Facial Action Unit Detection with Capsules

Maheen Rashid Yong Jae Lee
University of California, Davis
mhnrashid@ucdavis.edu

Abstract

In this paper we motivate the use of capsule networks for facial action unit detection. Action unit activations may be seen as local part deformations - for example AU 1,2, and 4 are deformations of the part 'eyebrows'. Different part deformations in a regular convolutional network must be modeled and represented as separate neurons since each neuron has a scalar activation. However, with capsule networks different part deformations can be modeled by the same capsule since it has a more expressive vector based representation. We argue that this property makes capsule networks well suited for the task of action unit recognition and demonstrate this by developing and testing capsule networks for the AU recognition. We achieve state-of-the-art results on the BP4D and DISFA datasets. We analyse the learned capsules' properties and find that capsule magnitude correlates with expression intensity and that capsule pose captures varied attributes such as face size, lighting, pose, and skin color. Finally we use activation gradient ascent to visualize capsule direction, and find that a single capsule can represent multiple deformations of the same part, while a single convolution neuron does not.

1. Introduction

Facial Action Coding System (FACS) is a system to define and name facial movements by their appearance on the face. Informed by the underlying muscular structure of a face, FACS annotation can be reliably used for describing as well as identifying facial expressions. There are 24 main facial action units to describe the human face. Additionally, action units can also be coded for intensity on a 4 to 5 point scale. While, extremely useful, manual action unit coding is a cumbersome process that can only be carried out by trained experts. Due to this hurdle, automatic action unit detection is an important problem for computer vision research.

Facial action unit detection requires identifying subtle deformations on parts of the face. Consequently, features that capture local movements around key parts of the face have been used to train machine learning systems. For example, in [1], a seminal work on emotion and facial action unit understanding, Gabor features were extracted around keypoints of the face to capture local muscle deformations. Developing features that capture part deformations well has also motivated more recent work [42, 19] where separate convolutional filters are trained to correspond to different parts of the face. The motivation behind these works is similar – the better we are able to model how parts of a face look and change, the better we can detect action units.

Recently, capsule networks [30, 13] were introduced. In theory, capsules have two primary advantages over regular convolutional neural networks (CNNs). Intuitively, neurons in a CNN can represent an attribute of the input image – such as the presence of an eye – and the activation of a neuron represents a confidence value in whether that attribute can be found in the image. With capsules the expressive power is increased - its activation can represent a confidence value in its presence or absence, and its direction can represent properties of the attribute. For example, the direction of the capsule can indicate how rotated the eye is, whether it is open or closed, etc. The second advantage of capsules is that the additional representative capacity allows for complex routing procedures. The pose of a capsule can be used to determine how it is propagated through the network. This is in contrast to neuron activations in a CNN that are propagated solely on the basis of its scalar value. As a result, the routing procedure followed by capsule networks can mimic the effect of a much deeper convolution neural network, trained with various data transformed augmentation techniques.

We believe that the higher expressive power capsules afford to each visual attribute of an image can directly translate to better modeling of local part deformations. Action unit activations can be seen as local part deformations, and therefore capsules can be better at detecting and modeling action units. To give a naive parallel example, if a capsule learns to detect lips, its pose can represent the type of deformation the lips are in – so different capsule poses can represent action units 12, 14, 15. At the same time, for a convolutional network, if a neuron comes to be associated

with lips, it cannot express the pose the lips are in. It can simply be either active - indicating the presence of lips – or inactive – indicating that the lips are not present. In order to represent lips in AU-12 or in AU-15, the network must learn to associate separate neurons.

Contributions. Our main contributions are:

1) We present results that indicate that capsules are indeed better than CNNs at modeling local part deformations - and therefore action units. Previous work [30, 13] have shown that capsules can model global deformations - so the network can generalize well across image level deformations (affine transformations of MNIST), or viewpoint deformations (azimuth changes on small-NORB). In this paper, we show that capsules can also capture small deformations well - and may therefore be useful in other areas such as fine-grained classification.

2) We develop a capsule network for action unit detection that gives state-of-the-art results across two large action unit datasets. On BP4D dataset we outperform the closest baseline architecture by 14.1% in AUC. We replicate similar performance gains on DISFA. We additionally present results on emotion recognition for CK+ dataset, and find that our architecture generalizes well.

3) To the best of our knowledge, capsule networks have not been used to perform facial action unit detection. To this end, we thoroughly analyze and visualize the learned capsule networks. We visualize the effect of changing capsule magnitude as well as capsule direction via a reconstruction network. We find that capsules are able to model face pose, shape, lighting, and skin color, and that capsule magnitude is correlated with action unit or emotion intensity. Finally, we use activation maximization visualization on capsule features and compare them with regular convolutional networks and find that a single capsule can capture visually dissimilar part deformations across identities and across parts of the face.

2. Related Work

Capsules Capsule networks were first proposed in [30]. The network replaces scalar neurons with higher dimension capsules - so that activation and neuron attributes can be modeled jointly. In addition, capsule direction or pose can be used to route capsules between higher layers - which replaces pooling based routing in convolutional networks. In [13], the authors propose vector capsules whose magnitude represents the activation of a capsule. Iterative routing is done using a simple agreement between lower and higher level capsule directions. In [13] the authors introduce matrix capsules, where a separate value represents the capsule activation. Routing is done using an EM algorithm, such that the probability distributions of higher and lower level capsules between consecutive layers are in agreement. Furthermore, the authors introduce convolutional capsules, whereas [30] only worked with fully-connected capsules. In this paper, we use vector capsules, with dynamic routing, and work with fully-connected capsules only.

Facial Action Unit Understanding Papers in action unit understanding have focused on two broad sub problems - action unit intensity estimation, and action unit detection.

A number of traditional non-deep approaches improve action unit understanding by exploiting the co-occurrence patterns between action units - either by developing a learning model that can help capture inter-AU relations [35, 38, 7], by developing a model based on prior knowledge of AU relationships and semantics [33, 20], or by using a data-driven approach to learn important AU relationships [45]. In particular, [41] jointly learns to identify important patches, and positive and negative correlations between action units for understanding action units.

Traditional approaches have also learned action units by assistance from facial keypoints - features extracted around keypoints are used for action unit detection. Some examples of such approaches are [7, 18, 3, 34, 37].

Deep learning has also been applied to the problem of action unit detection with great success [9, 10, 14, 11, 42, 19]. Among these, two papers in particular require discussion. In [42], the authors develop a 'region layer' that splits the incoming convolution map into a grid and develops separate convolutional maps for each grid section. The resulting map is concatenated spatially and propagated through the network. In a similar vein, [19] also explicitly design their deep network to develop features for parts of a convolutional map. Both methods explore a similar idea - to develop separate features for parts of a face - as is based on the intuition that different areas of the face correspond to different AU activations that require their own unique set of features for identification.

More recently, [12] develops separate neural networks to detect each action unit where convolutional filter size is learned during training, while [40, 29] do weakly-supervised action unit recognition.

In concurrent work [5], input images are cropped into patches that are in turn used to predict AUs via fusion of multiple deep neural networks' predictions and a message passing based structure inference module. In [31], additional annotated data is used to do joint face alignment and AU detection while using an adaptive attention module.

Modeling Facial Expressions Facial expression prediction is a well-explored topic of research in computer vision. We primarily focus on action unit detection, but also show qualitative and quantitative results on expression detection. Some approaches that do not use deep learning are, [1, 8, 32, 44], of which [1] is of particular note for creating a pipeline based on extracting features around facial

keypoints, detecting action units, and fusing action unit detections temporally for emotion detection.

A number of papers also explore emotion understanding in a simple deep feed-forward classification network setting [23, 16, 26, 43, 6]. [21, 22] attempt to enforce AU understanding to the end of emotion classification. Of these, [16] is notable for impressive results on expression detection and demonstrating the importance of data augmentation for the task of expression understanding. [6] is also an important paper that proposes a two-stage training pipeline to transfer VGG-Face [28] features for the task of expression classification. Also noteworthy is [15], which uses facial keypoint locations over time to train a network that is meant to capture temporal deformations alongside a traditional image-based CNN. Lastly, [17] proposes an encoder-decoder architecture that learns from pairs of neutral/non-neutral expressions to develop features that are discriminative for expression classification. Similarly, in [36] learns to identify facial expressions as the residue between faces showing non-neutral expressions, and their corresponding neutral expression.

3. Approach

Our proposed network comprises three parts - convolution layers, capsule layers, and a reconstruction module.

The first is a series of convolution layers that learn location invariant features and downsample the input. These are followed by a primary and a class capsule layer. Each primary capsule comprises multiple convolution filters whose outputs are joined to form a vector, then treated as the capsule activation. These primary capsules' vector outputs are then dynamically routed via direction agreement to the class capsules. There are n class capsules, each fully connected to the primary capsules, where n is the number of classes in the training data. Like primary capsules, each class capsule's activation is a vector, where its magnitude represents activation strength or the network's detection confidence for that class. Finally, the reconstruction network uses the class capsules' output to reconstruct the input image. Its input is the concatenated output of all class capsules, and its output is an image the same size as the input image. It comprises a series of fully-connected layers whose output is reshaped and bilinearly-upsampled to match the input size. We use the same modules as proposed in [30], to which we also refer the reader for details on the routing algorithm.

Intuitively, the convolution layers serve to develop low to mid level features for the task. They additionally scale down the feature map through pooling and strided convolutions for the ensuing computationally heavy capsule operations. The primary capsules further develop image features while simultaneously transforming scalar inputs from the convolution layers below to more complex vector representations. The class capsules collate the vector outputs of the primary capsules via routing to form the final class predictions. Additionally, the reconstruction network serves to 1) help visualize the properties learned by the class capsules, and 2) regularize the overall network and prevent it from over fitting.

We use a weighted modification of the margin loss from [30] to train our network. The loss for class c is:

$$L_c = w_c(T_c \max(0, m^+ - ||v_c||)^2 + \lambda(1 - T_c) \max(0, ||v_c|| - m^-)^2)$$

where $T_c = 1$ iff class c is present, $m^+ = 0.9$ and $m^- = 0.1$, v_c is the class capsule, and λ is a down weighting term for negative samples. The loss simultaneously encourages larger magnitude for positive class capsules, while encouraging negative class capsules to shrink. Given the high number of negative samples in our dataset, we set λ to 0.5, so that it is half as bad to predict a false positive as it is to predict a false negative. The class specific weight, w_c is the inverse of an action unit's frequency in the training data, normalized across all classes to sum to one. The addition of a class specific weight term balances the loss penalty so that infrequent action units are given the same importance during training as frequent action units.

Note that due to the 'squashing' procedure from capsule routing, capsule magnitudes necessarily lie between 0 and 1. Additionally, for single class classification (such as expression classification) a softmax operation is applied across all $||v_c||$ so that a single class has the maximum prediction.

While the margin loss serves to improve accuracy and train the network to identify action units (or expressions) correctly, the reconstruction loss serves to regularize the network while developing visually interpretive features. To this end, the reconstruction module is trained in a class agnostic manner; during training, the capsules for all classes apart from the ground-truth classes are zeroed-out and used as input to the reconstruction network. In this way, the reconstruction network does not, directly, affect classification accuracy. We use mean square error to supervise the reconstruction network.

The margin loss is averaged across all action unit instances in a batch, and added with average reconstruction loss for the batch. The final loss is:

$$L_{final} = L_{cls} + \alpha L_{recon} \tag{1}$$

where α is a weight parameter, L_{recon} is the average reconstruction loss, and L_{cls} is the average margin loss. We set α to bring the magnitude of the average reconstruction loss to be similar to that of the averaged margin loss at the beginning of training.

3.1. Architectures

We propose two different capsule architectures. Each is trained with three routing iterations between the primary

AU	LSVM[42]	JPML[41]	DRML[42]	CPM[37]	CNN+LSTM[4]	FERA[14]	OFS-CNN	Ours
1	23.2	32.6	36.4	43.4	31.4	28	41.6	**46.8**
2	22.8	25.6	**41.8**	40.7	31.1	28	30.5	29.1
4	23.1	37.4	43	43.4	71.4	34	39.1	**52.9**
6	27.2	42.3	55	59.2	63.3	70	74.5	**75.3**
7	47.1	50.5	67	61.3	77.1	78	62.8	**77.6**
10	77.2	72.2	66.3	62.1	45	81	74.3	**82.4**
12	63.7	74.1	65.8	68.5	82.6	78	81.2	**85**
14	64.3	65.7	54.1	52.5	**72.9**	75	55.5	65.7
15	18.4	**38.1**	36.7	34	33.2	20	32.6	33.7
17	33	40	48	54.3	53.9	36	56.8	**60.6**
23	19.4	30.4	31.7	39.5	38.6	41	**41.3**	36.9
24	20.7	42.3	30	37.8	37	-	-	**43.1**
Avg	35.3	45.9	48.3	50	53.2	51.7	53.7	**57.4**

AU	FVGG[19]	ROI[19]	Ours-VGG	Ours-VGGF
1	27.8	36.2	46.3	**47.3**
2	27.6	31.6	**41.6**	39.9
4	18.3	43.4	50.6	**52.8**
6	69.7	77.1	77	**77.9**
7	69.1	73.7	75.7	**79.9**
10	78.1	**85**	82.5	84
12	63.2	87	85.7	**88.1**
14	36.4	62.6	63.1	**67.2**
15	26.1	45.7	37.3	**49.2**
17	50.7	58	64.6	**65.4**
23	22.8	38.3	40	**47.7**
24	35.9	37.4	50.4	**55.1**
Avg	43.8	56.4	59.6	**62.9**

Table 1. F1-Frame results on BP4D dataset without(left) and with (right) external data.

AU	LSVM	JPML[41]	AlexNet	ConvNet	LCN	DRML	Ours	Ours-VGG	Ours-VGGF
1	20.7	40.7	34.9	49.4	51.9	55.7	65.7	65.3	**66.1**
2	17.7	42.1	25.8	51.3	50.9	54.5	56.0	**64.8**	63.7
4	22.9	46.2	36.1	47.4	53.6	58.8	70.2	70.0	**71.8**
6	20.3	40.0	48.3	52.2	53.2	56.6	71.3	**73.9**	73.7
7	44.8	50.0	54.3	64.8	63.7	61.0	60.6	62.4	**68.5**
10	73.4	**75.2**	54.3	61.4	62.4	53.6	70.8	68.3	70.1
12	55.3	60.5	50.0	60.2	61.6	60.8	74.6	**80.8**	**80.8**
14	46.8	53.6	47.7	29.8	**58.8**	57.0	56.7	55.8	57.5
15	18.3	50.1	34.9	50.6	49.9	56.2	59.4	61.5	**71.2**
17	36.4	42.5	48.5	53.5	48.4	50.0	66.1	69.7	**71.0**
23	19.2	51.9	40.5	49.5	50.3	53.9	61.6	64.4	**69.6**
24	11.7	53.2	31.7	52.5	47.7	53.9	67.6	73.9	**77.4**
Avg	32.2	50.5	42.2	51.8	54.4	56.0	65.0	67.6	**70.1**

AU	LSVM	APL	AlexNet	ConvNet	LCN	DRML	Ours	Ours-VGGF
1	21.6	32.7	47.8	44.2	44.1	53.3	58.2	**67.6**
2	15.8	27.8	52.1	37.3	52.4	53.2	61.4	**70.0**
4	17.2	37.9	44.0	47.9	47.7	60.0	72.4	**81.2**
6	08.7	13.6	44.3	38.5	39.7	54.9	64.9	**81.9**
9	15.0	64.4	48.7	49.5	40.2	51.5	66.1	**76.0**
12	93.8	**94.2**	55.3	54.8	54.7	54.6	76.1	85.5
25	03.4	50.4	50.2	48.4	48.6	45.6	75.6	**85.7**
26	20.1	47.1	45.8	45.8	47.0	45.3	**73.2**	72.7
Avg	27.5	46.0	49.1	45.8	46.8	52.3	68.5	**77.6**

Table 2. AUC scores on BP4D (left) and DISFA (right)

and class capsules.

Our first network is trained with inputs of size 96×96. The capsule network architecture comprises two convolution layers, with 64 and 128 filters, and kernel size of 5. Each is followed by max-pooling and ReLu. The convolution layers are followed by a primary capsule layer with 32 capsules of dimension 8, filter size 7, and stride 3. The resulting activation map is then fully connected to the class capsule layer with n capsules, each with dimension 32, where n is the number of output classes. The reconstruction network comprises 3 fully connected linear layers of dimension 512, 1024, and 1024. The last layer is reshaped to 32×32 and then bilinearly upsampled to 96×96. In experiments we refer to this model as 'Ours', and is trained from scratch for all experiments.

Additionally, we propose a larger capsule network with VGG convolution layers as its base. The network is identical to VGG-16 up to the end of its convolution layers. The last max-pooling layer is removed - resulting in an activation map of 14×14. This is followed by 32 primary capsules of size 8, kernel size of 3 and stride of 2. The following class capsules have dimension 32. The reconstruction network comprises three fully-connected layers with dimensions 512, 1024, and 9408. The output is resized to 56×56 and then bilinearly upsampled to 224 – the input image size. We show results of this model with the convolution layers initialized with both Imagenet pretrained weights (Ours-VGG) and VGG-Face pretrained weights (Ours-VGGF). For both variants, we found the network to be prone to over fitting despite usage of a reconstruction network and heavy data augmentation. We therefore add spatial dropout after the last convolution layer at 50% for 'Ours-VGG' and 70% for 'Ours-VGGF'.

Finally to balance the two losses in Equation 1 we set α to $1e-7$ at for the smaller network and $1e-8$ for the larger VGG based model.

4. Experiments

4.1. Action Unit Detection

Datasets We present results on two widely-used datasets. *BP4D* [39]: The dataset contains 328 videos of 31 subjects while completing eight different tasks designed to elicit emotion. Frames are annotated with 12 different action units. In total there are a little less than 140,000 frames that are usable. Following common procedure, we do 3 fold cross validation on subjects - train on 2 folds and test on the third. Results are collated across folds.

DISFA [25]: 26 subjects are recorded while watching videos. Action units and their intensity are annotated for each frame. Similar to BP4D, we conduct 3 fold cross validation, and collate results across folds.

For both datasets, we detect and align faces using [2]. For data augmentation, we randomly mirror, rotate, scale, translate, crop, and pixel augment the images.

Metrics We report F1-Frame score and AUC. The F1 score is the harmonic mean of precision and recall, and used by AU detection methods to report results. AUC is the area under the curve of the receiver operating curve and captures the relationship between true and false positives.

AU	LSVM[42]	APL[42]	DRML[42]	OFS-CNN[12]	Ours
1	10.8	11.4	17.3	**43.7**	[18.6]
2	10	12	17.7	**40**	[20.8]
4	21.8	30.1	37.4	**67.2**	[52.7]
6	15.7	12.4	29	**59**	[39.9]
9	11.5	10.1	10.7	**49.7**	[22.6]
12	[70.4]	65.9	37.7	**75.8**	64
25	12	21.4	38.5	**72.4**	[68.5]
26	22.1	26.9	20.1	**54.8**	[50.9]
Avg	21.8	23.8	26.7	**57.8**	[42.3]

AU	FVGG[19]	ROI[19]	Ours-VGGF
1	32.5	**41.5**	28.7
2	24.3	26.4	**29**
4	61	**66.4**	64.5
6	34.2	50.7	**58.6**
9	1.67	8.5	**51.2**
12	72.1	**89.3**	72.4
25	87.3	**88.9**	81.6
26	7.1	15.6	**54.5**
Avg	40.2	48.5	**55.1**

Table 3. F1-Frame results on DISFA dataset without (left) and with (right) external data.

Implementation Details For 'Ours' on BP4D, we train for ten epochs with a learning rate of 1e-4. For finetuning on DISFA, we transfer convolution features only, and train with a learning rate of 1e-4 for the first 5 epochs, and then at 1e-5 for the remaining 5 epochs.

For 'Ours-VGGF' we follow the procedure from ROI [19] and fix all convolution layers up to conv5. We finetune the conv5 layers at a 10 times lower learning rate, and train for 5 epochs. The capsule, and reconstruction layers are trained with a 1e-4 learning rate. For 'Ours-VGG', we additionally finetune the first four convolution layers, and train for 10 epochs. For DISFA finetuning, we initialize the convolution layers' weights with the best performing BP4D model (which is Ours-VGGF for the first fold of BP4D), and keep them fixed while training the capsule and reconstruction layers for 4 epochs at 1e-4 learning rate.

Results In Table 1, we compare F1 scores of our method against baselines. Without external data, our method outperforms the closest baseline by 4.2%, and with external data, our method outperforms ROI by 6.5%. Note that we do not compare against versions of ROI that use sequential input. In Table 2, we show AUC results for both datasets, and in Table 3, we compare F1 scores for DISFA. Without external data, while our method does not outperform OFS-CNN [12] we outperform the second closest baseline by 15.6%. With external data, our performance is similarly high with a 6.6% margin. On BP4D, OFS-CNN is not able to outperform our model.

Since OFS-CNN [12] relies on separate small neural networks to develop specialized features for each action unit, it can perform well on DISFA - a small dataset with relatively even distribution of different action units, but has less impressive performance on BP4D which may need a larger capacity network and joint learning across AUs. ROI [19] on the other hand uses a bigger model that can easily fall in to the trap of exploiting correlation information to make predictions without understanding the visual features of each AU. Our method on the other hand forms a happy medium - the capsule network is relatively lightweight and can be trained on a small dataset with careful regularization. At the same time, it is jointly trained to identify all AUs and can therefore learn features that are collectively useful. Lastly, our use of capsule units allows our network to learn and propagate visual properties that are complex and cannot be captured as easily by a convolution network of equivalent size. We further analyze this quality in Section 4.4.

Method	Accuracy
AURF[21]	92.2
AUDB[22]	93.7
Khorrami[16]	96.4
GCNet*[17]	97.28
DeRF*[36]	97.30
FN2EN*[6]	96.8
Ours	93.7
Ours-Max	96.2

	BP4D	DISFA
DSIN[5]	58.9	53.6
JAA-Net[31]	60	56
Ours-VGGF	62.9	55.1

Table 4. **(Left)** Results on 8 emotion classification against several state of the art methods. Starred methods use external data. **(Right)** Comparison of F1-Frame scores with concurrent work. Our method gives comparable results, while being simpler and easier to train.

Finally, in Table 4, we compare the F1 score of our best model against concurrent work for both datasets. Despite a simple feed forward architecture operating on the entire image unlike [5], and without the use of additional annotation unlike [31], our method has a comparable F1 score.

4.2. Emotion Detection

We also explore the use of capsules on the related task of emotion recognition on the Cohn-Kanade dataset [24]. We follow the established protocol of 10 fold cross validation, and average results across folds.

Table 4 (left) shows our results on 8 emotion classification against several state-of-the-art methods. Our results are comparable to the state-of-the-art. We found that test results were prone to fluctuate throughout training, and due to the small dataset size, some folds were prone to overfit. We therefore also report the best test accuracy we achieve during training as 'Ours Max' to provide an idea of the upper limit our model may achieve with more careful training and hyperparameter searching.

4.3. Visualizing Capsules by Reconstruction

Every class capsule is a 32 length long vector. This vector can be modified by rescaling its magnitude, or altering its direction. For each altered version of an input image's correct class capsule, we can use the reconstruction network to visualize the effect.

Magnitude Since capsule magnitude represents the confidence of our network in a capsule class's presence in the image, we expect increasing capsule magnitude to create reconstructions that represent that class even more. For example, we would expect the reconstruction of a 'surprise' capsule with less magnitude to show less surprise than that with a higher magnitude.

Figure 1 shows reconstructions of different expression capsules with increasing magnitude. From left to right the magnitude is increased from 0.1 to 0.9 at increments of 0.1.

Figure 1. We show the effect of altering magnitude of a class capsule. As activation strength of a capsule increases, the intensity of the facial expression also increases.

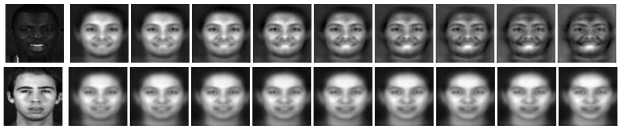

Figure 2. Reconstructions of test images with increasing capsule magnitude (left to right) on the BP4D dataset

The leftmost image is the input image. For each of the capsules, we see the expression become more pronounced and exaggerated. For example, for happiness (row 1), the smile in subsequent reconstructions becomes wider. Similarly, for surprise (row 2), a dark spot resembling an open mouth begins to appear, eventually resembling a full jaw drop.

When we repeat this experiment with our AU detection network, this time increasing the magnitude of all action units that are present in the image, the results are less clear. For BP4D Figure 2 increasing magnitude results in unique features of the image being exaggerated, such as skin color (1st row). At the same time, certain AUs also become prominent. The open mouth smile in the 1st row, the jaw drop in the 2nd row become more apparent as magnitude is increased. The subtlety of these changes can be due the difference in AU training data where expression changes are spontaneous/subtle and not posed/exaggerated as in the Cohn-Kanade dataset, and are therefore unlikey to affect the reconstruction loss enough to create a strong supervision signal.

Direction We can also keep capsule magnitude stable, while changing its direction. For this, we vary the value of each of its 32 dimensions between -0.5 and 0.5, and reset the capsule magnitude to its original magnitude. In Figure 3, we show the effect of changing capsule directions. The capsule dimensions are associated with attributes as varied as face shape, skin color, pose, or the visibility of teeth.

4.4. Capsules and Local Deformations

We hypothesize that a single primary capsule is capable of activating multiple types of part deformations. In other words, the same capsule can have a high magnitude on the lip region when a person is smiling or grimacing or pouting

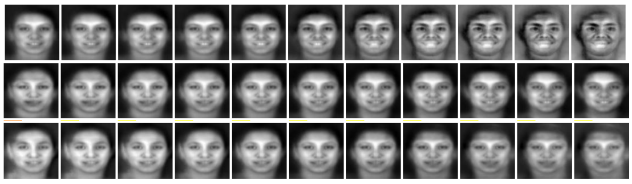

Figure 3. Visualizing capsule direction. The first row shows face features and skin change. In the second, teeth appear. The neck in the last row indicates that the capsules have learned pose attribute.

- but would have a different direction for each type of deformation. In this section, we find evidence to support our hypothesis, and use activation maximization visualization to gain insight into how capsules model local deformations.

In activation maximization visualization the input image is treated as a learnable layer, and changed by gradient ascent for a particular optimization function, such as the activation of a neuron in the network. As the network modifies the input image to increase a neuron's activation, the input image begins to show the visual attributes that the neuron has learned to identify. This process is popularly referred to as 'deep dreaming' - see [27] for an excellent overview.

Since the output of a capsule is a vector and not a scalar, we do gradient ascent on its magnitude. A naive application of this method is prone to create high frequency and nonsensical images, which we avoid by using gaussian blurring and random jittering between gradient ascent iterations as regularization. Recall that primary capsules comprise multiple convolution units that are applied to the input feature map in a sliding window fashion. By activating a primary capsule response at a certain feature map's spatial location, we can understand how the capsule interprets deformations of part of the face in its receptive field. Note that since we align our input images during training, the correspondence between primary capsules output maps' spatial locations and parts of the input face is easier to establish.

In order to test our hypothesis that a single capsule can model different part deformations via changes in direction, we need to ensure three conditions.

First, we need to ensure that the capsule is actually capable of modeling different types of part deformations - that

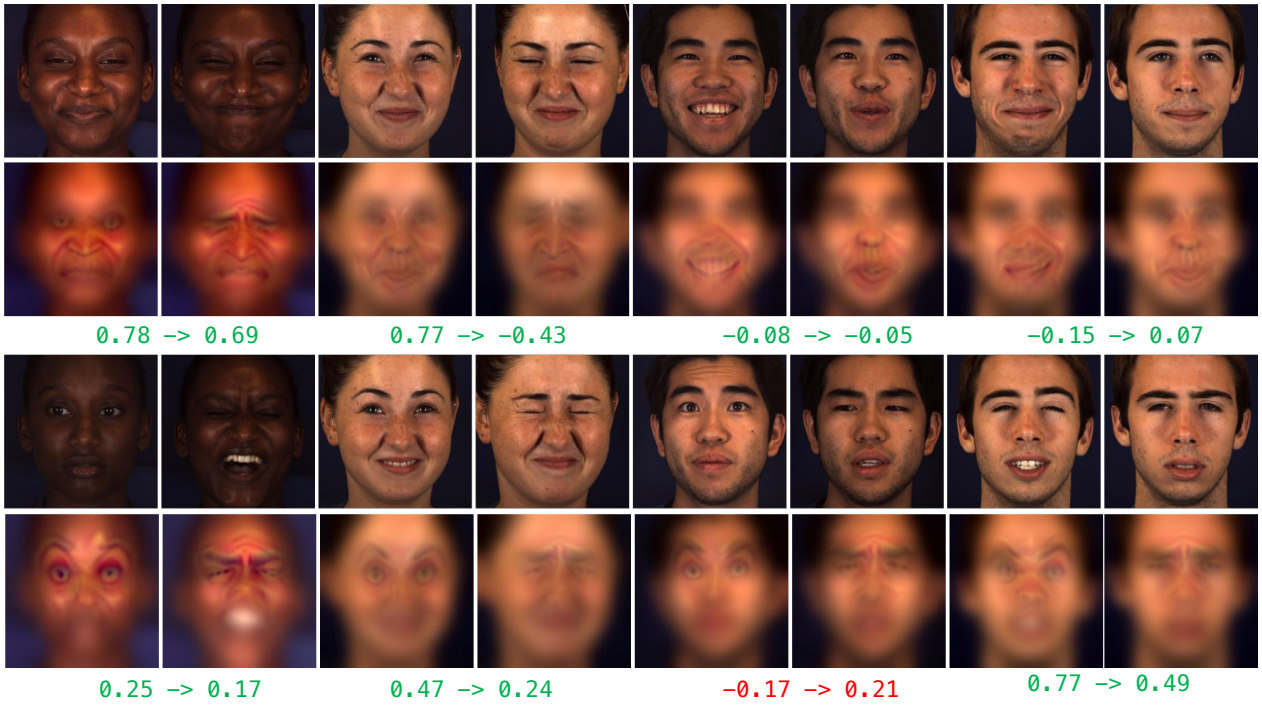

Figure 4. Activation maximization visualizing primary capsules. The first and third row show input test images. The second and fourth row show the result of activation maximizing for primary capsule 21 around the mouth (top) and capsule 5 around the brows (bottom) respectively. The results show that a single capsule is able to model multiple deformations. The numbers show cosine similarity before and after activation maximization - green indicating increase in orthogonality, and red indicating decrease.

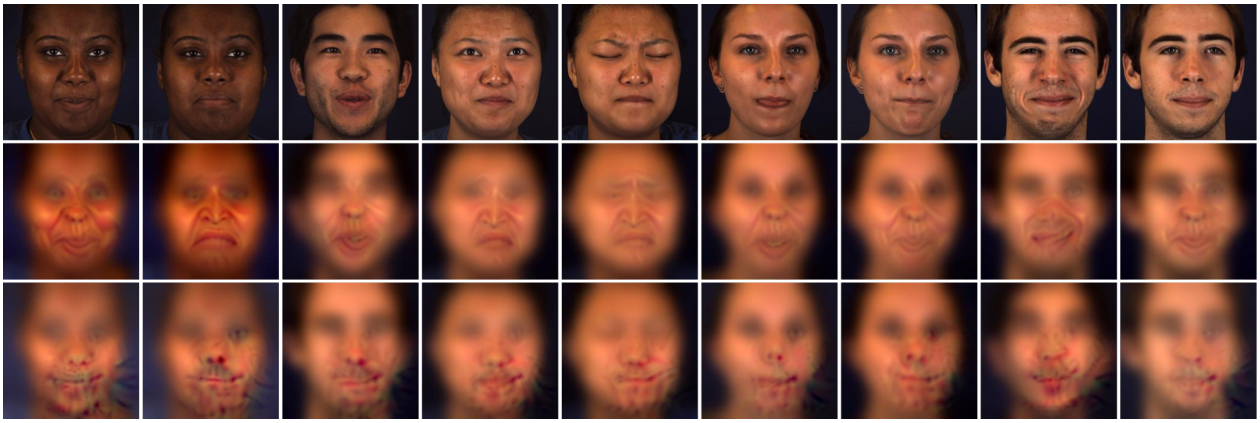

Figure 5. Comparing convolution and capsule units. The top row shows the input images. The second row shows the result of performing gradient ascent visualization on capsule 21 on the mouth area. The bottom row shows the results on the same input images with convolution unit 498 from a finetuned VGG-Face network. While a single capsule can have high magnitude with many different types of mouth positions, a single convolution capsule is only able to maximize activation for a thin mouth with an upturned corner.

it can achieve high magnitude with different types of part poses. To test this we perform gradient ascent on the primary capsule activation at the relevant part in a spatial map, and see if the facial part's deformation - e.g. how puckered the lips are - at that location is exaggerated and becomes more pronounced. This helps us verify what the capsule is 'seeing' in the input image. By repeating this process for different types of input part poses, we are able to test if the same capsule can identify different kind of part poses.

Second, we need to factor out noisy activations caused by network error. We therefore use test images that have 100% accurate action unit detection, and have the highest confidence (or class capsule activation magnitude) per test subject and per action unit. This also allows us to get variation in capsule direction naturally.

Third, we need to factor out changes in activation direc-

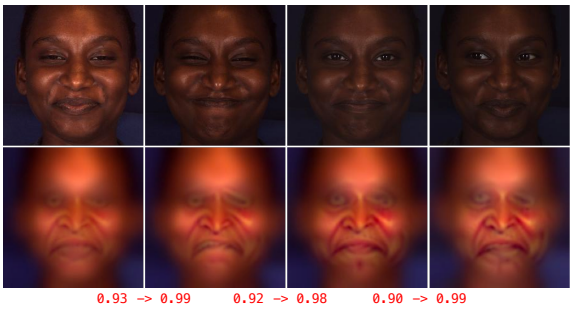

Figure 6. Activation maximization on the mouth area for capsule 5. While this capsule is able to model changes in brow position well (Figure 4) it is 'sticky' towards a downturned mouth position. More similar mouth deformations after gradient ascent also bring activation directions across images closer.

tion that may be caused by factors other than local part deformations - such as changes in skin tone, or facial features. We therefore compare activations of a primary capsule on the same part of the face (or activation map) for the same person, but with different expressions. Though not perfect, this allows us to factor out variations in primary capsule activation direction caused by changes in person identity. At the same time, we compare primary capsules activations across different test subjects in order to ensure that they are able to model deformations in an identity agnostic manner.

In Figure 4 we show qualitative evidence supporting our hypothesis. By performing gradient ascent on the mouth part of different images for primary capsule 21's activation (top), we are able to see very different mouth deformations become apparent in the input image. Similarly, for capsule 5 we see two distinct brow changes become apparent (bottom). At the same time, these changes are apparent across different test subjects.

To understand how primary capsules direction is changed by part deformation, we take the cosine similarity between primary capsule activation at a spatial location for two images of the same person with a different part deformation before and after modifying the input images with gradient ascent. We find that capsule activations becomes more orthogonal (closer to zero) when part deformations after gradient ascent are visually different from each other (Fig. 4), and less orthogonal when part deformations after gradient ascent are less different from each other (Fig. 6).

For comparison, we perform similar activation maximization visualization on convolution units. We finetune a VGG-Face network for action unit detection using the same training data. We use the activations of the last convolution layer after ReLU and max-pooling for this purpose since it allows us to have comparable receptive window size as the primary capsules, and lets us ignore convolution units that are turned off by ReLU and therefore irrelevant to the final prediction. We forward the same selected test images through the network and record the convolution units with the highest activation at the specific spatial locations in the activation map for each action unit. We then perform gradient ascent on the activations of these particular convolution units at selected spatial locations and show the resulting images in Figure 5. We find that unlike capsules, individual convolution units are not able to model dramatically different deformations, and either do not activate (post ReLU activation is 0) or exaggerate a fixed type of attribute exclusively. This is not surprising given that convolution units have scalar outputs, while capsules have vector outputs that are capable of representing more complex information. As a result, while a single primary capsule is able to have high activations with downturned, puckered, open, and smiling - among others - mouths, the convolution unit maximizes activation with thin mouths with a sharp right upturn across a range of different type of input images.

Lastly, we find that primary capsules may be more sensitive to some facial parts than others. In Figure 6 we show activation maximization results on the mouth area for different mouth deformations on the same person for capsule 5. While primary capsule 5 is able to model brow deformations well in Figure 4, the same capsule is not able to 'see' different mouth deformations, and seems to identify only down turned mouths regardless of the input.

5. Conclusions

In this paper, we explore whether capsule networks are able to model local part deformations in faces. We tested this hypothesis by using capsules for action unit detection, and found that capsules are indeed able to model action unit activations. Our results demonstrated state-of-the-art results on action unit detection on two widely-used datasets. While previous work has shown that capsules are able to model global deformations, we showed that capsules can also capture local deformations. This indicates that capsule networks may also be useful for other tasks where parts of an object need to be modeled well - such as fine-grained classification, or human pose estimation and tracking.

In the future, we plan to work on automatic animal facial expression understanding. For a setting such as animal facial expression understanding where data is scarce and difficult to both collect and annotate, it becomes critical to work with models such as capsule networks that are able to extract rich feature representations with fewer overall parameters. In addition, the added ease with which capsule properties can be visualized makes capsule networks an appropriate model working with limited, possibly noisily annotated data.

References

[1] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568–573. IEEE, 2005. 1, 2

[2] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 4

[3] W.-S. Chu, F. De la Torre, and J. F. Cohn. Selective transfer machine for personalized facial action unit detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3515–3522. IEEE, 2013. 2

[4] 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. 4

[5] C. A. Corneanu, M. Madadi, and S. Escalera. Deep structure inference network for facial action unit recognition. *ECCV*, 2018. 2, 5

[6] H. Ding, S. K. Zhou, and R. Chellappa. Facenet2expnet: Regularizing a deep face recognition net for expression recognition. *Automatic Face and Gesture Recognition (FG)*, 2017. 3, 5

[7] 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. 2

[8] X. Feng, M. Pietikäinen, and A. Hadid. Facial expression recognition based on local binary patterns. *Pattern Recognition and Image Analysis*, 17(4):592–598, 2007. 2

[9] 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. 2

[10] 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

[11] S. Han, Z. Meng, A.-S. Khan, and Y. Tong. Incremental boosting convolutional neural network for facial action unit recognition. In *Advances in Neural Information Processing Systems*, pages 109–117, 2016. 2

[12] S. Han, Z. Meng, Z. Li, J. O'Reilly, J. Cai, X. Wang, and Y. Tong. Optimizing filter size in convolutional neural networks for facial action unit recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 5

[13] S. Hinton, Geoffrey E snd Sabour and N. Frosst. Matrix capsules with em routing. In *ICLR*, 2018. 1, 2

[14] 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. 2, 4

[15] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim. Joint fine-tuning in deep neural networks for facial expression recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2983–2991, 2015. 3

[16] P. Khorrami, T. Paine, and T. Huang. Do deep neural networks learn facial action units when doing expression recognition? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 19–27, 2015. 3, 5

[17] Y. Kim, B. Yoo, Y. Kwak, C. Choi, and J. Kim. Deep generative-contrastive networks for facial expression recognition. *arXiv preprint arXiv:1703.07140*, 2017. 3, 5

[18] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *IEEE transactions on pattern analysis and machine intelligence*, 32(11):1940–1954, 2010. 2

[19] W. Li, F. Abtahi, and Z. Zhu. Action unit detection with region adaptation, multi-labeling learning and optimal temporal fusing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 4, 5

[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. 2

[21] 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, 5

[22] M. Liu, S. Li, S. Shan, and X. Chen. Au-inspired deep networks for facial expression feature learning. *Neurocomputing*, 159:126–136, 2015. 3, 5

[23] 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. 3

[24] 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. 5

[25] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013. 4

[26] 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. 3

[27] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. 6

[28] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. 3

[29] G. Peng and S. Wang. Weakly supervised facial action unit recognition through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[30] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017. 1, 2, 3

[31] Z. Shao, Z. Liu, J. Cai, and L. Ma. Deep adaptive attention for joint facial action unit detection and face alignment. *ECCV*, 2018. 2, 5

[32] 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

[33] 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. 2

[34] M. F. Valstar, T. Almaev, J. M. Girard, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. F. Cohn. Fera 2015-second facial expression recognition and analysis challenge. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 6, pages 1–8. IEEE, 2015. 2

[35] Z. Wang, Y. Li, S. Wang, and Q. Ji. Capturing global semantic relationships for facial action unit recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3304–3311, 2013. 2

[36] H. Yang, U. Ciftci, and L. Yin. Facial expression recognition by de-expression residue learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 5

[37] J. Zeng, W.-S. Chu, F. De la Torre, J. F. Cohn, and Z. Xiong. Confidence preserving machine for facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3622–3630, 2015. 2, 4

[38] X. Zhang and M. H. Mahoor. Simultaneous detection of multiple facial action units via hierarchical task structure learning. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1863–1868. IEEE, 2014. 2

[39] 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. 4

[40] Y. Zhang, W. Dong, B.-G. Hu, and Q. Ji. Weakly-supervised deep convolutional neural network learning for facial action unit intensity estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[41] 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. 2, 4

[42] 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. 1, 2, 4, 5

[43] X. Zhao, X. Liang, L. Liu, T. Li, Y. Han, N. Vasconcelos, and S. Yan. Peak-piloted deep network for facial expression recognition. In *European Conference on Computer Vision*, pages 425–442. Springer, 2016. 3

[44] 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

[45] Y. Zhu, S. Wang, L. Yue, and Q. Ji. Multiple-facial action unit recognition by shared feature learning and semantic relation modeling. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1663–1668. IEEE, 2014. 2