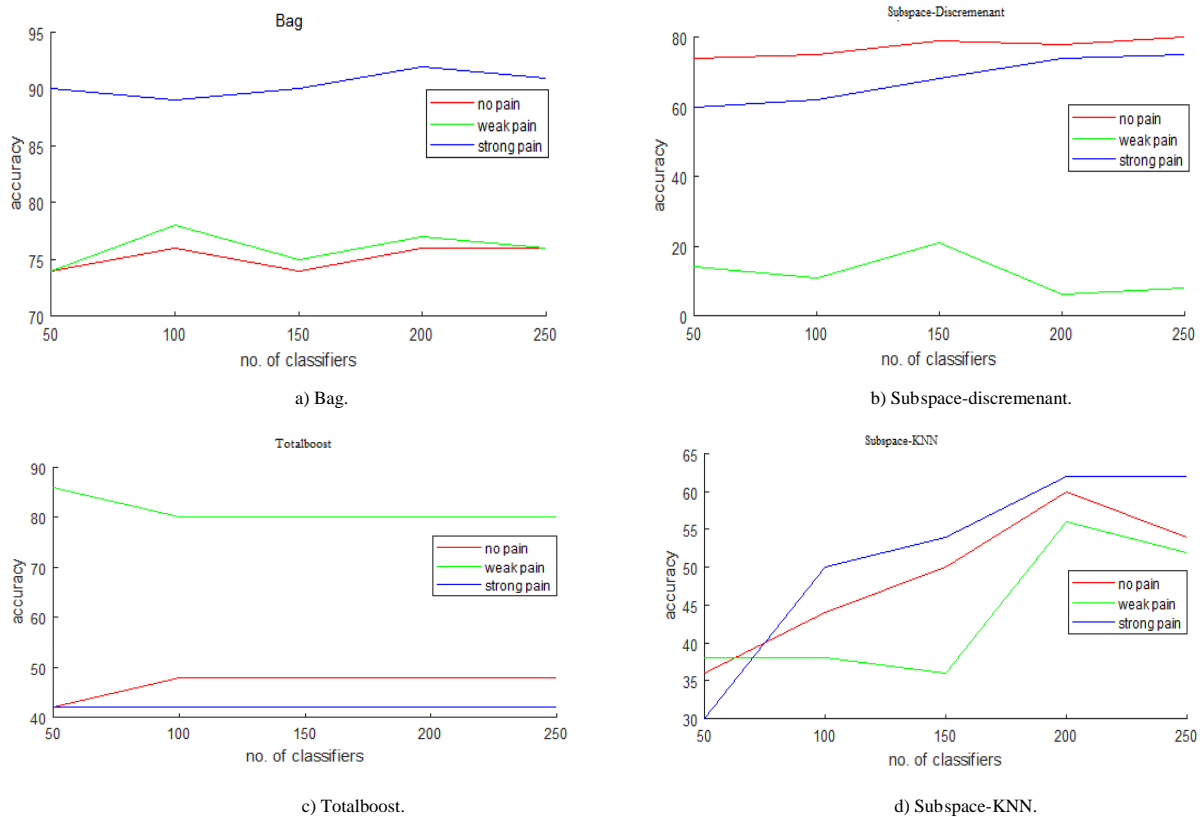b) Subspace-discremenant.

c) Totalboost.

d) Subspace-KNN.

Figure 4. Pain classification accuracy for the various algorithms.
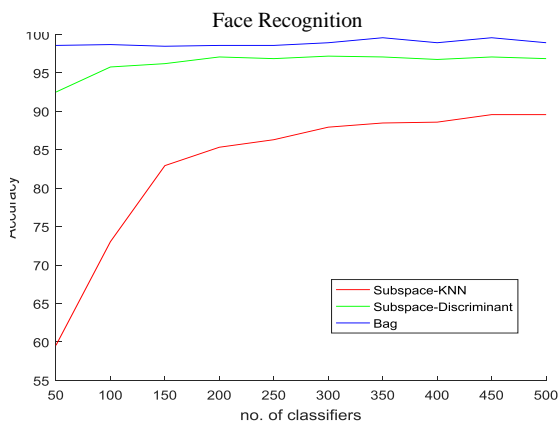

Figure 5. Face recognition accuracy using several Adaboost algorithms.

Table 2. Comparison between the proposed framework and previously published works for pain detection.

Authors	Algorithm	Accuracy
Ashraf *et al.* 2009 [1]	AAM + SVM	79.5%
Hammal *et al.* 2012 [10]	Log-normal filter + SVM	73%
Shier *et al.* 2016 [22]	Gabor filter + SVM	74%
Our framework using Logitboost	Gabor filter + Relieff-SADE + Logitboost	89%
Our framework using KNN	Gabors filter + Relieff-SADE + KNN	91%

Table 3 shows a comparison between the proposed framework and some previous works for pain classification. From these results, the proposed framework comes out with promising results and outperforms other works. It is also noticed that the proposed framework using KNN for pain classification introduces better results than using Bag algorithm in no pain and weak pain results. However, Bag outcomes KNN in strong pain results.

Table 3. Comparison between the proposed framework and previous systems of pain classification.

Authors	Algorithm	No pain	Weak pain	Strong pain
Hammal *et al.* 2012 [10]	Log-normal filter + SVM	65%	36%	70%
Irani *et al.* 2015 [11]	Separable steerable filters	77%	62%	70%
Shier *et al.* 2016 [22]	Gabor filter + SVM	74%	30	78%
Proposed framework using Bag	Gabor filter + Relieff-SADE + Bag	76%	77%	92%
Proposed framework using KNN	Gabor filter + Relieff-SADE +KNN	78%	92%	88%

4. Conclusions

In this paper, an automatic pain detection/classification

framework was proposed based on Gabor filter for feature extraction, Relieff and self-adaptive differential evolution for optimized feature selection, and Adaboost and KNN algorithms for classification. The proposed framework shows promising results compared to previously published works, and outperforms all of them. It has accomplished 91% accuracy in pain detection and 76%, 77%, and 92% in pain classification with the three-levels of pain (no pain, weak pain, and strong pain), respectively.

Future work can focus on enhancing the feature selection stage while choosing various important features using the SADE algorithm or any other feature selection algorithm. This framework must to be tested using real-time applications instead of using the database to assess the recognition rate and its integration.

References

[1] Ashraf A., Lucey S., Cohn J., Chen T., Ambadar Z., Prachin K., and Solomon P., "The Painful Face-Pain Expression Recognition Using Active Appearance Models," *Image and Vision Computing*, vol. 27, no. 12, pp. 1788-1796, 2009.

[2] Bargshady G., Zhou X., Deo R., Soar J., Whittaker F., and Wang H., "Enhanced Deep Learning Algorithm Development to Detect Pain Intensity from Facial Expression Images," *Expert Systems with Applications*, vol. 149, 2020.

[3] Bargshady G., Soar J., Zhou X., Deo R., Whittaker F., and Wang H., "A Joint Deep Neural Network Model for Pain Recognition from Face," *in Proceedings of IEEE 4th International Conference on Computer and Communication Systems*, Singapore, pp. 52-56, 2019.

[4] Chehrehgosha A., Emadi M., Boda R., and Priyadarsini M., "Face Detection and Tracking Using KLT and Viola Jones," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 23, pp. 13472- 13476, 2016.

[5] Capela N., Lemaire E., and Baddour N., "Feature Selection for Wearable Smartphone-based Human Activity Recognition with Able bodied, Elderly, and Stroke Patients," *PlosOne*, vol. 10, no. 4, 2015.

[6] Chehrehgosha A. and Emadi M., "Face Detection Using Fusion of LBP and Adaboost," *Journal of Soft Computing and Applications*, vol. 2016, pp. 1-10, 2016.

[7] Ghosh A., Datta A., and Ghosh S., "Self-Adaptive Differential Evolution for Feature Selection in Hyperspectral Image Data," *Application of Soft Computing Journal*, vol. 13, no. 4, pp. 1969-1977, 2013.

[8] Freuend Y. and Schapire R., "Experiments with a New Boosting Algorithm," *in Proceedings of the*
13th International Conference on Machine Learning, Bari, pp.148-156, 1996.

[9] Gupta A. and Goel L., "Heuristic Approach for Face Recognition using Artificial Bee Colony Optimization," *in Proceedings of the Intelligent System Technologies and Applications*, Jaipur, pp. 209-223, 2016.

[10] Hammal Z. and Cohn J., "Automatic Detection of Pain Intensity," *in Proceedings of the 14th ACM International Conference on Multimedia Interaction*, Santa Monica, pp. 47-52, 2012.

[11] Irani R., Nasollahi K., and Moeslund T., "Pain Recognition using Spatiotemporal Oriented Energy of Facial Muscles," *in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Boston, pp. 80-87, 2015.

[12] Khan R., Meyer A., Konik H., and Bouakaz S., "Pain Detection Through Shape and Appearance Features," *in Proceedings of the IEEE International Conference on Multimedia and Expo*, San Jose, pp.1-6, 2013.

[13] Lajevardi S., "Structural Similarity Classifier for Facial Expression Recognition," *Signal, Image and Video Processing*, vol. 8, pp. 1103-1110, 2014.

[14] Kononenko I., "Estimating Attributes: Analysis and Extensions of Relieff," *in Proceedings of the International Conference on Machine Learning*, Catania, pp. 171-182, 1994.

[15] Kumar B., "Boosting Techniques in Rarity Mining," *International Journal of Advanced Research in Computer Science and Software Engineering*, no. 2, pp. 27-35, 2012.

[16] Lucey P., Cohn J., Matthews I., Lucey S., Sridharan S., Howlett J., and Prkachin K., "Automatically Detecting Pain in Video Through Facial Action Units," *IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 41, no. 3, pp. 664-674, 2011.

[17] Lucey P., Cohn J., Prachin K., Solomon P., and Matthews I., "Painful Data: The UNBC-McMaster Shoulder Pain Expression Archive Database," *in Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition and Workshops*, Santa Barbara, pp. 57-64, 2011.

[18] Pedersen H., "Learning Appearance Features for Pain Detection Using the UNBC-McMaster Shoulder Pain Expression Archive Database," *in Proceedings of International Conference on Computer Vision Systems*, Copenhagen, pp. 128-136, 2015.

[19] Rathee N. and Ganotra D., "A Novel Approach for Pain Intensity Detection Based on Facial Feature Deformations," *Journal of Visual Communication and Image Representation*, vol. 33, pp. 247-254, 2015.

[20] Rodriguez P., Cucurull G., Gonz`alez J., Gonfaus J., Nasrollahi K., Moeslund T., and Roca F., "Deep Pain: Exploiting Long Short-Term Memory Networks for Facial Expression Classification," *IEEE Transactions on Cybernetics*, no. 99, pp. 1-11, 2017.

[21] Satange D., Alsubari A., and Ramteke R., "Composite Feature Extraction based on Gabor and Zernike Moments for Face Recognition," *IOSR Journal of Computer Engineering*, pp. 17-23, 2017.

[22] Shier W. and Yanushkevich S., "Pain Recognition and Intensity Classification Using Facial Expression," *in Proceedings of The International Joint Conference on Neural Networks*, Vancouver, pp. 3578-3583, 2016.

[23] Singh A. and Pandey B., "Diagnosis of Liver Disease Using Correlation Distance Metric Based K-Nearest Neighbor Approach," *in Proceedings of International Symposium on Intelligent Systems Technologies and Applications*, Jaipur, pp. 845-856, 2016.

[24] Singh S., Tiwari S., Abidi A., and Singh A., "Prediction of Pain Intensity using Multimedia Data," *Multimedia Applications and Tools, Springer*, vol. 76, no. 18, pp. 19317-19342, 2013.

[25] Skurichena M. and Duin R., "Bagging, Boosting and the Random Subspace Method for Linear Classifiers," *Pattern Analysis and Applications*, Springer, vol. 5, no. 2, pp. 121-135, 2002.

[26] Storn R. and Price K., "Differential Evolution- a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[27] Soni L., Datar A., and Datar S., "Viola-Jones Algorithm Based Approach for Face Detection of African Origin People and Newborn Infants," *International Journal of Computer Trends and Technology*, vol. 51, no. 2, pp. 75-81, 2017.

[28] Tavakolian M. and Hadid A., "A Spatiotemporal Convolutional Neural Network for Automatic Pain Intensity Estimation from Facial Dynamics," *International Journal of Computer Vision*, vol. 127, no. 10, pp. 1413-1425, 2019.

[29] Tuysuzoglu G. and Berant D., "Enhanced Bagging (eBagging): A Novel Approach for Ensemble Learning," *The International Arab journal of Information Technology*, vol. 17, no. 4, pp. 515-528, 2019.

[30] Zafar Z. and Khan N., "Pain Intensity Evaluation through Facial Action Units," *in Proceedings of the 22nd International Conference on Pattern Recognition*, Stockholm, pp. 4696-4701, 2014.

[31] Zainudin M., Sulaiman Md., Mustapha N., Perumal T., Nazre A., Mohamed R., and Abd Manaf S., "Feature Selection Optimization using Hybrid Relieff with Self-Adaptive Differential Evolution," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 3, pp. 21-29, 2017.

Fatma Elgendy was born in Kafrelshiekh, Egypt, in 1983. She received the B. Sc and M. Sc in Computer Engineering from the Faculty of Engineering, Tanta University, in 2005, and 2014, respectively. She is working as Assistant lecturer in the Department of Computer Engineering and Automatic Control, Kafrelshiekh Higher institute for Engineering and Technology, Egypt. Her interests are in the area of: Image processing, Object Recognition, Cryptography, Healthcare, and IOT applications.

Mahmoud Alshewimy was born in Tanta, Egypt, in 1977. He received the M.Sc. degree in computer engineering and automatic control from Tanta University (Egypt) in 2006 and Ph.D. from Istanbul University in 2014. He is working as Associate Professor in the Department of Computer Engineering and Automatic Control, Tanta University, Egypt. His research interests include Software/Hardware Co-design, Object Recognition & Image Processing, and IOT applications.

Amany Sarhan, received the B.Sc degree in Electronics Engineering, and M.Sc. in Computer Engineering from the Faculty of Engineering, Mansoura University, in 1990, and 1997, respectively. She awarded the Ph.D. degree as a joint research between Tanta Univ., Egypt and Univ. of Connecticut, USA. She is working now as a Full Prof. and head Computers and Control Dept., Tanta Univ., Egypt. Her interests are in the area of: Network, Distributed Systems, Image and video processing, GPU and Distributed Computations.