Using Synthetic Data to Improve Facial Expression Analysis with 3D Convolutional Networks

Iman Abbasnejad[⋆,†], Sridha Sridharan[⋆], Dung Nguyen[⋆], Simon Denman[⋆],
Clinton Fookes[⋆], and Simon Lucey[†]

Queensland University of Technology[⋆]
{s.sridharan, d.nguyen, s.denman, c.fookes}@qut.edu.au

Carnegie Mellon University[†]
{imanaba,slucey}@andrew.cmu.edu

Abstract

Over the past few years, neural networks have made a huge improvement in object recognition and event analysis. However, due to a lack of available data, neural networks were not efficiently applied in expression analysis. In this paper, we tackle the problem of facial expression analysis using deep neural network by generating a realistic large scale synthetic labeled dataset. We train a deep 3-dimensional convolutional network on the generated dataset and empirically show how the presented method can efficiently classify facial expressions. Our method addresses four fundamental issues: (i) generating a large scale facial expression dataset that is realistic and accurate, (ii) a rich spatial representation of expressions, (iii) better spatiotemporal feature learning compared to recent techniques and (iv) with a simple linear classifier our learned features outperform state-of-the-art methods.

1. Introduction

Facial expression analysis is a challenging problem and has received increasing attention from computer vision researchers due to its potential in a number of applications such as human computer interaction, behavioral science and marketing. Facial expressions can be coded and defined using facial Action Units (AU) and the Facial Action Coding System (FACS), which was first introduced by Ekman et al. [17]. Typically, facial AU analysis can be done in four steps: (i) face detection and tracking; (ii) alignment and registration; (iii) feature extraction and representation; and (iv) AU detection and expression analysis. Due to the recent advances that have been made in the face tracking and alignment steps, most approaches focus on feature extraction and classification methods (interested readers may refer to [2, 29, 41] for comprehensive reviews).

Generally an ideal automated Action Unit recognition system should consist of: (i) Spatial feature representation: which must be efficient and be able to generalize to any arbitrary subject regardless of the recording environment and (ii) Spatio-temporal modeling: that should extract and learn all the temporal correlations and dynamics among the video frames.

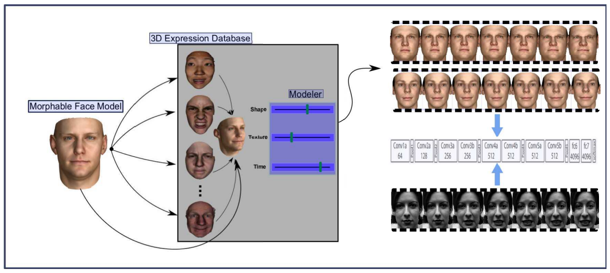


Figure 1: Our proposed model. We are able to synthetically generate a large scale facial expression dataset that enables us to train a deep neural network. Once we fit the face template on the scan faces we estimate the expressions parameters and generate different sequences with different facial textures in different lengths. We first pre-train our model on the synthetic faces and then fine tune it on the real data.

One method to address the above issues is to train and test separate classifiers with each subject to discriminate positive examples from negative ones. In particular, these methods are mostly based on finding the best classifier on the testing samples according to the mismatch between the distribution of training and testing samples [14]. One problem with this approach is that, for each subject an enormous quantity of training data is required to train the best classifier. In order to tackle this limitation, many methods use data from multiple subjects. However, when a classifier is trained on all training subjects it cannot perform efficiently on the unseen test subjects. The main reason of such a problem is that the spatial and temporal properties are varied among different videos and the current classifier and feature representation techniques are not able to fully capture these properties.

One idea is to utilize a rich feature representation, i.e. Deep Convolutional Neural Networks of the input examples in order to improve the detection accuracy. Although deep neural architectures outperform other feature representation methods in many computer vision applications, in the area of temporal analysis, utilizing only deep features is not sufficient [1, 3, 4, 40, 45]. To overcome this limitation Tran et al. [40] proposed to learn spatio-temporal features using a deep 3D Convolutional Network (C3D). They train a deep convnet on a large scale labeled dataset and show that their C3D architecture outperforms other event detection techniques. This approach raises the question of whether such a model can be applied to expression analysis. The most recent approach which used C3D for expression analysis is [32] and they show that C3D can significantly improve the expression classification performance. However as is shown in Tran et al. [40] the performance of C3D is highly influenced with the small amount of training data. Generally, there are limitations with applying C3D effectively on the problem of expression analysis; the amount of labeled instances available in expression analysis for training a deep network is limited; generating a large scale dataset on expression analysis is time consuming and requires special facilities and laboratories; asking a large number of participants for different expressions is expensive and due to the head pose variation of the participants the performance of deep neural networks may be affected.

In this work, in order to tackle the problem of expression recognition we develop an end-to-end model for efficient expression analysis. At the core of this model is a C3D network that learns spatial representation of expressions and the spatiotemporal information among frames. In order to address the limitation of the lack of sufficient training data, we parametrically create and generate accurate faces that are able to deform naturally for different action units and expressions over time. This framework enables us to synthetically create different action units and expressions. The novelty of our method enables us to generate a large scale synthetic facial expression dataset that helps us to train neural networks. Figure 1 shows an overview of our method.

2. Related Works

In this section we review some recent advances that use deep networks for facial expression analysis.

2.1. Feature Extraction

Feature extraction is a crucial step in facial expression analysis and plays an important role in obtaining higher classification accuracy. As presented in the literature current feature extraction methods can be categorized into three types: *shape, appearance* and *dynamic*:

Shape based features: Geometric features contain information about shape and locations of salient facial features such as eyes, nose and mouth. Standard approaches rely on first detecting and tracking faces over the video sequence and then localizing and tracking the key facial components using Constrained Local Models [8] or Parameterized Appearance Models (PAMs) [15, 26, 30]. The output is a set of coordinates which corresponds to the salient parts of the face. Shape based features follow the movement of key parts or points and capture movement, as a sequence of observations over time. Although geometric features perform well in capturing the temporal features, they have difficulty in detecting subtle expressions and are highly vulnerable to registration error [11].

Appearance based features: Over the past few years, appearance features have become increasingly popular in facial expression analysis. Appearance features extract the facial skin texture details and represent them in a higher dimensional feature space for better representation. One popular method for appearance features is SIFT [50]. The SIFT descriptor computes the gradient vector for each pixel in the neighborhood of the interest points and builds a normalized histogram of gradient directions. For each pixel within a subregion, SIFT adds the pixel's gradient vector to a histogram of gradient directions by quantizing each orientation to one of 8 directions and weighting the contribution of each vector by its magnitude. Similar to SIFT, DAISY [50], Gabor jets [9], LBP [49], Bag-of-Words model [34, 37], compositional [46] and others [18] are efficient feature descriptors that are used for feature extraction. The most recent approaches are [19, 43], where a CNN is used for detection and intensity estimation of multiple AUs. As presented in the literature (see De la Torre et al. [16] for a comparison), appearance features outperform shape only features for AU detection.

Dynamic features: In this strategy different sets of features from different modalities are combined in order to create the feature vector. For example Gunes et al. [20] combine body features with the facial features for expression analysis and Zhu et al. [50] uses mixture of SIFT and temporal features and presents an efficient AU detection framework.

2.2. Classification

After extracting the facial features we need a classifier that can accurately classify the expression without overfitting. The literature on facial expression classifiers can be categorized into two main groups, *static* and *temporal*.

Static classifiers: One popular method of expression detection is to learn a discriminative expression detection function which is linearly applied to the observed data. Although there are many benefits in maintaining a linear relationship between the data domain and the classifier [2, 4], there are still some drawbacks with this model: (i) the performance in this model is strongly influenced by the quality of the input features; (ii) decreasing the amount of training data reduces the classification accuracy; (iii) it fails to capture temporal information among the frames in the observed videos; (iv) since the filter has a fixed size in such a presentation, it cannot be applied on videos with different durations. Representative approaches include Neural Networks [21], Adaboost [9], SVMs [28,35,48], and Deep Networks [24].

Temporal classifiers: To address the limitations with the static classifiers, some methods consider temporal approaches. The key intuition behind temporal approaches is to present a classifier that learns the spatio-temporal dependencies among frames. For instance, Tong et al. [39] used Dynamic Bayesian Networks (DBN) with appearance features to model the dependencies among AUs and temporal properties between frames. Other variants of DBN include Hidden Markov Models [33] and Conditional Random Fields (CRF) [10]. Abbasnejad et al. [2] used Dynamic Time Warping to align all the training sequences and learn the temporal correlations.

2.3. CNN Based Facial Expression Approaches

Deep networks have dramatically improved the performance of vision systems, including object detection [23] and face verification [38]. In the field of facial expression analysis, Kim et al. [22] used a convolutional neural network based model for a hierarchical feature representation in the audiovisual domain to recognise spontaneous emotions; Liu et al. [24] used convolutional models to learn discriminative local regions for holistic expressions. They introduced an AU aware receptive field layer in a deep network, and show improvement over the traditional handcrafted image features such as LBP, SIFT and Gabor. Gudi et al. [19] utilized a CNN framework with 3 convolutional and 1 max-pooling layers that is jointly trained for detection and intensity estimation of multiple AUs. Nguyen et al. [32] used a C3D model to learn the spatio temporal features for multi-modal emotion recognition. Chu et al. [13] used CNNs to extract the spatial features and then feed the CNN features to a Long Short-Term Memory (LSTM) to model the temporal dependencies between the frames and. Walecki [43] presented a deep CNN-CRF model to capture the output structure of CNN features by means of a CRF.

One common problem with the previous CNN based methods is, the networks do not learn the spatio-temporal information (which is crucial in the task of event analysis [40]) among frames. This makes models vulnerable to facial expressions with a high temporal dependency. In addition, due to the lack of data, previous CNN based methods mostly pre-trained their models on large scale object classification datasets and fine-tune them on the expression data [43]. The main problem with these methods is that since they are pre-trained on the object based datasets, they cannot fully learn the facial expression features.

3. Setting the problem

Recently deep convolutional networks have become a popular technique in different applications of computer vision, such as object tracking [23] and event detection [40]. The current success can be traced back to the ImageNet Challenge. ImageNet contains several hundred images for any given class, such as *"dog", "cat"* or *"plane"*. During the contest in 2015 and the ImageNet Challenge neural networks were finally able to surpass by recognizing 96% of images, compared to humans recognizing of 95%.

Although deep networks perform well in different applications such as object recognition and event detection, in the task of facial expression recognition they have still not advanced sufficiently [13, 24, 43]. One problem stems from the fact that in contrast to the other applications such as object detection and event analysis, there are small labeled datasets for training a deep network. Furthermore, it is expensive and time consuming to collect facial expression of many different subjects since it is hard to verify the action unit motions. In addition, due to noise and head pose variation the data needs pre-processing and cleaning before training.

In this work we move beyond the previous methods and apply a deep network to the problem of expression analysis. Since there is not enough labeled expression data for training a deep network, we synthetically generate a new large scale expression dataset. Since the data is generated synthetically we can confidentially create faces that have different levels of saturation in expression and have accurate movement in their action units. In our synthetic data generation we are not worried about the number of participants. Unlike the previous CNN based methods that use a network with small number of layers, our framework helps us to train a deep network with 16 layers for expression analysis.

4. Synthetic Data Generation

In this section we describe our synthetic data generation method. Our method consists of two stages [5]: (i) *Face Model*, that represents the face template and the process of generating different faces with various textures; (ii) *Expression Model*, that explains the face fitting process and expression data generation.

4.1. Face Model

The 3D Face Model consists of two parametric models: the *shape* and *texture* models. By manipulating the shape and texture parameters we can create different subjects in different expressions. This section explains the theoretical details of our approach.

Shape Model: Let us denote the $3D$ mesh (shape) of an object with $N = 53490$ vertices as a $3N \times 1$ vector,

$$\mathbf{s} = [s_1^T, s_2^T, \dots, s_N^T]^T, \tag{1}$$

where the vertices $s_i = (x_i, y_i, z_i)^T \in \mathbb{R}^3$ are the object-centered Cartesian coordinates of the i-th vertex. A $3D$ shape model can be built by first transferring a set of $3D$ training meshes into dense correspondence such that for any given i, the i-th vertex corresponds to the same location on all face scans. Once the correspondence between the vertices of all scans and the corresponding meshes is established, $\{\mathbf{s}_i\}$ are then brought into a shape space by applying Generalized Procrustes Analysis and then Principal Component Analysis (PCA) is performed. The shape is modeled by the mean shape vector $\bar{\mathbf{s}}$ and the first n_s orthonormal basis of the principal components, $\mathbf{U}_s \in \mathbf{R}^{3N \times n_s}$. Then the new shape can be created using the functions $\mathcal{S} : \mathbf{R}^{n_s} \rightarrow \mathbf{R}^{3N}$,

$$\mathcal{S}(\mathbf{p}_i) = \bar{\mathbf{s}} + \mathbf{U}_s \mathbf{p}_i, \tag{2}$$

where $\mathbf{p}_i = [p_1, ..., p_{n_s}]^T$ are the first n_s shape parameters.
Texture Model: Texture-vector which represents the texture of a face is defined as,

$$\mathbf{t} = [R_1, G_1, B_1, ..., R_N, G_N, B_N]^T \in \mathbf{R}^{3N}, \tag{3}$$

where the texture vector contains the R, G, B color values of N corresponding vertices. The $3D$ texture model is then constructed using the set of training examples. Texture is extracted by applying PCA to the registered faces which results in $\{\bar{\mathbf{t}}, \mathbf{V}\}$, where $\bar{\mathbf{t}} \in \mathbf{R}^{3N}$ is the mean texture vector and $\mathbf{V} \in \mathbf{R}^{3N \times n_t}$ is the first n_t principal components. Then the new texture example will be established using the functions $\mathcal{T} : \mathbf{R}^{n_t} \rightarrow \mathbf{R}^{3N}$ as,

$$\mathcal{T}(\mathbf{b}_i) = \bar{\mathbf{t}} + \mathbf{V}_t \mathbf{b}_i, \tag{4}$$

where $\mathbf{b} = [b_1, ..., b_{n_t}]^T$ are the first n_t texture parameters.

4.2. Expression Model

The assumption we consider in this paper is that the facial expression space can not be independent from the face space. Each expression can be modeled by manipulating the shape parameters in the face space. Therefore the facial expression can be generated by changing the weights of the n_s PCA components of $\mathbf{U}_s \in \mathbb{R}^{3N \times n_s}$. To define the facial expression sequence we need a $3D$ template mesh of a face, the shape parameters, and the animation sequence. To define the mesh topology, we use a $3D$ mesh of a face explained in Section 4.1, and to estimate the shape parameters we fit a face template to six scanned facial expressions to create the synthetic facial expression models.

4.2.1 Face Registration

In order to accurately create facial expression sequences we need to estimate the shape parameters in Eq. 2 that are accurately defined for each expressions. To do so, we fit the face shape template to the scan face models of six different expressions, e.g. *Anger, Disgust, Fear, Happy, Sad* and *Surprise* to establish shape parameters of each expression. The face models that are used in this paper are from BU-4DFE dataset [47] and are accurately labeled by experts.

The fitting algorithm used in this paper is the robust nonrigid ICP as presented in Amberg et al. [6]. This model is a variant of nonrigid ICP [7]. However, the main difference is that they use a statistical deformation model to capture the details of the scan faces. Also during the optimization, they use an iterative method to solve the cost function. The cost function for our optimization problem can be defined as follows,

$$E(\mathbf{R}, \mathbf{t}, \mathbf{p}) = \sum_{i=1}^{N} \|\bar{\mathbf{s}}_i + \mathbf{U}_s \mathbf{p}_i + \mathbf{t}' - \mathbf{R}' \mathbf{m}_i\|_2^2 + \lambda \|\mathbf{p}\|_2^2, \tag{5}$$

$$\mathbf{t}' = \mathbf{R}^{-1}\mathbf{t}, \qquad \mathbf{R}' = \mathbf{R}^{-1},$$

where $\mathbf{R}$ is the rotation matrix, $\mathbf{t}$ is the transition vector and $\mathbf{m}$, is the scan face surface model,

$$\mathbf{M} = [\mathbf{m}_1^T, \dots, \mathbf{m}_N^T]^T,$$

This function can be solved by a Gauss-Newton least square optimization, using an analytic Jacobian and Gauss-Newton Hessian approximation. The gradient and Jacobian matrices are defined as,

$$E_i = \bar{\mathbf{s}}_i + \mathbf{U}_{si} + \mathbf{t}' - \mathbf{R}'_{r_{x,y,z}} \mathbf{m}_i, \tag{6}$$

$$\frac{\partial E_i}{\partial \mathbf{s}_i} = \mathbf{U}, \quad \frac{\partial E_i}{\partial \mathbf{t}'} = \mathbf{I}_3, \quad \frac{\partial E_i}{\partial r_i} = \frac{\partial \mathbf{R}'_{x,y,z}}{\partial r_i} \mathbf{m}_i, \tag{7}$$

$$\mathbf{J} = [\mathbf{J_c} \;\mid\; \mathbf{J_d}], \tag{8}$$

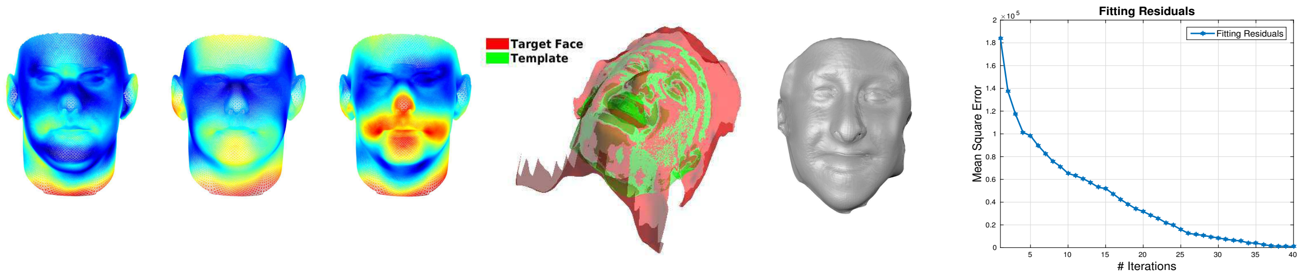
$$\mathbf{J}_c = \begin{bmatrix} \mathbf{U} & \mathbf{1} \otimes \mathbf{I}_3 \\ \mathbf{I} & 0 \end{bmatrix}, \tag{9}$$

$$\mathbf{J}_d = \begin{bmatrix} (\mathbf{I} \otimes \frac{\partial \mathbf{R}'}{\partial r_x})\mathbf{m}^T & (\mathbf{I} \otimes \frac{\partial \mathbf{R}'}{\partial r_y})\mathbf{m}^T & (\mathbf{I} \otimes \frac{\partial \mathbf{R}'}{\partial r_z})\mathbf{m}^T \\ 0 & 0 & 0 \end{bmatrix}, \tag{10}$$

where $\otimes$ refers to the tensor product. The Hessian matrix is estimated as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{J}_c^T \mathbf{J}_c & (\mathbf{J}_c^T \mathbf{J}_d)^T \\ \mathbf{J}_c^T \mathbf{J}_d & \mathbf{J}_d^T \mathbf{J}_d \end{bmatrix}. \tag{11}$$

By pre-calculating the constant parts of the matrices we can reduce the computational time and make the convergence faster. Figure 2 shows an example of fitting the template to a scan face model from the BU-4DFE dataset.



(a) (b)

Figure 2: (**a**) The first three faces show the distance between the template and the scanned face over three face registration steps for the smile expression. The fourth and fifth faces show the registered face results (for smile expression). (**b**) The fitting residuals.

After fitting the template to the scan model we need to calculate the corresponding shape parameters, $\mathbf{p}_i$, of each expression. The parameters can be computed as:

$$E_p(\mathbf{m}, \mathbf{U}_s, \mathbf{p}_i) = \min_{\mathbf{p}_i} \|\mathbf{M} - \mathbf{U}_s \mathbf{p}_i\|^2, \tag{12}$$

the optimum value of that minimizes Eq. 12, $\mathbf{p}_i^*$ gives us the shape parameters of different expressions,

$$\mathbf{p}_i^* = (\mathbf{U}_s^T \mathbf{U}_s)^{-1} (\mathbf{U}_s \mathbf{M}). \tag{13}$$

4.2.2 Expression Generation

As explained earlier by changing the weights of $\mathbf{U}_s$ we can generate different expressions. In this section we explain the details of generating the expression sequences. We represent each facial expression sequence as $\mathcal{G}(f, \mathcal{T}, \mathcal{S}, \omega, \mathbf{p}_i)$, where f is the length of the sequence and ω is a weight that controls the facial expression level in each frame,

$$\mathbf{p}_i(w, f) = \mathbf{p}_0 + (\frac{\mathbf{p}_i^* - \mathbf{p}_0}{f}) * w \tag{14}$$
$$w \in [0, \dots, f].$$

From Eq. 14 we can see at the first frame we start from a neutral face $\mathbf{p}_0$. Over time we increase the shape weights until the last frame which has the peak, $\mathbf{p}_i^*$.

In order to create different facial expression subjects, we need to create different faces with different facial texture. Therefore, we randomly generate the texture parameters from the following distribution,

$$prob(\mathbf{b}) \sim exp[-\frac{1}{2}\sum_{i=1}^{n_t}(b_i/\sigma_i^2)].$$

Light. The faces are illuminated using the Phong lighting model.

Camera. The projective camera has a resolution of 648×490, focal length of 60mm and sensor size of 32mm. The camera is located exactly in front of the faces and during recording it is not moved or rotated.

Background. Since most of the available expression datasets are from laboratories with a simple background, in this work we render the faces in front of a white background.

5. Proposed Architectures and Training Method

Most recent video representations for temporal analysis are based on two different CNN architectures: (i) 3D spatio-temporal convolutions [40,42] that learn complicated spatio-temporal dependencies and (ii) Two-stream architectures [36] that decompose the video into motion and appearance streams, train separate CNNs for each stream and at the end fuse the outputs. In this work, we establish our model based on the 3D *ConvNet* architecture which was introduced for action recognition [40]. We believe this model is a better representation for action unit detection since 3D *ConvNet* consists of 3D convolution and 3D pooling, which are used to observe the appearance of the faces and learns the temporal dependency among frames.

C3D has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$, with stride 1 in both spatial and temporal dimensions. The convolution layers consist of, $64, 128, 256, 256, 512, 512, 512$ and 512 filters respectively and the last two fully connected layers have 4096 outputs. All pooling kernels are $2 \times 2 \times 2$ except the first pooling which is $1 \times 2 \times 2$. The network input is a 3-channel RGB video of size $16 \times 128 \times 128$ croped and scaled to fit a face bounding box. We use binary-cross entropy loss defined on all frames for classification. We evaluate our network on two tasks: expression recognition and facial action unit detection. For the task of expression classification we have six classes and for the task of action unit detection we have 11 outputs.

In the training phase we first pre-train the network on the synthetic expressions we generated in Section 4. We trained for $50K$ iterations and we use the RMSprop algorithm with mini-batches of size 16 and a learning rate of 10^{-3}. After the pre-training step, we fine-tune our network on the real datasets.

6. Dataset

In this section we explain details of the datasets we use in this paper. We evaluate our method on the CK+ [27] and BU-4DFE [47] datasets.

Synthetic facial expression: As we mentioned earlier one of the aims of this paper is to generate a synthetic expression dataset for efficient expression analysis. We generate in total $12,000$ facial expression sequences where each expression sequence contains 16 frames ($f = 16$). We generate six different expressions *Anger, Disgust, Fear, Happiness, Sadness, Surprise*. For each expression we generate $2,000$ subjects.[1]

Cohn-Kanade: The CK+ Database is a facial expression database. It contains 593 facial expression sequences from 123 participants. Each sequence starts from a neutral face and ends at the peak frame. Sequences vary in duration between 4 and 71 frames and the location of 68 facial landmarks are provided along with database. Facial poses are frontal with slight head motions.

BU-4DFE: This dataset consists of both 3D and 2D facial expression videos that are captured at a video rate of 25 frames per second. For each subject, there are six model sequences showing six prototypic facial expressions *Anger, Disgust, Fear, Happiness, Sadness, Surprise*, respectively. Each expression sequence contains about 100 frames. The database contains 606 three dimensional facial expression sequences captured from 101 subjects, with a total of approximately $60,600$ frames.

7. Evaluation

In this section we provide details about the evaluation settings and our results on the presented datasets. We evaluate our method with generic and alternative approaches using two scenarios for facial expression recognition: *within-dataset, cross-dataset*, and *synthetic-model*. We report results separately for each scenario [14]. We also evaluate our method on the task of facial action unit detection.

7.1. Evaluation Setting

Preprocessing. Since each sequence in the proposed datasets varies between 4 to approximately 100 frames, we need to limit the sequence length to 16 frames. For those sequences which are less than 16 frames we simply repeat the last frame until we reach 16 frames and for those which are more than 16 frames, we eliminate the frames with respect to the following frame ratio,

$$r = round(\frac{f}{f-16}),$$

where f is the length of sequence.

Train/Test split. We first pre-train the network on the generated synthetic faces. Overall, 2000 subjects with $192,000$ frames are used to pre-train the model. After the pre-training stage, we fine-tune our network on the real datasets. We use 10-fold cross validation to evaluate the results.

Face tracking registration. To make sure all the faces contain the same bounding box, we pre-processed all videos by extracting facial landmarks. For CK+ we use the landmark set which is provided by the dataset and for the synthetic and BU-4DFE datasets we use the CML method presented in [8] to extract the facial landmarks. The tracked faces are then cropped into 128×128 using the coordinates of eyebrows, jaws and chin.

Evaluation metrics. To evaluate the performance, we report the area under ROC curve, and the maximum F_1-score. The F_1-score is defined as,

$$F_1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}},$$

and conveys the balance between the precision and recall.

7.2. Examples of synthetic faces

Figure 3 shows examples of the registered template to the scan faces using Eq. 5. For fitting the template to the scan faces we use six subjects in their peak facial expression frame from BU-4DFE dataset [47].

In order to create different subjects we change the texture parameters in Eq. 4. Figure 3 shows five examples of the generated data. In this figure five different subjects (with different facial textures) are shown for the *Anger* expression.

7.3. Within-dataset Evaluation

In this section we evaluate our method when the network is trained and tested on the same dataset. Here we first pre-train the C3D network on the synthetic dataset and then we fine-tune and test the network on the same real-world dataset. Table 1 and Table 2 show the Action Unit classification results on the CK+ and BU-4DFE datasets. The table also compares our method against, “SIFT + SVM” and “C3D + SVM” and “ITraj + SVM” methods. “C3D + SVM” refers to the case when the features are extracted from the last fully connected layer of the fine-tuned C3D model and

[1] This dataset will be released publicly and interested parties are requested to contact the authors directly

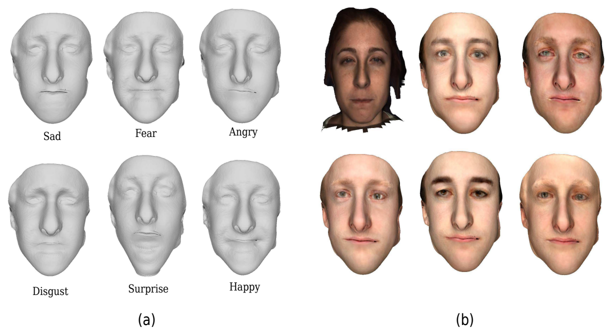


Figure 3: (a), Examples of registered template to the scan faces from BU-4DFE dataset. (b), Comparison between different synthetic subjects that are performing, *anger* expression in the peak frame and an example from the scanned face.

AU	Area Under ROC Curve				F_1-score			
	C3D	C3D + SVM	SIFT + SVM	ITraj + SVM	C3D	C3D + SVM	SIFT + SVM	ITraj + SVM
2	**98.37**	96.21	92.60	79.22	**83.71**	82.87	76.03	50.22
4	**97.82**	94.46	91.25	72.21	**75.52**	70.42	64.26	62.02
5	**97.19**	95.53	90.91	80.29	**81.15**	71.11	60.19	56.32
6	**98.79**	95.42	91.01	82.63	**84.59**	79.24	70.13	51.01
9	**96.41**	88.65	86.17	79.09	**85.38**	77.91	69.12	53.22
12	**98.89**	97.42	90.31	87.76	**93.33**	91.21	80.44	71.81
14	**97.62**	95.25	89.77	89.22	**88.28**	85.17	79.29	74.12
15	**98.13**	97.17	92.42	83.52	**92.02**	88.11	72.59	67.73
17	**98.31**	95.92	91.49	79.01	**93.51**	88.18	73.66	68.92
18	**98.83**	91.82	89.81	72.91	**93.42**	89.33	65.21	54.28
20	**96.29**	90.62	89.37	77.37	**81.59**	70.53	69.71	62.82
Mean	**97.87**	94.41	90.46	80.29	**86.59**	81.28	70.97	61.13

Table 1: Results on the CK+ dataset.

are fed to a linear SVM [2] for classification. "SIFT + SVM" refers to the case when the SIFT features are representing video frames and SVM is the classifier. "ITraj + SVM" refers to the experiment when "Improved Dense Trajectory" features [44] are fed to a linear SVM classifier for action unit classification. We also use our method for expression classification. Figure 4 demonstrates the results for expression classification on the proposed dataset. In this part we also follow the same procedure as is mentioned for training and testing the classifier.

7.4. Cross-dataset Evaluation

In this experiment we use our network to extract the facial features. At the first stage of this experiment we first pre-train the network on the generated synthetic data explained in Section 4. After the pre-training step, in order to extract features from CK+ dataset we fine-tune the network on the BU-4DFE and in order to extract features from BU-4DFE dataset we fine tune the network on CK+ datasets. Then the features are extracted from the last fully connected layer of the C3D network, and are used for action unit classification. In this experiment we use the linear SVM classifier for classification. The intuition behind this experiment is whether we can use the trained model of our network (which is pre-trained on the synthetic dataset and fine-tuned on the other dataset) as a blackbox. Table 3 demonstrates the detection results on CK+ and BU-4DFE datasets. From the table we can see that overall we obtain 86.43% accuracy, however with the *"Within-dataset"* approach we obtain 93.13%.

AU	Area Under ROC Curve				F_1-score			
	C3D	C3D + SVM	SIFT + SVM	ITraj + SVM	C3D	C3D + SVM	SIFT + SVM	ITraj + SVM
2	**88.72**	86.79	69.91	62.34	**73.02**	68.81	61.25	52.85
4	**96.31**	94.17	81.15	70.29	**89.75**	88.01	60.54	55.10
5	**91.54**	92.10	81.45	72.91	**79.82**	78.02	60.62	60.13
6	**97.61**	95.48	79.31	75.19	**96.03**	94.12	71.74	66.18
9	91.39	**91.68**	82.29	81.24	**87.13**	86.28	60.45	58.72
12	**98.59**	97.28	90.89	86.79	**95.01**	92.79	72.51	70.39
14	**91.14**	90.83	81.42	77.28	**89.27**	88.82	65.19	62.41
15	87.19	**89.38**	81.72	76.82	70.68	**71.93**	62.42	61.52
17	**90.29**	89.31	82.58	82.65	**84.20**	83.17	68.51	62.79
18	**88.14**	87.52	79.54	71.44	**78.62**	74.39	69.33	67.55
20	**82.49**	81.92	69.18	61.72	**76.27**	75.07	60.29	59.16
Mean	**91.22**	90.59	79.95	74.43	**83.62**	81.95	64.81	61.53

Table 2: Results on the BU-4DFE dataset.

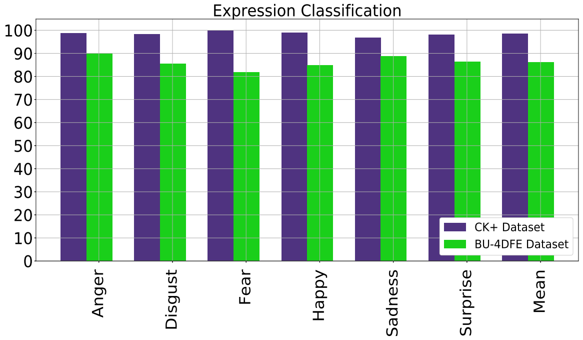


Figure 4: Expression classification accuracy on the proposed datasets.

7.5. Synthetic Model

In this experiment we investigate the efficiency of the pre-trained model for feature representation. Here, we first train the network on the generated synthetic dataset. Then the last fully connected layer of the C3D network is utilized for feature extraction and a linear SVM is used for classi-

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

	Area Under ROC Curve		F_1-score	
AU	CK+	BU-4DFE	CK+	BU-4DFE
2	92.34	85.42	81.12	65.29
4	90.62	88.16	68.27	84.53
5	91.25	84.59	71.43	75.72
6	94.94	87.34	75.62	90.45
9	82.59	82.93	71.78	82.26
12	95.61	86.31	82.21	87.61
14	92.43	85.77	74.85	59.24
15	94.75	88.23	84.65	81.58
17	94.42	83.27	86.94	73.83
18	93.76	85.03	84.51	71.43
20	88.39	81.51	71.28	69.39
Mean	91.92	85.32	77.52	76.49

Table 3: Cross-dataset experiment on the CK+ and BU-4DFE datasets.

	Area Under ROC Curve		F_1-score	
AU	CK+	BU-4DFE	CK+	BU-4DFE
2	87.31	60.08	55.44	49.16
4	84.73	58.27	56.95	46.66
5	84.29	56.48	52.39	46.67
6	88.43	69.03	58.51	51.24
9	80.49	71.53	53.30	52.19
12	84.76	75.69	66.27	63.93
14	81.78	60.37	62.31	50.11
15	88.03	56.71	70.19	51.38
17	89.28	58.01	70.54	50.83
18	83.72	70.23	66.67	59.26
20	75.24	55.27	49.22	52.13
Mean	84.37	62.88	60.16	52.14

Table 4: Synthetic Model experiment on the CK+ and BU-4DFE datasets.

fication. Table 4 shows the results of this experiment on the CK+ and BU-4DFE datasets. This experiment shows how much the synthetic traineing process could be efficient for the whole pre-training and fine-tuning. From Table 4 we can see the performance drops by approximately $\approx 8\%$ with respect to the *"Within-class"* experiment.

7.6. Comparison with the State-of-the-art

We also compare our method with the other models. The results for this comparison are given in Table 5. In Chu et al. [14] they used a Selective Transfer Machine (STM), to personalize a generic classifier, Liu et al. [25] used Boosted Deep Belief Network (BDBN), Chu et al. [12] utilied CNN to extract spatial features and the LSTM for temporal modeling, and Mollahosseini et al. [31] use a deep network for expression analysis. All the methods in the table are using the same training and testing protocol for evaluation (10-fold cross validation). As can be seen our method outperforms other methods on the both datasets.

	Area Under ROC Curve		F_1-score	
Method	CK+	BU-4DFE	CK+	BU-4DFE
Chu et al. [14]	91.30	-	-	-
Liu et al. [25]	96.70	-	-	-
Chu et al. [12]	-	-	-	82.50
Mollahosseini et al. [31]	93.20	-	-	-
Ours	**97.87**	91.22	86.59	**83.62**

Table 5: Comparing with the state-of-the-art.

	Area Under ROC Curve		F_1-score	
Method	CK+	BU-4DFE	CK+	BU-4DFE
C3D + Synthetic Data	97.87	91.22	86.59	83.62
C3D + CK+	79.43	-	60.52	-
C3D + BU-4DFE	-	72.27	-	62.86

Table 6: Comparison between using synthetic and without using synthetic data in training C3D.

7.7. Discussion

In this paper we show how using synthetic facial expression data can help us to train a C3D model for expression analysis and improve the detection performance. However, one might ask how will training C3D without synthetic data perform. Table 6 compares the classification performance in two scenarios: using the synthetic data and without using the synthetic data. In this evaluation we followed the exact network and evaluation protocol we explained in Section 5 and Section 7.1 respectively. As can be seen, the yntheticdataset improves the detection performance dramatically.

8. Conclusion

Since the introduction of the convolutional neural network, there have been great advances in the classification of objects and events, however CNN-based methods have not made an enormous impact on expression analysis. One reason stems from the lack of large-scale facial expression datasets. In this work in order to tackle this limitation we synthetically generate a large scale expression dataset. The generated dataset enables us to efficiently train a 3-dimensional convolutional network for expression analysis. We evaluate our model on two real-world expression datasets and we obtain state-of-the-art performance.

Acknowledgment

This research was funded and supported by Australian Research Council (ARC) Discovery Grant number DP140100793.

References

[1] E. Abbasnejad, A. Dick, and A. v. d. Hengel. Infinite variational autoencoder for semi-supervised learning. *arXiv preprint arXiv:1611.07800*, 2016. 2

[2] I. Abbasnejad, S. Sridharan, S. Denman, C. Fookes, and S. Lucey. Learning temporal alignment uncertainty for efficient event detection. In *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, pages 1–8. IEEE, 2015. 1, 2, 3

[3] I. Abbasnejad, S. Sridharan, S. Denman, C. Fookes, and S. Lucey. Complex event detection using joint max margin and semantic features. In *Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on*, pages 1–8. IEEE, 2016. 2

[4] I. Abbasnejad, S. Sridharan, S. Denman, C. Fookes, and S. Lucey. Joint max margin and semantic features for continuous event detection in complex scenes. *arXiv preprint arXiv:1706.04122*, 2017. 2

[5] I. Abbasnejad and D. Teney. A hierarchical bayesian network for face recognition using 2d and 3d facial data. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pages 1–6. IEEE, 2015. 3

[6] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3d face recognition with a morphable model. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2008. 4

[7] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 4

[8] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451, 2013. 2, 6

[9] M. S. Bartlett, G. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan. Automatic recognition of facial actions in spontaneous expressions. *Journal of multimedia*, 1(6):22–35, 2006. 2, 3

[10] K.-Y. Chang, T.-L. Liu, and S.-H. Lai. Learning partially-observed hidden conditional random fields for facial expression recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 533–540. IEEE, 2009. 3

[11] S. W. Chew, P. Lucey, S. Lucey, J. Saragih, J. F. Cohn, I. Matthews, and S. Sridharan. In the pursuit of effective affective computing: The relationship between features and registration. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1006–1016, 2012. 2

[12] W.-S. Chu, F. De la Torre, and J. F. Cohn. Modeling spatial and temporal cues for multi-label facial action unit detection. *arXiv preprint arXiv:1608.00911*, 2016. 8

[13] W.-S. Chu, F. De la Torre, and J. F. Cohn. Learning spatial and temporal cues for multi-label facial action unit detection. In *Automatic Face and Gesture Conference*, volume 4, 2017. 3

[14] W.-S. Chu, F. De la Torre, and J. F. Cohn. Selective transfer machine for personalized facial expression analysis. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):529–545, 2017. 1, 6, 8

[15] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001. 2

[16] F. De la Torre and J. F. Cohn. Facial expression analysis. In *Visual analysis of humans*, pages 377–409. Springer, 2011. 2

[17] P. Ekman and E. L. Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997. 1

[18] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern recognition*, 36(1):259–275, 2003. 2

[19] A. Gudi, H. E. Tasli, T. M. den Uyl, and A. Maroulis. Deep learning based facs action unit occurrence and intensity estimation. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 6, pages 1–5. IEEE, 2015. 2, 3

[20] H. Gunes and M. Piccardi. Affect recognition from face and body: early fusion vs. late fusion. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 4, pages 3437–3443. IEEE, 2005. 2

[21] A. Kapoor and R. W. Picard. Multimodal affect recognition in learning environments. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 677–682. ACM, 2005. 3

[22] Y. Kim, H. Lee, and E. M. Provost. Deep learning for robust feature generation in audiovisual emotion recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3687–3691. IEEE, 2013. 3

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[24] M. Liu, S. Li, S. Shan, and X. Chen. Au-aware deep networks for facial expression recognition. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013. 3

[25] P. Liu, S. Han, Z. Meng, and Y. Tong. Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812, 2014. 8

[26] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981. 2

[27] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010. 6

[28] S. Lucey, A. B. Ashraf, and J. F. Cohn. *Investigating spontaneous facial action recognition through aam representations of the face*. INTECH Open Access Publisher, 2007. 3

[29] A. Martinez and S. Du. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *Journal of Machine Learning Research*, 13(May):1589–1608, 2012. 1

[30] I. Matthews and S. Baker. Active appearance models revisited. *International journal of computer vision*, 60(2):135–164, 2004. 2

[31] A. Mollahosseini, D. Chan, and M. H. Mahoor. Going deeper in facial expression recognition using deep neural networks. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–10. IEEE, 2016. 8

[32] D. Nguyen, K. Nguyen, S. Sridharan, A. Ghasemi, D. Dean, and C. Fookes. Deep spatio-temporal features for multimodal emotion recognition. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 1215–1223. IEEE, 2017. 2, 3

[33] L. Shang and K.-P. Chan. Nonparametric discriminant hmm and application to facial expression recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2090–2096. IEEE, 2009. 3

[34] K. Sikka, T. Wu, J. Susskind, and M. Bartlett. Exploring bag of words architectures in the facial expression domain. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 250–259. Springer, 2012. 2

[35] T. Simon, M. H. Nguyen, F. De La Torre, and J. F. Cohn. Action unit detection with segment-based svms. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2737–2744. IEEE, 2010. 3

[36] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 5

[37] J. Sivic, A. Zisserman, et al. Video google: A text retrieval approach to object matching in videos. In *iccv*, volume 2, pages 1470–1477, 2003. 2

[38] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2754, 2015. 3

[39] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE transactions on pattern analysis and machine intelligence*, 29(10), 2007. 3

[40] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2, 3, 5

[41] M. F. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer. Meta-analysis of the first facial expression recognition challenge. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):966–979, 2012. 1

[42] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5

[43] R. Walecki, O. Rudovic, V. Pavlovic, B. Schuller, and M. Pantic. Deep structured learning for facial expression intensity estimation. *IMAVIS (article in press)*, 259:143–154, 2017. 2, 3

[44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013. 7

[45] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. *arXiv preprint arXiv:1703.07814*, 2017. 2

[46] P. Yang, Q. Liu, and D. N. Metaxas. Exploring facial expressions with compositional features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2638–2644. IEEE, 2010. 2

[47] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu. A high-resolution spontaneous 3d dynamic facial expression database. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013. 4, 6

[48] K. Zhao, W.-S. Chu, F. De la Torre, J. F. Cohn, and H. Zhang. Joint patch and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2207–2216, 2015. 3

[49] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas. Learning active facial patches for expression analysis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2562–2569. IEEE, 2012. 2

[50] Y. Zhu, F. De la Torre, J. F. Cohn, and Y.-J. Zhang. Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior. *IEEE transactions on affective computing*, 2(2):79–91, 2011. 2