Weakly Supervised Facial Analysis with Dense Hyper-column Features

Chenchen Zhu*, Yutong Zheng*, Khoa Luu, T. Hoang Ngan Le,
Chandrasekhar Bhagavatula, and Marios Savvides
CyLab Biometrics Center and the Department of Electrical and Computer Engineering,
Carnegie Mellon University, Pittsburgh, PA, USA
{chenchez, yutongzh, kluu, thihoanl, cbhagava}@andrew.cmu.edu, msavvid@ri.cmu.edu

Abstract

Weakly supervised methods have recently become one of the most popular machine learning methods since they are able to be used on large-scale datasets without the critical requirement of richly annotated data. In this paper, we present a novel, self-taught, discriminative facial feature analysis approach in the weakly supervised framework. Our method can find regions which are discriminative across classes yet consistent within a class and can solve many face related problems. The proposed method first trains a deep face model with high discriminative capability to extract facial features. The hypercolumn features are then used to give pixel level representation for better classification performance along with discriminative region detection. In addition, calibration approaches are proposed to enable the system to deal with multi-class and mixed-class problems. The system is also able to detect multiple discriminative regions from one image. Our uniform method is able to achieve competitive results in various face analysis applications, such as occlusion detection, face recognition, gender classification, twins verification and facial attractiveness analysis.

1. Introduction

Early computer vision and pattern recognition methods for face-related applications, e.g. facial age estimation, facial gender classification, face detection and face recognition, aim to build a robust classifier, e.g. Linear Discriminant Analysis (LDA), Convolutional Neural Network (CNN), Support Vector Machines (SVM), etc., on top of annotated facial regions with pre-defined windows. However, these methods are limited in practical applications. They are, indeed, unable to guarantee the optimal annotated window to achieve the highest classification results [10, 21, 22]. Moreover, these supervised machine learning methods usually require huge amount of training data with richly annotated labels, e.g. bounding boxes in the object detection, segmented regions in the object segmentation problems or even labeled regions of object parts. The manual labeling task is mostly impossible when these systems process large-scale databases, i.e. millions of images.

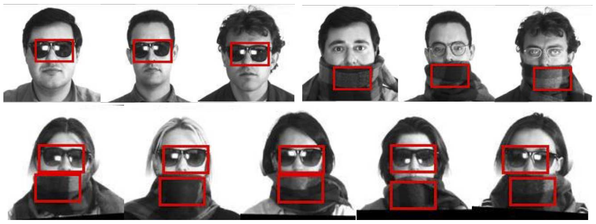


Figure 1. An example of weakly supervised facial analysis with dense hypercolumn features. The first row: the automatically detected discriminative regions between faces with sunglasses and faces with scarves; The second row: mixed-class multiple region detection on faces with both sunglasses and scarves.

Weakly supervised approaches have become prominent machine learning methods recently [14, 1, 16]. These approaches overcome the limitations of the traditional approaches mentioned above since they have the capability to relax the process of empirically selecting fixed sizes for cropping windows and the over-fitting possibilities in selected local optimal hyper-parameters in the classifiers. Furthermore, these methods do not require as detailed labels for their corresponding databases as their classical counterparts. Nguyen et al. [14] introduced a weakly supervised framework to simultaneously localize a single discriminative sub-window of the positive class and to distinguish it from the negative one. The data in this method are only annotated with binary labels indicating the presence of an object without its location. However, it has some critical limitations. Firstly, it is only applicable in the *binary classification* problem. Secondly, due to its optimization constraints, as shown in Eqn. (1), the method is only able to find a discriminative *sub-window in the positive class*, but not in the negative one. Thirdly, the method can localize

*These two authors contributed equally.

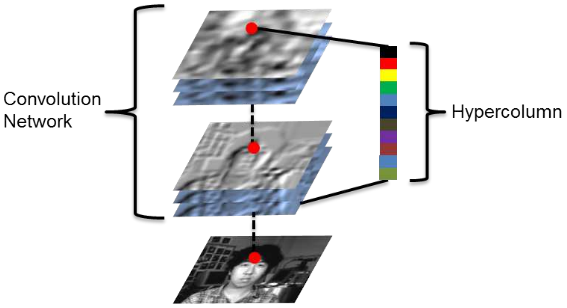


Figure 2. The hypercolumn feature. Given an input image at the bottom, a few feature maps of different layers in CNN can be computed by forward pass. The hypercolumn at one pixel is the vector of the activations of all units that lie above that pixel.

only one sub-window in input images.

Our work proposes a weakly supervised, self-taught method extending the work by Nguyen et al. [14] for simultaneously discriminative localization and classification with advanced properties highlighted in Figure 1. In general, our contributions in this work can be summarized as follows.

Firstly, our proposed approach employs dense hypercolumn features from a deep CNN model (Sections 3.1, 3.2). After pre-processing and training (Section 3.3), our hypercolumn-feature-based, binary classifiers outperform the state-of-the-art method in various facial analysis problems. These deep model based features are shown to extract more discriminative information so that the system can yield higher classification accuracy.

Secondly, instead of finding discriminative sub-windows only in positive class for binary classification tasks, our method can search across all classes for multi-class classification tasks, and has even proven to be effective for mixed-class classification tasks (see Figure 1). To deal with multi-class problems, a simple but non-trivial SVM calibration technique is employed to process numerous one-vs-all binary SVMs (Section 3.4). Therefore, our method can handle more general classification tasks than binary classifiers.

Moreover, our method is able to find any number of sub-windows in each class. The number of windows are pre-defined according to the specific application. This is achieved by removing the key points in the previous discriminative region to search for the next one. So it can enhance the discriminative capability between classes and consistency within classes (Section 3.5).

As the focus of our paper is on a general method for improving classification by localizing discriminative regions, we choose to show that our method can be successfully employed in various facial analysis applications, i.e. occlusion detection, face recognition, gender classification, twins verification and facial attractiveness analysis. Even though in many of these applications, the face is already generally localized, the presence of unnecessary information can throw off traditional methods whereas this unique framework can achieve better results in these applications.

2. Related Work

2.1. Joint Localization and Classification

The idea of joint localization and classification is to simultaneously localize sub-windows, the most discriminative between categories and the most consistent within each category, and to learn a classifier to distinguish them [14]. The data are only annotated with binary labels indicating the presence of an object *without* their locations. This joint detector and classifier can be achieved by solving the SVM optimization problem presented in Eqn. (1).

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \max_{\mathbf{x}\in LS(\mathbf{d}_i^+)} \mathbf{w}^T\varphi_i(\mathbf{x}) + b \geq 1, \forall i, \\ & \max_{\mathbf{x}\in LS(\mathbf{d}_i^-)} \mathbf{w}^T\varphi_i(\mathbf{x}) + b \leq -1, \forall i, \end{aligned} \tag{1}$$

where $\mathbf{d}_i^+$ belongs to the positive training data and $\mathbf{d}_i^-$ belongs to the negative training data. $LS(\mathbf{d})$ denotes the set of all possible sub-windows in image I. And $\varphi(\mathbf{x})$ is the feature vector representing the image sub-window $\mathbf{x}$. The constraints in Eqn. (1) guarantee each positive image has at least one sub-window classified as positive and each negative image to have all sub-windows classified as negative.

Given a query data $\mathbf{d}$, first the sub-window $\hat{\mathbf{x}}$ with the highest SVM score will be computed as follows:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in LS(\mathbf{d})} \mathbf{w}^T\varphi(\mathbf{x}) + b \tag{2}$$

Then if the score $(\mathbf{w}^T\varphi(\hat{\mathbf{x}}) + b)$ is positive, $\hat{\mathbf{x}}$ will be considered as the detected object in $\mathbf{d}$. Otherwise, there is not any object found in $\mathbf{d}$. In previous work by Nguyen et al. [14], the feature vector $\varphi(\mathbf{x})$ is the histogram of visual words associated with descriptors inside the sub-window $\mathbf{x}$. When using this kind of region representation, the search for the sub-window $\mathbf{x}$ with the highest SVM score can be implemented efficiently using the branch-and-bound algorithm [7].

2.2. Hypercolumns

“Hypercolumn” is a term from neuroscience used to describe a set of V1 neurons sensitive to edges at multiple orientations and multiple frequencies arranged in a columnar structure [6]. Bariharan et al. [3] borrowed this term in their work to describe a pixel level deep CNN feature, not only edge detectors but also more semantic units.

Most classification algorithms based on CNN use the output of the last layer as the features. However, this kind of feature contains mostly semantic and not spatial information. That may not be enough information to achieve precise detection. Hypercolumns, on the other hand, are constructed with both earlier CNN layers for localization capability as well as the later layers for semantic information. More specifically, a hypercolumn at a pixel is a vector of activations of all CNN nodes above that pixel as illustrated in Figure 2. These hypercolumn features are used in our proposed approach as the descriptor of key points in facial images for the tasks of searching and localization.

2.3. SVM Calibration

A calibrated multi-class classifier outperforms running binary classifiers independently when the data or features are, to a large extent, overlapping, and hard to separate by single binary classifiers [2]. Calibration is also used frequently in Exemplar SVMs [12] that apply a joint method for calibration.

One popular calibration method is Platt scaling [18] that takes the scores from the classifier and turns it into a probability. Platt scaling uses a sigmoid function,

$$p(y=1|x) = \frac{1}{1+\exp\{af(x)+b\}} \tag{3}$$

where $a, b \in \mathbb{R}$. Though Platt scaling is proposed for binary classification problems, it is able to be further extended to multi-class problems by replacing sigmoid function with softmax function.

3. Our Proposed Method

3.1. Deep Face Model Training

We adopt the architecture proposed by [23] to train a deep CNN model for face recognition. There are four convolution layers with max-pooling layers included in the CNN model. The max-pooling also includes an element-wise maximum between two sub-tensors sliced from the feature map tensor. Two fully-connected layers follow, combining with the final softmax layer, to give a probabilistic distribution over classes, as shown in Figure 3. Dropout with a rate of 0.7 is applied between the two fully-connected layers. Training data are normalized and cropped to 144 × 144 based on facial landmarking points.

To further enrich the input data, the data layer of the CNN randomly crops each image into 128 × 128 pieces. The maximum number of iterations is 2,000,000. The Casia WebFace dataset [20] with 494,414 images of 10,575 subjects are used for training. The testing dataset is Labeled Faces in the Wild [5]. Our trained model performs quite competitively against the state-of-the-art method [20] on randomly picked 3,000 pairs of LFW images with an average precision of 99.15%.

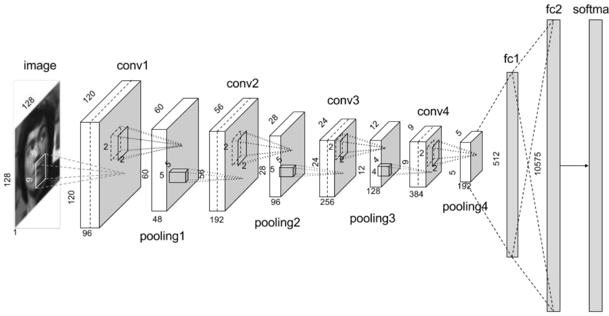


Figure 3. CNN model for face verification.

3.2. Dense Hyper-column Feature Extraction and Clustering

This section describes how to extract hypercolumn features from dense key points and how to quantize these features by clustering. Given an aligned face image F_i, where $i \in \{1...N\}$ indexing through all the images, it is fed into the CNN model described in 3.1 and the outputs of all layers are computed. A dense grid of key points is generated on top of the image. For each key point, p_{ij}, where $j \in \{1...K\}$ indexing through all the key points on ith image, its hypercolumn representation, $\mathbf{h}_{ij}$, is computed by concatenating all the intermediate outputs from feature maps above the location of p_{ij}. Similar to [3], each feature map is resized to the size of the input image and the fully connected layers are treated as a 1 × 1 feature map. Eventually, we have a vector representation for each key point. Unlike [3], in our approach, the sub-vectors extracted from each layer are normalized to have unit length. This is intended for better quantization during the clustering process.

In this work, the hypercolumn is constructed from the output feature maps of *conv1* (96 channels), *conv2* (192 channels), *conv3* (256 channels), *conv4* (384 channels) and *fc1* (512 channels), which results in a 1440 dimensional vector. In the quantization steps, all hypercolumns from all images in the dataset are stacked into a matrix $\mathbf{H} = [\ldots, \mathbf{h}_{ij}, \ldots]$, and k-means clustering is performed based on the ℓ_2 distance. This $\mathbf{H}$ is usually very large, so the approximated nearest neighbors algorithm [13] is employed to accelerate the sample-to-center comparisons. Finally, each $\mathbf{h}_{ij}$ is assigned to a center with an index I_{ij} pointing to that center in the dictionary. Finally, each p_{ij} can be presented as $[x_{ij}, y_{ij}, I_{ij}]$, where x and y are the location coordinates in the ith image.

3.3. Simultaneous Localization and Classification

In this section, we focus on simultaneously learning a sub-window SVM classifier and searching for the most discriminative sub-windows in the positive images among two classes by solving Eqn. (1). The window feature $\varphi(\mathbf{x})$ is the histogram computed by counting the I_{ij} as described in 3.2 inside the window $\mathbf{x}$. A coordinate descent method is applied to solve this non-convex problem. The whole procedure is presented in Algorithm 1, where $FW(\mathbf{d})$ is the full window of $\mathbf{d}$.

Algorithm 1 Single window localization and SVM learning

```
Initialize SVM weights $\mathbf{w}$
Initialize positive features: $\varphi_{i0}(\mathbf{x} \in FW(\mathbf{d}_i^+)), \forall i$
Objective function: $f = \frac{1}{2}\|\mathbf{w}\|^2$
repeat
    Initialize negative features: $\varphi_{i0}(\mathbf{x} \in FW(\mathbf{d}_i^-)), \forall i$
    repeat
        Setup positive and negative constraints using positive and negative features.
        Solve Eqn. (1) to get $\mathbf{w}$ and $b$
        Add maximum violated negative features to negative constraints: $\varphi(\arg\max_{\mathbf{x} \in LS(\mathbf{d}_i^-)} \mathbf{w}^T \varphi(\mathbf{x})), \forall i$
    until The deviation of $\mathbf{w}$ is less than a tolerance
    Replace positive features with maximum positive features $\varphi(\arg\max_{\mathbf{x} \in LS(\mathbf{d}_i^+)} \mathbf{w}^T \varphi(\mathbf{x})), \forall i$
until Change of $f$ is less than a tolerance
Output $\mathbf{w}$ and $b$
Output subwindows: $\hat{\mathbf{x}}_i = \arg\max_{\mathbf{x} \in LS(\mathbf{d}_i)} \mathbf{w}^T \varphi(\mathbf{x}) + b, \forall i$
```

3.4. Extension to Multi-Class with SVM Calibration

So far, the system is able to detect regions in single positive class by a SVM classifier. However, the system is not well designed to deal with data samples with different regions containing multiple classes, e.g. $classes(\mathbf{x}^{(i)}) = \{k_1, k_2, ...\}$. Therefore, we propose two calibration strategies to tune multiple SVMs in an integrated system. Here we present the method step by step. The overall approach is shown in Algorithm 2.

Training data for calibration: Suppose we already trained K binary SVMs in an one-vs.-all manner, one for each of K classes, and we use SVM$_k$ to denote the SVM treating class k as the positive class. Given a training data sample $\mathbf{x}^{(i)}$, we can first extract regions with high SVM scores within the sample. Different strategies can be applied to define a high SVM score. In our experiments, we find two regions with the two highest scores regarding each SVM$_k$. Ignoring the biases, we have a vector of regional SVM scores of data sample i,

$$\mathbf{s}^{(i)} = \begin{bmatrix} \mathbf{w}_1^T \varphi_1(\mathbf{x}^{(i)}) & \ldots & \mathbf{w}_{R_i}^T \varphi_{R_i}(\mathbf{x}^{(i)}) \end{bmatrix}^T \tag{4}$$

where $\mathbf{s}^{(i)} \in \mathbb{R}^{R_i}$ and $R_i \in \mathbb{R}$, denoting the number of regions extracted by all SVMs; $\varphi_j(\mathbf{x}^{(i)})$ denotes the feature vector of the region extracted by an SVM, and $\mathbf{w}_j$ is the corresponding SVM weight vector. Note that the regions extracted by SVM$_k$ may contain nonsense if $k \notin classes(\mathbf{x}^{(i)})$.

The corresponding class indicator vector is

$$\mathbf{y}^{(i)} = \begin{bmatrix} y_1^{(i)} & \ldots & y_{R_i}^{(i)} \end{bmatrix}^T \tag{5}$$

where $y \in \mathbb{R}^{R_i}$ and $y_j^{(i)} = k$ if $k \in classes(\mathbf{x}^{(i)})$ and the region j is extracted by SVM$_k$, else $y_j^{(i)} = 0$.

Goal of calibrated SVM: Our final goal is to assign any extracted region, if it contain a class marker, to the real class. The calibration framework is basically doing a softmax regression over all K classes. The calibration parameters are $\mathbf{a}, \mathbf{b} \in \mathbb{R}^K$ with entry a_k, b_k associated with SVM$_k$.

Algorithm 2 Calibration of Binary SVMs

```
Input: A data sample $\mathbf{x}^{(i)}$.
Output: $classes(\mathbf{x}^{(i)})$ and the set of bounding boxes $B^{(i)}$ containing the marker regions of every entry in $classes(\mathbf{x}^{(i)})$.
procedure EXTRACT CANDIDATE REGIONS
    for each binary SVM$_k$ where $k \in \{1...K\}$ do
        Find high SVM scores $\mathbf{w}_k^T \varphi_k(\mathbf{x}^{(i)})$
        Add high scores to score set $\mathbf{s}^{(i)}$
    end for
end procedure
procedure SVM CALIBRATION
    for each binary SVM$_k$ where $k \in \{1...K\}$ do
        for each entry $s_j^{(i)} \in \mathbf{s}^{(i)}$, where $j \in \{1...R_i\}$ do
            Find region $B_j^{(i)}$ corresponding to $s_j^{(i)}$
            if softmax$(a_k s_j^{(i)} + b_k) > threshold$ then
                Add $B_j^{(i)}$ to output set $B^{(i)}$
                Add $k$ to $classes(\mathbf{x}^{(i)})$
            end if
        end for
    end for
end procedure
```

Training SVM calibration: Similar to the training of a softmax regression, our objective is

$$\min_{\mathbf{a},\mathbf{b}} \quad \sum_i \sum_{j=1}^{R_i} \sum_{k=1}^{K} \frac{\exp\{a_k s_j^{(i)} + b_k\}}{\sum_{k'=1}^{K} \exp\{a_{k'} s_j^{(i)} + b_{k'}\}} - Y_{j,k}^{(i)} \tag{6}$$

where $s_j^{(i)} = \mathbf{w}_j^T \varphi_j(\mathbf{x}^{(i)})$ are the SVM scores without biases, $Y^{(i)} \in \mathbb{R}^{R_i \times K}$ is the ground truth class indicator function, which has multiple different definitions discussed below.

Note that the sum of ground truth class indicator has to be $\sum_{k=1}^{K} Y_{j,k}^{(i)} = 1$ so that it is a probability distribution

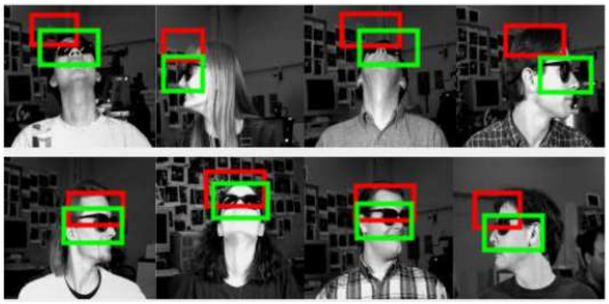


Figure 4. The first row shows challenging cases in face localization where the faces are highly non-frontal, making sunglasses barely visible. The DLC method (red boxes) fails to localize the sunglasses. However, our approach (green boxes) still robustly finds the precise locations. The second row shows clearly that our method still gives a tighter and precise crop, though both the methods find the correct location.

over all classes. Some of the regions, as mentioned previously, may contain nonsense if the data sample doesn't belong to the class where it was extracted. Two strategies, regarding different treatment of these regions, are applied for the calibration.

SVM based Ignoring Strategy (SVM-IS): This strategy simply ignores the nonsense regions, i.e. we only consider the regions extracted by SVMs corresponding to existing classes in a sample. Hence all regions contain the meaningful marker of a class. In other words, instead of $j \in \{1, 2, ..., R_i\}$, we have $j \in \{j' : y_{j'}^{(i)} \neq 0, \forall j' \in \{1, 2, ..., R_i\}\}$ and $Y_{j,k}^{(i)} = \mathbb{1}\{y_j^{(i)} = k\}$.

SVM based Uncertainty Strategy (SVM-US): This strategy treats the ground truth class indicator of nonsense regions as a uniform distribution, instead of an indicator function. That means the system is uncertain where the nonsense regions coming from.

$$Y_{j,k}^{(i)} = \begin{cases} \mathbb{1}\{y_j^{(i)} = k\} & \text{if } y_j^{(i)} \neq 0 \\ \frac{1}{K} & \text{if } y_j^{(i)} = 0. \end{cases} \tag{7}$$

3.5. Extension to Multi-Window Localization

Our proposed framework is also extended to multi-window localization to find multiple discriminative regions and to learn multiple classification SVMs. It is implemented by a simple yet effective strategy. After a new discriminative SVM is learned, the optimal sub-window corresponding to this SVM is computed in each image. Then all the key points in the sub-window are removed so that they will not contribute to the histogram computing during the next learning circle. Then another SVM will be trained on features extracted from sub-windows searched on the updated set of key points. This strategy will be used for $(N-1)$ times to find N sub-windows, where N is a predefined number by the user. The whole pipeline is presented in Algorithm 3.

Algorithm 3 Multi-window localization and SVM learning

```
for $n = 1, \cdots, N$ do
    Initialize SVM weights $\mathbf{w}^{(n)}$
    Initialize positive features: $\varphi_{i0}^{(n)}(\mathbf{x} = FW(\mathbf{d}_i^+)), \forall i$
    Objective function: $f = \frac{1}{2}\|\mathbf{w}^{(n)}\|^2$
    repeat
        Initialize negative features: $\varphi_{i0}^{(n)}(\mathbf{x} = FW(\mathbf{d}_i^-)), \forall i$
        repeat
            Setup positive and negative constraints using positive and negative features
            Solve Eqn. (1) to get $\mathbf{w}^{(n)}$ and $b^{(n)}$
            Add maximum violated negative features to negative constraints:
            $\varphi(\arg\max_{\mathbf{x} \in LS(\mathbf{d}_i^-)} \mathbf{w}^{(n)T}\varphi(\mathbf{x})), \forall i$
        until The deviation of $\mathbf{w}$ is less than a tolerance
        Replace positive features with maximum positive features $\varphi(\arg\max_{\mathbf{x} \in LS(\mathbf{d}_i^+)} \mathbf{w}^{(n)T}\varphi(\mathbf{x})), \forall i$
    until Change of $f$ is less than a tolerance
    Output $\mathbf{w}^{(n)}$ and $b^{(n)}$
    Output sub-windows:
    $\hat{\mathbf{x}}_i^{(n)} = \arg\max_{\mathbf{x} \in LS(\mathbf{d}_i)} \mathbf{w}^{(n)T}\varphi(\mathbf{x}) + b^{(n)}, \forall i$
    Remove key points in $\hat{\mathbf{x}}_i^{(n)}$ of $\mathbf{d}_i, \forall i$
end for
```

4. Experiments

4.1. Single Window Localization and Classification

A single window classification experiment is implemented on the CMU Face Images dataset [1] using Algorithm 1. This data consists of 640 black and white face images of 20 subjects taken with varying pose (straight, left, right, up), expression (neutral, happy, sad, angry), eyes (wearing sunglasses or not). The experiment is setup to simultaneously distinguish faces with and without sunglasses and localize the sunglasses. For a fair comparison, we split the dataset the same way as [14], i.e. training on the first 8 subjects and testing on the last 12 subjects. Overall, there are 254 training images, 126 with sunglasses and 128 without sunglasses, and 370 testing images, 185 with sunglasses and 185 without sunglasses.

Table. 1 shows the classification results of our method measured by accuracy and ROC area. We benchmark our method against several other approaches. They are bag-of-words [15] using a 10-nearest neighbor classifier, SVM-F denoting the traditional SVM in which each image is represented by the histogram of the words in the full image, Efficient Sub-window Search (ESS) [7] and Discriminative Localization and Classification (DLC) [14]. Notice that our method can not only achieve better classification accuracy,

[1] http://archive.ics.uci.edu/ml/datasets/CMU+Face+Images

Table 1. Comparison results on the CMU Face dataset.

Measure	BoW	SVM-F	ESS	DLC	Ours
Acc. (%)	80.11	82.97	86.79	90.0	**94.32**
ROC Area	n/a	0.90	0.94	0.96	**0.99**

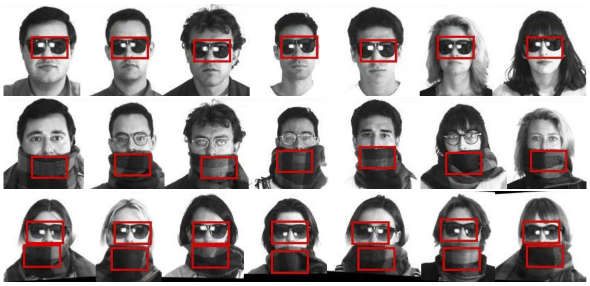


Figure 5. Discriminative facial region of each class indicated with red boxes, implemented on AR dataset [11]. First row is discriminative for the sunglasses. Second row is discriminative for scarf. Third row is discriminative for both wearing the sunglasses and the scarf.

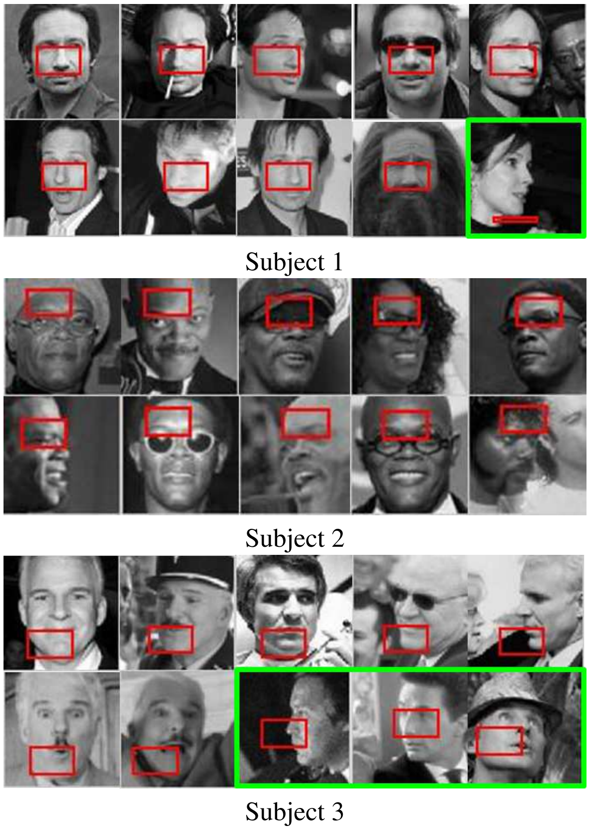
but also find a more precise location of sunglasses. In some extreme cases where DLC fails to find the sunglasses, our method can still give an accurate region hypothesis (see Figure 4).

4.2. Multi-class Localization and Classification with SVM Calibration

A single window multi-class classification experiment is implemented on AR dataset [11], which contain multiple positive classes, i.e. faces with sunglasses and faces with scarf. The classifiers are binary SVMs trained in a one-vs-all manner, calibrated with Algorithm 2. Our method is able to detect the discriminative region as well as classify the images. During training, we apply SVM-US to deal with nonsense regions. In testing, any region with a calibrated softmax value of a positive class over 0.9 is considered a discriminative region and the image is classified to the associated positive class. If an image is classified as the negative class, i.e. bare face, the system will disregard the region selected. Note that neither the binary classifier applied in [14] or the method in previous experiments can do this multi-class task.

Further, we synthesize a set of mixed-class images, i.e. the faces are now wearing sunglasses as well as a scarf for testing. Keeping exactly the same framework and training dataset as the above single window multi-class classification experiment, the system can detect multiple windows with respect to multiple classes.

The result is shown in Figure 5. The system can successfully detect the discriminative regions we want and the results are intuitive. The method achieves 100% accuracy on AR dataset of 120 testing images, i.e. 30 images from each class of sunglasses, scarf, bare face and mixed classes.



Subject 1

Subject 2

Subject 3

Figure 6. Discriminative facial region of each subject. Red windows indicate discriminative regions on subjects, which are very interesting and intuitive. Subjects are discriminative for (1) his pointing nose, (2) the reflection on his forehead and (3) his special patterns on his chin. Examples marked within the green boxes are the failure cases in the testing set. It can be shown that one case of Subject (1) is due to a mislabeling in the dataset itself. It seems that the three cases of Subject (3) are caused by invisibility of the discriminative region.

In the second experiment, we aim to show the efficiency of our proposed calibration strategies on multi-class regions in the application of face recognition. Our method will be compared against other baseline classifiers, i.e. Linear Discriminant Analysis (LDA), Locality Preserving Projections (LPP) [4], Unsupervised Discriminant Projection (UDP) [25], etc. This experiment aims to show the capability of our method in the problem of multi-class classification where the baseline method [14] is unusable. Our method doesn't aim to achieve the state-of-the-art performance on the large-scale face recognition problems as shown in [22, 21].

The system also implements Algorithm 2 on three subjects randomly selected from the CASIA WebFace dataset [26]. The ultimate goal of the method is to automatically find what is the most discriminative facial region of one subject against others. The experiment takes 150 images from each subject for training and 50 images from each for

Table 2. Classification accuracy on three subjects from the CASIA WebFace [26] dataset. Each subject has 50 testing images, examples shown in Figure 6. The best classification accuracy is **97.33%**, achieved by the uncertainty strategy.

Method	Sub 1	Sub 2	Sub 3	Average
LDA	84%	92%	82%	86.00%
LPP [4]	84%	92%	84%	86.67%
UDP [25]	80%	62%	86%	76.00%
Our SVM-IS	96%	100%	84%	93.33%
Our SVM-US	**98%**	**100%**	**94%**	**97.33%**

testing. Two strategies described in 3.4 are implemented for calibration. An upper constraint is imposed on the region size in order to speedup the learning procedure. Given the biological properties of the human face, we restrict the height and the width of possible regions to not exceed 30 and 50 pixels respectively. The classification results are presented in Table 2. Figure 6 gives some examples of detected discriminative regions by the SVM calibrated with the uncertainty strategy.

Briefly, our method can localize a facial region of one subject discriminative to other subjects, yet consistent within the subject class. This approach is significantly better than other classifiers operated on the full face and is robust to different facial poses, illumination condition, occlusion, expressions, and the interference of other faces. The four failure testing cases are due to noise in the dataset or invisibility of regions.

4.3. Gender Classification

Soft biometric classification is very useful in reducing the possible search space for other recognition techniques. To that end, we ran an experiment in gender classification using our proposed method on frontal mugshot style photos. The database was collected from several online sources, including three mugshot databases and a set of Olympic athlete photos. The dataset was balanced between both genders and the three most common ethnicities in the data, i.e. Black, White and Asian. There are a total of 8,748 images (1,458 images in each gender/ethnic group). A subset of 3,870 images (630 images in each gender/ethnic group) was used in training both our method and baseline methods used in the comparison. These images were from one of the four mugshot databases. The rest of the data was used for testing in order to account for any possible dataset bias. All data is aligned using hand clicked eye locations.

The experiment is setup based on Algorithm 1. Again taking the biological properties of face and the searching efficiency into consideration, we restrict the height and the width of possible regions to not exceed 30 and 50 pixels respectively. Figure 7 illustrates the discriminative regions for both male and female examples from the Olympic dataset. The regions selected for males tend to be in a region that indicates the presence or absence of facial hair. This could be a result of the fact that mugshot images of males tend to have facial hair more often than not. However, the method is still able to achieve high accuracy. We report the classification accuracy of out method and compare it with some baselines, i.e. Linear Discriminant Analysis (LDA), SVM and Random Forest (RF), in Table 3. An important note is that even though the other baseline methods do not seem to be able to generalize from mugshot style pictures to Olympic athlete pictures, our method is able to do a much better job, maintaining a more consistent accuracy across datasets.

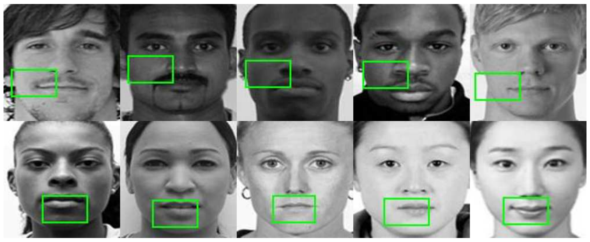


Figure 7. Discriminative facial region of each subject. Green windows indicate discriminative regions for males (top row) and females (bottom row). The discriminative region for males is on a region that tends to have facial hair when it is present. This is a good indicator of gender when it is present but does not help when it is not. The discriminative region for the females tends to be on the lips which is an area of the face that can be distinctive between men and women.

Table 3. Gender classification accuracy (%) on the two mugshot datasets not used in training (M1 and M2) and the Olympic athlete dataset (O). The best overall classification accuracy is **94.09%**, achieved by our weakly supervised method.

Method	M1	M2	O	Overall
LDA	94.92	94.86	83.88	88.58
SVM	91.51	90.67	80.67	85.20
RF	86.68	87.30	73.62	79.25
Ours	**95.60**	**96.03**	**92.90**	**94.09**

4.4. Twins Verification

This experiment addresses the twins verification problem using our unified framework of simultaneously detection and classification presented above. Experiments are conducted on face images of identical twins from the University of Notre Dame ND-Twins database [17] acquired at the 2009 and 2010 Twins Days Festivals in Twinsburg, Ohio. The experiment shows that our proposed method achieve the state-of-the-art performance compared against recent works in [8] and [9] in two separated Twins datasets of 2009 and 2010. The experiments are performed on the category of neutral faces without glasses. There are 90 pairs of identical twins collected in 2009 and 107 pairs of identical twins collected in 2010. For each pair of twins, there are two images ($Twin_A$, $Twin_B$) selected for training, and

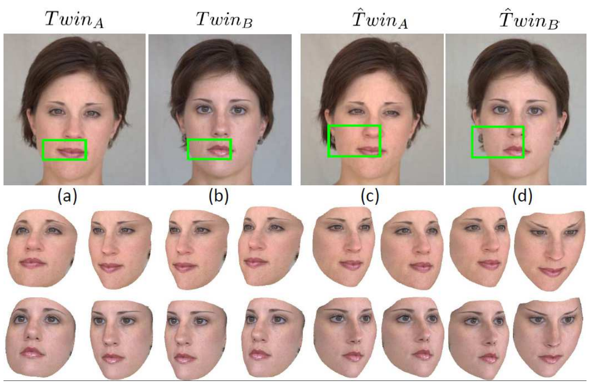


Figure 8. Examples of twin classification: The first row: (a) and (b) are the pair of twins, i.e. $Twin_A$ and $Twin_B$, in training set, (c) and (d) are the corresponding pair of twins, i.e. $\hat{T}win_A$ and $\hat{T}win_B$, in the testing set; green boxes show the discriminative regions for distinguishing $Twin_A$ from $Twin_B$; the second row: synthesized 3D faces of twin $Twin_A$ for training; The third row: synthesized 3D faces of twin $Twin_B$ for training.

two other images ($\hat{T}win_A$ and $\hat{T}win_B$) selected for testing. The accuracy rates in this experiment are computed as the ratio of the number of correct matches to the sum of the number of correct and incorrect matches.

Since our proposed method requires multiple training images to train the system, numerous variations of these two twins images in the training set will be generated. Therefore, the 3D Generic Elastic Model (3D-GEM) method [19] is employed to generate multiple off-angle face images from a single frontal face image as shown in Figure 8. In this method, it is observed that fairly accurate 3D models can be generated by using only one single frontal image in a computationally cheaper way compared to other 3D approaches. For each single face image in the training set, there are 143 synthesized face images varied in yaw angles from -30^o to 30^o in ranges of 5^o and pitch angles from -25^o to 25^o in ranges of 5^o. Table 4 shows the performance obtained of images collected in 2009 and 2010 in the ND-Twins database by aging features based method (Gabor and HOG), facial asymmetry decomposition based methods (SVD-AD and Procrustes-AD) against to our approach. The first two best results are emphasized in bold. Compared to the other twin identification methods, our proposed approach achieves the best performance on both two twins databases. The experiment not only proves that the propose approach is a good solution for twins identification problem but also shows that our method is able to work on synthesized data (which is generated from 3DGEM on still images).

4.5. Facial Attractiveness Analysis

This experiment shows the ability of our method to find more than one discriminative regions in the positive class.

Table 4. Accuracy obtained of images collected in 2009 and 2010 in the ND-Twins database by aging features based method, facial asymmetry decomposition based methods and our approach.

Approaches	2009	2010
Gabor based Facial Aging	**96.7%**	97.7%
HOG based Facial Aging	93.9%	**98.6%**
SVD-AD based Asymmetry	86.1%	96.7%
Procrustes-AD based Asymmetry	85%	95.8%
Our approach	**95.6%**	**99.1%**

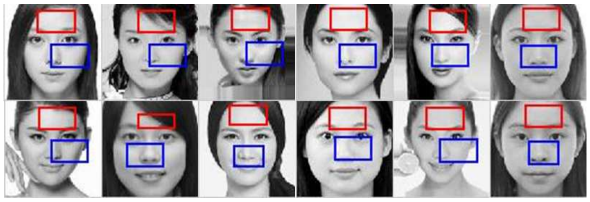


Figure 9. Examples of attractive regions found on the positive class. Blue boxes represent the regions found in the first loop, which are located at the nose and the cheek. Red boxes are the regions found in the second loop which are focusing on the forehead. Overall, these two regions can be seen as the two discriminative areas of facial attractiveness.

The experiment implements Algorithm 3 on an interesting dataset: SCUT-FBP [24]. It is a novel face dataset with attractiveness ratings and developed for automatic facial beauty perception. This dataset contains 500 different Asian female subjects with attractiveness ratings, all of which have been verified in terms of rating distribution, standard deviation, consistency, and self-consistency.

The images are sorted based on their attractiveness score in descending order. The top 200 images are selected as the positive group and the last 200 images as the negative group. Then 150 samples from each group are randomly picked to build the training set. The rest of them are used for evaluation. The number of the searching sub-windows are set to 2. Thus, we want to find what are the two most commonly attractive regions of a female face. It turns out that the region combining the nose and the cheek and the region of forehead are the two most distinguish area representing attractiveness, as illustrated in Figure 9. Our method achieved a 75% classification accuracy.

5. Conclusion

This work has presented a simultaneous localization and classification method using discriminative facial feature in a weakly supervised framework. Our proposed unified framework can be used to solve various face analysis problems. The hypercolumn features extracted from our deep model allows for higher classification results. In addition, our presented SVM calibration can help to deal with multi-class categorization and the mixed-class problems. Finally, our framework has the ability to handle multi-region analysis.

References

[1] A. Bergamo, L. Bazzani, D. Anguelov, and L. Torresani. Self-taught object localization with deep networks. *arXiv preprint arXiv:1409.3964*, 2014.

[2] H. Caesar, J. Uijlings, and V. Ferrari. Joint calibration for semantic segmentation. *arXiv preprint arXiv:1507.01581*, 2015.

[3] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[4] X. He and P. Niyogi. Locality preserving projections. In *NIPS*. MIT Press, 2003.

[5] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[6] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[7] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[8] H. N. Le, K. Seshadri, K. Luu, and M. Savvides. A facial aging approach to identification of identical twins. In *BTAS*, pages 91–98, 2012.

[9] H. N. Le, K. Seshadri, K. Luu, and M. Savvides. Facial aging and asymmetry decomposition based approaches to identification of twins. *Journal of Pattern Recognition*, 48, 2015.

[10] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[11] A. M. Martinez. The ar face database. *CVC Technical Report*, 24, 1998.

[12] D. Modolo, A. Vezhnevets, O. Russakovsky, and V. Ferrari. Joint calibration of ensemble of exemplar svms. *arXiv preprint arXiv:1503.00783*, 2015.

[13] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP*, 2, 2009.

[14] M. H. Nguyen, L. Torresani, F. de la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *Computer Vision, International Conference on*, pages 1925–1932, 2009.

[15] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168. IEEE, 2006.

[16] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free?–weakly-supervised learning with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 685–694, 2015.

[17] P. J. Phillips, P. J. Flynn, K. W. Bowyer, and R. W. V. Bruegge. Distinguishing Identical Twins by Face Recognition. In *Intl. Conf. on Automatic Face and Gesture Recognition and Workshops (FG)*, pages 185–192, 2011.

[18] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[19] U. Prabhu, J. Heo, and M. Savvides. Unconstrained Pose-Invariant Face Recognition using 3D Generic Elastic Models. *TPAMI*, 33(10):1952–1961, oct 2011.

[20] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[21] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[22] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[23] X. Wu, R. He, and Z. Sun. A lightened cnn for deep face representation. *arXiv preprint arXiv:1511.02683*, 2015.

[24] D. Xie, L. Liang, L. Jin, J. Xu, and M. Li. Scut-fbp: A benchmark dataset for facial beauty perception.

[25] J. Yang, D. Zhang, J. Yang, , and B. Niu. Globally maximizing, locally minimizing: Unsupervised discriminant projection with applications to face and palm biometrics. *IEEE TTPAMI*, 29:650–664, 2007.

[26] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.