A Compact Gesture Recognition for Visually Challenged People Using Machine Learning

RAMESH.S[*1], ABINAYA.P[2] , FAHIMA.A[3], NIVETHA.V[4], REHANA.Y[5]

[1]*Assistant Professor,Department of Computer Science & Engineering, Krishnasamy College of Engineering & Technology, S.Kumarapuram, Cuddalore.*

[2,3,4,5]*UG Project Research Students, Department of Computer Science & Engineering, Krishnasamy College of Engineering & Technology, S.Kumarapuram, Cuddalore.*

Abstract- Hand Gesture algorithms are key enabling technologies for Human-Computer Interaction (HCI) systems. State of the art approaches for automatic detection of body movements and for analyzing emotions from facial features heavily rely on advanced machine learning algorithms. Most of these methods are designed for the average user, but the assumption "one-size-fits-all" ignores diversity in cultural background, gender, ethnicity and personal behaviour and limits their applicability in real world scenarios. A possible solution is to build personalized interfaces, which practically implies learning person-specific classifiers and usually collecting a significant amount of labeled samples for each novel user. As data annotation is a tedious and time-consuming process, in this paper we present a framework for personalizing classification models which does not require labeled target data. Personalization is achieved by devising a novel transfer learning approach. Specifically, we propose a regression framework which exploits auxiliary (source) annotated data to learn the relation between person-specific sample distributions and the parameters of the corresponding classifiers. Then, when considering a new target user, the classification model is computed by simply feeding the associated (unlabeled) sample distribution into the learned regression function. We evaluate the proposed approach in different databases demonstrating the generality of our method with respect to different input data types and basic classifiers. We also show the advantages of our approach in terms of accuracy and computational time both with respect to user-independent approaches and to previous personalization techniques.

Keywords – Hand Gesture algorithms, regression framework, novel transfer learning approach, user-independent approaches

I. INTRODUCTION

The term digital image refers to processing of a two-dimensional picture by a digital computer. In a broader context, it implies digital processing of any two-dimensional data. A digital image is an array of real or complex numbers represented by a finite number of bits. An image given in the form of a transparency, slide, photograph or an X-ray is first digitized and stored as a matrix of binary digits in computer memory. This digitized image can then be processed and/or displayed on a high-resolution television monitor. For display, the image is stored in a rapid-access buffer memory, which refreshes the monitor at a rate of 25 frames per second to produce a visually continuous display.

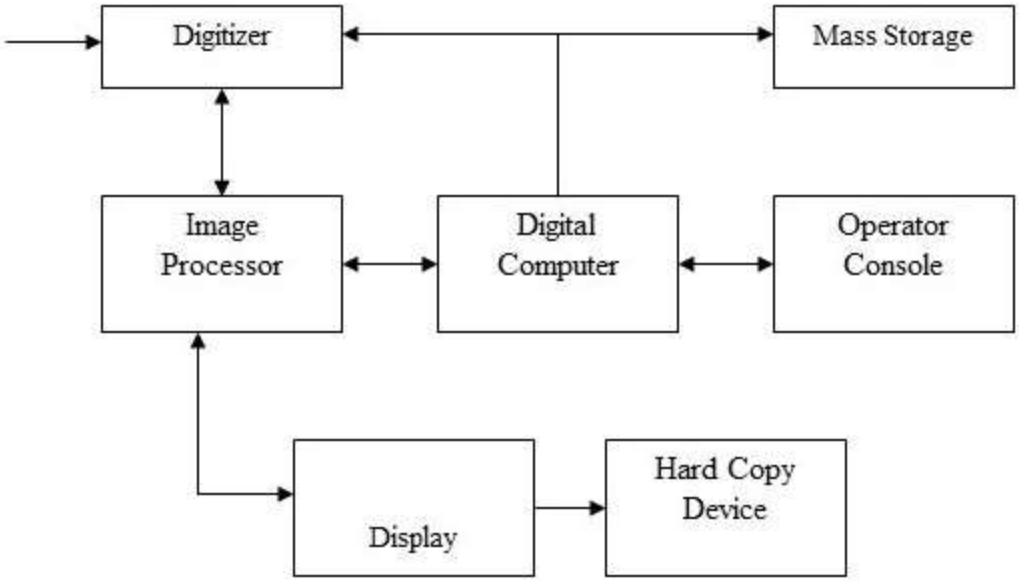

Figure 1 Block Diagram for Image Processing System

DIGITIZER

A digitizer converts an image into a numerical representation suitable for input into a digital computer. Some common digitizers are

1. Microdensitometer
2. Flying spot scanner
3. Image dissector
4. Videocon camera
5. Photosensitive solid- state arrays.

IMAGE PROCESSOR

An image processor does the functions of image acquisition, storage, preprocessing, segmentation, representation, recognition and interpretation and finally displays or records the resulting image. The following block diagram gives the fundamental sequence involved in an image processing system.

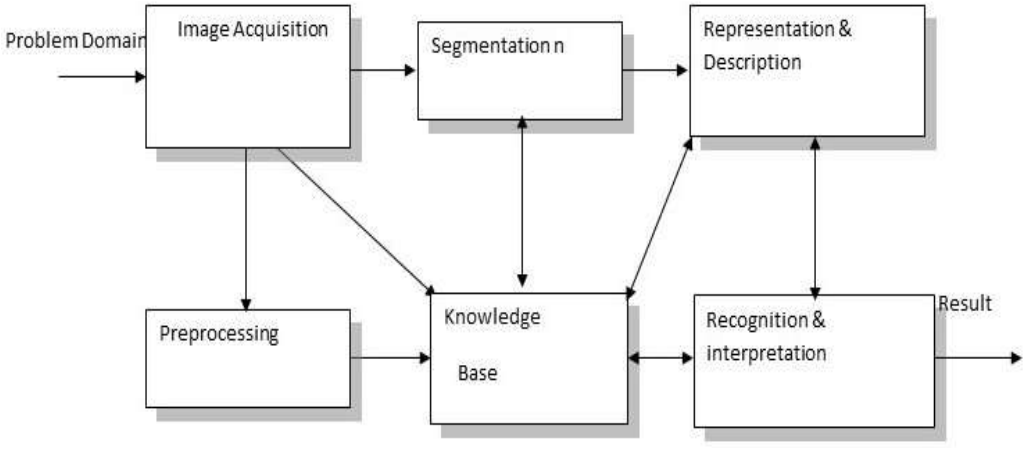

Figure 2 Block Diagram of Fundamental Sequence Involved in an Image Processing System

As detailed in the diagram, the first step in the process is image acquisition by an imaging sensor in conjunction with a digitizer to digitize the image. The next step is the preprocessing step where the image is improved being fed as an input to the other processes. Preprocessing typically deals with enhancing, removing noise, isolating regions, etc. Segmentation partitions an image into its constituent parts or objects. The output of segmentation is usually raw pixel data, which consists of either the boundary of the region or the pixels in the region themselves. Representation is the process of transforming the raw pixel data into a form useful for subsequent processing by the computer. Description deals with extracting features that are basic in differentiating one class of objects from another. Recognition assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects. The knowledge about a problem domain is incorporated into the knowledge base. The knowledge base guides the operation of each processing module and also controls the interaction between the modules. Not all modules need be necessarily present for a specific function. The composition of the image processing system depends on its application. The frame rate of the image processor is normally around 25 frames per second.

II. LITERATURE SURVEY

According to [6], current unsupervised domain adaptation works can be differentiated into instance transfer and feature transfer methods. Conversely, the proposed method aims to directly transfer the parameters of the classifiers from the source to the target domain. According to the type of information transferred from source to target domains, the methods are categorized into parameter transfer, feature transfer and instance transfer approaches. Parameter transfer methods aim to find a set of parameters or priors shared between the source and the target models. Feature transfer methods operate by looking for a shared feature representation for source and target data. In [13] Yang et al. extended standard Support Vector Machines (SVMs) and proposed Adaptive-SVMs. Adaptive-SVMs employ a regularization term to impose the target classifier to be similar to the source one. However, these methods usually require annotated target data, which are typically not easily obtained in HCI. Feature transfer methods operate by looking for a shared feature representation for source and target data. For instance, in [14] the input feature vector is augmented by obtaining a novel descriptor composed of a shared, a source-specific and a target-specific part. The Heterogeneous Feature Augmentation method is presented in [15] to tackle the problem of knowledge transfer when the data from the source and the target domain are represented by heterogeneous features with different dimensions. In [16] a shared representation for source and target data in terms of visual attributes is proposed. In [17] both source and target data are projected to a common subspace where each target sample can be represented by some combination of source samples. In [18] deep structures are exploited for learning a discriminative feature representation to alleviate the cross-domain discrepancy problem. Instance transfer approaches [19], [20] are commonly adopted when the target data are unlabeled. For instance, in [19] Gretton et al. proposed to compute the centroids of the source and the target distributions and to estimate those source sample weights which reduce the inter-centroid distance in a Reproducing Kernel Hilbert Space. These weights are then used to assign importance to the source samples when training a model for classification on target data. The drawback with most instance transfer approaches [19], [20] is that computing the distance between centroids may poorly approximate the real discrepancy between distributions. We overcome this issue by adopting more accurate approaches to quantify the difference between source and target distributions which are based on specific kernels for distributions. Moreover, most instance transfer methods rely on a computationally intense training phase. In the lastfew years, research on facial expression analysis from visual data has made significant progress. Many approaches have proved to be effective for recognizing simple facial expressions (e.g. happiness, sadness, anger, etc.) or alternatively for detecting Action Units (AUs) [23], [24]. However, most state-of-the-art methods are trained and tested in laboratory conditions, with datasets mainly consisting of frontal face images and posed emotions [23], [24], [25]. Very little attention has been paid to realistic scenarios and personalized systems. Notable exceptions are the works in [26], [27], [28] which focused on recognizing spontaneous facial expressions and non-basic emotions, e.g. pain. However, they are based on user-independent models, i.e. on detectors trained on a generic dataset, which aim tobe sufficiently representative of many possible sources of variability (e.g. illumination conditions, target appearance, etc.). Unfortunately, having at disposal only datasets with few hundreds or thousands of images, generalization to arbitrary conditions is hard to achieve.

To cope with this issue, few works have proposed solutions to integrate weakly labeled or unlabeled data. In [29] Sikka et al. adopted a Multiple Instance Learning approach for training a pain expression classifier using video-level labels where frame-level labels are not available. The problem of pain detection is also addressed in [21] where an extension of AdaBoost for user-personalization is proposed both in a supervised and in an unsupervised setting. However, in the unsupervised case, the proposed method did not achieve a significant improvement in terms

of accuracy with respect to the user independent classifier. In [20] Selective Transfer Machine (STM) is proposed for person-specific AU detection. STM is based on the Kernel Mean Matching algorithm [19], which is modified usingan iterative minimization procedure where labeled source data drive a progressive movement of the generic SVM hyperplane toward the target space. Even if effective, this approach is very slow at training time, as the underlying optimization strategy is very time consuming. On the other hand, user-specific adaptation algorithms are required to be computationally efficient to be used in HCI applications. Our method is mainly motivated by this need and our experiments confirm that it is much faster than [20], being its accuracy comparable and even better.

III. OBJECTIVE AND SCOPE OF THE PROJECT

In this paper we propose a novel transfer learning framework to build personalized models without resorting to user-specific labeled data. Our approach relies on learning a regression function which captures the relation between a data distribution and the classifier's parameters learned using the samples generated from that distribution. Specifically, our method is based on three phases. In the first phase, given N Auxiliary source users and the associated labeled training samples, we learn a set of N classifiers, parametrized by the vectors $\theta_1, \ldots, \theta_N$. In the second phase a regression function f(.), which relates the unlabeled data distribution of the ith source user with the associated classifier θ_i , is learned. Importantly, once f(.) is obtained, labeled data are not required anymore. Finally, given a novel target user, it suffices to apply f(.) to the associated data distribution to obtain the personalized classifier θ^t.

IV. EXISTING SYSTEM

Nowadays, the importance of adaptive and personalized human-computer interfaces, as opposite to systems designed for an "average" user, is widely recognized in a large variety of applications. Machine learning algorithms for automatic analysis of hand expressions and body movements are currently employed in many HCI systems. However, surprisingly, few of these systems adapt the learned models to specific users. The issue with personalization is that typically a significant amount of labeled data is required to train user-specific classifiers. This is practically infeasible in many real-world applications as collecting a large number of annotated samples is very time consuming.

In the last few years several transfer learning methods have recently become popular in the multimedia and the computer vision fields [10], [11], [12] to solve or alleviate the so-called dataset bias problem. Transfer learning aims to improve the learning performance in a target domain using knowledge extracted from related source domains.

DISADVANTAGES OF EXISTING SYSTEM

- Less accuracy
- Need of large dataset
- High computation time

V. PROPOSED SYSTEM

The proposed approach in this work relies on learning a regression function which captures the relation between a data distribution and the classifier's parameters learned using the samples generated from that distribution. The proposed method is based on three phases. In the first phase, given N auxiliary source users and the associated labeled training samples, we learn a set of N classifiers, parametrized by the vectors $\theta_1, \ldots, \theta_N$. In the second phase a regression function f(.), which relates the unlabeled data distribution of the ith source user with the associated classifier θ_i , is learned. Importantly, once f(.) is obtained, labeled data are not required anymore. Finally, given a novel target user, it suffices to apply f(.)to the associated data distribution to obtain the personalized classifier θ^t.

BLOCK DIAGRAM

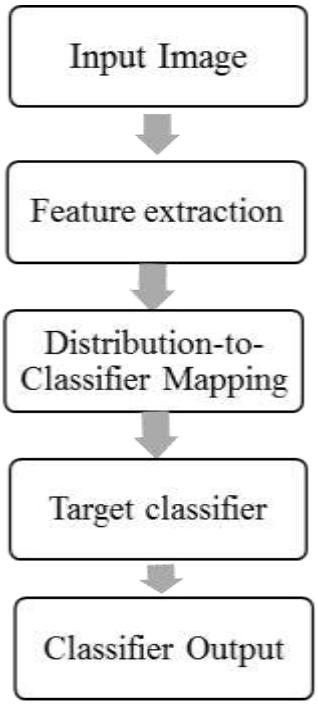

Figure 3. Block diagram of proposed system

PROPOSED SYSTEM ADVANTAGES

- The proposedmethod automatically classify the action based on input.
- The overall time for learning as well as testing is less and involve less computation.
- The proposed method can be applied even to database with small number of sources.

VI. PROJECT DESCRIPTION

The proposed TPT approach is based on three main phases . In the first phase, N userspecific classifiers are learned, one for each source training setD_i^s. Each classifier is defined by a parameter vectorθ_i. Then a regression algorithm is proposed in order to learn the relation between the marginal distributions$\pi_i^s$and θ_i. Finally, the desired target classifier θ^t is obtained by applying the learned distribution-to-classifier mapping and using as input the distribution. In the following, the three phases are further detailed.

FEATURE EXTRACTION

We also use Gabor Wavelet to extract features from fingerprint images for texture analysis. Gabor filters have optimal localization properties in both the frequency and spatial domain, and have been successfully used in many applications to extract discriminative features [42]. In fingerprint images, the local ridge characteristics are extracted via a set of Gabor filters whose frequency corresponds to the inter-ridge spacing in fingerprints.The Gabor wavelets (kernels, filters) can be defined as follows :

$$\psi_{\mu,\upsilon}(z) = \frac{\|k_{\mu,\upsilon}\|^2}{\sigma^2}\epsilon^{(-\|k_{\mu,\upsilon}\|^2\|z^2\|/2\sigma^2)}\left[\epsilon^{ik_{\mu,\nu}z} - \epsilon^{-\sigma^2/2}\right]$$

where μ and υ define the orientation and scale of the Gabor kernels, z = (x, y), ||.||denotes the norm operator, and the wave vector kμ,vis defined as follows:

$$k_{\mu,\upsilon} = k_\nu \epsilon^{i\phi_\mu}$$

Where kμ = kmax/fand φ = πμ/8.kmax is the maximum frequency, and is the spacing factor between kernels in the frequency domain . The Gabor kernels are all self-similar since they can be generated from one filter, the mother

wavelet, by scaling and rotation via the wave vector kμ,v. Each kernel is a product of a Gaussian envelope and a complex plane wave, while the first term in the square brackets in (1) determines the oscillatory part of the kernel and the second term compensates for the DC value. The effect of the DC term becomes negligible when the parameter, which determines the ratio of the Gaussian window width to wavelength, ha sufficiently large values.

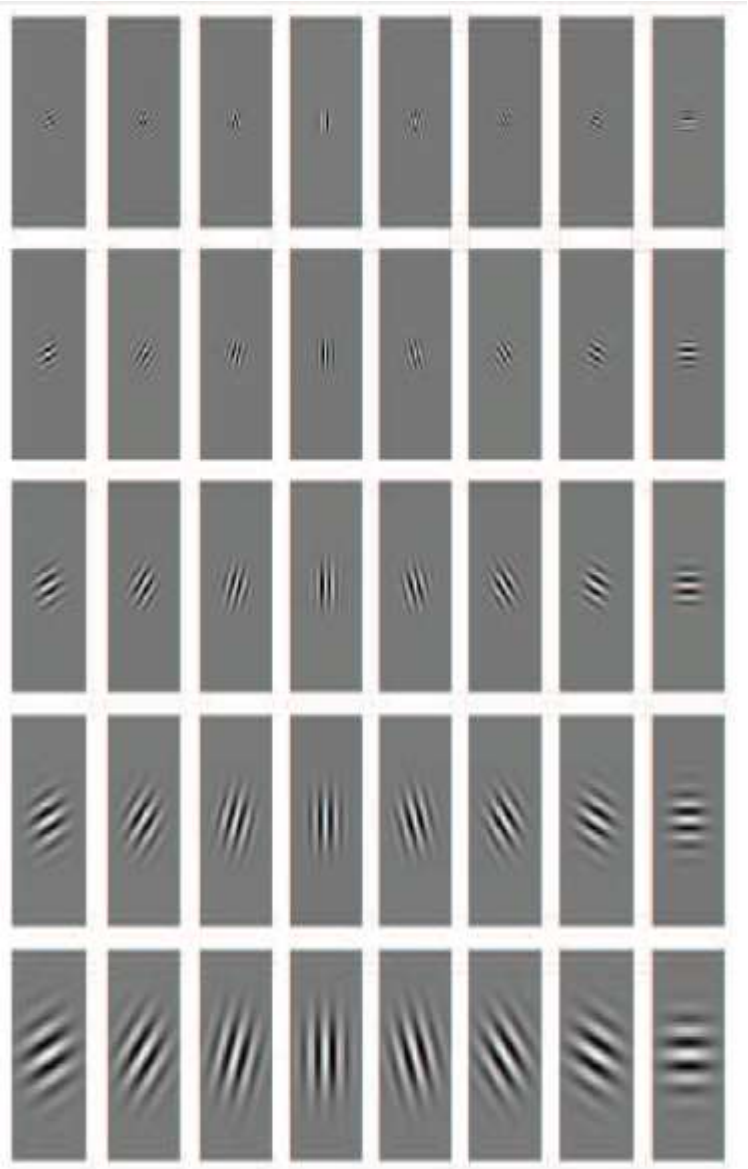

Figure 4 GABOR KERNEL

In our experiments we have used Gabor wavelets of five different scales, $v \in \{0, 1, 2, 3, 4\}$, and eight orientations, $\mu \in \{0, 1, 2, 3, 4, 5, 6, 7\}$. Figure. 4 shows the real part of the Gabor kernels for a sample image at five scales and eight orientations. The following parameters of Gabor Wavelets ($\sigma = 2\pi$, $kmax = \pi/2$, and $f = \sqrt{2}$) were used. The kernels exhibit desirable characteristics of spatial frequency, spatial locality, and orientation selectivity. The Gabor wavelet representation of an image is the convolution of the image with a family of Gabor kernels . Let I(x, y) be the gray level distribution of an image, the convolution of image I and a Gabor kernel defined as follows;

$$O_{\mu,\nu}(z) = I(z) * \psi_{\mu,\nu}(z)$$

where z = (x, y), ∗ denotes the convolution operator, and $O_{u,v}(z)$ is the convolution result corresponding to the Gabor kernel at orientation μ and scale υ. Therefore, the set of O contain μ ∈ {0, 1, 2, 3, 4, 5, 6, 7}, v ∈ {0, 1, 2, 3, 4} forms the Gabor wavelet representation of the image I(z). Applying the convolution theorem, we can derive each O from (3) via the Fast Fourier Transform (FFT).

TRANSDUCTIVE PARAMETER TRANSFER

Let X and Y denote the feature and the label spaces, respectively. For classification, $Y = \{-1,1\}$in the binary setting, while in case of multiple classes$Y = \{1,2,\dots..C\}$. Given an unlabeled target dataset $X^t = \{x_j^t\}_{j=1}^{n^t}$and N labeled source datasets$D_1^s, \dots..D_N^s, D_i^s = \{x_{ij}^s, y_{ij}^s\}_{j=1}^{n_i^s}$, $x_{ij}^s, x_j^t \in X, y_{ij}^s \in Y$we want to learn a classifier on the target dataX^t without acquiring labeled information. In the context of user personalization D_i^scontains all the training samples corresponding to the i[th] source person, while X^t is the set of (unlabeled) data of the target individualfor whom we aim to construct the personalized classifier. Moreover,$X_i^s = \{x_{ij}^s\}_{i=1}^{n_i^s}$ denotes the set of points inD_i^sobtained by discarding the label information.

We assume that the vectors in X_i^s are generated by a marginal distribution π_i^sdefined on X and similarly the vectors $X^t$are generated by π^t. We generally assume that $\pi_i^s \neq \pi^t$. Finally, we call P the space of all possible distributions on X and we assume that are π_i^s, π^ti.i.d. sampled on P .In the following $(.)'$ denotes the transpose operator.

Phase 1: Learning User-specific Source Classifiers

In TPT the source datasets $D_i^s$are used to learn N independent classifiers $\theta_i$by solving N different problems:

$$\min_{\boldsymbol{\theta}\in\Theta} \Omega(\boldsymbol{\theta}) + \lambda_L \Lambda(D_i^s; \boldsymbol{\theta}) \qquad (1)$$

Each weight vector θ_i represents a personalized classifier since it is learned using user-specific samplesD_i^s. While our framework is general and different choices can bemade for the regularization and the loss terms in (1), here we consider a set of linear SVMs. Therefore, defining $X \equiv IR^M$ the optimal decision hyperplane$\theta_i = [w_i, b_i]'$. For each source dataset D_i^scan be computed by solving:

$$\min_{\mathbf{w}_i, b_i} \frac{1}{2}||\mathbf{w}_i||^2 + \lambda_L \sum_{j=1}^{n_i^s} \ell(\mathbf{w}_i' \mathbf{x}_{ij}^s + b_i, y_{ij}^s) \qquad (2)$$

Phase 2: Learning a Distribution-to-Classifier Mapping

In the second phase of TPT, we propose a regression framework in order to learn a mapping $f: P \rightarrow \Theta$between a sample distribution and its associated classifier. The intuition is straightforward: if we are able to learn the relationship between the underlying distribution $\pi_i^s$and the corresponding hyperplaneθ_i, then when computing the optimal hyperplane on the target data we do not need labeled samples since we can simply apply the learned mapping f(.),i.e$\theta^t = f(\pi^t)$. However, since $\pi_i^s, \pi^t$are unknown, we need to approximate distributionsusing the empirical data at disposal. In particular in this paper we use all the samples in $X^t$to approximate π^t while for π_i^swe evaluate two possibilities: (i) all the data in $X_i^s$are considered and (ii) only the Support Vectors obtained by solving (2) for each of the i[th] source tasks are used as a proxy for π_i^s .In the following, to simplify the notation, we assume that $Z_i$is the set of points chosen to approximate π_i^s , either$Z_i = X_i^s$ *or* $Z_i = \hat{X}_s^i$.

Given a training set $T = \{(Z_i, \theta_i)\}_{i=1}^N$we propose to learn a mapping $\hat{f}: 2^X \rightarrow \Theta$ which approximates $\hat{f}(.)$. In this paper we investigate two possibleapproaches to compute $f(.)$, i.e. learningM+1 independent scalar regressors$\hat{f}_k(X)$:

$$\hat{f}(X) = (\hat{f}_1(X), \dots, \hat{f}_{M+1}(X)) \qquad (3)$$

In the caseof independent regression models, we compute each $\hat{f}_k(.)$using the ϵinsensitive Support Vector Regression (SVR)framework [36] which, in our setting, can be formulated as follows. Each is$\hat{f}_k(.)$ defined by a set of parameters v_k, u_k:

$$\widehat{f}_k(X) = \sum_{i=1}^{N} \delta_i^k \kappa(Z_i, X) + u_k.$$

Where $k(Z_i, Z_l)$is the kernel function defined by equation:

$$\kappa_{DE}(Z_i, Z_j) = \frac{1}{nm} \sum_{p=1}^{n} \sum_{q=1}^{m} \kappa_{\mathcal{X}}(\mathbf{z}_p^i, \mathbf{z}_q^j), \qquad (19)$$

Phase 3: Learning the Target Classifier

In the last phase of TPT the optimal target classifier $\theta^t$is computed considering the unlabeled target samples X^t and using$\theta^t = \hat{f}(X^t)$. For M-SVR we use (11), while (7) and(3) are used in the case of independent regression models. Note that the computations which involve the source data(Phase 1-2) are performed only once. Then, for every new user, only Phase 3 needs to be repeated. This is very advantageous in real world applications, where it is desirable to accomplish personalization in a limited timeframe. Algorithm 1 summarizes the main phases of TPT.

Test Phase

Once $\theta^t = [(w^t)', b^t]'$ has been computed, the test phase is a standard classification with linear SVMs. Given a new target feature vector x, the corresponding label y is predicted as $y = sig(x'w^t + b^t)$.It is worth noting that our TPT framework can be trivially extended to a multi-class setting adopting a one-versus-all scheme.

VII.MATERIALS AND METHODS

The work presented in this study consists of three major modules:

1. **FEATURE EXTRACTION**
2. **DISTRIBUTION-TO-CLASSIFIER MAPPING**
3. **TARGET CLASSIFIER**

MODULE DESCRIPTION

MODULE 1: FEATURE EXTRACTION

From the training and test images first the face region is detected and cropped. Then Gabor filter is applied tom extract features from hand region. The extracted features are concatenated to form vectors.

MODULE 2: DISTRIBUTION-TO-CLASSIFIER MAPPING

In the training images first SVM classifier is applied to obtain individual classifier parameter then using regression mapping function is learned.

ALGORITHM

Input:$D_1^s, \ldots\ldots D_N^s$,X^t and appropriate parameters .
Phase 1. Learning User-specific Source Classifiers
Compute θ_i ,$\forall i = 1 \ldots\ldots N$ using (2)
Phase 2. Learning a Distribution-to-classifier Mapping
Create training set $T = \{(Z_i, \theta_i)\}_{i=1}^N$where $Z_i = X_i^s$ or $Z_i = \hat{X}_s^i$.
Compute the source kernel matrix $K, K_{il} = k(Z_i, Z_l)$ using 19 .
Given K, T compute $\hat{f}(.)$ solving 3 $\forall k = 1 \ldots\ldots K + 1$
Phase 3.Computing target classifier
Compute target kernel vector $K^t, K_i^t = k(Z_i, X^t)$
Using K^t compute $\theta^t = \hat{f}(X^t)$.
Output:θ^t

MODULE 3: TARGET CLASSIFIER

From the training images kernel function is extracted and then applied to mapping function to get classified output

VIII. DEVELOPING ALGORITHMS AND APPLICATIONS

MATLAB provides a high-level language and development tools that let you quickly develop and analyze your algorithms and applications.

The MATLAB Language

The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development and execution. With the MATLAB language, you can program and develop algorithms faster than with traditional languages because you do not need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for 'for' loops. As a result, one line of MATLAB code can often replace several lines of C or C++ code.

At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, object-oriented programming (OOP), and debugging features.

MATLAB lets you execute commands or groups of commands one at a time, without compiling and linking, enabling you to quickly iterate to the optimal solution. For fast execution of heavy matrix and vector computations, MATLAB uses processor-optimized libraries. For general-purpose scalar computations, MATLAB generates machine-code instructions using its JIT (Just-In-Time) compilation technology.

This technology, which is available on most platforms, provides execution speeds that rival those of traditional programming languages.

Development Tools

MATLAB includes development tools that help you implement your algorithm efficiently. These include the following:

MATLAB Editor

Provides standard editing and debugging features, such as setting breakpoints and single stepping

Code Analyzer

Checks your code for problems and recommends modifications to maximize performance and maintainability

MATLAB Profiler

Records the time spent executing each line of code

Directory Reports

Scan all the files in a directory and report on code efficiency, file differences, file dependencies, and code coverage

Designing Graphical User Interfaces

By using the interactive tool GUIDE (Graphical User Interface Development Environment) to layout, design, and edit user interfaces. GUIDE lets you include list boxes, pull-down menus, push buttons, radio buttons, and sliders, as well as MATLAB plots and Microsoft ActiveX® controls. Alternatively, you can create GUIs programmatically using MATLAB functions.

ANALYZING AND ACCESSING DATA

MATLAB supports the entire data analysis process, from acquiring data from external devices and databases, through preprocessing, visualization, and numerical analysis, to producing presentation-quality output.

Data Analysis

MATLAB provides interactive tools and command-line functions for data analysis operations, including:

- Interpolating and decimating
- Extracting sections of data, scaling, and averaging

- Thresholding and smoothing
- Correlation, Fourier analysis, and filtering
- 1-D peak, valley, and zero finding
- Basic statistics and curve fitting
- Matrix analysis

Data Access

MATLAB is an efficient platform for accessing data from files, other applications, databases, and external devices. You can read data from popular file formats, such as Microsoft Excel; ASCII text or binary files; image, sound, and video files; and scientific files, such as HDF and HDF5. Low-level binary file I/O functions let you work with data files in any format. Additional functions let you read data from Web pages and XML.

Visualizing Data

All the graphics features that are required to visualize engineering and scientific data are available in MATLAB. These include 2-D and 3-D plotting functions, 3-D volume visualization functions, tools for interactively creating plots, and the ability to export results to all popular graphics formats. You can customize plots by adding multiple axes; changing line colors and markers; adding annotation, Latex equations, and legends; and drawing shapes.

2-D Plotting

Visualizing vectors of data with 2-D plotting functions that create:

- Line, area, bar, and pie charts.
- Direction and velocity plots.
- Histograms.
- Polygons and surfaces.
- Scatter/bubble plots.
- Animations.

3-D Plotting and Volume Visualization

MATLAB provides functions for visualizing 2-D matrices, 3-D scalar, and 3-D vector data. You can use these functions to visualize and understand large, often complex, multidimensional data. Specifying plot characteristics, such as camera viewing angle, perspective, lighting effect, light source locations, and transparency.

3-D plotting functions include:

- Surface, contour, and mesh.
- Image plots.
- Cone, slice, stream, and isosurface.
-

IX. PERFORMING NUMERIC COMPUTATION

MATLAB contains mathematical, statistical, and engineering functions to support all common engineering and science operations. These functions, developed by experts in mathematics, are the foundation of the MATLAB language. The core math functions use the LAPACK and BLAS linear algebra subroutine libraries and the FFTW Discrete Fourier Transform library. Because these processor-dependent libraries are optimized to the different platforms that MATLAB supports, they execute faster than the equivalent C or C++ code.

MATLAB provides the following types of functions for performing mathematical operations and analyzing data:

- Matrix manipulation and linear algebra.
- Polynomials and interpolation.
- Fourier analysis and filtering.
- Data analysis and statistics.
- Optimization and numerical integration.
- Ordinary differential equations (ODEs).
- Partial differential equations (PDEs).
- Sparse matrix operations.

MATLAB can perform arithmetic on a wide range of data types, including doubles, singles, and integers.

X. APPLICATIONS OF DIGITAL IMAGE PROCESSING

Digital image processing has a broad spectrum of applications, such as remote sensing via satellites and other spacecrafts, image transmission and storage for business applications, medical processing, radar, sonar and acoustic image processing, robotics and automated inspection of industrial parts.

MEDICAL APPLICATIONS:

In medical applications, one is concerned with processing of chest X-rays, cineangiograms, projection images of transaxial tomography and other medical images that occur in radiology, nuclear magnetic resonance (NMR) and ultrasonic scanning. These images may be used for patient screening and monitoring or for detection of tumors' or other disease in patients.

SATELLITE IMAGING:

Images acquired by satellites are useful in tracking of earth resources; geographical mapping; prediction of agricultural crops, urban growth and weather; flood and fire control; and many other environmental applications. Space image applications include recognition and analysis of objects contained in image obtained from deep space-probe missions.

COMMUNICATION:

Image transmission and storage applications occur in broadcast television, teleconferencing, and transmission of facsimile images for office automation, communication of computer networks, closed-circuit television-based security monitoring systems and in military communications**.**

RADAR IMAGING SYSTEMS:

Radar and sonar images are used for detection and recognition of various types of targets or in guidance and maneuvering of aircraft or missile systems.

DOCUMENT PROCESSING:

It is used in scanning, and transmission for converting paper documents to a digital image form, compressing the image, and storing it on magnetic tape. It is also used in document reading for automatically detecting and recognizing printed characteristics.

DEFENSE/INTELLIGENCE:

It is used in reconnaissance photointerpretation for automatic interpretation of earth satellite imagery to look for sensitive targets or military threats and target acquisition and guidance for recognizing and tracking targets in real-time smart-bomb and missile-guidance systems.

XI.CONCLUSION

In this paper we proposed Transductive Parameter Transfer, a framework for building personalized classification models, and we demonstrated its effectiveness on three different application domains: accelerometer-based gesture recognition, pain classification from facial expressions and AU detection. The proposed method is based on a regression framework which is trained to learn the relation between the unlabeled data distribution of a given person and the parameters of her/his personalized classifier. We experimentally showed that our TPT outperforms both user independent and previous domain adaptation approaches and achieves state of the art performance on public benchmarks. As far as we know, this is the first transductive parameter transfer approach in transfer learning literature. The main advantage of our method is that, using a pre-trained regression function, its computational cost is much lower than other domain adaptation algorithms. This makes TPT an appealing candidate for building personalized systems.

REFERENCES

[1] G. Costante, L. Porzi, O. Lanz, P. Valigi, and E. Ricci, "Personalizing a smartwatch-based gesture interface with transfer learning," in EUSIPCO, 2014.

[2] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," Pervasive and Mobile Computing, vol. 5, no. 6, pp. 657– 675, 2009.

[3] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews, "Painful monitoring: Automatic pain monitoring using the unbc-mcmaster shoulder pain expression archive database," Image and Vision Computing, vol. 30, no. 3, 2012.

[4] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," PAMI, vol. 28, no. 12, pp. 2037–2041, 2006.
[5] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, an I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in CVPRW, 2010.
[6] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2010.
[7] E. Sangineto, G. Zen, E. Ricci, and N. Sebe, "We are not all equal: Personalizing models for facial expression analysis with transductiveparameter transfer," in ACM'MM, 2014.
[8] G. Zen, E. Sangineto, E. Ricci, and N. Sebe, "Unsupervised domain adaptation for personalized facial emotion recognition," in ICMI, 2014.
[9] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," arXiv preprint arXiv:1306.6709, 2013.
[10] S. D. Roy, T. Mei, W. Zeng, and S. Li, "Towards cross-domain learning for social video popularity prediction," IEEE Transactions on Multimedia, vol. 15, no. 6, pp. 1255–1267, 2013.
[11] S. Xia, M. Shao, J. Luo, and Y. Fu, "Understanding kin relationships in a photo," IEEE Transactions on Multimedia, vol. 14, no. 4, pp. 1046–1056, 2012.
[12] Z. Guo and Z. J. Wang, "An unsupervised hierarchical feature learning
framework for one-shot image recognition," IEEE Transactions
on Multimedia, vol. 15, no. 3, pp. 621–632, 2013.
[13] J. Yang, R. Yan, and A. G. Hauptmann, "Adapting svm classifiers to data with shifted distributions," in Int. Conf. on Data MiningWorkshops, 2007.
[14] H. Daum´e III, "Frustratingly easy domain adaptation," in ACL, 2007.
[15] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," IEEE Transactions on PAMI, vol. 36, no. 6, pp. 1134–1148, 2014.
[16] C. H. Lampert, H. Nickisch, and S.Harmeling, "Learning to detectunseen object classes by between-class attribute transfer," in CVPR, 2009.
[17] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," IJCV, vol. 109, no. 1-2, pp. 74–93,2014.
[18] Z. Ding, M. Shao, and Y. Fu, "Deep low-rank coding for transfer learning," in IJCAI, 2015.
[19] A. Gretton, A. Smola, J. Huang, M.Schmittfull, K. Borgwardt, and B. Sch¨olkopf, "Covariate shift by kernel mean matching," Dataset shift in machine learning, vol. 3, no. 4, p. 5, 2009.
[20] W.-S. Chu, F. D. L. Torre, and J. F. Cohn, "Selective transfer machine for personalized facial action unit detection," in CVPR, 2013.
[21] J. Chen, X. Liu, P. Tu, and A. Aragones, "Learning person-specific models for facial expression and action unit recognition," Pattern Recognition Letters, vol. 34, no. 15, pp. 1964–1970, 2013.
[22] J. M¨antyj¨arvi, J. Kela, P. Korpip¨a¨a, and S. Kallio, "Enabling fast andeffortlesscustomisation in accelerometer based gesture interaction,"in Int. Conf. on Mobile and Ubiquitous Multimedia, 2004.
[23] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," PAMI, vol. 31, no. 1, pp. 39–58, 2009.
[24] A. Martinez and S. Du, "A model of the perception of facial expressions of emotion by humans: Research overview and perspectives," J. of Machine Learning Research, vol. 13, no. 1, pp. 1589–1608, 2012.
[25] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer, "Thefirst facial expression recognition and analysis challenge," in FG,2011.
[26] H. Dibeklio˘glu, T. Gevers, A. A. Salah, and R. Valenti, "A smilecan reveal your age: Enabling facial dynamics in age estimation," in ACM'MM, 2012.
[27] G. C. Littlewort, M. S. Bartlett, and K. Lee, "Faces of pain: automated measurement of spontaneousallfacial expressions of genuineand posed pain," in ICMI, 2007.
[28] M. F. Valstar, M. Pantic, Z. Ambadar, and J. F. Cohn, "Spontaneous vs. posed facial behavior: automatic analysis of brow actions," in ICMI, 2006.
[29] K. Sikka, A. Dhall, and M. Bartlett, "Weakly supervised pain localization using multiple instance learning," in Int. Conf. on Automatic Face and Gesture Recognition, 2013.
[30] G. Bieber, T. Kirste, and B. Urban, "Ambient interaction by smart watches," in Int. Conf. on PErvasive Technologies Related to AssistiveEnvironments, 2012.
[31] M. Khan, S. I. Ahamed, M. Rahman, and J.-J. Yang, "Gesthaar: An accelerometer-based gesture recognition method and its application in nui driven pervasive healthcare," in Int. Conf. on Emerging Signal Processing Applications, 2012.

[32] L. Porzi, S. Messelodi, C. M. Modena, and E. Ricci, "A smart watch based gesture recognition system for assisting people with visual impairments," in Int. Workshop on Interactive Multimedia on Mobile& Portable Devices, 2013.
[33] H. Junker, O. Amft, P. Lukowicz, and G. Tr¨oster, "Gesture spotting with body-worn inertial sensors to detect user activities," Pattern Recognition, vol. 41, no. 6, pp. 2010–2024, 2008.
[34] S.-J. Cho, E. Choi, W.-C. Bang, J. Yang, J. Sohn, D. Y. Kim, Y.-B. Lee, S. Kim et al., "Two-stage recognition of raw acceleration signals for 3-d gesture-understanding cell phones," in Int. Workshop on Frontiers in Handwriting Recognition, 2006.
[35] D. Tuia, J. Verrelst, L. Alonso, F. P´erez-Cruz, and G. Camps- Valls, "Multioutput support vector regression for remote sensing biophysical parameter estimation," IEEE Geosci. Remote Sens. Lett.,vol. 8, no. 4, pp. 804–808, 2011.
[36] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," NIPS, 1997.
[37] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," Int. Journ.of Computer Vision, vol. 40, no. 2, pp. 99–121, 2000.
[38] E. Ricci, G. Zen, N. Sebe, and S. Messelodi, "A prototype learning framework using emd: Application to complex scenes analysis," PAMI, vol. 35, no. 3, pp. 513–526, 2013.
[39] M. R. Daliri, "Kernel earth mover's distance for eeg classification," Clinical EEG and neuroscience, vol. 44, no. 3, pp. 182–187, 2013.
[40] T. Jaakkola, D. Haussler et al., "Exploiting generative models in discriminative classifiers," NIPS, 1999.
[41] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in CVPR, 2007.
[42] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," NIPS, 2011.
[43] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in NIPS, 2002.
[44] K. Q. Weinberger and L. K. Saul, "Fast solvers and efficient implementations for distance metric learning," in ICML, 2008.
[45] T. Joachims, "Transductive inference for text classification usingsupport vector machines," in ICML, 1999.
[46] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," PAMI, vol. 32, no. 5, pp. 770–787, 2010.
[47] K. M. Prkachin and P. E. Solomon, "The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain," Pain, vol. 139, no. 2, pp. 267–274, 2008.