Journal of Engineering
Sciences Assiut University
Faculty of Engineering
Vol. 48, No. 6
November 2020
PP. 1051-1071

FACE RECOGNITION FROM SMALL DATASETS USING KERNEL SELECTION OF GABOR FEATURES

Alyaa A.S. Gad-Elrab[1], Yasser F. O. Mohammad[1], Moumen T. El-Melegy[1]

[1] *Electrical Engineering Department, Faculty of Engineering, Assiut University, Egypt*

Received: 10 September 2020; Revised: 9 November 2020; Accepted: 9 November 2020

ABSTRACT

Recent advances in face recognition are mostly based on deep learning methods that require large datasets for training. For smaller datasets, we propose a method that combines Gabor feature extraction and aggressive kernel selection to achieve low error rates while keeping computational cost at a minimum. The paper compares the proposed method against traditional feature selection approaches in terms of the recognition accuracy and model compression and show that the proposed method can achieve the same or higher accuracy with significantly lower computational cost. Moreover, we evaluated combining multiple feature selection algorithms to derive our proposed kernel selection method achieving an error rate of 0.025 on the Yala face dataset.

Keywords: Face recognition, Gabor wavelet transform, feature selection, KNN, Neural Network.

1. Introduction

Face recognition [3] is one of the most relevant applications of image analysis. Recent advances in the field achieve human-like recognition capability using deep learning methods which in turn require large amounts of training data.
Moreover, pre-trained deep models typically require color information. This paper focuses on the problem of face recognition from a small training dataset of low-quality grayscale images. The input of a face recognition system is always an image or video stream.
The output is an identification or verification of the subject or subjects that appear in the image or video. And due to its nonintrusive and natural characteristics, face recognition (FR) has been the prominent biometric technique for identity authentication and has been widely used in many areas, such as military, finance, public security and daily life. From the early 1990s until late of 2012 traditional methods attempted to solve FR problem by one- or two-layer representation, such as filtering responses or histogram of the feature codes. The research community studied intensively to separately improve the preprocessing, local descriptors, and feature transformation, which improve face recognition accuracy slowly. By the continuous improvement of a decade, “shallow" method only improve the accuracy of the LFW (Labeled Faces in the Wild) [2] benchmark to about 95%, which indicates that "shallow" methods are insufficient to extract stable identity feature against unconstrained facial variations. Due to the technical insufficiency, facial recognition systems were often reported with unstable performance or failures with countless false alarms in real-world applications. But all that changed in 2012 when AlexNet [18] won the ImageNet competition by a large margin using a technique called deep learning [17]. Deep learning methods, such as convolutional neural networks, use a cascade of multiple layers of processing units for feature extraction and transformation. They learn multiple levels of representations that correspond to different levels of abstraction. The levels form a hierarchy of concepts, showing strong invariance to the face pose, lighting, and expression changes. For example, the first layer of the deep neural network is somewhat similar to the Gabor feature found by human scientists with years of experience. The second layer learned more complex texture features. The features of the third layer are more complex, and some simple structures have begun to appear such as high-bridged nose and big eyes. In the fourth, the network output is enough to explain a certain facial attribute, which can make a special response to some clear abstract concepts such as smile, roar, and even blue eye. Finally, the combination of this higher-level abstraction represents facial identity with unprecedented stability. In 2014, Deep Face [16] achieved the state-of-the-art accuracy on the famous LFW (Labeled Faces in the Wild) benchmark,

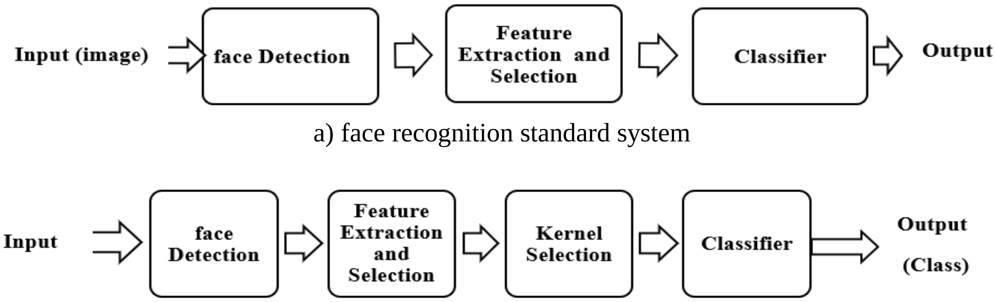
approaching human performance on the unconstrained condition for the first time (Deep Face: 97:35% vs. Human: 97:53%).
Since then, research focus has shifted to deep-learning-based approaches, and the accuracy was dramatically boosted to above 99:80% in just three years [17]. For some applications - like admission control to buildings -- it is possible to control the environment leading to a simpler FR scenario. In such cases, we need a fast method that can easily be implemented in an embedded system. So, we use Gabor-wavelets [1] and notice that Gabor wavelet transformation leads to several relatively similar features which may not be optimal for the classifier in terms of both speed and accuracy. The main contributions of this paper are: Firstly, we evaluate six different feature selection methods in terms of the balance between accuracy and the number of features required. Secondly, we propose "kernel selection" as an alternative to simple feature-selection for face recognition based on the Gabor transform and show that it is capable of achieving error rates as low as 3.4% with a compression rate of 75%. The rest of this paper is organized as follows: Section 2 describes the problem. Section 3 provides details of the face recognition pipeline employed and the proposed solution. Section 4 describes the basic feature selection approaches compared in this study. Section 5 evaluates the proposed system and describes datasets and procedure of algorithms that they performed in our lab and experimental results. The paper is then concluded.

2. Problem Statement

The problem tackled in this paper is achieving accurate face recognition with limited computational resources. By “limited" computational resources we mean low computational power (i.e. memory, CPU ops) during both system training and evaluation. Noted that we are not competing against deep learning systems in term of accuracy, but we provided a middle ground between hand-coded fast feature extraction and learning based deep learning in terms of both speed and accuracy. To achieve this goal, we propose “kernel selection" as the main method to reduce the dimensionality of the classification problem faced by the final classifier in the FR system. Kernel selection is the process of eliminating less important Gabor kernels for classification while keeping the level of accuracy acceptable. Kernel selection differs from traditional feature selection in measuring the value of complete kernels consisting of several features together. Because of its structured nature, Kernel selection has the advantage of eliminating the need to evaluate complete Gabor kernels reducing the computational cost of the system compared with traditional feature selection methods.

3. Proposed Solution

A face recognition system as a standard contains at least three stages: Face detection, feature learning (typically using a deep neural network) and classifier training (typically combined with the second step) - as shown in figure 1A. In this paper we employ a four stages face recognition system: Face Detection, Feature Extraction, kernel selection and classifier training as shown in Figure 1B.



a) face recognition standard system

b) Proposed face recognition system

Figure 1: face recognition stages

3.1 Feature Extraction Using Gabor Transform

Before the widespread utilization of deep learning for face recognition, many FR systems relied on feature extraction using the Gabor transform [1]. The wavelet transform was employed to perform multi-resolution time-frequency analysis. The tunable kernel size results in different time-frequency resolution pair and the size is related to the analytical frequency. For example, smaller kernel size (in time domain) has higher resolution in time domain but lower resolution in frequency domain, and is used for the analysis of fast changes; while bigger kernel size has higher resolution in frequency domain but lower resolution in time domain, and is used for the analysis of slow changes.
The real part of the 2-D Gabor function is defined as:

$$\varphi(x, y) = exp(-\left(\frac{x_r^2+\gamma^2 y_r^2}{2\sigma^2}\right)cos(2\pi\frac{x_r}{\lambda}+\varphi) \qquad (1)$$

$$x_r = x\cos\theta + y\sin\theta \qquad , \qquad y_r = x\sin\theta + y\cos\theta$$

where the arguments x and y specify the position of a light impulse in the visual field and σ, γ, λ, $\theta$and φ are parameters as follows:
In equation (1) ,λ represents wavelength of sinusoidal factor, θ the orientation of the normal to the parallel strips of a Gabor function, φ is the phase offset ,σ

is the sigma/stander deviation of the Gaussian envelope and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function [11].

Finally the parameterφ, which is a phase offset in the argument of the cosine factor in Eq (1), determines the symmetry of the concerned Gabor function: for φ =0 degrees and φ =180 degrees the function is symmetric, or even; for φ =-90 degrees and φ =90 degrees, the function is ant-symmetric, or odd, and all other cases are asymmetric mixtures. As shown in figure 2 observes that Gabor wavelet [1] with 4 scales and 8 orientations and we got 40 kernels.

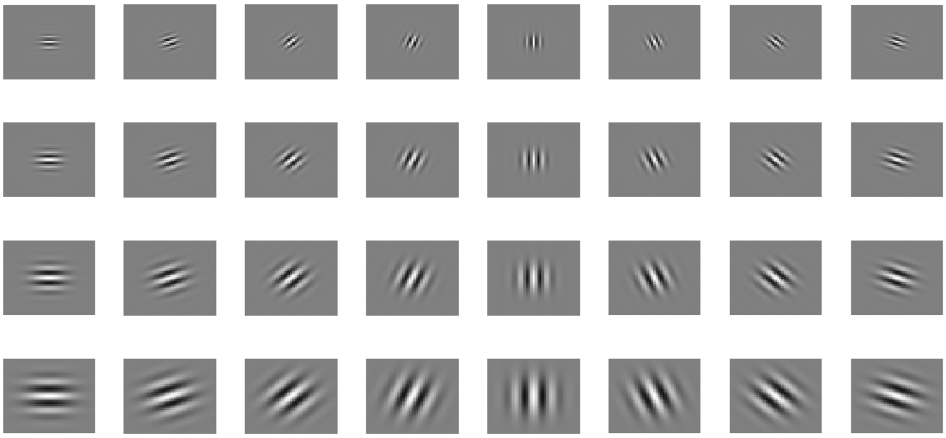


Figure 2: an example of the real part of Gabor wavelets with 4 scales and 8 Orientations

3.2 Feature Selection

A "feature" or "attribute" or "variable" refers to an aspect of the data. Usually before collecting data, features are specified or chosen. Feature selection (FS) [9] methods can be used in data pre-processing to achieve efficient data reduction. This is useful for finding accurate data models. Since exhaustive search for optimal feature subset is infeasible in most cases, many search strategies have been proposed in literature. The usual applications of FS are in classification, clustering, and regression tasks. This review considers most of the commonly used FS techniques. Particular emphasis is on the application aspects. In addition to standard filter, wrapper, and embedded methods, we also provide insight into FS for recent hybrid approaches and other advanced topics.

Ladha and Deepa [15] summarized the advantages of feature selection as:

- It reduces the dimensionality of the feature space, to limit storage requirements and increase algorithm speed.
- It removes the redundant, irrelevant, or noisy data.

- The immediate effects for data analysis tasks are speeding up the running time of the learning algorithms.
- Improving the data quality.
- Increasing the accuracy of the resulting model.
- Feature set reduction, to save resources in the next round of data collection or during utilization.
- Performance improvement, to gain in predictive accuracy.
- Data understanding, to gain knowledge about the process that generated the data or simply visualize the data

Features can be discrete, continuous, or nominal. Generally, features are characterized as [15]:

- **Relevant** these are features which have an influence on the output and their role cannot be assumed by the rest.
- **Irrelevant** features are defined as those features not having any influence on the output, and features could be irrelevant if they are not correlated with the target.
- **Redundant** A redundancy exists whenever a feature can take the role of another (perhaps the simplest way to model redundancy).

High dimensional feature set can negatively affect the performance of pattern or image recognition systems. In other words, too many features sometimes reduce the classification accuracy of the recognition system since some of the features may be redundant or irrelevant. Different combinatorial set of features should be obtained in order to keep the best combination to achieve optimal accuracy. In machine learning and statistics, feature selection, which is also called variable selection, attribute selection or variable subset selection, is the process of obtaining a subset of relevant features (probably optimal) for use in machine model construction. There are lots of techniques available for obtaining such subsets. Some of these techniques include Genetic Algorithm (GA) [7], Mutual information algorithm [14], Univariate feature selection [6, 23], recursive feature elimination [5,6,26] and Feature selection using Embedded method (linear [6,12] and tree [6,8]).

There are three general classes of feature selection algorithms:

Filter Methods: Relying on the characteristics of data, filter models evaluate features without utilizing any classification algorithms [22]. A typical filter algorithm consists of two steps. In the first step, it ranks features based on certain criteria. Feature evaluation could be either univariate or multivariate. In the univariate scheme, each feature is ranked independently of the feature space, while the multivariate scheme evaluates features in a batch way. Therefore, the multivariate scheme is naturally capable of handling redundant features. In the second step, the features with highest rankings are chosen to induce classification models. In the past decade, a number of performance criteria have been proposed for filter-based feature selection such as Fisher score [8], methods based on mutual information [14] as we used in this paper in section 4.1.1 and Univariate feature selection as observe at section 4.1.2. Filter feature selection methods [19, 20] apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the dataset. The methods are often univariate and consider the feature independently, or with regard to the dependent variable. Example of filter methods include the mutual information algorithm (see Section 4.1.1) and the univariate feature selection algorithm (see Section 4.1.2). Filter methods provide superior performance in many applications [27].

Wrapper Methods: wrapper models [19, 20] utilize a specific classifier to evaluate the quality of selected features and offer a simple and powerful way to address the problem of feature selection, regardless of the chosen learning machine. Given a predefined classifier, a typical wrapper model will perform the following steps:

- Step 1: searching a subset of features,
- Step 2: evaluating the selected subset of features by the performance of the classifier,
- Step 3: repeating Step 1 and Step 2 until the desired quality is reached.

Wrappers consider feature subsets by the quality of the performance on a modelling algorithm, which is taken as a black box evaluator. Thus, for classification tasks, a wrapper will evaluate subsets based on the classifier performance (e.g. SVM [5]), while for clustering, a wrapper will evaluate subsets based on the performance of a clustering algorithm. The evaluation is repeated for each subset, and the subset generation is dependent on the search strategy, in the same way as with filters. Wrappers are much slower than filters in finding sufficiently good subsets because they depend on the resource demands of the modelling algorithm. The feature subsets are also biased towards the modelling algorithm on which they were evaluated (even when using cross-validation). Therefore, for a reliable generalization, it is necessary that both an independent validation sample and another modelling algorithm are used after the final subset is found. On the other hand, it has been empirically proven that wrappers obtain subsets with better performance than filters because the subsets are evaluated using a real modelling algorithm. Practically any combination of search strategy and modelling algorithm can be used as a wrapper, but wrappers are only feasible for greedy search strategies and fast modelling algorithms such as linear SVM [5]. Example of wrapper methods include genetic algorithms at section 4.2.1 and recursive feature elimination at section 4.2.2.

Advantages and disadvantages of wrapper algorithms are

- Wrapper models obtain better predictive accuracy estimates than filter models. However, wrapper models are very computationally expensive compared to filter models. It produces better performance for the predefined classifier since we aim to select features that maximize the quality therefore the selected subset of features is inevitably biased to the predefined classifier.
- Slow execution: must train a classifier for each feature subset (or several tanning's if cross validation is used)
- Lack of generality: the solution lacks generality since it is tied to the bias of the classifier used in the evaluation function.
- Ability to generalize: Since they typically use cross-validation measures to evaluate classification accuracy, they have a mechanism to avoid over fitting.
- Accuracy: Generally, achieve better recognition rates than filters since they find a proper feature set for the intended classifier.

Embedded Methods: Filter models select features that are independent of the classifier and avoid the cross-validation step in a typical wrapper model, therefore they are computationally efficient. Wrapper models utilize a predefined classifier to evaluate the quality of features and representational biases of the classifier are avoided by the feature selection process. However, they have to run the classifier many times to assess the quality of selected subsets of features, which is very computationally expensive. Embedded Models [19] embedding feature selection with classifier construction, have the advantages of wrapper models - they include the interaction with the classification model and (2) filter models - they are far less computationally intensive than wrapper methods. Examples of Embedded Methods are L1 (LASSO) regularization as observe in section 4.3.1 and decision tree as observe in section 4.3.2.

3.3 Proposed Kernel Selection

- In standard feature selection, all features are treated similarly by the feature selection algorithm. Applying this naively to the problem of selecting Gabor features for an image, will lead in most cases to selecting features that are scattered on all the kernels which entails that all kernels must be computed then the features not selected by the algorithm are discarded. This has an obvious computational cost that defeats the purpose of using Gabor features to speed up the process of feature recognition for small devices not capable of modern deep learning.
- In this paper we propose a structured method of selection in which a complete

- Gabor kernel is either kept or discarded (including all its features).
- This proposed approach has a clear advantage in real-time speed as the discarded kernels need not be computed at all (rather than being computed and partially discarded in the naive approach).
- In section 4, we compare several feature selection methods and evaluate their relative performance in a face recognition task with a small training dataset.

3.4 Classification:

We employ a simple K-nearest neighbors (KNN) [13] as our primary classifier. This is one of the simplest possible classifiers. It just stores the all the training samples. When asked to classify a new sample, it finds the distance between this sample and all stored samples, finds the nearest k stored samples and classifies the new sample as belonging to the same class as the majority of these k nearest samples. This simple classifier was chosen to make sure that the accuracy of the complete system depends mostly on the quality of feature extraction and selection (the main focus of this paper). Moreover, we compare the KNN [13] classifier with a shallow neural network (NN) [10] showing that there is no significant difference in accuracy between the two classifier which lends support to our claim that the proposed feature extraction/selection methodology can produce good enough features that the simplest of classifiers can achieve acceptably good performance.

One of the main disadvantages of KNN [13] is that its computational cost increases linearly with the size of the training set (in terms of classification time, and storage requirement). Nevertheless, this paper focuses on small-datasets for which this is a non-issue. For large datasets, existing deep learning-based approaches can be employed.

4. Feature Selection Algorithms

This section describes the basic feature selection approaches compared in this study. We describe two method of filter methods in section 4.1, we describe two method of wrapper methods in section 4.2 and we describe two methods of embedded methods in section 4.3.

4.1 Filter Methods

4.1.1 Mutual Information Algorithm (MI):

Mutual information algorithm (MI) [14] is one of the information metrics used to measure the relevance of features considering the higher order statistical structures existing in the data. Pedregosa et al [6]. define mutual information as the amount of information shared by two variables. For variables X and Y, it is computed as:

$$\mathrm{MI(X, Y)} = -\sum_{\mathrm{y}\in\mathrm{Y}}\sum_{\mathrm{x}\in\mathrm{X}}\mathrm{p(x, y)log(p(x, y)/p(x)p(y))} \qquad (2)$$

Where p (x, y) gives the joint probability of X and Y random variables and p(x), p(y) are the probability density functions of variable X and Y respectively.

A large value of MI signifies high correlation of two variables. Zero value indicates that two variables are not correlated. Conditional MI is defined as the amount of information shared by two variables when the third is known.

The conditional MI between variables X and Y given Z is computed as:

$$\mathrm{MI(X, Y \parallel Z)} = \mathrm{H}_c\mathrm{(X \parallel Z)} - \mathrm{H}_c\mathrm{(X \parallel Y, Z)} \qquad (3)$$

This gives the information added by Y about X which is not contained in Z. Where $\mathrm{H}_{\boldsymbol{c}}$the entropy of X variable is after observing the values of another variable Z is called conditional entropy. MIFS (Mutual Information based Feature Selection) that utilized MI to reduce the number of features. Pedregosa et al [6] suggested that a good set of features are not relevant individually but also non redundant with respect to each other. That means features should be highly correlated with target class variable and not be correlated with each other [14]. The evaluation function used for selection of feature subset was

$$EvalFunc = MI(X_n, Y) - B^* \sum_{k=1}^{n} MI(X_n, Y_k) \qquad (4)$$

Where $\boldsymbol{X_n}$, $\boldsymbol{Y_k}$ are the input variables (features),the first factor of this expression gives the feature relevance and second factor measures the penalty for correlation of the feature with each other. Here, β is a parameter, to be determined empirically that varies between 0 and 1, that controls how important is mutual information between features. Features are selected by first ranking them using their value then selecting either a predetermined fraction of them or the ones with a value above some predetermined threshold. With small β, many correlated features are selected.

4.1.2 Univariate Feature Selection:

In the univariate scheme, each feature is ranked independently of the feature space, while the multivariate scheme evaluates features in a batch way. Therefore, the multivariate scheme is naturally capable of handling redundant features. In the second step, the features with highest rankings are chosen to induce classification models [6, 23].

According to ranking method we used Select Best function [6] that need Chi squared stats of non-negative features for classification tasks.

A choice of feature selection ranking methods depending on the nature of [20]:

- the problem (dependencies between variables, linear or nonlinear relationships between variables and target)
- the available data (number of examples and number of variables, noise in data)
- The available tabulated statistics.

Advantage of this type of feature selection is computational and statistical scalability. Disadvantage of univariate method is redundant subset: same performance could possibly be achieved with a smaller subset of complementary variables that does not contain redundant features.

4.2 Wrapper Methods

4.2.1 Genetic Algorithm:

Genetic Algorithms (GA) [7] can be defined as population-based and algorithmic search heuristic methods that mimic natural evolution process of man. GA iteratively employ the use of one population of chromosomes (solution candidates) to get a new population using a method of natural selection combined with genetic functional such as crossover and mutation. In comparative terminology to human genetics, chromosomes are the bit strings, gene is the feature genotype is the encoded string, and phenotype is the decoded genotype. The fitnesses of the chromosomes are evaluated using a function commonly referred to as Objective function or fitness function (in other words, the fitness function (objective function) reports numerical values which are used in ranking the chromosomes in the population. In this paper we use the fitness function as follow:

$$\mathrm{Fit} = \mathrm{KNN\ classifier\ accuracy} + \left(1 - \frac{\mathrm{NF}}{\mathrm{Nn}}\right) \tag{5}$$

Where NF is number of features selected by algorithm and Nn is number of features in dataset. The five important issues in the GA are chromosome encoding, population initialization, fitness evaluation, selection and criteria to stop the GA. The GA operates on binary search space as the chromosomes are bit strings. The GA manipulates the finite binary population in similitude of human natural evolution. First, an initial population is created randomly and evaluated using a fitness function. As regards binary chromosome used in this work, a gene value ’1’ indicates the feature indexed by the position of the ’1’ is selected. If it is ’0’, the feature is not selected for evaluation of the chromosome concerned.

In this paper we used 10 generation for 1725 chromosomes for Genetic feature selection algorithm. Each row is a chromosome containing genes valued as either 0 or 1. The chromosomes are then ranked and based on the rankings, the top n fittest kids are selected to survive to the next generation. After the elite (these children are given pushed automatically into the next generation) individuals are moved to the next generation, the remaining individuals in the current population are used to produce the rest of the next generation through crossover and mutation. Crossover is basically, combination of two individuals to form a crossover kid. Mutation operator on the other hand, depicts a genetic perturbation of the genes in each chromosome through flipping of bits depending on the mutation probability. Following the previous steps until algorithm reached to 10 generation as we initialized in this paper, the steps involved in using the GA for feature selection are explained in this section.

The crossover operator in the GA genetically combines two individuals (parents) to form children for the next generation. Two parent’s chromosomes are needed to carry out crossover operation. Two chromosomes are taken from tournament selection.

The mutation is an operator which allows diversity. During the mutation stage, a chromosome has a probability $\mathbf{p_{mut}}$ to mutate. If a chromosome is selected to mutate, we choose randomly a number n of bits to be flipped then n bits are chosen randomly and flipped. To create a large diversity, we set $\mathbf{p_{mut}}$ around 10% and n ∈ [1, 5]. In Selection step, we implement a probabilistic binary tournament selection. Tournament selection holds n tournaments to choose n individuals. Each tournament consists of sampling 2 elements of the population and choosing the best one with a probability p ∈ [0.5, 1].

4.2.2 Recursive Feature Elimination:

Recursive feature elimination [6, 5] use Basic Backward Selection algorithm [24] firstly fits the model to all of the features. Each feature is then ranked according to its importance to the model. Let S be a sequence of ordered numbers representing the number of features to be kept (S1 > S2 > S3...). At each iteration of feature selection algorithm, the Si top raked features are kept, the model is refit and the accuracy is assessed. The value of Si with the best accuracy is assessed and the top Si features are used to fit the final model.

Describe subsequent steps of this procedure are Basic Recursive Feature Elimination Train the model using all features after that it determine model’s accuracy then it determine feature’s importance to the model for each feature and each subset size Si , i = 1. . . N, it keep the Si most important features and it remove others the train the model using Si features after that determine model's accuracy then Calculate the accuracy profile over the Si and determine the appropriate number of features finally it use the optimal subset used to train the final model. Corresponding to the optimal Si Model building process is composed of few successive steps and feature selection is one of these. Due to that, we used resampling methods (e.g. cross-validation [25]) that contribute to this process when calculating model's accuracy. It has been showed that improper use of resampling when measuring accuracy can result in model's poor performance on new samples. Note that Recursive feature elimination [6] with cross-validation: in this paper we used a recursive feature elimination with automatic tuning of the number of features selected with cross-validation to detect over fitting.

4.3 Embedded Methods

4.3.1 L1-Based Feature Selection Algorithm:

L1 (LASSO (Least Absolute Shrinkage and Selection Operator)) [12] regression for generalized linear models are one of the simplest ways to predict output using a linear function of input features.

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \cdots \dots . . + w[n] * x[n] + b \tag{6}$$

In the equation (6) above, we have shown the linear model based on n features x[n]. Linear regression looks for optimizing *w* and *b* such that it minimizes the cost function. The cost function can be written as

$$\sum_{i=1}^{M} (\mathrm{y}_i - \hat{y_i})^2 = \sum_{i=1}^{M} (y_i - \sum_{j=1}^{p} w_j * x_{ij})^2 \tag{7}$$

Where $\mathbf{y}_i$ are desired output values and equation (7) assumes the data-set has M instances and p features Using linear regression on a data-set divided into training and testing sets can give us a rough idea about whether the model is suffering from over-fitting or under-fitting [6].

4.3.2 Tree Based Feature Selection Algorithm:

This algorithm that summaries training data in the form of a decision tree. Along with systems that induce logical rules, decision tree algorithms have proved popular in practice. This is due in part to their robustness and execution speed, and to the fact that explicit concept descriptions are produced, which users can interpreter. Nodes in the tree correspond to features, and branches to their associated values [8].

5. Evaluation

This section describes the dataset used in this experiment, the evaluation procedure employed and the results we obtained.

5.1 Dataset

We used a subset of the Extended Yale Face Database B [4] with 50 different persons. The dataset contains images with different facial expressions making it a challenging dataset. Each person had between 3 and 134 images, with an average of 34.5 images per person (std. dev =0.21). Total number of images in database equal 1725 image all grayscale and of low resolution (50*50 pixels). All images represented faces detected in natural images as shown in figure 3.

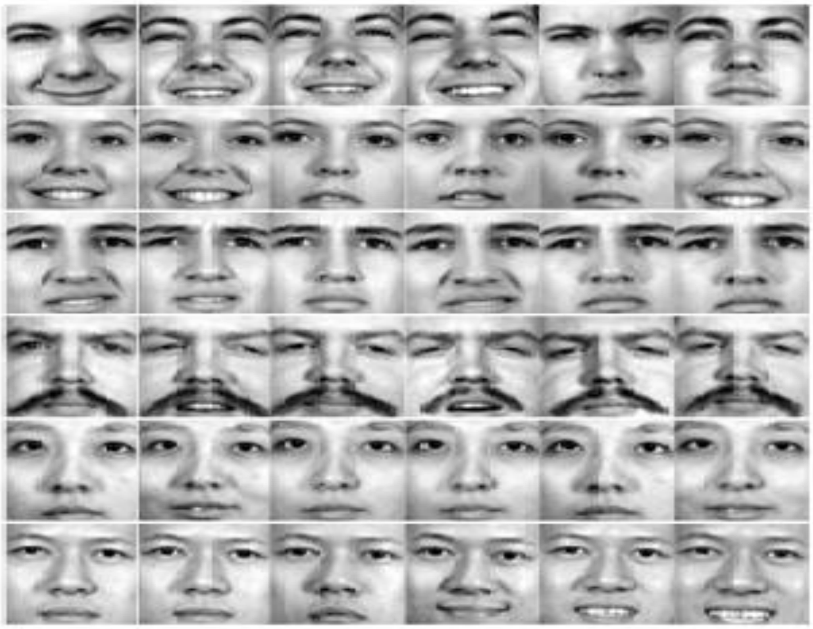


Figure 3: Examples of some faces in data set

5.2 Evaluation Criteria

Two evaluation criteria were used in this paper: Error Rate (ER) defined as the number of incorrect results divided by the number of test cases (i.e. 1 - accuracy), and the Compression Ratio (CR) defined as one minus the fraction of kernels that need to be calculated.

5.3 Procedure

- Feature extraction was performed on all the images in the database using Gabor wavelets (See Equation 1) with 5 scales and 8 orientations (40 kernels). This resulted in a total of 100,000 features per image. Some of the Gabor features had almost no variance (the exact threshold was 0.1). These features were removed leading to a set of 38,031 features.
- The data was split randomly into a 1207 (representing 50 people) and 518 testing samples (representing 50 people).
- Feature Selection using the six methods mentioned in section 4 were performed with the results shown in (Table 1).
- We performed conservative kernel selection by keeping all kernels from which even a single feature was selected by the FS algorithm (Table 1).
- We performed aggressive kernel selection by keeping the top 10 kernels based on the number of selected features by each algorithm (See Table 2 and Figure 4)

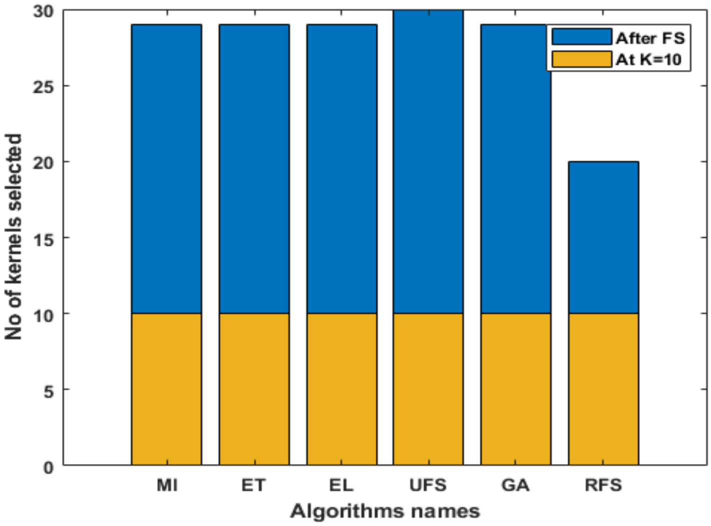


Figure 4: Number of kernels selected from each algorithm after FS and KS=10

- Figure 4 shows that number of kernels selected by Recursive feature elimination with cross validation had the least value of the six algorithms equal only 20 kernels, so it had the highest CR.
- We run the KNN classifier on the features and kernels selected by every algorithm and we calculated Error Rate (ER) as shown in (Figure 5) that it observes error rate for all of six algorithms after Feature selection and kernel selection.

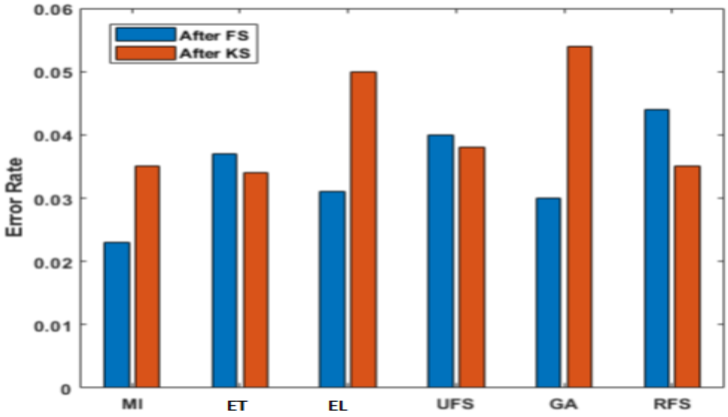


Figure 5: ER Calculated after FSAs and KSA

- Figure 5 shows that the error rate resulted from MI algorithm had the least value after both of feature selection and kernel selection.
- We used simple counting to fuse the results of different feature selection algorithms keeping only the features that were selected i times where (1≤ i ≤ 6), and for each i times we found kernels for features that were selected. Then we used these features for each i times to calculate accuracy and error rate as shown in Table 3.

5.4 Experimental Results

This paragraph reports analysis of our results. Fig. 6 shows the mutual information between features and the target (in descending order). It is clear that relevance of features for predicting the target class drops considerably after the first few thousand features.

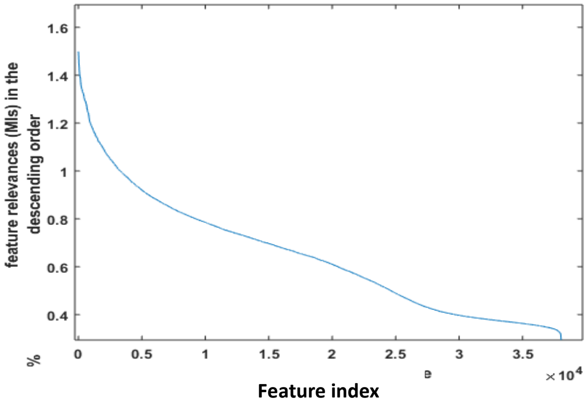


Figure 6: MI between features and the target

Fig. 7 shows the total feature relevance of all features with each kernel in descending order. The figure shows that the first few kernels carry most of the information about the target justifying our procedure of kernel selection.

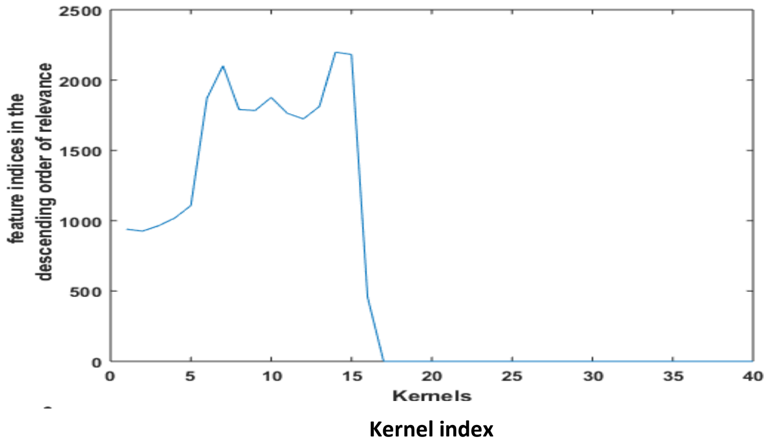


Figure 7: The total feature relevance of all features with each kernel in descending order

Table 1 shows that, out of the six feature selection algorithms employed in this study, MI achieves the lowest error rate (highest accuracy) with an ER of 0.023, while the tree-based Embedded achieved the highest compression ratio (0.956) with a corresponding error rate of only 0.037 (just 48% higher than MI). An important point to note in Table 1 is that the features selected are distributed among more than half the kernels (CR ≥0.5) for all algorithms.

Table 1: Error Rate, and Compression Ratio for Conservative Kernel Selection

Algorithm Name	Number of Features	Number of Kernels	ER	CR FS	CR KS
MI	27000	29	0.023	0.29	0.275
Embedded (Tree)	1655	29	0.037	0.956	0.275
Embedded (Linear)	11423	29	0.031	0.7	0.275
UFS	32000	30	0.040	0.16	0.25
GA (10 Generations)	18426	29	0.03	0.52	0.275
RFS(CV)	7607	20	0.044	0.8	0.5

Table 2 shows the results of the proposed aggressive kernel selection. Here we fix the number of selected kernels to 10 for all algorithms. The tree-based embedded method now achieves the lowest error rate (0.034) with the highest compression ratio (98%).

Table 2: Error Rate, and Compression Ratio for Aggressive Kernel Selection

Algorithm Name	Number of Features	Number of Kernels	ER	CR FS	CR KS
MI	13764	10	0.035	0.64	0.75
Embedded (Tree)	835	10	0.034	0.98	0.75
Embedded (Linear)	8012	10	0.05	0.79	0.75
UFS	18301	10	0.038	0.79	0.75
GA (10 Generations)	10981	10	0.054	0.71	0.75
RFS(CV)	4537	10	0.035	0.88	0.75

Comparing the results of Table 1 and Table 2 shows that aggressive kernel selection can achieve a similar accuracy to standard feature selection for most methods while providing a higher effective compression ratio (this comparison is shown in Fig. 4).

Table 3 shows the results of fusing the six-feature selection method by simple counting. As expected, the accuracy and compression ratios improve with increased threshold for feature selection (achieving a best error rate of 0.025 and CR of 0.96 by using KNN and achieving also best error rate of 0.027 and CR of 0.775 by using NN) with a threshold of 4. Beyond this limit, higher thresholds led to better compression ratios at the expense of higher error rates. Taken together, these results show that aggressive kernel selection using a constant kernel threshold (10 in our case), or combination using counting of feature selection results lead to higher compression ratios that translate to faster processing without loss of recognition accuracy. Aggressive kernel selection using a fixed threshold led to a minimum error rate of 0.34 while fusion led to the best results in this experiment (error rate of 0.025).

Table 3: The performance after fusing results from different feature extractors using AND operation.

Number of used algorithms	Number of Features	Number of kernels	ER by KNN	CR FS by KNN	CR KS by NN	ER by NN
6	11	3	0.63	0.99	0.925	0.56
5	253	5	0.058	0.99	0.875	0.042
4	1661	9	0.025	0.96	0.775	0.027
3	5345	9	0.03	0.86	0.775	0.039
2	11552	13	0.035	0.7	0.675	0.056
1	19582	16	0.035	0.37	0.6	0.06

6. Conclusions

We proposed the use of kernel selection as an alternative to simple feature-selection for face recognition based on Gabor transform. Then, the selected features are used to achieve efficient face recognition for situations in which computational constraints precludes the use of more advanced deep learning methods. Six feature selection methods in the proposed approach for face recognition using Gabor features.

References

[1] Zhang, Qian, et al. "Feature extraction of face image based on LBP and 2-D Gabor wavelet transform." Mathematical Biosciences and Engineering: MBE 17.2 (2019): 1578-1592.

[2] Huang, G., Ramesh, M., Berg, T. and Learned-Miller, E., 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments Univ. Massachusetts, Amherst. MA, Tech. Rep. 07-49.

[3] Umer, Saiyed, Bibhas Chandra Dhara, and Bhabatosh Chanda. "Face recognition using fusion of feature learning techniques." Measurement 146 (2019): 43-54.

[4] Georghiades, A.S., Belhumeur, P.N. and Kriegman, D.J., 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE transactions on pattern analysis and machine intelligence, 23(6), pp.643-660.

[5] Guyon, I., Weston, J., Barnhill, S. and Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine learning, 46(1-3), pp.389-422.

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, pp.2825-2830.

[7] Babatunde, O.H., 2015. A neuro-genetic hybrid approach to automatic identification of plant leaves.

[8] PEH, D. "RO Duda, PE Hart, and DG Stork, Pattern Classification." (2001): 305-307.

[9] Jović, A., Brkić, K. and Bogunović, N., 2015, May. A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1200-1205). IEEE.

[10] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[11] Weilun Chao. Gabor wavelets transform and its application. R98942073 (TFA and WT final project), 2010.

[12] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

[13] Yihua Liao and V Rao Vemuri. Use of k-nearest neighbor classifier for intrusion detection. Computers and security, 21(5):439–448, 2002.

[14] Amiri, F., Yousefi, M.R., Lucas, C., Shakery, A. and Yazdani, N., 2011. Mutual information-based feature selection for intrusion detection systems. Journal of Network and Computer Applications, 34(4), pp.1184-1199.

[15] L Ladha and T Deepa. Feature selection methods and algorithms. International journal on computer science and engineering, 3(5):1787–1797, 2011.

[16] Taigman, Y., Yang, M., Ranzato, M.A. and Wolf, L., 2014. Deep face: Closing the gap to human-level performance in face verification. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).

[17] Ng, Andrew, Jiquan Ngiam, Chuan Yu Foo, Yifan Mai, Caroline Suen, Adam Coates, Andrew Maas et al. "Deep learning tutorial." Univ. Stanford (2015).

[18] Mark Andrew Hall. Correlation-based feature selection for machine learning. 1999.

[19] Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. Data classification: Algorithms and applications, page 37, 2014.

[20] Guyon I, Gunn S, Nikravesh M, Zadeh LA, editors. Feature extraction: foundations and applications. Springer; 2008 Nov 16.

[21] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. Machine earning, 46(1-3):389–422, 2002.

[22] H. Liu and H. Motoda. Computational Methods of Feature Selection. Chapman and Hall/CRC Press, 2007.

[23] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[24] Giorgos Borboudakis and Ioannis Tsamardinos. Forward-backward selection with early dropping. The Journal of Machine Learning Research, 20(1):276–314, 2019.

[25] Jun Shao. Linear model selection by cross validation. Journal of the American statistical Association, 88(422):486–494, 1993

[26] Max Kuhn and Kjell Johnson. Feature engineering and selection: A practical approach for predictive models. CRC Press, 2019.

[27] Ghosh, Manosij, et al. "A wrapper-filter feature selection technique based on ant colony optimization." Neural Computing and Applications (2019): 1-19.

التعرف على الوجه من مجموعات البيانات الصغيرة باستخدام اختيار KERNEL لميزات GABOR

ملخص عربي

تعتمد التطورات الحديثة في التعرف على الوجوه في الغالب على أساليب التعلم العميق التي تتطلب مجموعات بيانات كبيرة للتدريب. بالنسبة لمجموعات البيانات الأصغر، نحن نقترح طريقة تجمع بين استخراج الميزات باستخدام خوارزميه Gabor واختيار النواة القوي لتحقيق معدلات خطأ منخفضة مع الحفاظ على التكلفة الحسابية عند الحد الأدنى. هذه الورقة تقدم الطريقة المقترحة مع مناهج اختيار الميزات التقليدية من حيث دقة التعرف وضغط النموذج وتوضح أن الطريقة المقترحة يمكن أن تحقق نفس الدقة أو أعلى بتكلفة حسابية أقل بكثير. علاوة على ذلك، قمنا بتقييم الجمع بين خوارزميات اختيار الميزات المتعددة لاشتقاق طريقة اختيار النواة المقترحة الخاصة بنا والتي تحقق معدل خطأ قدره ٠,٠٢٥ في مجموعة بيانات وجه تسمى Yala.