FACSCaps: Pose-Independent Facial Action Coding with Capsules

Itir Onal Ertugrul[1], László A. Jeni[1], Jeffrey F. Cohn[1,2]
[1]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[2]Department of Psychology, University of Pittsburgh, Pittsburgh, PA, USA
iertugru@andrew.cmu.edu, laszlojeni@cmu.edu, jeffcohn@pitt.edu

Abstract

Most automated facial expression analysis methods treat the face as a 2D object, flat like a sheet of paper. That works well provided images are frontal or nearly so. In real-world conditions, moderate to large head rotation is common and system performance to recognize expression degrades. Multi-view Convolutional Neural Networks (CNNs) have been proposed to increase robustness to pose, but they require greater model sizes and may generalize poorly across views that are not included in the training set. We propose FACSCaps architecture to handle multi-view and multi-label facial action unit (AU) detection within a single model that can generalize to novel views. Additionally, FACSCaps's ability to synthesize faces enables insights into what is leaned by the model. FACSCaps models video frames using matrix capsules, where hierarchical pose relationships between face parts are built into internal representations. The model is trained by jointly optimizing a multi-label loss and the reconstruction accuracy. FACSCaps was evaluated using the FERA 2017 facial expression dataset that includes spontaneous facial expressions in a wide range of head orientations. FACSCaps outperformed both state-of-the-art CNNs and their temporal extensions.

1. Introduction

Facial expression communicates emotion, intentions, and physical states [27]. Automatic detection of facial expressions is crucial to multiple domains that include mental and physical health, education, and human-computer and robot-human interaction. The most comprehensive method to annotate facial expression is the anatomically-based Facial Action Coding System (FACS). FACS "action units" alone or in combinations can describe nearly all possible facial expressions. Automatic detection of FACS action units (AU) has been an active area of research.

Approaches have included both shallow- and deep learning. For the former, hand-crafted features have included SIFT, HOG, LBP, LGBP and geometric features. More recently, deep-learning approaches have been proposed [11, 10]. Convolutional Neural Networks (CNNs) learn representations and estimate AU occurrences. While CNNs have often outperformed shallow-learning approaches in AU detection [5, 38], except for the recent FERA Challange, pose variation has been limited to frontal or nearly frontal views. In natural environments in which moderate to large head rotation is common, generalizability to non-frontal views is critical.

Recently, some studies have performed multi-view facial AU detection. The approaches have included hand-crafted features [19], CNN [26, 4], and LSTM [12]. A limitation to all these studies is that they fail to illuminate the underlying representations. They are unable to reveal or interpret what is learned by their architectures visually. With respect to accuracy, CNNs performed well but have two main drawbacks [23, 32]. First, they fail to represent spatial hierarchy between object parts. If the existence of parts is satisfied, a CNN model outputs the existence of whole object by ignoring the spatial orientation among parts. It would yield false positives. Second, they lack rotational invariance. Since pose is not independent from the internal representation of an object, the same object observed from a different orientation would be recognized as a different object, leading to false negatives.

To address these problems, Sabour et al. [23] proposed Capsule Networks (CapsNets), where capsule is a group of neurons that encapsulates all significant information about the state of the features in a form of vector, rather than the scalar neuron outputs, which are common to nearly all other neural network approaches. The most crucial property of CapsNets is *routing by agreement*, which means capsules at lower levels predict the outcome of capsules at higher levels, and higher level capsules become activated only if these predictions agree. Dynamic routing and reconstruction regularization enable CapsNets to model spatial hierarchy and invariance to rotation [23]. They can learn viewpoint invariant relationship between the parts of an object and whole object. They have been shown to be more successful than

Figure 1: Different views included in the FERA 2017 dataset

CNNs for digit [23] and object [32, 14] classification.

Most of the studies aiming to recognize facial expressions or AUs focus only on the classification without considering the variations in the data. On the contrary, variants of variational autoencoders [13, 18] aim to learn the hierarchical representations and variations in the dataset in an unsupervised manner or they apply classifiers learned separately from the encoded representations. Linh Tran et al. [21] proposed to jointly learn the hierarchical representations and perform AU intensity estimation only on near-frontal images.

In this study, we propose an architecture called FACSCaps to both detect AUs from facial images having multiple views and model the variation in the data simultaneously. We perform multi-label AU detection for the classification part and we model the variation in the data by reconstructing the images from unmasked capsules. We test FACSCaps on FERA 2017 challenge dataset which has frames with 9 different views. Our results outperform previous approaches which rely on hand-crafted features or CNN and their temporal extensions. In addition to performance improvement, we can visualize what is learned by capsules and manipulate the learned representations. Finally, we perform cross-pose experiments to explore how FACSCaps will perform on novel viewpoints for the first time. We test our architecture with frames having faces with unseen views in the dataset. Cross-pose experiment results validate that FACSCaps is good at learning internal representations invariant to pose.

2. Related Work

AU detection has been a popular field in the past years. Several studies have focused on designing and extracting features [9, 3, 25, 22] and generating novel classifiers [34, 8, 6] for AU detection. Readers are referred to the surveys [7, 24] for further information. In the rest of this section, we will review the literature most relevant to our work.

Convolutional Neural Networks (CNNs) have been used to learn representations from facial images and detect AUs. A number of studies [16] trained separate CNN architectures to detect individual AUs. However, contrary to the existing research suggesting strong AU correlations [31, 20], these studies do not take AU correlations into account. In order to model the correlations among AUs, several studies [5, 37, 10, 11] perform multi-label AU detection, in which AUs are detected concurrently. Yet, most of the methods have been developed for multi-label AU detection from frontal or close to frontal face images.

Several studies focused on detection of AUs using multi-view face images. Among them, Toser et al. [28] used 3D information to augment BP4D-spontaneous dataset [35] and trained CNNs with the resulting face images having large head poses. Li et al. [19] used combination of hand-crafted (LBP-TOP) and CNN features with a late fusion mechanism applied over multiple AUs. Batista et al. [4] estimated pose, AU occurrence and AU intensity in a single CNN architecture. Tang et al. [26] fine-tuned VGG-Faces network using all views for each AU separately.

Following the recent success of CapsNets compared to CNNs on classification of digits in MNIST dataset [23] and object recognition [23, 14], a number of studies employed capsules in various fields. Afshar et al. [1] reported that CapsNets outperform CNNs for brain tumor type classification. Moreover, Jaiswal et al. [15] proposed Generative Adversarial Capsule Networks (CapsuleGAN) for modeling image data and showed that CapsuleGAN outperforms convolutional-GAN at modeling image data distribution on the MNIST dataset. Wang et al. [30] proposed RNN-Capsule for sentiment analysis and obtained state-of-the-art performance on sentiment classification. Furthermore, Andersen [2] showed that CapsNet is a reliable architecture for Deep Q-Learning based algorithms for game AI.

3. Multi-view Multi-label AU Detection using FACSCaps

3.1. Dataset

In this study we use FERA 2017 challenge dataset [29]. It contains sequences of BP4D-spontaneous [35] and BP4D+ [36] datasets which are 3D rotated by -40, -20 and 0 degrees yaw and -40, 0 and 40 degrees pitch. Therefore, original videos are synthesized into 9 different head poses as shown in Figure 1. Dataset is divided into training, development and test partitions containing videos from 41, 20 and 30 participants, respectively.

In this dataset frame-level AU occurrence labels are provided for 10 AUs namely AU1 (inner brow raiser), AU4

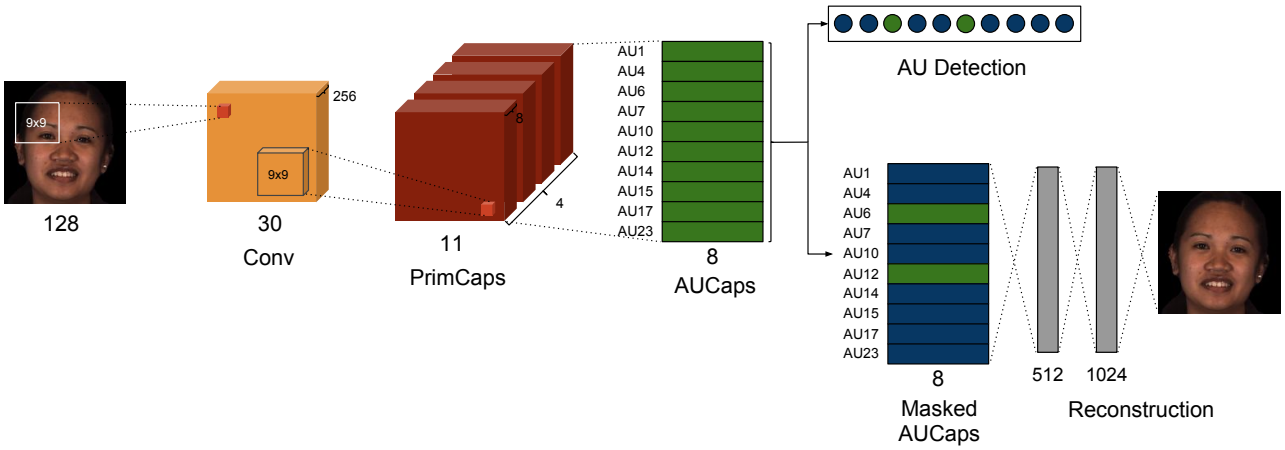

Figure 2: Overview of the proposed FACSCaps architecture for AU detection.

(brow lowerer), AU6 (cheek raiser), AU7 (lid tightener), AU10 (upper lip raiser), AU12 (lip corner puller), AU14 (dimpler), AU15 (lip corner depressor), AU17 (chin raiser) and AU23 (lip tightener).

3.2. Proposed FACSCaps Architecture

In this study we introduce FACSCaps architecture, which aims to detect occurrence of multiple AUs concurrently from multi-view facial images. As illustrated in Figure 2, the architecture takes input images having multiple views during training. Input layer is followed by a convolutional layer, which detects the basic features in the image and converts pixel intensities to the activations of local feature detectors. Then, the output of convolutional layer is fed to primary capsules.

Primary capsule is a convolutional capsule, in which information about the state of the features are encapsulated in a form of vector as opposed to the scalar outputs of basic neurons in artificial neural networks. In other words, the activity of neurons in a capsule represents the instantiation parameters (size, orientation, etc.) of a given entity.

Let j denote a capsule at a higher layer and i denote a capsule at a layer below. The activation of capsule j depends on the activations from the layer below. Based on the degree of agreement between the capsules at the higher layer and layer below, coupling coefficients c_{ij} between capsule i and capsule j are computed using the following routing softmax function:

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})} \tag{1}$$

where b_{ij} is the log probability representing whether capsule i at lower layer should be coupled with capsule j. b_{ij} is initially set to 0 and with the iterative dynamic routing process c_{ij} is estimated. Input to capsule j (denoted as $\mathbf{x}_j$) is computed as follows:

$$\mathbf{x}_j = \sum_i c_{ij} \mathbf{W}_{ij} \mathbf{u}_i \tag{2}$$

where $\mathbf{u}_i$ represents the output vector of capsule i, $\mathbf{W}_{ij}$ represents the weight matrix between capsule i and capsule j.

Length of the output vector of a capsule represents the probability that the entity is present in the input, while the orientation of the output vector represents the properties of the entity. When the position, scale or other state of a detected feature changes in the image, the length of the vector remains the same, yet its orientation changes. We employ a squashing function to map the length of the output vectors between the interval $[0, 1]$ such that short vectors have values close to zero whereas long vectors get shrunk to a value close to 1. By applying the squashing function, we obtain the output of capsule j, (denoted as $\mathbf{v}_j$) as follows:

$$\mathbf{v}_j = \frac{||\mathbf{x}_j||^2}{1 + ||\mathbf{x}_j||^2} \frac{\mathbf{x}_j}{||\mathbf{x}_j||} \tag{3}$$

In FACSCaps, outputs of primary capsules are fed to AU Capsules and the outputs of AU capsules are used both for AU detection and for reconstruction of the input image. In AU detection part, the architecture learns to estimate occurrences of multiple AUs in the given image simultaneously. In reconstruction part, the architecture tries to reconstruct the input image from the representations in AU capsules. During training, AU capsules are masked such that only the capsules whose AUs are existent in the image are used to reconstruct the image. Outputs of AU capsules are fed to fully connected layers which are followed by ReLU layers.

3.3. Network Optimization

Since the architecture both estimates the multi-label AU occurrence and reconstructs the input image, the network is optimized by minimizing the following loss function:

$$L^{net} = L^{margin} + \alpha L^{reconstruction} \tag{4}$$

Table 1: Multi-view AU detection results (F1-score)

	Basline [29]	CNN	Fusion [19]	AUMPNet [4]	CNN-BLSTM [12]	FACSCaps
AU1	0.147	0.199	0.215	**0.219**	0.198	0.196
AU4	0.044	0.051	0.044	0.056	0.043	**0.067**
AU6	0.630	0.750	0.755	**0.785**	0.747	0.766
AU7	0.755	0.810	0.805	**0.816**	0.784	0.791
AU10	0.758	0.821	0.810	0.838	0.816	**0.840**
AU12	0.687	0.805	0.753	0.780	0.809	**0.819**
AU14	0.668	0.698	0.750	0.747	0.691	**0.764**
AU15	0.220	0.244	0.208	0.145	0.208	**0.247**
AU17	0.274	0.386	0.286	0.388	**0.398**	0.349
AU23	0.342	0.365	0.356	0.286	0.374	**0.413**
Mean	0.453	0.513	0.498	0.506	0.507	**0.525**

where α is the weight determining the ratio of effect of losses. If an image contains an AU, we want the corresponding output vector to be long, whereas the if an AU is non-existent in the image, we expect the output vector of the corresponding AU to be short. In order to enforce that, we use the following margin loss for each AU a:

$$L_a^{margin} = T_a \max(0, m^+ - ||\mathbf{v}_a||)^2 + \lambda_a(1 - T_a)\max(0, ||\mathbf{v}_a|| - m^-)^2 \quad (5)$$

where $T_a = 1$ if AU a exists in the image and $T_a = 0$ if a does not exist in the image. λ_a specifies the effect of losses obtained when the AU is present or absent in the image. Finally, margin loss is provided to be zero if $||\mathbf{v}_a|| > m^+$ when $T_a = 1$ and $||\mathbf{v}_a|| < m^-$ when $T_a = 0$. We compute the margin loss obtained from all AUs as:

$$L^{margin} = \sum_a L_a^{margin} \quad (6)$$

In reconstruction part, we compute $L^{reconstruction}$ as the sum of squared differences between the sigmoid outputs of the last fully connected layer and pixel intensities on images. We select α in Equation (4) to be a small value so that the reconstruction loss does not dominate margin loss.

4. Experiments

4.1. Settings

In our architecture, the Conv layer has 256 9×9 kernels with a stride of 4 and ReLU activation. In the PrimCaps layer, we have 4 channels of convolutional 8D capsules where each capsule has 8 convolutional units with a 9×9 kernel. Capsules in $[11 \times 11]$ grid share weights and we obtain $[4 \times 11 \times 11]$ 8D capsule outputs. Each capsule in PrimCaps layer is connected to each capsule in AUCaps layer by a weight matrix $\mathbf{W}_{ij}$ of size $[8 \times 8]$. We have 10 AUCaps for each AU. During reconstruction we use three fully connected layers of sizes 512, 1024 and 16384 (128×128), respectively. The first two fully connected layers have ReLU activation function while the last layer has Sigmoid activation function. Margins of L^{margin} are selected as $m^+ = 0.9$ and $m^- = 0.1$ as suggested in [23].

During optimization we used Adam optimizer with a learning rate 0.001 and decaying learning rate weight 0.9. We select $\alpha = 0.0005$ to use reconstruction as a regularizer. We used 3 iterations of dynamic routing.

4.2. Multi-label AU detection

We resized the frames in FERA2017 to 128×128. In order to compare our results with the ones in the literature, we have followed the protocol of the FERA2017 challenge. We have trained our FACSCaps architecture using only the frames in the training set of FERA2017 challenge dataset and we report result on the test set.

4.2.1 Comparison.

We compare our results with FERA2017 baseline [29] and the recent studies which perform fusion of multiple features and AUs (Fusion) [19], employ a CNN architecture to estimate pose and AU occurrence together (AUMPNet) [4], combine CNN and BLSTM (CNN-BLSTM) [12] on FERA2017 challenge dataset. We also trained a standard CNN having a similar architecture our FACSCaps as suggested in [23]. The baseline CNN we trained has three convolutional layers having 256, 256, 128 channels with 5×5 kernels. Since we perform multi-label detection of multiple AUs, we used binary cross-entropy loss. We also trained the CNN architecture with Adam optimizer [17].

In Table 1 we share the average F1-score values over nine poses for each AU separately. Results reflect that, on average of 10 AUs, our FACSCaps architecture performs

Table 2: AU detection results (F1-score) for each pose and each AU

		Pose								
	Skew	1	2	3	4	5	6	7	8	9
AU1	14.28	0.205	0.186	0.132	**0.230**	0.203	0.228	0.213	0.207	0.160
AU4	37.23	0.074	**0.085**	0.066	0.055	0.055	0.080	0.074	0.076	0.038
AU6	1.29	0.759	0.754	0.737	0.763	0.776	0.767	0.770	**0.786**	0.777
AU7	0.57	**0.806**	0.794	0.792	0.789	0.783	0.790	0.782	0.789	0.793
AU10	0.64	0.836	0.821	0.825	0.854	0.846	0.832	**0.858**	0.850	0.835
AU12	0.98	0.798	0.812	0.811	0.816	0.829	0.815	**0.841**	0.840	0.811
AU14	0.66	0.751	0.767	0.783	0.739	0.772	0.768	0.747	0.758	**0.789**
AU15	6.87	0.188	0.219	0.269	0.228	**0.276**	0.274	0.238	0.263	0.263
AU17	5.60	0.395	**0.442**	0.407	0.372	0.367	0.333	0.293	0.265	0.265
AU23	3.33	0.448	0.445	0.454	**0.491**	0.458	0.472	0.306	0.328	0.312
Mean	7.15	0.526	0.533	0.528	0.534	**0.537**	0.536	0.512	0.516	0.504

the best among others. Moreover, FACSCaps achieves the best F1-score values in six out of ten AUs (AU4, AU10, AU12, AU14, AU15 and AU23). For three AUs (AU1, AU6 and AU7), AUMPNet achieves the best performance. Note that, a subset of development set was used for early stopping evaluation during the training of AUMPNet.

In [26], authors fine-tuned the pre-trained VGG-Faces architecture with training set of FERA2017 and obtained an average F1-score of 0.574. Since the pre-trained architecture is trained with millions of faces having lots of views and all other methods used only the training set of FERA, it would not be fair to compare results of this method with others including ours.

4.2.2 Detailed results for each pose.

In addition to the comparison of FACSCaps with other methods, we provide results for each pose and each AU separately in Table 2. In the second column, we also share the degree of skew (ratio of negative samples to positive samples). Results show that the best average F1-score is obtained in pose 5 (small yaw variation). However, different AUs achieve their best results in different AUs. For example AU10 and AU12 have the best F1-scores on pose 7, whereas AU1 and AU23 have the best F1-scores on pose 4. We can also infer from Table 2 that changes in the F1-scores of AUs for different poses are small.

4.2.3 Cross-pose AU detection.

Since we expect capsules to learn the internal representations independent from the view angle, we would expect our architecture to detect AUs in a view it has never seen before. To verify this, we perform cross-pose experiments, in which we train the architecture using eight of the nine poses of training set and test it with the remaining pose of test set. We report the F1-score results of cross-pose experiments in Table 3. We hypothesize that, detecting AUs in more extreme poses such as pose 1 (-40 degrees yaw, -40 degrees pitch), pose 3 (0 degrees yaw, -40 degrees pitch), pose 7 (-40 degrees yaw, 40 degrees pitch) and pose 9 (0 degrees yaw, 40 degrees pitch) would be more difficult using only the remaining poses since they are more difficult to interpolate from other poses. On the other hand, detection of AUs in poses which are in the middle of two other poses would be easier. For example, pose 2 (-20 degrees yaw, -40 degrees pitch) can be interpolated using pose 1 (-40 degrees yaw, -40 degrees pitch) and pose 3 (0 degrees yaw, -40 degrees pitch). Since frames from pose 1 and pose 3 are included in the training set, we expect higher AU detection results compared to more extreme poses.

Results in Table 3 reflect that we obtain lower average F1-scores for extreme poses 1 (0.469), 3 (0.509), 7 (0.448) and 9 (0.475) compared to others. Moreover, the cross-pose results obtained for poses which are in the middle of other poses are better compared to their corresponding surrounding poses. Pose 2, pose 5 and pose 8 are in the middle of pose pairs 1-3, 4-6 and 7-9 in terms of yaw, respectively. Similarly, pose 4, pose 5 and pose 6 are in the middle of pose pairs 1-7, 2-8 and 3-9 in terms of pitch, respectively. It can be seen that results obtained by these poses in the middle are better than the ones obtained for surrounding pose pairs. We can also note that pose 5 is in the middle of poses 4 and 6 in terms of yaw angle and in the middle of poses 2 and 8 in terms of pitch angle. Therefore, AUs in pose 5 are the easiest to detect leading to the best average F1-score value (0.533) in Table 3. Note that, since other studies including FERA 2017 challenge paper [29] do not report cross-pose experiment results, we cannot compare our cross-pose results with other methods.

Table 3: Cross-pose F1-score results

	Pose to be tested								
	1	2	3	4	5	6	7	8	9
AU1	0.155	0.180	0.130	0.188	**0.194**	0.186	0.160	0.167	0.169
AU4	0.056	0.052	**0.076**	0.070	0.064	0.039	0.046	0.032	0.039
AU6	0.674	**0.789**	0.720	0.736	0.775	0.784	0.655	0.773	0.733
AU7	0.618	0.764	**0.796**	0.768	0.756	0.769	0.789	0.782	0.750
AU10	0.706	0.801	0.786	0.846	0.846	**0.847**	0.835	0.830	0.800
AU12	0.771	0.806	0.748	0.798	0.829	0.810	0.688	**0.842**	0.768
AU14	0.750	0.774	0.746	0.681	**0.779**	0.775	0.556	0.770	0.684
AU15	0.115	0.251	0.243	0.222	0.248	0.240	0.227	0.262	**0.280**
AU17	0.366	**0.423**	0.408	0.362	0.388	0.344	0.250	0.319	0.197
AU23	**0.481**	0.430	0.439	0.406	0.448	0.462	0.270	0.356	0.325
Mean	0.469	0.527	0.509	0.508	**0.533**	0.526	0.448	0.513	0.475

4.3. Perturbing dimensions of a capsule

During training, we pass the encoding of AUs existent in the input image and mask out the AUCaps of absent AUs while reconstructing the input image. Therefore, we expect AUCaps to learn the variations in the given AU classes and dimensions of AUCaps to learn to span the space of variations. In order to understand what each individual dimension of each AU capsule has learned, we perturb only a single dimension of activity vector of the corresponding capsule and then reconstruct the image from the perturbed vector. We plot the examples of synthesized image from perturbed activity vectors in Figure 3.

We can observe that, all of the frames in a given row contain the corresponding AU denoted in front of the row but frames in the same row vary in one or more dimensions. We explain the variations from the first image to last image in a row. In the examples of AU1 and AU4, we observe changes in the yaw angle and AU intensity as we perturb individual dimensions. For AU6, variations appear in pitch and yaw directions and illumination. The most obvious variation for the AU10 sample is the size of the head. We plot four example rows for AU12, each denoting variations in different dimensions. In the first row of AU12, pose is changing from pose 1 to pose 3 in FERA2017 dataset and mouth opens as the intensity of AU12 increases. Faces in second row of AU12 are mainly frontal but smile intensity increases. In the third row of AU12, size of the head, yaw angle and identity changes. Fourth row of AU12 represent variation from pose 7 to pose 9 of FERA2017 dataset. Moreover, as the face rotates in yaw dimension in the row of AU14, dimple becomes more visible. In the first row of AU17, pitch and yaw directions change together as the intensity of AU17 decreases, while in the second row of AU17, head size and pose changes. Finally, synthesized perturbed images obtained for AU23 are blurry, but the variation in the pose and shape of the mouth is visible.

From these synthesized frames we can infer that, individual dimensions of corresponding AU capsules learned to represent variations in instantiation parameters such as pose, head size, AU intensity, etc.

4.4. Occlusion Sensitivity Maps

We generated Occlusion Sensitivity Maps [33] for different poses and different AUs. We modified the pixel values of patches having size 15×15 in the original image with 0.5 (gray color). We slide the patch over the image of size 128×128 with a stride 2. Therefore, we obtain modified images in which the patch resides in 57×57 different positions. For each AU and each pose, we select 25 images from all of the test participants (750 frames) containing the specified AU. Then, we test the modified frames (nearly 2.4 million frames for each pose) and obtained accuracy values for each position of the patch on the image. After an interpolation step, the resulting grid of accuracy values gives us occlusion sensitivity maps as shown in Figure 4. In these maps, darker red colors represent the lowest accuracy of correctly estimating positive samples while darker blue colors represent the parts, whose occlusion do not affect the accuracy a lot. Therefore, the significant regions for each AU are the ones colored with red.

From the maps we can infer that FACSCaps architecture correctly learns where to look in the image for most of the AUs. Since the variation in the yaw angle is only in one direction in the dataset, occlusion sensitivity maps are generally not symmetric. It can be observed that for mainly AU1, AU6 and AU10, the regions our model mainly looks at are on the right side of the face.

In the top row of Figure 4, for poses 4-6, the model focuses on upper eyebrows to detect AU1. Yet, for other

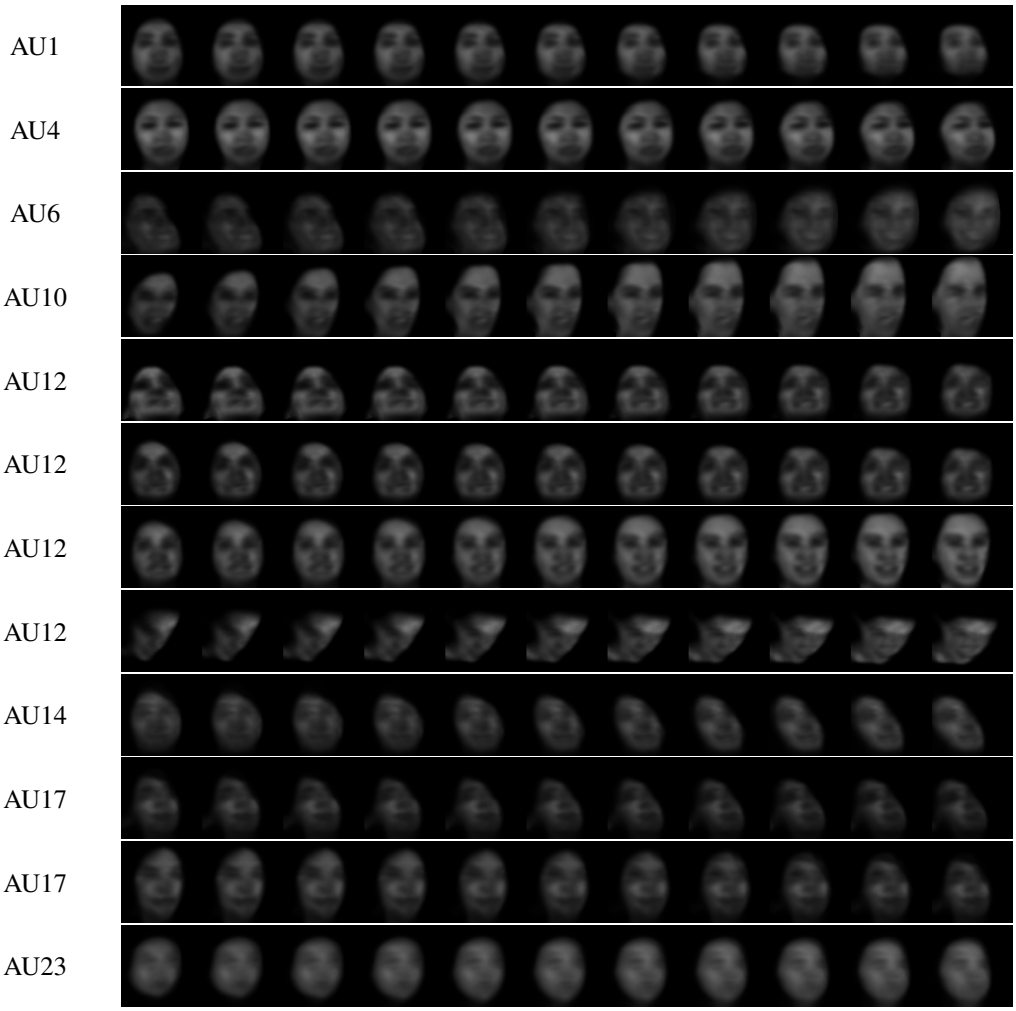

Figure 3: Synthesized images obtained by perturbing individual dimensions of activity vectors of AU capsules.

poses, whose yaw angle is not 0, the model looks at larger regions. For AU6, the model mainly looks at regions close to cheek and for poses 1-3 the focus region also contains lip corners. We can also observe weakly focused parts on eye corner regions where wrinkles appear during AU6. Occlusion sensitivity maps of AU10 mainly highlight upper lip regions as expected. Moreover, for AU12 and AU14, the model looks at regions covering mouth and lip corners. For AU15, the model looks at a larger region around mouth, covering both lip corners and chin. This can be explained by the fact that while moving the lip corners down, AU15 may flatten or cause bulges to appear on the chin boss. For AU17, significant regions contain mouth and chin regions as expected. Finally, our classifier looks at mainly mouth region for AU23.

Although focused parts contain upper eyebrows and nose, maps obtained for AU4 do not reflect regions very specific to brows. It can be explained by the fact that the dataset is highly skewed in terms of AU4, which also leads to very low F1-scores. Moreover, since AU7 does not lead to an obvious change in the appearance, the model could not learn where to look at for AU7 correctly. Since the F1-score values are high for AU7, we can say that while classifying AU7, our model considers other visual changes on the face, which are caused by the AUs that co-occur with AU7.

5. Conclusions

In this paper, we proposed FACSCaps architecture for multi-view, multi-label AU detection. FACSCaps architecture both estimates the occurrence of multiple AUs and learns to reconstruct the input image. With the help of masking during reconstruction, AUCaps are forced to learn the variations in the corresponding AUs. When multiple AUs are present in training images, corresponding AUCaps remain unmasked, which provides multi-label learning of AUCaps representations. Similarly, the architecture per-

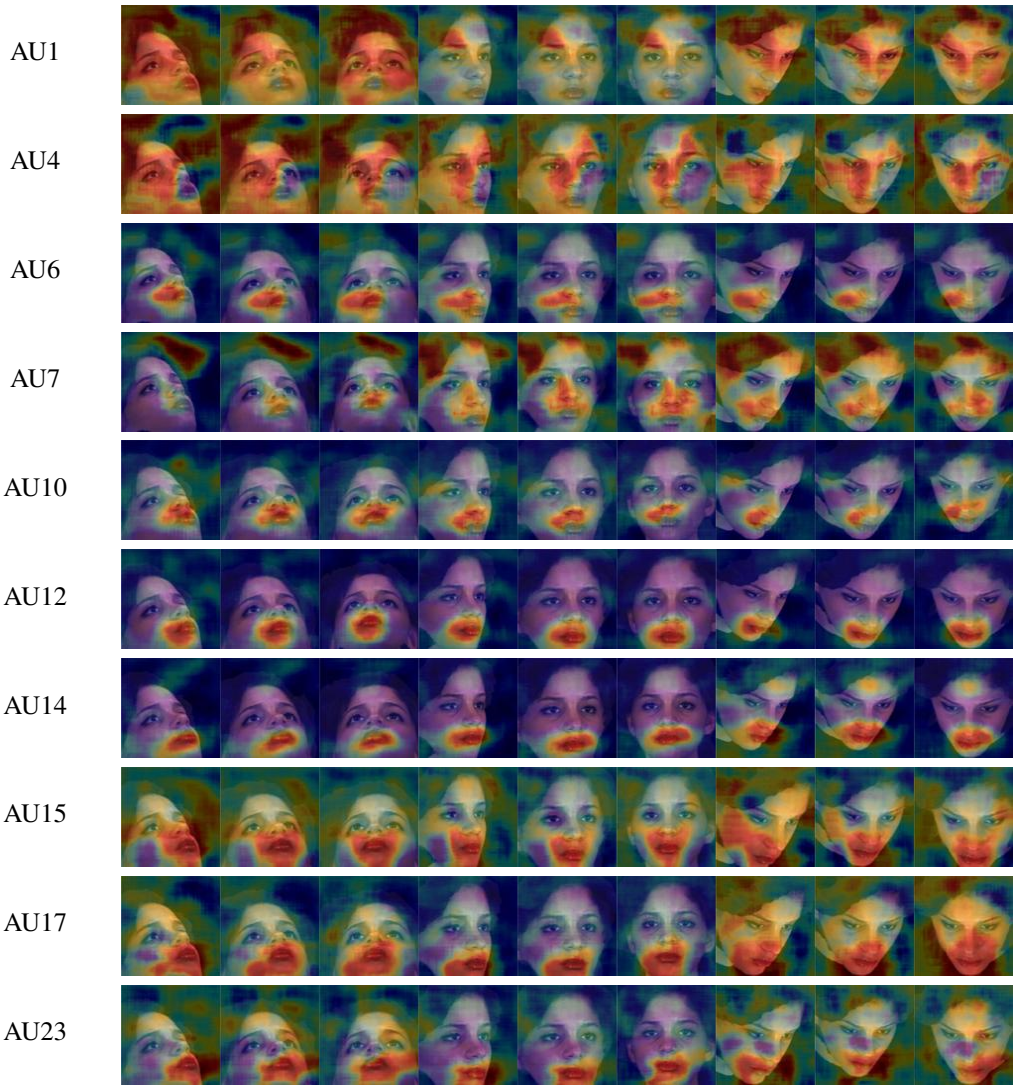

Figure 4: Occlusion sensitivity maps overlaid on a neutral frame for poses 1-9. The modified images are evaluated for accuracy of positive examples. Performance is color coded at the central pixel of the patch. Best viewed in color.

forms multi-label AU detection, which considers the correlation between AUs.

We showed that FACSCaps outperforms baseline approaches employing hand-crafted features and CNNs. Moreover, our cross-pose experiment results reflect that FACSCaps can detect AUs from the face images having unseen viewpoints. By perturbing the individual dimensions, we could interpret and visualize what is learned by each dimension of capsules. We observed that variation in pose is represented in many dimensions and as we perturb each dimension, multiple instantiation parameters such as pose, head size, AU intensity, illumination, etc. changes.

Finally we occluded parts of frames using a sliding patch and measured the decrease in the accuracy of positive samples by testing the occluded frames. Occlusion sensitivity maps are generally consistent with the expected regions related to AUs and pose. We can conclude that, employing capsule networks for AU detection both provides promising results and brings more interpretability on the variation of data. A future direction would be integration of AU intensity estimation to the architecture and joint estimation of AU intensity and AU occurrence. Moreover, it is worth exploring that whether training capsule networks with thousands of faces having lots of views would lead better representations compared to CNN-based VGG-Faces architecture.

Acknowledgements: Preparation of this article was supported in part by the National Institute of Mental Health of the National Institutes of Health under Award Number MH096951.

References

[1] P. Afshar, A. Mohammadi, and K. N. Plataniotis. Brain tumor type classification via capsule networks. *arXiv preprint arXiv:1802.10200*, 2018.

[2] P.-A. Andersen. Deep reinforcement learning using capsules in advanced game environments. *arXiv preprint arXiv:1801.09597*, 2018.

[3] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Openface: an open source facial behavior analysis toolkit. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–10. IEEE, 2016.

[4] J. C. Batista, V. Albiero, O. R. Bellon, and L. Silva. Aumpnet: simultaneous action units detection and intensity estimation on multipose facial images using a single convolutional neural network. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 866–871. IEEE, 2017.

[5] 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 & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 25–32. IEEE, 2017.

[6] 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.

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

[8] S. Eleftheriadis, O. Rudovic, M. P. Deisenroth, and M. Pantic. Gaussian process domain experts for model adaptation in facial behavior analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–26, 2016.

[9] S. Eleftheriadis, O. Rudovic, and M. Pantic. Multi-conditional latent variable model for joint facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3792–3800, 2015.

[10] S. Ghosh, E. Laksana, S. Scherer, and L.-P. Morency. A multi-label convolutional neural network approach to cross-domain action unit detection. In *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pages 609–615. IEEE, 2015.

[11] 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.

[12] J. He, D. Li, B. Yang, S. Cao, B. Sun, and L. Yu. Multi view facial action unit detection based on cnn and blstm-rnn. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 848–853. IEEE, 2017.

[13] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2017.

[14] G. Hinton, N. Frosst, and S. Sabour. Matrix capsules with em routing. 2018.

[15] A. Jaiswal, W. AbdAlmageed, and P. Natarajan. Capsulegan: Generative adversarial capsule network. *arXiv preprint arXiv:1802.06167*, 2018.

[16] S. Jaiswal and M. Valstar. Deep learning the dynamic appearance and shape of facial action units. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE, 2016.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.

[19] X. Li, S. Chen, and Q. Jin. Facial action units detection with multi-features and-aus fusion. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 860–865. IEEE, 2017.

[20] Y. Li, J. Chen, Y. Zhao, and Q. Ji. Data-free prior model for facial action unit recognition. *IEEE Transactions on affective computing*, 4(2):127–141, 2013.

[21] D. Linh Tran, R. Walecki, S. Eleftheriadis, B. Schuller, M. Pantic, et al. Deepcoder: Semi-parametric variational autoencoders for automatic facial action coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3190–3199, 2017.

[22] S. Lucey, A. B. Ashraf, and J. F. Cohn. Investigating spontaneous facial action recognition through aam representations of the face. In *Face recognition*. InTech, 2007.

[23] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017.

[24] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1113–1133, 2015.

[25] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost. Facial action recognition combining heterogeneous features via multikernel learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):993–1005, 2012.

[26] C. Tang, W. Zheng, J. Yan, Q. Li, Y. Li, T. Zhang, and Z. Cui. View-independent facial action unit detection. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 878–882. IEEE, 2017.

[27] Y.-I. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 23(2):97–115, 2001.

[28] Z. Tősér, L. A. Jeni, A. Lőrincz, and J. F. Cohn. Deep learning for facial action unit detection under large head poses. In *European Conference on Computer Vision*, pages 359–371. Springer, 2016.

[29] M. F. Valstar, E. Sánchez-Lozano, J. F. Cohn, L. A. Jeni, J. M. Girard, Z. Zhang, L. Yin, and M. Pantic. Fera 2017-addressing head pose in the third facial expression recognition and analysis challenge. In *Automatic Face & Gesture*

Recognition (FG 2017), 2017 12th IEEE International Conference on, pages 839–847. IEEE, 2017.

[30] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu. Sentiment analysis by capsules. 2018.

[31] Z. Wang, Y. Li, S. Wang, and Q. Ji. Capturing global semantic relationships for facial action unit recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3304–3311. IEEE, 2013.

[32] E. Xi, S. Bing, and Y. Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017.

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[34] X. Zhang and M. H. Mahoor. Task-dependent multi-task multiple kernel learning for facial action unit detection. *Pattern Recognition*, 51:187–196, 2016.

[35] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014.

[36] Z. Zhang, J. M. Girard, Y. Wu, X. Zhang, P. Liu, U. Ciftci, S. Canavan, M. Reale, A. Horowitz, H. Yang, et al. Multimodal spontaneous emotion corpus for human behavior analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3438–3446, 2016.

[37] 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 *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 2207–2216. IEEE, 2015.

[38] K. Zhao, W.-S. Chu, and H. Zhang. Deep region and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3391–3399, 2016.