A Scalable Metric Learning-Based Voting Method for Expression Recognition

Shaohua Wan and J.K. Aggarwal
Computer Vision Research Center/Dept. of ECE
The University of Texas at Austin
shaohuawan@utexas.edu aggarwaljk@mail.utexas.edu

***Abstract*— In this paper, we propose a facial expression classification method using metric learning-based k-nearest neighbor voting. To achieve accurate classification of a facial expression from frontal face image, we first learn a distance metric structure from training data that characterizes the feature space pattern, then use this metric to retrieve nearest neighbors from training dataset, and finally output the classification decision accordingly. An expression is represented as a fusion of face shape and texture. This representation is based on registering a face image with landmarking shape model and extracting Gabor features from local patches around landmarks. This type of representation achieves robustness and effectiveness by using an ensemble of local patch feature detector at a global shape level. A naive implementation of metric learning-based k-nearest neighbor would incur a time complexity proportional to the size of the training dataset, which precludes this method being used with enormous dataset. To scale to potential larger databases, an approximate yet efficient variant scheme of ML-based kNN voting is further devised based on Locality Sensitive Hashing (LSH). A query example is directly hashed to the bucket of a pre-computed hash table where candidate nearest neighbors can be found and there is no need to search the entire database for nearest neighbors. Experimental results on Cohn-Kanade database and Moving Faces and People database show that both ML-based kNN voting and its LSH approximation outperform the state-of-the-art, demonstrating the superiority and scalability of our method.**

Keywords- Metric Learning; K-Nearest Neighbor; Gabor feature; Locality sensitive Hashing; emotion recognition

I. INTRODUCTION

Human emotion recognition has long been an actively researched topic in Human Computer Interaction (HCI). Unlike other types of non-verbal communication, human face is truly expressive and facial expressions are closely tied to emotional state. The ability to interpret non-verbal face gestures is key to a wide range of HCI applications. The promising future of emotion-aware machine intelligence has propelled researchers to build computer systems to understand and use this natural form of human communication [3].

An effective representation of facial expression is a vital component of any successful facial expression recognition system. Various models and methods have been proposed to attack this problem. Seeing from a geometric perspective, model-based approaches, as in [8]–[11], iteratively register face with a deformable shape model and capture the holistic geometric variation aspects of an expression. Appearance-based approaches consider the varying pattern of pixel intensities as the distinguishing traits of an expression and design feature detector for local skin patches, examples of which include SIFT [12], Local Binary Pattern [13], Local Directional Pattern [14], etc. To draw on the descriptive power of both shape and texture, [27], [28] use a combination of shape and texture model.

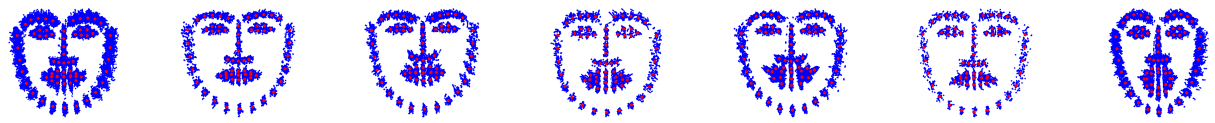
In our method, a hybrid representation of facial expression is also used by fusing a shape model of landmark points and an underlying appearance model of local patches of face images.

A good expression recognition methodology should consider classification as well as representation issues [1]. Donohue et al. [4] used the back-propagation algorithm to train a neural network, and a recognition rate of 85% based on 20 test cases was reported. Kotsia et al. [5] used Support Vector Machine to classify geometric deformation features. In [22], Condition Random Fields are used to model the temporal variations of face shapes and make classification accordingly. Previous methods have demonstrated satisfactory categorization performance on exaggerated expressions. As categorization samples the semantic space more densely and naturally induced expressions are involved, expression classification becomes quite difficult and previous methods are subject to severe accuracy degradation.

We consider facial expression classification in the framework of measuring similarities. While nearest neighbor is a natural choice in this setting, its classification resolution is limited since there exists much overlapping between subtle expressions. Thus, a metric structure that adapts to the feature space embeddings is preferred over the default Euclidean metric as a measure of similarity. Inspired by the success of Metric Learning (ML) in semantic image classification [24], we propose an expression classification method based on ML. In particular, a generalized Mahalanobis distance matrix is learned that satisfies pairwise similarity/dissimilarity constraints on distance between expression feature vectors. Afterwards, a kNN classifier equipped with this distance metric is used to assign majority class label to query expression.

J.K. Aggarwal is with the Faculty of Electrical & Computer Engineering, University of Texas at Austin, 1 University Station C0803 Austin, TX 78712-0240 aggarwaljk@mail.utexas.edu

Shaohua Wan is with the Department of Electrical & Computer Engineering, University of Texas at Austin,1 University Station C0803 Austin, TX 78712-0240 shaohuawan@utexas.edu



(a) Neutral (b) Anger (c) Disgust (d) Fear (e) Hapiness (f) Sadness (g) Surprise

Fig. 1: Face shape of seven basic expressions (including neutral). Red denotes the mean shape of that particular expression.

As far as we know, this is the first time ML has been used for expression recognition. Our ML-based kNN classifier can be used either as an expression detector, where a single category is discriminatively trained against all other categories, or as a multi-class classifier, where the metric structure for multiple expressions are simultaneously learned. The latter approach has the advantage of sharing useful metric structures across different expression categories.

kNN classifier tends to Bayesian optimal as the dataset size tends to infinity [6]. However, the time cost of searching all examples for nearest neighbors would become the bottleneck as more instances are added to the database. To scale up to larger database, an approximate yet fast classification technique based on Locality Sensitive Hashing (LSH) [7] is proposed as a variant of ML-based kNN voting method. This variant achieves efficiency by narrowing search for candidate nearest neighbors down to only items sharing the same hash key in a pre-computed hash table.

The main contributions of this paper are summarized below:

1) Metric learning is employed to train a domain-specific distance metric that is able to capture the inherent embeddings of feature space and yields more accurate and robust expression classification results;
2) An approximation scheme based on LSH is used to speed up the nearest neighbor search process of ML-based kNN, making our method a real-time one and applicable to enormous database;
3) We also perform comparative experimental studies of various expression classification algorithms on two databases. Our ML-based kNN voting method compares favorably in terms of recognition rate, especially when it comes to subtle expressions.

This paper is structured in the following way. Section II will introduce the feature representation of facial expression. We then formulate expression classification as a metric learning problem. In section IV, how ML-based kNN can be sped up via the use of LSH is described. Finally, we present various experimental results and discussions followed by conclusions.

II. Expression Representation

This section outlines the method for representing facial expression. We use a fusion of face shape and texture as the representation of facial expression. This hybrid representation is able to incorporate local pixel intensity variation pattern while still adhering to shape constraint at a global level, proving to be robust and effective. Necessary preprocessing steps prior to constructing such a representation are also described.

A. Face Shape

In our work, a face shape is represented by a set of 68 points known as landmarks. To be invariant to scale, orientation and reference point, Procrutes Analysis is employed to align these landmarks to the mean shape. Example alignment results from the CK+ dataset [20] are shown in Fig. 1. In total, shapes of seven basic expressions (including neutral) defined in [2] are given in seven subfigures [15]. Each subfigure is a superimposition of face shapes of all examples of a specific expression from the CK+ dataset [20]. Red landmarks denote the mean shape.

B. Texture Feature

A number of research works on expression recognition using Gabor features have reported improved recognition rate [16]–[18]. In our work, Gabor features are used as the texture descriptor. In this subsection, the procedures for extracting Gabor features are described.

1) Face Image Normalization: Face appearance can vary greatly among instances of subjects due to skull sizes, lighting conditions, image noise, and intrinsic sources of variability. To minimize geometric and luminance variances, two normalization techniques are applied to raw images before the actual extraction of Gabor features. First, the face image is shifted, scaled and rotated so that the face shape in this image is aligned with the mean shape. Then from this affine-transformed image we calculate self-quotient image to attenuate variation of illumination. This is accomplished by first convolving the transformed image with a Gaussian smooth filter and then dividing it by its smoothed version.

2) Gabor Feature Extraction: A family of Gabor kernel can be expressed as a Gaussian modulated sinusoid in the spatial domain. Our Gabor filter bank consists of filters at 5 scales and 8 orientations.

Since it has been shown that mouth contributes the most to a particular expression, followed by canthus and eyebrows, we crop a total number of 7 patches from the self-quotient image to serve as expression identification regions, as shown in Fig. 2. Gabor filter bank is then applied to these 7 patches respectively, resulting in a Gabor feature vector of dimension 560. To remove redundancy, Principal Component Analysis (PCA) is employed to reduce data dimension to 80 while retaining 98% of energy. Denoting the face shape vector and

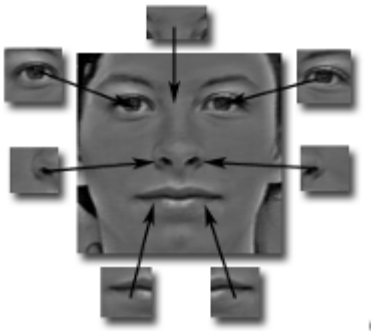


Fig. 2: Image patches where Gabor features are extracted.

the Gabor feature vector as s and g respectively, a particular expression could be represented as a concatenation s of g:

$$x = [s^T, \lambda \cdot g^T]^T$$

where λ is a weighting factor balancing the relative importance of shape and texture. To further reduce data dimension, PCA is performed on x to derive the final representation of facial expression. Without causing confusion, we will still use x to represent facial expression in the later parts of this paper.

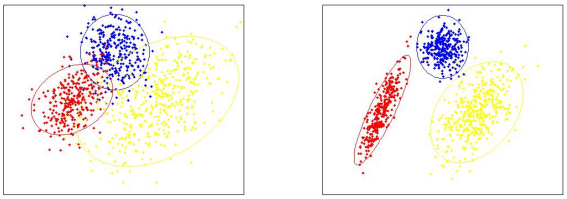
To select proper λ such that s and g are commensurate, we estimate the effect of varying s on g using a similar method in [8]. To do this, we displace s from its ground truth position and the RMS change in g per unit RMS change in s is recorded. The weighting factor λ is set as the inverse of the average value of RMS change of all training example.

III. Distance Metric Learning

Many algorithms in pattern recognition rely on some distance metric for measure of similarity between two objects. A good metric should supply high similarity for objects of the same category, and a low one for those of different categories. L_p norm is a frequently used metric due to its simplicity. Kernel methods can be seen as an attempt to transform default Euclidean geometry with a non-linear kernel operation to a high dimensional feature space and has achieved wide applicability in pattern recognition community. Other methods like Linear Discriminant Analysis seek to project data to subspace that maximizes inter-class variance while keeping intra-class variance as small as possible.

For a kNN classifier, the class label is determined by the consensus of k nearest neighbors. Traditionally, in the absence of prior knowledge on the statistical regularities in data, Euclidean distance is used to measure the dissimilarity between instances. However, as shown by some researchers [25], [26], kNN performance can be significantly improved by exploiting the inherent data embeddings and learning a distance metric accordingly.

Distance metric learning is an emerging method that allows more flexible transformation of feature space so that in the derived feature space, similar examples are closer to each other while dissimilar examples are separated by a large margin. The learning dynamics of distance metric



(a) Before metric learning (b) After metric learning

Fig. 3: Schematic illustration of data distribution before and after metric learning. Class label is denoted by the color. The distance metric is optimized so that similarly labeled data is tightly clustered and differently labeled data is separated by a large margin.

learning are illustrated in Fig. 3. Particularly, given a set of points $X = \{x_1, x_2, ..., x_n\}$, where x_i denotes the feature vector of an expression, we seek a matrix A that parameterizes the (squared) generalized Mahalanobis distance between two expressions x_i and x_j:

$$d_A(x_i, x_j) = (x_i - x_j)^T A (x_i - x_j)$$

In supervised expression classification, the objective is to learn a new distance matrix A regularized by A_0 that satisfies pairwise constraints imposed by label information:

$$\begin{aligned} &\min_{A \geq 0} \; D_{ld}(A, A_0) \\ s.t. \quad &d_A(x_i, x_j) \leq l_{i,j} \quad if (i,j) \in S \\ &d_A(x_i, x_j) \geq u_{i,j} \quad if (i,j) \in D \end{aligned}$$

where $A \geq 0$ requires A to be semi-positive definite, $D_{ld}(A, A_0)$ is the LogDet divergence that regularizes A to be as close to A_0 as possible, S is the set of all similar pairs of training instances, D is the set of all dissimilar pairs of training instances, and $l_{i,j}$ and $u_{i,j}$ are the lower and upper bound for similar and dissimilar pairs respectively.

In our work, Information-Theoretic Metric Learning (ITML) from [19] is used to solve the above optimization problem. A_0 is chosen to be the inverse of the covariance matrix of the training data. For examples from two different categories, $u_{i,j}$ is set as the 80^{th} percentiles of the sample histogram of distances between all dissimilar pairs of these two categories. For examples from the same category, $l_{i,j}$ is set as the 20^{th} percentile of the sample histogram of distances between similar pairs within category. In total, we use 7 lower bounds and 21 upper bounds for a 7-class expression classification problem. To classify an unseen example, k nearest neighbors are first retrieved based on the learned metric, and weighting is further applied to the votes of these k nearest neighbors to determine the final wining expression category.

The algorithmic steps used to perform the ML-based kNN voting are described as a two-stage process. In the very first stage, the bootstrapping stage, the model is contructed and the distance metric is learned. After the training is

finished, our voting method proceeds to stage 2, where the classification decision is made.

Stage 1 Bootstrapping

Input: The training set of face images with different expressions $I = \{I_i\}$, the set of face shapes $S = \{s_i\}$ and the set of expression labels $C = \{c_i\}$, the weighting factor λ to balance the relative importance of face shape and texture.

Output: Distance metric A.

1: Compute mean shape $\overline{m}$ from S in an iterative manner until convergence;
2: For each $s_i \in S$, align s_i to $\overline{m}$ using affine transform $aftr_i$, then transform $I_i \in I$ with $aftr_i$. Still denote the resulting face image set and face shape set as I and S respectively for convenience;
3: For each $I_i \in I$, extract Gabor features g_i;
4: Concatenate s_i and g_i to derive the final representation of expression $x_i = [s_i^T, \lambda \cdot g_i^T]^T$;
5: According to expression label information C, form the set of similar pairs S and the set of dissimilar pairs D and calculate pairwise constraints $u_{i,j}$ and $l_{i,j}$;
6: Optimize the distance metric A with the goal of satisfying pairwise constraints imposed by $u_{i,j}$ and $l_{i,j}$ using ITML algorithm.

Stage 2 Classification

Input: Distance metric A, query facial expression x, the number k of nearest neighbors to extract, weighting sequence $\{w_i(i = 1,2,...,k)\}$ controlling the decay of k votes of the nearest neighbors.

Output: Winning expression label c.

1: For x, calculate its distance to all examples in the dataset using A;
2: Retrieve the k nearest neighbors, rank them according to their similarity to x. Denote the corresponding expression label sequence as $\{c_i^{'}(i = 1,2,...,k)\}$;
3: Denote the score for each expression label as $score_j$ and initialize each score to 0.0. Apply $\{w_i(i = 1,2,...,k)\}$ to the votes of $\{c_i^{'}(i = 1,2,...,k)\}$ to derive the final score for each expression using the following routine:

$$\text{for } c_i^{'} \text{ in } \{c_i^{'}(i = 1,2,...,k)\}$$
$$score_{c_i^{'}} = score_{c_i^{'}} + w_i$$

4: Set c as the one that has the maximum score.

In implementation, k is chosen to be 100 and $\{w_i\}$ is chosen to be geometric progression with a common ratio of 0.9. This makes intuitive sense since the vote of a nearest neighbor with lower ranking should be weighted progressively less.

IV. Scaled ML-based KNN via LSH

A kNN classifier would require linear scan of all examples in the database in order to make a classification, thus being computational expensive. While our ML-based k-NN classifier working at the scale of 10,000 facial images can reach a processing speed of 17fps on a Dell desktop with 3.6GHz Intel Core i7 CPU and 4G memory, this method would virtually become computationally infeasible as a REAL-TIME algorithm as new data-rich collections with more facial images and expression categories are continuously being introduced. To gear toward future large-scale data set, Locality Sensitive Hashing (LSH) [7] is adopted to trade off classification accuracy with computation speed.

The basic idea of LSH is to compute a hash key for each example x in the database using a family of hashing functions $h(x) \in F$ so that similar examples will have a higher probability of collision in the hash table. The hash function $h(x)$ should satisfy the locality sensitive hashing property:

$$Pr_{h \in F}[h(x_i) = h(x_j)] = sim(x_i, x_j)$$

where $sim(x_i, x_j)$ is the similarity between x_i and x_j.

Commonly used similarity function and corresponding hash function family are inner product similarity and random hyperplane projection defined as follows:

$$sim(x_i, x_j) = 1 - \frac{1}{\pi} cos^{-1}\left(\frac{x_i^T x_j}{\| x_i \| \| x_j \|}\right)$$

$$h_r(x) = \begin{cases} 1, & \text{if } r^T x > 0 \\ 0, & \text{otherwise} \end{cases}$$

where r is a random vector drawn from multivariate normal distribution with zero mean and identity variance $N(0, I)$. To estimate similarity between two examples, the hash keys are formed by concatenating the output of l hash functions drawn from F, and the hamming distance between these two keys are calculated.

To account for the effect of the learned metric from previous section, we have adapted the similarity function and hash function to have the following form:

$$sim(x_i, x_j) = 1 - \frac{1}{\pi} cos^{-1}\left(\frac{x_i^T A x_j}{\| x_i L \| \| x_j L \|}\right)$$

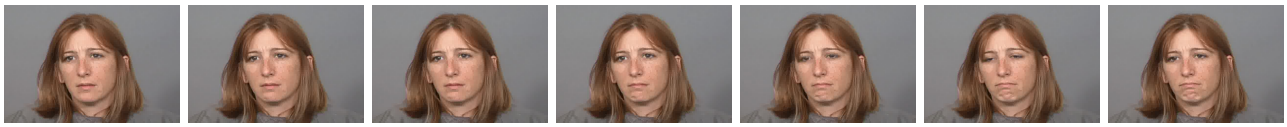
$$h_r(x) = \begin{cases} 1, & \text{if } r^T L x > 0 \\ 0, & \text{otherwise} \end{cases}$$

where L is the Cholesky decomposition of A satisfing $A = LL^T$.

At query time, the vector representing an expression is hashed directly to a specific position in the hash table and all examples that are in the same place as the query vector are returned as similar candidates. The k nearest neighbors are then selected by a linear search through similar candidates. The computational cost is incurred the most when taking the sequence of examples that collided and sorting them by their similarity to the query. Since the range of search for nearest neighbors is significantly reduced, LSH makes possible faster search of high-dimensional feature space.

(a) (b) (c) (d) (e) (f) (g)

Fig. 4: Image sequences from CK+ database showing the formation of sadness from onset to peak.



(a) (b) (c) (d) (e) (f) (g)

Fig. 5: Image sequences from MFP database showing the formation of sadness from onset to peak.

V. Experiments and Results

Our primary goal is to verify that the proposed metric learning method can indeed learn a metric that adapts to the feature embeddings of facial expressions, and improve the hit rate when retrieving nearest neighbors. To this end, we perform a comparative study of several widely used methods, including standard kNN, SVM and LDCRF (LDCRF [15] is reported to give the highest recognition rate for a 7-class expression recognition problem). Experiments show that our method outperforms the state-of-the-art. In particular, we empirically derive the average recognition rate of 5 classification methods on 2 different dataset and contrast the confusion matrix obtained from LDCRF [15] and our ML-based kNN. An interesting plot of the first 3 principal components of facial expressions before and after metric learning further demonstrates the discriminative power of our method on subtle expressions. We use LIBSVM [23] for experiment on SVM and implementation of LDCRF is based on [15].

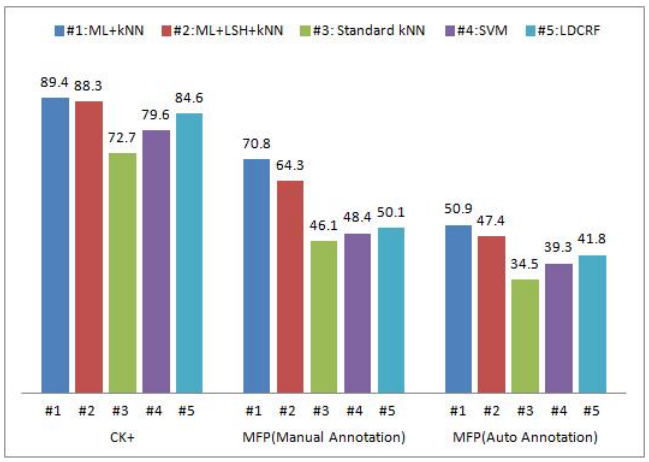


Fig. 6: Average recognition rate of various methods for expression classification (including neutral).

A. Overview of Dataset

We evaluate our algorithm on two datasets, the Extended Cohn-Kanade (CK+) dataset [20] and the Moving Faces and People (MFP) dataset [21]. The first one, CK+, is one of the most widely used test-bed for face analysis algorithms and consists of AU-coded and expression-labeled face images of single persons, taken under relatively controlled viewpoint and illumination conditions. Current state-of-the-art expression recognition systems have saturated in performance on this dataset, and we include evaluation on it for the purpose of comparison of our method against other systems. We also evaluate our algorithm on a more challenging dataset, the MFP dataset. It contains a variety of still images and videos of individuals in natural context. Human expressions exhibited in MFP are all naturally induced by scenes from movies and television programs rather than posed ones, thus being subtle and more difficult to recognize. Fig. 4 and Fig. 5 show two image sequences demonstrating the formation of sadness from onset to peak, with the first from CK+ and the second from MFP. To the best of our knowledge, no previous experimental results of expression recognition on MFP are available, most probably due to the fact that it is a dataset of spontaneous, hard-to-classify facial expressions.

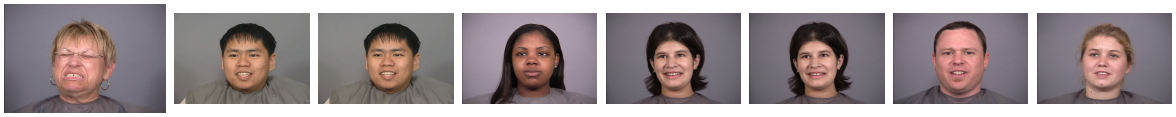
One difference between CK+ and MFP motivates us to take different approaches when evaluating the performance of our method: CK+ carries ground-truth landmarks with itself but MFP does not. Hence, we implement the following three approaches of training and testing: 1) Training and testing on CK+ with ground truth facial landmarks and expression labels provided by the dataset itself; 2) Training and testing on MFP with ground truth facial landmarks and expression labels obtained by manual annotation; 3) Training and testing on MFP with ground truth facial landmarks and expression labels obtained from automatic annotation based on [9]. To maximize the amount of training and testing data, a five-fold cross-validation configuration is used.

B. Experimental Results

1) CK+: CK+ contains 593 sequences from 123 subjects. Out of the 593 sequences 309 were labeled as one of the six basic expressions. Since all the sequences start from the neutral pose to the peak formation of the expression, to train a discriminative model, we split each continuous sequence into two halves: the first half is labeled as neutral, and the

(a) Query Image (b) Dsgt#1 (c) Dsgt#2 (d) Nutr#3 (e) Dsgt#4 (f) Nutr#5 (g) Dsgt#6 (h) Dsgt#7

Fig. 7: Ranked list of nearest neighbors obtained using ML-based kNN along with the groundtruth labels.



(a) Query Image (b) Hpns#1 (c) Hpns#2 (d) Nutr#3 (e) Hpns#4 (f) Nutr#5 (g) Nutr#6 (h) Nutr#7

Fig. 8: Ranked list of nearest neighbors obtained using standard kNN along with the groundtruth labels.

second half is labeled as expressive. As a result, we set up a 7-class classification experiment for CK+.

Average recognition rate of our method as well as the state-of-the-art methods is given in Fig. 6. The confusion matrices of our method and LDCRF are given in Table 1 and Table 2 respectively (LDCRF [15] reports the best performance among the state of the art, so other methods' confusion matrices are not given here).

From the comparison between Table 1 and Table 2, it's clear that ML-based kNN achieves less confusion between subtle expressions such as neutrality, anger, fear, etc, which is exactly the reason why our method outperforms others with a recognition rate of 89.4%. Of course, LDCRF is a probabilistic method of modeling dynamically varying patterns whereas our method tries to uncover the interrelationships between different examples in a static manner. The optimal way for the classification task at hand is to incorporate metric learning into the LDCRF model in a unified framework; this is subject for future research.

TABLE I: Confusion Matrix for 7-Class Expression Classification Using ML-based KNN on CK+

	Ntr	Agr	Dsg	Fer	Hpn	Sdn	Spr
Ntr	**96.7**	0.7	0.0	0.3	1.0	1.3	0.0
Agr	11.1	**84.9**	0.0	0.0	0.0	0.0	0.0
Dsg	15.2	0.0	**83.1**	0.0	1.7	0.0	0.0
Fer	20.0	0.0	0.0	**80.0**	0.0	0.0	0.0
Hpn	4.3	0.0	0.0	0.0	**95.7**	0.0	0.0
Sdn	7.2	0.0	0.0	0.0	0.0	**92.8**	0.0
Spr	5.6	0.0	0.0	0.0	0.0	0.0	**94.4**

TABLE II: Confusion Matrix for 7-Class Expression Classification Using LDCRF on CK+

	Ntr	Agr	Dsg	Fer	Hpn	Sdn	Spr
Ntr	**73.5**	6	1.6	1.9	2.6	9.2	5.2
Agr	20.6	**76.6**	1.1	0.0	1.6	0.0	0.0
Dsg	2.7	6.2	**81.5**	0.0	9.6	0.0	0.0
Fer	0.0	0.0	0.0	**94.4**	0.0	4.2	1.4
Hpn	0.5	1.0	0.0	0.0	**98.6**	0.0	0.0
Sdn	21.5	0.0	0.0	1.3	0.0	**77.2**	0.0
Spr	0.9	0.0	0.0	0.0	0.0	0.0	**99.1**

2) MFP: MFP is a database of static images and video clips of human faces and people. Of more interest to our investigation are the Dynamic Facial Expressions video clips that show spontaneous expressions of subjects watching a 10 minute video clip. There are several drawbacks working directly with these video clips: a) Expressions in each video vary in length. Some occur over a few frames, others may last many seconds; b) These expressions are not verified and subjects may respond to stimulus with a non-intended expression (e.g. aiming at inducing fear but actually getting disgust; c) Some clips contain more than one expression (e.g. a fear expression may be accompanied by a surprise); d) All these clips come with neither landmarks nor expression labels.

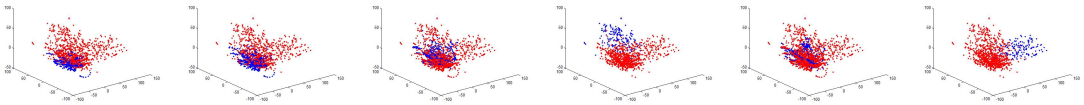
To suit our needs, the following steps are taken to validate the dataset: 1) Manually examine each clip and discard those containing non-intended expressions; 2) For each valid clip, cut it short so that it contains exactly the formation of an expression from onset to peak; 3) Manually annotate frames in each cut-short clip. Table 3 gives detailed statistics about the validated dataset. Note that anger is excluded from our experiment and we only perform a 6-class classification on MFP dataset since we found no valid examples of anger at all.

TABLE III: Statistics of the MFP Dataset after Validation

Expression	Ntr	Agr	Dsg	Fer	Hpn	Sdn	Spr
Number of Examples	374	N/A	96	15	134	30	99

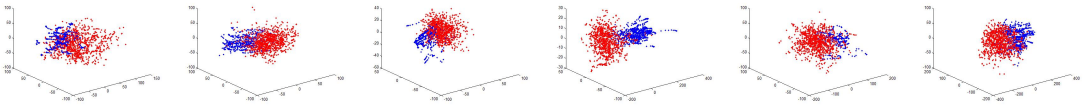
TABLE IV: Confusion Matrix for 6-Class Expression Classification Using ML-based KNN Trained on Manually Annotated MFP

	Ntr	Dsg	Fer	Hpn	Sdn	Spr
Ntr	**94.6**	0.0	0.0	2.1	0.0	0.0
Dsg	40.6	**56.2**	0.0	3.2	0.0	0.0
Fer	80.0	0.0	**20.0**	0.0	0.0	0.0
Hpn	3.5	0.0	0.0	**96.5**	0.0	0.0
Sdn	0.0	0.0	0.0	0.0	**100.0**	0.0
Spr	15.2	0.0	0.0	0.0	0.0	**84.8**



(a) Anger (b) Disgust (c) Fear (d) Hapiness (e) Sadness (f) Surprise

Fig. 9: Plot of the first 3 principal components of different expression vectors BEFORE applying ML-based transformation. Blue denotes the expression of interest. Red denotes all expressions of non-interest.



(a) Anger (b) Disgust (c) Fear (d) Hapiness (e) Sadness (f) Surprise

Fig. 10: Plot of the first 3 principal components of different expression vectors AFTER applying ML-based transformation. Blue denotes the expression of interest. Red denotes all expressions of non-interest.

TABLE V: Confusion Matrix for 6-Class Expression Classification Using LDCRF Trained on Manually Annotated MFP

	Ntr	Dsg	Fer	Hpn	Sdn	Spr
Ntr	**68.6**	1.3	13.4	3.1	4.1	9.5
Dsg	8.4	**81.9**	1.2	0.0	7.7	0.8
Fer	56.0	0.0	**27.5**	10.1	6.4	0.0
Hpn	0.9	1.0	0.0	**98.1**	0.0	0.0
Sdn	25.7	17.3	9.7	6.8	**40.5**	0.0
Spr	15.2	27.7	12.9	4.8	6.8	**32.6**

With a validated MFP dataset, we first test the 5 expression recognition methods with the manually annotated landmarks, i.e. training and testing using approach 2. Furthermore, we also test the 5 expression recognition methods in a real-time setting using the auto face image annotator from [9], i.e. training and testing using approach 3. The average recognition rates of these 5 methods are given in Fig. 6. Confusion matrices for ML-based kNN and LDCRF obtained using approach 2 are given in Table 4 and Table 5 respectively (Confusion matrices obtained using approach 3 are not given here).

Since MFP is a database of spontaneous expressions rather than acted ones, one should not be surprised that recognition rates on MFP decline a lot compared to those on CK+. However, ML-based kNN still gives much better result than other methods when tested on MFP. By careful examination of Table 4 and Table 5, it is again evidenced that our method is conducive to high classification accuracy by being the most successful in removing indiscrimination between dissimilar examples while reinforcing alikeness between similar examples.

We could also see in Fig. 6 that approach 3 yields the lowest recognition accuracy among the three training/testing approaches. This is in part due to the subtlety of expression and in part due to the not-so-good CLM-based auto face registration. For this very reason, we call for improvement on current face registration and alignment techniques.

What is noteworthy is that the recognition rate of LSH approximation of ML-based kNN voting is consistently better than all but ML-based kNN. In our experiment, the bit length of hash key is selected to be 100, for which a frame rate of around 25 fps could be attained. In fact, the length of hash key controls the tradeoff between accuracy and speed. We could further improve the accuracy of LSH approximation by using longer hash key, at the expense of higher computational complexity and lower frame rate. If one chooses to distribute the matching process across several machines, computational complexity should not be an issue. It should be noted that we do not observe any significant performance improvement beyond bit length of 130, where the program runs at a frame rate of 20 fps. The superior performance of the LSH approximation scheme demonstrates scalability of our ML-based kNN voting method and qualifies itself as applicable to real world problems.

3) Visualization of feature embedding transformation: To exemplify the retrieval performance of ML-based kNN, we show in Fig. 7 and Fig. 8 the nearest neighbors retrieved by ML-based kNN and standard kNN respectively given the same query image. As can be seen, ML-based kNN gives quite satisfactory results whereas disgust is completely confused with happiness and neutrality by standard kNN. To visualize the transformation effect of metric learning on feature space, we further show a comparative plot of the first 3 principal components of facial expressions in CK+ before and after applying ML-based transformation in Fig. 9 and Fig. 10 respectively. Much overlap could be observed in Fig. 9 between dissimilar expressions. The overlap is even exacerbated for subtle expressions such as anger, disgust, fear and sadness. In contrast, Fig. 10, which gives PCA plot of expressions after ML-based transformation, shows a much better separated point distribution for differently labeled expressions, consequently facilitating higher recognition accuracy.

VI. Conclusions

We present a new expression classification method using Metric Learning-based k-Nearest Neighbor voting. The metric is optimized with the goal that all similarly labeled inputs have small pairwise distances, while all differently labeled inputs have large pairwise distances. This method alleviates confusion between subtle expressions such as neutral, angry and fear, etc., thus outperforming the state-of-the-art methods. To speed up our method, an approximate yet efficient variant scheme of ML-based kNN voting is further devised based on Locality Sensitive Hashing. LSH allows fast indexing of similar examples with the help of a precomputed hash table and significantly accelerates the nearest neighbor matching process.

Experiments show that ML-based kNN demonstrates better classification especially when it comes to subtle expressions. Also, our LSH approximation scheme gives superior classification performance than the state-of-the-art, and more importantly, works at a faster speed, demonstrating the scalability and capability of our method.

VII. Acknowledgements

The authors would like to thank Alice O' Toole for making available the Moving Faces and People dataset, and Changbo Hu for making valuable comments to this paper. This work is partially supported by Instituto de Telecomunicacoes and the UT Austin/Portugal Program CoLab grant (FCT) UTA-Est/MAI/0009/2009 (2009) supported by the Portuguese government.

References

[1] Chengjun Liu and Harry Wechsler. Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition. IEEE Trans. Image Processing, vol. 11, no. 4, pp. 467-476, 2002.

[2] Ekman P., Friesen W.V. Unmasking the Face: A guide to recognizing emotions from facial clues. Consulting Psychologists Press 1975.

[3] Dornaika, Fadi and Bogdan Raducanu. "Facial Expression Recognition for HCI Applications." Encyclopedia of Artificial Intelligence. IGI Global, 2009. 625-631. Web. 6 Jul. 2012. doi:10.4018/978-1-59904-849-9.ch095.

[4] B.A. Donohue, J.D. Bronzino, J.H. DiLiberti, D.P. Olson, L.R. Schweitzer, P. Walsh, Application of a neural network in recognizing facial expression, in: IEEE Proceedings of the Seventh Annual Northeast Bioengineering Conference, Hartford, CT, USA, 1991, pp. 206-207.

[5] Irene Kotsia, Ioannis Pitas. Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines. IEEE Transactions on Image Processing 16(1): 172-187 (2007).

[6] T. M. Cover. Estimation by the Nearest Neighbor Rule. IEEE Trans. on Information Theory, 14(1):50-55, 1968.

[7] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In Proceedings of the thirtieth annual ACM symposium on Theory of computing, STOC98, pages 604-613, New York, NY, USA, 1998. ACM.

[8] T.F. Cootes, G.J Edwards, and C,J. Taylor. Active Appearance Models, Proc. European Conference on Computer Vision 1998.

[9] Jason M. Saragih, Simon Lucey, and Jeffrey Cohn, Face Alignment through Subspace Constrained Mean-Shifts, International Conference of Computer Vision (ICCV), September, 2009.

[10] D. Cristinacce and T. F. Cootes. Feature Detection and Tracking with Constrained Local Models. In EMCV, pages 929-938, 2004.

[11] L. Gu and T. Kanade. A Generative Shape Regularization Model for Robust Face Alignment. In ECCV08, 2008.

[12] Stefano Berretti, Alberto Del Bimbo, Pietro Pala, Boulbaba Ben Amor, Mohamed Daoudi: A Set of Selected SIFT Features for 3D Facial Expression Recognition. ICPR 2010: 4125-4128.

[13] Yuxiao Hu; Zhihong Zeng; Lijun Yin; Xiaozhou Wei; Xi Zhou; Huang, T.S. Multi-view facial expression recognition. FG 2008. Page(s): 1-6.

[14] Taskeed Jabid, Md. Hasanul Kabir, and Oksam Chae. Robust Facial Expression Recognition Based on Local Directional Pattern. ETRI Journal, vol.32, no.5, Oct. 2010, pp.784-794.

[15] Suyog Jain, Changbo Hu, J. K. Aggarwal. Facial Expression Recognition with Temporal Modeling of Shapes. 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), page(s): 1642-1649.

[16] S.M. Lajevardi, M. Lech. Averaged Gabor Filter Features for Facial Expression Recognition. Digital Image Computing: Techniques and Applications, 2008 (DICTA '08). 71-76.

[17] W. Fellenz, J. Taylor, N. Tsapatsoulis, S. Kollias, Comparing template-based, feature-based and supervised classification of facialexpressions from static images, Proceedings of Circuits, Systems, Communications and Computers (CSCC99), Nugata, Japan, 1999, pp. 5331-5336.

[18] Shishir Bashyal, Ganesh K. Venayagamoorthy: Recognition of facial expressions using Gabor wavelets and learning vector quantization. Eng. Appl. of AI 21(7): 1056-1064 (2008).

[19] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, Inderjit S. Dhillon: Information-theoretic metric learning. ICML 2007: 209-216.

[20] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kande Dataset (CK+): A complete facial expression dataset for action unit and emotion-specified expression. Third IEEE Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010).

[21] Alice J. OToole, Joshua Harms, Sarah L. Snow, Dawn R. Hurst, Matthew R. Pappas, Janet H. Ayyad, Herve Abdi. A Video Database of Moving Faces and People. IEEE Transactions on Pattern Analysis And Machine Intelligence, Vol. 27, No. 5, May 2005.

[22] A. Kanaujia and D. N. Metaxas. Recognizing facial expressions by tracking feature shapes. In International Conference on Pattern Recognition, 33-38, 2006.

[23] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[24] Prateek Jain, Brian Kulis, Kristen Grauman. Fast Image Search for Learned Metrics. CVPR 2008.

[25] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems 17, 513-520, Cambridge, MA, 2005. MIT Press.

[26] C. Domeniconi, D. Gunopulos, and J. Peng. Large margin nearest neighbor classiers. IEEE Transactions on Neural Networks, 16(4):899-909, 2005.

[27] Xudong Xie, Kin-Man Lam. Facial expression recognition based on shape and texture. Pattern Recognition. Volume 42, Issue 5, May 2009, Pages 1003-1011.

[28] I. Kotsia and I. Pitas. Facial expression recognition using shape and texture information. IFIP International Federation for Information Processing, 2006, Volume 217, Artificial Intelligence in Theory and Practice, Pages 365-374.