Customized Expression Recognition for Performance-Driven Cutout Character Animation

Xiang Yu[†] Jianchao Yang[‡] Linjie Luo[‡] Wilmot Li[§] Jonathan Brandt[§] Dimitris Metaxas[♯]
[†]NEC Laboratories America [‡]Snapchat [§]Adobe [♯]Rutgers University

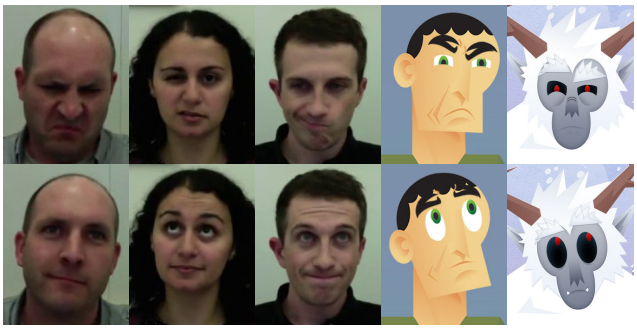
Abstract

Performance-driven character animation enables users to create expressive results by performing the desired motion of the character with their face and/or body. However, for cutout animations where continuous motion is combined with discrete artwork replacements, supporting a performance-driven workflow has some unique requirements. To trigger the appropriate artwork replacements, the system must reliably detect a wide range of customized facial expressions that are challenging for existing recognition methods, which focus on a few canonical expressions (e.g., angry, disgusted, scared, happy, sad and surprised). Also, real usage scenarios require the system to work in realtime with minimal training.

In this paper, we propose a novel customized expression recognition technique that meets all of these requirements. We first use a set of handcrafted features combining geometric features derived from facial landmarks and patch-based appearance features through group sparsity-based facial component learning. To improve discrimination and generalization, these handcrafted features are integrated into a custom-designed Deep Convolutional Neural Network (CNN) structure trained from publicly available facial expression datasets. The combined features are fed to an online ensemble of SVMs designed for the few training sample problem and performs in realtime. To improve temporal coherence, we also apply a Hidden Markov Model (HMM) to smooth the recognition results. Our system achieves state-of-the-art performance on canonical expression datasets and promising results on our collected dataset of customized expressions.

1. Introduction

Animating virtual characters has become a critical task in the production of movies, television shows, computer games, and many other types of digital media. Traditional character animation typically involves keyframing of animation parameters that define how the character moves. While keyframe-based animation gives the user fine-grained control, it requires a large amount of time, effort and skill to produce high quality results. More recently, advances in motion capture technology have enabled performance-driven workflows where users control characters by acting out the desired motions with their faces and/or bodies. This authoring modality allows users to quickly create expressive character animations without having to explicitly define how each individual animation parameter changes over time.



(a) Customized expressions (b) Cutout characters

Figure 1. *Performance-driven cutout character animation.* Actors perform customized expressions in (a) e.g. "disdainful" (top) and "daydreaming" (bottom) to animate the expressions of various cutout characters in (b). Note that the large inter-person expression variations even within the same expression category.

In most performance-driven systems, the continuous motion of the user is directly transferred to the virtual character. While this approach is suitable in some animation scenarios (e.g., creating realistic motion for virtual characters in live action movies), continuous motion alone is not sufficient for all styles of animation. In particular, *cutout* animation is a popular style of 2.5D animation that combines continuous transformations of visual elements with discrete replacements of artwork. These replacements allow animators to drastically alter the appearance of certain visuals and are often used to change the expression of a character (see Figure 1 and Figure 5). Since most existing systems do not support performance-based triggering of artwork replacements, they cannot directly support the creation of cutout character animations.

In this work, we propose a customized facial expression

recognition method that enables authoring of cutout character animations via facial performance . We focus on facial animation since it is a critical component of most character animation scenarios. Our approach addresses the following unique challenges of building a practical performance-driven cutout character animation system:

Wide range of expressions. Expressive cutout animation characters exhibit many different facial expressions that help define the unique personality of the character. It is thus important for the expression recognition algorithm to handle a wide range of expressions. Moreover, since animators often use different expressions for different characters, the algorithm must be flexible enough to handle a customizable rather than predefined set of expressions.

Minimal training. One way to support customized expressions is to allow actors to train the system online to recognize specific expressions. Training frames are recorded in a short period and thus very few. Given the wide range of expressions used in a typical animation, it is important to minimize the required training effort.

Real-time recognition. A key benefit of performance-driven animation is that actors can quickly experiment with different timings and motions by acting out a few variations of a performance and evaluating the resulting animations. To realize this benefit, the animation system should be able to recognize expressions in real-time so that the user receives immediate feedback on the results.

Facial expression recognition is a widely explored topic in computer vision. Significant efforts have been made to boost recognition accuracy through better feature representations [38, 25, 20, 39] and better strategies to discriminate expression categories [3, 1, 21, 23]. However, most of these techniques are designed to recognize just the canonical expressions, i.e. angry, disgusted, scared, happy, sad and surprised. As explained above, a practical performance-driven cutout animation system must support a much wider range of expressions. Moreover, non-canonical expressions often exhibit far more inter-person variations, even within a single expression category, which indicates the need for customized recognition. For example, Figure 1 shows the non-canonical expressions "disdainful" and "daydreaming" performed by three different people who have very different interpretations of these sentiments.

We propose a novel facial expression recognition method that addresses the aforementioned challenges. The input to our algorithm is a set of 1-2 second customized expression frames recorded by a single user. We extract a combination of *handcrafted features* and regularized Deep Convolutional Neural Network (CNN) features for the expression classification. The handcrafted features consist of the geometric features derived from facial landmarks and the patch-based appearance features through group sparsity-based facial component learning. To further boost capabilities of discrimination and generalization, the Deep CNN feature is regularized by the handcrafted features. Then an online ensemble of SVM classifiers is introduced to recognize the customized expressions. We also apply a Hidden Markov Model (HMM) online sequential smoothing to improve the temporal coherence of the recognition results. Our system is evaluated on both canonical and customized expression datasets and achieves state-of-the-art performance.

To summarize, our contributions in this work are:

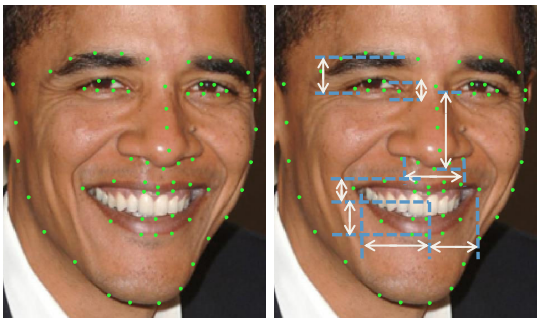
- A customized expression recognition system for performance-driven cutout character animation.
- A novel set of effective features to enable accurate and robust recognition of a variety of customized expressions.
- An online few-shot SVM ensemble with an HMM-based temporal filtering algorithm which addresses the few-training-sample problem and improves temporal coherence of expression prediction.

2. Related Work

From approach point of view, facial expression recognition is generally divided into two mainstreams: emphasizing feature extraction and designing classifiers. Most of the features are handcrafted features, i.e. Gabor wavelets [38, 25], Haar feature [31, 29], Local Binary Patterns (LBP) [20, 39, 26], which are all extracted from patch appearance. Geometric handcrafted features are also proposed in the literature [38, 4]. In contrast to the handcrafted features, the learning based strategies are more and more developed recently, e.g. methods utilizing sparse representations [41, 34, 12, 40, 24]. Some learning based features directly model the dynamic sequential information such as boosted coded dynamics [31]. With powerful representation ability, recently deep Neural Networks are also employed in the expression recognition task [18, 13, 14], especially combining the multi-modality to help improve the effectiveness [9, 8].

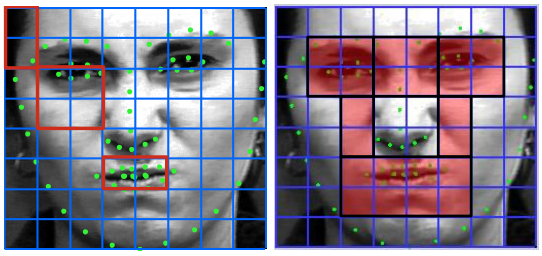
Fusion of features is an important branch of feature representation. Many researchers created a number of fusion algorithms to boost the recognition performance [22, 34, 23, 37]. However, to the best of our knowledge, those fusion methods are based on the above mentioned appearance features, i.e. Gabor, LBP, etc. There may be an upper bound of the performance by combining the appearance features. If we explore the appearance feature fused with Neural Network features, due to the CNN's strong representability [11], an improved performance can be expected. Moreover, our method is an embedded structural fusion, not a simple concatenation, which provides a new channel for feature fusion.

With finely designed expression representations, Support Vector Machines (SVMs) [1, 20, 40] is the most common and effective method for recognition. Some variants are proposed to extend its applicability [2, 15]. There are some other classifier modelings, such as laplacian ordinal regression [19]. The static image based approaches are suspicious



(a) (b)

Figure 2. Geometric feature definition. (a) Facial image with detected facial key points in green dots from a state-of-the-art face alignment method [36]. (b) The defined geometric parameters, left/right eyebrow height, left/right eyelid height, nose height, nose width, upper lip height, lower lip height, left mouth corner to mouth center distance and right mouth corner to mouth center distance.



(a) (b)

Figure 3. Selected region for appearance feature by the facial region selection. (a) The normalized facial image with detected facial key points in green dots, 8x8 patches in blue lines and blocks defined in red rectangles. Images are consistently normalized by aligning facial components, i.e. eyebrows and eyes are normalized into corresponding patches. (b) The selected regions in red. The regions are selected by evaluating on the frequency of each block's being selected based on multiple independent optimization processes.

to perfectly solve the problem because of the lack of utilizing the dynamic temporal information. Thus, many dynamic models are proposed such as Hidden Markov Model (HMM) and its variants [3, 21, 23], and latent conditional random fields [7]. Some other methods model the spatial-temporal cube as a longitudinal atlas [6] or as expressionlets forming a spatial-temporal manifold [14].

3. Proposed Method

We first describe the design of facial features that can capture a wide range of expressions. Then, we present our customized expression recognition framework for cutout character animation.

3.1. Designing Facial Expression Features

We consider three different types of features for representing facial expressions: geometric features, which describe the spatial deformations of facial landmarks; appearance features, which capture the appearance of the most discriminative facial regions for expression recognition; and CNN-based features that we extract from a deep neural network trained to recognize generic facial expressions. We also consider a concatenation of the geometric and appearance features, which we refer to as our handcrafted feature vector.

3.1.1 Geometric Features

To capture deformations caused by the activation of facial muscles, we define geometric features that capture the 2D configuration of facial landmarks (Figure 2). The facial landmarks are provided from some accurate real-time facial feature localization methods [35, 36]. Since expressions are mainly controlled by muscles around the mouth, eyes and eyebrows [5], we focus on features that characterize the shape and location of these parts of the face. Specifically, our features include the following measurements: the left/right eyebrow height (vertical distance between top of the eyebrow and center of the eye), left/right eyelid height (vertical distance between top of an eye and bottom of the eye), nose height (vertical distance between bottom of the nose and center of both eyes), nose width (horizontal distance between leftmost and rightmost nose landmarks), upper lip height (vertical distance between top and center of the mouth), lower lip height (vertical distance between bottom and center of the mouth), left mouth corner to mouth center distance, and right mouth corner to mouth center distance. To ensure that these measurements are consistent across different images, we transform each face into a frontal view (via an affine deformation) and normalize the scale based on inter-ocular distance. Note that a similar set of geometric features has been validated in [4].

3.1.2 Appearance Features by Facial Region Selection

While geometric features capture spatial deformations of facial landmarks, they do not consider the appearance changes caused by such deformations. We define patch-based appearance features using a method inspired by Zhong et al. [40]. First, we partition the face image into a uniform grid of 8x8 image patches, and then we consider all 2×1, 2×2 and 1×2 blocks or regions of patches covering the entire image (Figure 3) allowing overlap. We then compute HoG features on each block and concatenate these features into an integrated feature vector. While we could potentially use this integrated feature vector directly to represent appearance, only a subset of the concatenated HoG features are actually meaningful for distinguishing between different expressions. We use the following data-driven approach to select the best set of features to include.

For the training data, we assume a set of face images, each of which is labeled with one of T expression cate-

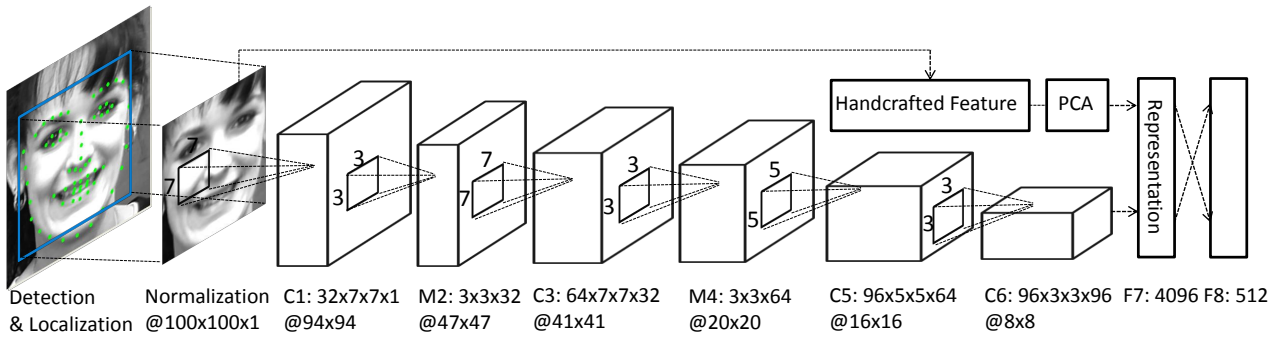


Figure 4. Outline of the fused CNN architecture. The input image is normalized as 100x100 pixels. The convolutional layers, max pooling layers and fully connected layers are denoted as C, M and F followed by the layer number. The number of channels are illustrated by the width of cuboid. They are also denoted as number of filters by horizontal filter size by vertical filter size by channels. Local receptive fields of neurons are illustrated by small squares in each layer.

gories. For each expression category t, we create a set of tuples (x_i^t, y_i^t) where $x_i^t \in \mathbb{R}^M$ is the integrated feature vector for the i^{th} image and $y_i^t \in \{-1, 1\}$ indicates whether the image is a positive ($y_i^t = 1$) or negative ($y_i^t = -1$) example of category t. For each category t, we define a weight vector w^t that represents a separating hyperplane such that $y = (w^t)^T x_i^t + b^t$ is the classification prediction for x_i^t. We define an overall weight matrix $W \in \mathbb{R}^{T \times M}$ for all expression categories by setting its t-th row $W(t,:) = (w^t)^T$. Then, we decompose the matrix into a concatenation of sub matrices $W = [w_{C_1}, ..., w_{C_K}]$, where w_{C_j} corresponds to the weights for the j-th block across all T expression categories and C_j indicates the patches that belong to j-th block.

During training, we try to minimize the classification error over all the expression categories while requiring that W satisfies a structured group sparsity property. We can formulate this problem as multi-task sparse learning, where recognizing each of the T independent expression categories represents the individual tasks. Specifically, we define the problem as follows:

$$\arg\min_{W \in \mathbb{R}^{T \times M}} \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} L(W, X^t, Y^t) + \lambda R(W), \quad (1)$$

where n is the number of training face images, X^t is a matrix with $\{x_i^t\}$ as columns, and Y^t is the concatenated label vector for all examples for category t. $L(W, X^t, Y^t)$ is the loss evaluation over expression t classification and $R(W)$ is the regularization term selecting the block-wise patches. We choose the loss function as logistic loss as shown in Eq. 2.

$$L(W, X^t, Y^t) = \log(1 + \exp(-Y^t \odot (WX^t))) \quad (2)$$

where $\odot$ refers to element-wise product. For regularization, we use $l_{1,2}$ to enforce group sparsity as shown below.

$$R(W) = \sum_{j=1}^{K} \|w_{C_j}\|_2. \quad (3)$$

To solve this multi-task sparse learning problem, an accelerated algorithm can be referred to [30]. After solving the optimization, by thresholding $\|w_{C_j}\|_2$, the facial components are selected for the our classification. Independent such pursuits are conducted based on randomly chosen training sets from the canonical facial expression datasets multiple times. The robust selection is shown in Figure 3 (b) red regions. As expected, the selected regions are surrounding important facial areas, such as eyes, eye brows, and mouth. Once the facial regions are fixed, LBP and HoG features are extracted from each region and the final appearance feature is obtained by concatenating all of them.

3.1.3 Standard and Fused CNN Features

By combining the geometric and appearance features into a single handcrafted feature vector, we can capture much of the relevant variation across different expressions. However, recent results have shown that the features extracted from deep CNN can also be useful for a variety of image understanding tasks. We experimented with two CNN structures for defining facial expression features.

CNN. This structure consists of multiple convolutional layers followed by max-pooling layers and several full-connected layers as in [11]. The network parameters are detailed in the bottom path of Figure 4 where "C" denotes convolutional layers, "M" denotes max pooling layers and "F" denotes fully connected layers. The softmax layer is not shown in the figure.

Fused CNN (f-CNN). Since handcrafted features typically demonstrate good generalization behavior, we introduce the fused CNN (f-CNN) structure in Figure 4. The proposed f-CNN has two paths: the top path extracts handcrafted features followed by PCA dimensionality reduction, and the bottom path is the standard CNN; the two paths are fused in the fully connect layer F7. Trained from scratch, the f-CNN learns a deep model combines the best of both worlds; CNN-based features that perform well on constrained recognition tasks and handcrafted features that generalize

well to more categories.

Network Training. The overall training involves two parts. One is the convolutional layers, which act as feature extractor. The other is the fully-connected layers, acting as classifier. To train the convolutional layers, the more data included, the less likely the training is overfitting. Thus, 3DFE and 4DFE as mentioned above may be included. As the readers may suspect, including other datasets is not fair because the compared methods in the experiments do not use other datasets for training. Thus, our training is: apply sufficient many datasets to train a network and keep the convolutional layers for feature extractor. To train each of the fully-connected layers, only the training data in that database is applied, which guarantees that the classification comparison is consistent.

To learn deep CNN models that generalize well across a wide range of expressions, ideally we need sufficient training data with a large number of expression categories. Unfortunately, all publicly available facial expression databases only include six canonical expressions — angry, disgusted, scared, happy, sad, and surprised. A CNN trained with these datasets would be tuned to classify these six expressions, which may hurt its ability to generalize to customized expressions. Moreover, creating ground truth datasets with additional expressions may be challenging, since non-canonical expressions tend to have larger inter-person variations that make accurate labeling a difficult task. Thus, our approach is to use existing canonical expression databases, including CK+ [10, 16], MMI [27], 3DFE [33] and 4DFE [32], for training both the standard and fused CNN models. As shown in Section 4, we indeed find that, while the standard CNN model perform extremely well on the six canonical expressions, exceeding state-of-the-art methods significantly, they do not generalize well to arbitrary customized expressions, especially when we use the last fully connected layer output as our features. Instead, we use the outputs of C6 for standard CNN, and the combination of C6 outputs and handcrafted features for f-CNN. Our experimental results show that f-CNN performs the best for both canonical expression recognition and customized expression recognition tasks.

To train our CNN models, we augment the canonical expression datasets by generating variations of each face via cropping, horizontal flipping, and perturbing aspect ratios. In the end, we obtain around 1 million data samples from the existing facial expression datasets mentioned above. We normalize the detected faces to 100x100 as the inputs to our network models. Considering the forward propagation, the output of each layer is the linear combination of the inputs non-linearly mapped by an activation function:

$$u^{k+1} = f((\mathcal{W}^{k+1})^T u^k) \tag{4}$$

where u^k indicates the k^{th} layer output, $\mathcal{W}^k$ indicates the weights that connect to each output node and $f(.)$ is the nonlinear activation function, for which we use rectified linear unit (ReLu) as in [11]. To update the weights of each layer, back propagation is applied:

$$\delta^k = (\mathcal{W}^k)^T \delta^{k+1} \frac{\partial f}{\partial u^k}, \tag{5}$$

where δ^k is the increment of weights at layer k. For training the f-CNN, we split the weights connecting F7 into two parts: weights for the handcrafted features $\mathcal{W}_h^7$ and weights for C6 $\mathcal{W}_c^7$. We initialize $\mathcal{W}_c^7$ to 0 and only update the weights connecting F7 and F8 according to the handcrafted feature inputs. Upon convergence, we fix $\mathcal{W}_h^7$ and update the whole CNN network. In this way, the CNN generates features that are complementary to the handcrafted features and improve the overall classification accuracy. As mentioned before, we then combine the handcrafted features with the output of C6 as our f-CNN feature.

3.2. Cutout Character Animation

Here, we describe a customized expression recognition framework that uses the features described above to support performance-driven cutout character animation. In our approach, an animator first demonstrates all the customized expressions that the system should recognize by recording a few seconds of video for each expression. These demonstrations act as training data for a set of SVM-based ensemble classifiers, one for each expression. To animate a character, the animator simply performs the desired motion. The ensemble classifiers recognize the current expression in real-time, and the system uses the detected expression to trigger the appropriate artwork replacements in the character. We also apply continuous deformations to the character based on the motion of the tracked facial landmarks on the actor.

3.2.1 Online Classifier Ensemble Learning

For each of the T expressions that the user demonstrates to the system, we train an ensemble classifier as follows. We take all the n_i training frames from the demonstration of expression i as positive samples and treat the recorded frames from all the other expressions as negative samples. Note that n_i is typically far less than $\sum_{j \neq i} n_j$. Thus, we randomly split all the negative samples into $N = \frac{\sum_{j \neq i} n_j}{n_i}$ piles, each of which has approximately n_i samples, and then train N independent SVM classifiers. We repeat this procedure independently t times to produce tN classifiers, which we combine linearly to obtain the final ensemble classifier for expression i:

$$F_N(x) = \sum_{j=1}^{tN} \omega_j f_j(x), \tag{6}$$

where f_j is the j-th SVM classifier trained using the positive samples and the j-th pile of negative samples, and ω_j is its associated weight that is initialized as $\frac{1}{tN}$. During online testing, among the tN classifiers, some of the classifiers may produce results that conflict to the final classification

Table 1. Expression recognition average accuracy on geometric feature (Geo), appearance feature (App), the geometric and appearance combined handcrafted feature (HC), CNN and fused CNN (f-CNN) feature testing on CK+ and MMI datasets. Some state-of-the-art methods, i.e. ITBN, CSPL and LFEA are also listed for comparison.

Method	CK+							MMI						
	Angry	Disgust	Fear	Happy	Sad	Surprise	Ave.	Angry	Disgust	Fear	Happy	Sad	Surprise	Ave.
Geo	0.84	0.76	0.58	0.88	0.66	0.75	0.81	0.35	0.75	0.45	0.92	0.85	0.94	0.71
App	0.87	0.96	0.97	0.87	0.93	0.87	0.91	0.62	0.80	0.48	0.95	0.84	0.97	0.78
HC	0.96	0.97	0.95	0.96	0.99	0.90	0.96	0.62	0.97	0.67	1.00	0.96	1.00	0.87
ITBN [28]	0.91	0.94	0.83	0.89	0.76	0.91	0.87	0.47	0.55	0.57	0.71	0.66	0.63	0.60
CSPL [40]	0.71	0.95	0.81	0.95	0.88	0.98	0.88	0.50	0.79	0.67	0.83	0.60	0.89	0.71
LFEA [6]	0.95	0.98	0.95	0.99	0.97	0.99	0.97	0.92	0.95	0.94	0.97	0.92	0.94	0.94
CNN	0.98	0.98	0.99	0.99	0.98	0.99	0.99	**0.99**	0.98	0.99	0.99	0.97	0.98	0.98
fused-CNN	0.98	**0.99**	0.99	0.99	0.98	0.99	0.99	0.98	0.98	0.99	0.99	**0.98**	**0.99**	**0.99**

output of F_N. To give our classifier a certain amount of online adaptation ability, we penalize those violating classifiers by decreasing their weights with a small amount of decay β,

$$\omega_j = (1-\beta)\omega_j. \tag{7}$$

Then all the weights of tN classifiers are normalized to unit sum for next iterations, i.e.,

$$\omega_j = \frac{1}{\sum_k \omega_k}\omega_j. \tag{8}$$

By adjusting the contributions of the ensemble of classifiers, our algorithm can achieve robustness to slight mismatches between the few recorded training samples and the same expression demonstrated in a performance. Note that the proposed ensemble of classifiers can be regarded as a generalization of exemplar-based SVM [17] in order to gain some robustness in case of scarce training samples.

3.2.2 Temporal Smoothing with HMM

Without considering temporal information, frame-by-frame classification using the ensemble classifier could produce jittering artifacts (i.e., flipping rapidly between two or more expressions). To smooth the classification results, we apply an online sequential Hidden Markov Model (HMM). The HMM maximizes the joint probability of the current hidden state s_t and all the previous observations $x_{\{1,2.,...,t\}}$. Here, the hidden state s_t is the underlying expression category while the data observations are the captured facial expressions. We denote the joint probability as $\alpha(s_t) = p(s_t, x_{\{1,2,...,t\}})$. By Bayesian inference, the recursion function of updating the joint probability is shown bellow.

$$\alpha(s_t) = p(x_t|s_t)\sum_{s_{t-1}} p(s_t|s_{t-1})\alpha(s_{t-1}) \tag{9}$$

where $p(x_t|s_t)$ is the expression recognition posterior and $p(s_t|s_{t-1})$ is the state transition probability. In the transition matrix, for each non-neutral expression, the probability of a self-transition (i.e., remaining in the same expression) and a transition to the neutral expression are the same. In addition, transitions from the neutral expression to every other non-neutral expression are equally likely. The probability of a self-transition from the neutral expression is independent. Between one non-neutral expression and another non-neural expression, we always assume there are neutral frames. Thus, the transition matrix contains 4 independent variables. It can be obtained through cross validation with multi-dimensional line search. For the posterior $p(x_t|s_t)$, according to Bayes' rule, $p(z_t|s_t) \propto p(s_t|z_t)$ (uniform prior on all customized expressions), where the likelihood $p(s_t|z_t)$ can be approximated by converting our classifier outputs in Equation 6 into probabilities with the softmax function.

4. Evaluation

In this section, we evaluate our algorithm on both canonical expression recognition datasets including CK+ [10, 16] and MMI [27], as well as our customized expression recognition dataset for cutout character animation.

4.1. Evaluation on Canonical Expression Datasets

The CK+ dataset contains 327 labeled expression sequences, containing seven expressions, i.e., anger, contempt, disgust, fear, happiness, sadness and surprise. For single image based expression detection, we exclude the neutral expressions from all the categories as the state-of-the-art methods do [40, 6]. For the remaining 5180 images, we conduct a 10-fold cross validation by randomly dividing the dataset into equal 10 folds. The performance is evaluated as the average over all the 10 experiments. The MMI dataset includes 205 video sequences from 30 subjects with different ages, gender and ethnicity. Six canonical expressions are defined as anger, disgust, fear, happiness, sadness and surprise. Similar to CK+, a 10-fold cross validation is conducted on MMI for evaluation. To make all comparisons consistent, we remove the contempt category from CK+ to form the same 6 canonical expression categories.

We evaluate each component in our combined feature set on these two datasets and summarize the results in Ta-

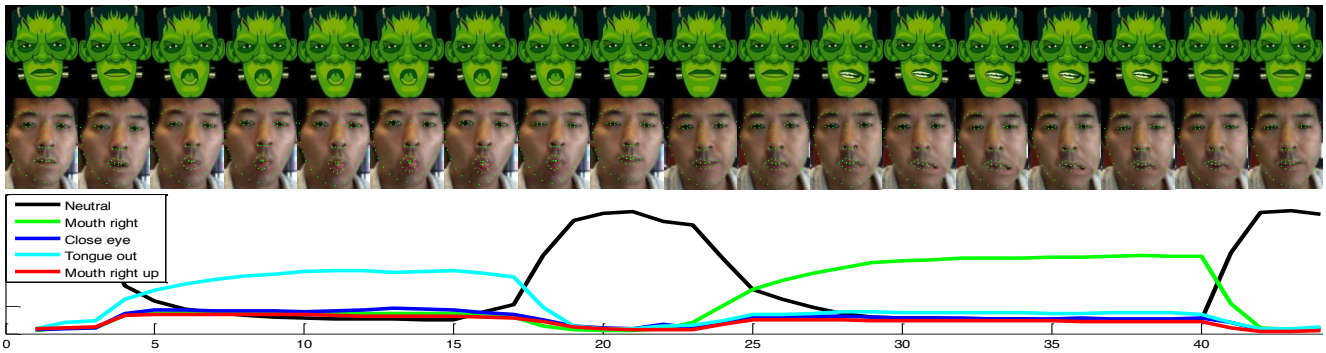


Figure 5. A cutout character animation generated by our system based on a test facial performance. The output probabilities of the trained expressions (neutral, mouth right, close eye, tongue out and mouth right up) for each selected frame are plotted in colored lines.

ble 1. The combined geometric feature and appearance feature produces significant performance boosted from either geometric or appearance feature alone. Even compared to some state-of-the-art methods listed in the table [28, 40, 6], the handcrafted features already achieve competitive performance. Using our CNN and f-CNN features, the performance on both CK+ and MMI improves significantly, which demonstrates CNN's capability in the canonical expression recognition task. The f-CNN features improve the result of CNN marginally since both are working extremely well. However, the advantages of f-CNN becomes clearer in the following section when being evaluated on the customized expression dataset.

4.2. Evaluation on Customized Expression Dataset

In order to evaluate our framework on customized facial expression task, we collect a customized expression dataset under the following protocol. Our dataset contains 30 subjects with different races and ages, 53% of which are female. Each subject contains 5 to 10 training sequences. In total, there are 186 training sequences and 30 testing sequences. Moreover, each testing sequence includes 2000 to 3000 frames, in which each expression is guaranteed to appear more than twice. Overall, there are around 80,000 frames for the whole dataset.

Comparing different features. We first evaluate different subsets and variants of the proposed feature set on the collected dataset with the same classification method in Section 3.2.1. We summarize the comparison results in Table. 2 in terms of precision, recall, F1 score and correction ratio (C-Ratio). The correction ratio is defined as the number of incorrect detected expression intervals over the number of groundtruth intervals that fail to yield a higher-than-threshold overlap with a groundtruth expression interval. For each metric, we show the mean and standard deviation across the test dataset. Note that our handcrafted feature achieves higher precision and recall compared to the CNN feature, which achieves almost perfect results on CK+ and MMI. This suggests that CNN training is overfitted to the canonical facial expressions and thus generalizes poorly to the other customized expressions. This is evident as c6 layer feature is much better than f7 layer feature, where the latter is more tuned for recognizing the six canonical expressions. Simply combining handcrafted feature with CNN feature, however, does not improve the performance (similar F1 scores and correction ratios). In contrast, with our f-CNN structure, we can learn features that are complementary to handcrafted features and the fused feature outperforms both.

Table 2. Precision/Recall, F1 score and correction ratio (C-Ratio) comparison on geometric feature (Geo), appearance feature (App), handcrafted feature (HC) combining Geo and App, CNN feature of C6 layer (CNN-c6), CNN feature of F7 layer(CNN-f7), simple combination HC + CNN-fc7 and HC + CNN-c6, and our fused CNN (f-CNN) feature. The classifier for all features is HMM.

Feature	Precision	Recall	F1 Score	C-Ratio
Geo	0.66±0.14	0.63±0.13	0.65	0.19±0.16
App	0.85±0.08	0.85±0.11	0.85	0.13±0.10
HC	0.86±0.08	0.89±0.10	0.87	0.12±0.10
CNN-f7	0.79±0.11	0.78±0.13	0.79	0.25±0.20
CNN-c6	0.82±0.08	0.79±0.17	0.80	0.15±0.15
HC+CNN-f7	0.87±0.06	0.84±0.13	0.85	0.14±0.14
HC+CNN-c6	0.89±0.05	0.85±0.11	0.87	0.12±0.11
f-CNN	**0.90±0.06**	**0.89±0.09**	**0.89**	**0.10±0.09**

Table 3. Precision/Recall, F1 score and correction ratio (C-Ratio) comparison on kNN, ensemble of SVMs (eSVM), HMM with observation from kNN (HMM-kNN) and HMM with observation from ensemble of SVMs (HMM-eSVM). The features for all classifiers are the f-CNN feature.

Classifier	Precision	Recall	F1 Score	C-Ratio
kNN	0.85±0.08	0.81±0.19	0.83	0.17±0.14
eSVM	0.86±0.13	0.81±0.15	0.83	0.11±0.09
HMM-kNN	0.86±0.08	0.89±0.10	0.87	0.13±0.11
HMM-eSVM	**0.90±0.06**	**0.89±0.09**	**0.89**	**0.10±0.09**

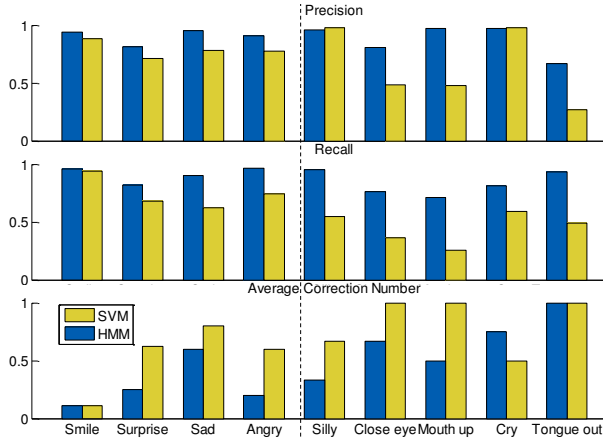


Figure 6. Single expression comparison of canonical expressions and customized expressions. The precision, recall and average correction number are listed among smile, surprise, sad, angry, silly, close eye, mouth up, cry and tongue out 9 expressions. The first 4 are canonical expressions and the last 5 after the vertical dash line are the customized ones.

Comparing different classifiers. To justify our classifier choice, we compare it to ensemble of SVMs (denoted as eSVM) as well as a baseline classifier k-nearest neighbor (kNN). We list the comparison results in Table 3 for all different combinations of these techniques (eSVM, kNN, HMM-kNN and HMM-eSVM) using f-CNN features. While kNN and eSVM show little difference in Precision/Recall, eSVM is better in the correction ratio, which indicates that e-SVM predicts better expression occurrences than kNN does. After combining with HMM, significant improvements are achieved both from kNN to HMM-kNN and from eSVM to HMM-eSVM (more than 4% in terms of F1 score). These prove that the HMM play a positive role in boosting the performance by incorporating the temporal coherence prior. A sample sequence of customized expression recognition for animation is shown in Figure 5.

4.3. Discussion

While the canonical expressions among different people exhibit large degrees of consistencies, customized expressions can have very large inter-person variations (Figure 1). This presents a big challenge for collecting labeled data for a large number of expression categories in order to learn features that are more effective to arbitrary expression recognition, even for targeting user-specific customized expression recognition. In our collected dataset, there are user expressions that belong to both the canonical categories, as we did not constrain the user on what expressions to perform, and the customized or even unnameable expressions. We select the most common canonical expressions from the video dataset, including smile, surprise, sad and angry, and some recognizable user-specific expressions, including silly, close eye, mount lift up, cry and tongue out. We summarize the results on these nine expressions in Figure 6, where the left four are canonical expressions and the right five are user defined expressions. Comparing the performance between the two groups, the correction numbers in the left group are notably smaller than the right group, meaning that the canonical expressions can be recognized with better performance even when our training is totally adaptive to specific users. This is not surprising, as our f-CNN features are trained using the canonical expression face dataset. But it indeed suggests that coming up with facial expression dataset with more variety or a better method that has better generalization ability is important for real-life expression recognition and thus in general for HCI.

5. Conclusion

In this paper, we provide an initial investigation over the customized expression recognition for performance-driven cutout character animation. We propose several types of features including handcrafted features by combining geometric features derived from facial landmarks and region-based appearance features selected with sparse learning, and Deep CNN features learned with regularization from the handcrafted features. We demonstrate that the proposed features can achieve state-of-the-art performance on conventional canonical expression recognition benchmarks. Then an online ensemble SVM classifiers is proposed to recognize customized expressions from few samples. A sequential HMM online smoothing is applied to further boost the recognition performance by incorporating temporal coherency. Experiments on our collected customized expression dataset demonstrate promising results. For future work, we plan to collect even more customized expression data from professional animators for the study of expression recognition in a wide range. We believe this will benefit the Human Computer Interaction (HCI) research as a whole. We also would like to investigate the robustness of our algorithm in terms of recording environment changes for particular users, e.g., a user may record his defined expressions under one environment setting, and would like to use them later in different environment settings.

References

[1] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: Machine learning and application to spontaneous behavior. In *CVPR*, pages 568–573, 2005.

[2] S. Chew, S. Lucey, P. Lucey, S. Sridharan, and J. Cohn. Improved facial expression recognition via uni-hyperplane classification. In *CVPR*, pages 2554–2561, 2012.

[3] I. Cohen, N. Sebe, L. Chen, A. Garg, and T. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *Computer Vision and Image Understanding*, 91(1-2):160–187, 2003.

[4] H. Dibeklioglu, A. Salah, and T. Gevers. Like father, like son: Facial expression dynamics for kinship verification. In *ICCV*, pages 1497–1504, 2013.

[5] G. Duchenne. Mecanisme de la physionomie humaine, 1862.

[6] Y. Guo, G. Zhao, and M. Pietikainen. Dynamic facial expression recognition using longitudinal facial expression atlases. In *ECCV*, pages 631–644, 2012.

[7] S. Jain, C. Hu, and J. Aggarwal. Facial expression recognition with temporal modeling of shapes. In *ICCVW*, pages 1642–1649, 2011.

[8] S. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, R.Ferrari, M. Mirza, D. Warde-Farley, A. Courville, P. Vincent, R. Memisevic, C. Pal, and Y. Bengio. Emonets: Multimodal deep learning approaches for emotion recognition in video. In *arXiv preprint arXiv:1503.01800*, 2015.

[9] S. Kahou, C. Pal, X. Bouthillier, P. Froumenty, C. Gulcehre, R. Memisevic, P. Vincent, A. Courville, and Y. Bengio. Combining modality specific deep neural networks for emotiokn recognition in video. In *ACM International Conference on Multimodal Interaction*, 2013.

[10] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *FG*, pages 46–53, 2000.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[12] Y. Lin, M. Song, D. Quynh, Y. He, and C. Chen. Sparse coding for flexible robust 3d facial expression synthesis. 32(2):76–88, 2012.

[13] M. Liu, S. Li, S. Shan, and X. Chen. Au-aware deep networks for facial expression recognition. In *FG*, pages 1–6, 2013.

[14] M. Liu, S. Shan, R. Wang, and X. Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *CVPR*, pages 1749–1756, 2014.

[15] P. Liu, J. Zhou, I. Tsang, Z. Meng, S. Han, and Y. Tong. Feature disentangling machine - a novel approach of feature selection and disentangling in facial expression analysis. In *ECCV*, pages 151–166, 2014.

[16] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete expression dataset for action unit and emotion-specified expression. In *CVPR Workshops*, pages 94–101, 2010.

[17] T. malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[18] S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza. Disentangling factors of variation for facial expression recognition. In *ECCV*, pages 808–822, 2012.

[19] O. Rudovic, V. Pavlovic, and M. Pantic. Multi-output laplacian dynamic ordinal regression for facial expression and intensity estimation. In *CVPR*, pages 2634–2641, 2012.

[20] C. Shan, S. Gong, and P. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27:803–816, 2009.

[21] L. Shang and K.-P. Chan. Nonparametric discriminant hmm and application to facial expression recognition. In *CVPR*, pages 2090–2096, 2009.

[22] X. Tan and B. Triggs. Fusing gabor and lbp feature sets for kernel-based face recognition. In *FG*, pages 235–249, 2007.

[23] H. Tang, M. Hasegawa-Johnson, and T. Huang. Non-frontal view facial expression recognition based on ergodic hidden markov model supervectors. In *ICME*, 2010.

[24] U. Tariq, J. Yang, and T. Huang. Multi-view facial expression recognition analysis with generic sparse coding feature. In *ECCV Workshop*, 2012.

[25] Y. Tian, T. Kanade, and J. Cohn. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *FG*, pages 229–234, 2002.

[26] M. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer. Meta-analysis of the first facial expression recognition challenge. *IEEE TSMC-B*, 42(4):966–979, 20012.

[27] M. Valstar and M. Pantic. Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In *LREC Workshop on Emotion*, pages 65–70, 2010.

[28] Z. Wang, S. Wang, and Q. Ji. Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. In *CVPR*, pages 3422–3429, 2012.

[29] J. Whitehill, M. Bartlett, G. Littlewort, I. Fasel, and J. Movellan. Towards practical smile detection. *IEEE TPAMI*, 31(11):2106–2111, 2009.

[30] C. Xi, P. Weike, J. Kwok, and J. Carbonell. Accelerated gradient method for multi-task sparse learning problem. In *ICDM*, pages 746–751, 2009.

[31] P. Yang, Q. Liu, and D. Metaxas. Boosting coded dynamic features for facial action units and facial expression recognition. In *CVPR*, pages 1–6, 2007.

[32] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *FG*, pages 1–6, 2008.

[33] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato. A 3d facial expression database for facial behavior research. In *FG*, pages 211–216, 2006.

[34] Z. Ying, Z. Wang, and M. Huang. Facial expression recognition based on fusion of sparse representation. 6216:457–464, 2010.

[35] X. Yu, J. Huang, S. Zhang, W. Yan, and D. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *ICCV*, 2013.

[36] X. Yu, Z. Lin, J. Brandt, and D. Metaxas. Consensus of regression for occlusion-robust facial feature localization. In *ECCV*, 2014.

[37] T. Zavaschi, A. B. Jr., L. Oliveira, and A. L. Koerich. Fusion of feature sets and classifiers for facial expression recognition. *Expert Systems with Applications*, 40(2):646–655, 2013.

[38] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *FG*, pages 454–459, 1998.

[39] G. Zhao and M. Pietiainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE TPAMI*, 29(6):915–928, 2007.

[40] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. Metaxas. Learning active facial patches for expression analysis. In *CVPR*, pages 2562–2569, 2012.

[41] M. Zhou, K. Veon, S. Mavadati, and J. Cohn. Facial action unit recognition with sparse representation. In *FG*, pages 336–342, 2011.