Deep Learning the Dynamic Appearance and Shape of Facial Action Units

Shashank Jaiswal
The University of Nottingham
psxsj3@nottingham.ac.uk

Michel Valstar
The University of Nottingham
michel.valstar@nottingham.ac.uk

Abstract

Spontaneous facial expression recognition under uncontrolled conditions is a hard task. It depends on multiple factors including shape, appearance and dynamics of the facial features, all of which are adversely affected by environmental noise and low intensity signals typical of such conditions. In this work, we present a novel approach to Facial Action Unit detection using a combination of Convolutional and Bi-directional Long Short-Term Memory Neural Networks (CNN-BLSTM), which jointly learns shape, appearance and dynamics in a deep learning manner. In addition, we introduce a novel way to encode shape features using binary image masks computed from the locations of facial landmarks. We show that the combination of dynamic CNN features and Bi-directional Long Short-Term Memory excels at modelling the temporal information. We thoroughly evaluate the contributions of each component in our system and show that it achieves state-of-the-art performance on the FERA-2015 Challenge dataset.

1. Introduction

Automatic facial expression recognition has attracted considerable attention in the past few years [6, 18, 20, 23] due to their potential application in the field of psychology, mental health, human-computer interaction, etc. The Facial Action Coding System (FACS) developed by Ekman and Friesen [4], provides a systematic and objective way to study any kind of facial expression, by representing them as a combination of individual facial muscle actions known as Action Units (AU). Automatic detection of facial Action Units is a challenging problem primarily due to the high degree of variation in the visual appearance of human faces (caused by person specific attributes, multiple poses, etc.), low intensity activation of spontaneous expressions, non-additive effects of co-occurring AUs and the scarcity of training data.

Recently, deep learning algorithms have shown significant performance improvements for object detection tasks [13, 7]. The recent success of deep learning algorithms has been attributed to three main factors: (a) Efficient methods for training deep artificial neural networks, (b) Availability of high performance computational hardware e.g. GPUs and (c) Availability of large amounts of labeled training data. Although deep learning algorithms have been shown to produce state of the art performance on object recognition tasks, there has been considerably less work on using deep learning techniques in action recognition, facial expression recognition, and in particular facial AU recognition. With the increasing availability of large databases for AU recognition [16, 23, 29], it would be interesting to see if deep learning algorithms can give a similar leap in performance in the field of facial expression/AU recognition.

Traditional AU recognition algorithms have used hand-crafted appearance features (e.g. Gabor, HoG, LBP) and/or shape/geometric features computed from the locations of facial landmarks. Since these hand-crafted features are not tuned to a specific task, they limit the performance of the classifier learnt on these features. Deep learning techniques on the other hand allow a multistage approach in which the features are learnt directly from the pixel values in combination with the classifier. Therefore, in addition to providing an algorithm which can be trained directly from pixels to labels, the features learnt in the intermediate stages are designed specifically for the target task.

Good automatic facial AU recognition involves the analysis of three facial features: face shape, appearance and dynamics. Each of these can be considered a source of complimentary information for the modelling of facial action unit detectors. We hypothesise that learning all the three features jointly can produce highly accurate models for facial AU recognition. However, the performance of the models depends a lot on how one fuses these. In this work, we present a deep learning based framework for facial AU detection in images. In particular we use Convolutional Neural Networks (CNNs) to model the appearance, shape and the dynamics of facial regions for AU detection. In contrast to previous approaches, our system learns all the key features (appearance, shape and dynamics) jointly in a deep CNN. We also shy from a fully naïve deep learning approach, which would have us learn directly from pixel data.

Instead we make full use of past progress in face and facial point detection to guide our CNN.

We introduce a novel way to encode shape of facial parts by computing binary image masks using the locations of facial landmarks. This enables us to learn the relevant shape features instead of using hand-crafted geometric features. We also use a novel method to model the temporal information using a combination of time-windowed CNN and Bi-directional Long Short-Term Memory (BLSTM) [10, 8]. BLSTM is a recurrent neural network architecture which is capable of storing information over extended time intervals. It consists of memory enabled blocks to which access is controlled by multiplicative gates. BLSTMs have previously been used for continuous emotion prediction from audio-visual data [25, 17] and have shown promising performance. It has also been used in combination with CNNs for activity recognition and predicting visual description [3]. To the best of our knowledge, our work is the first to use CNN in combination with BLTSM for facial expression recognition. We show that both BLSTMs and dynamic features extracted from a time-windowed CNN improve recognition accuracy, independently from each other, and with best result achieved when both are used. Our final system outperforms the winners of the recent FERA 2015 challenge [23] by a margin of 10%.

In summary, our main contributions are:

- A deep CNN based framework for jointly learning dynamic appearance and shape features for facial Action Unit detection.
- A novel way to encode shape features in a CNN, by using binary image masks computed from the locations of automatically detected facial landmarks.
- Learning temporal information for facial AU detection, using a combination of time-windowed CNN and BLSTM.
- Achieving new state-of-the-art performance on the FERA-2015 Challenge dataset.

2. Previous work

Fasel [5] was one of the first to use CNNs for the task of facial expression recognition. He used a 6 layer CNN architecture (2 convolutional layers, 2 sub-sampling layers and 2 fully connected layers) for classifying 7 facial expressions (6 basic emotions + 1 neutral). He experimented with 2 versions of his architecture. In the first version, the filter size in the first convolutional layer was fixed to 5x5 pixels, i.e. the features at the first layer were extracted at a single scale. In the second version, the features at the first layer were extracted at multiple scales, using filters of 3 different sizes (5x5, 7x7 and 9x9 pixels). This CNN consisted of 3 different data streams corresponding to the three scales which are connected to each other only at the fully connected layer of the network.

Gudi et al.[9] used a deep CNN consisting of 3 convolutional layers, 1 sub-sampling layer and 1 fully connected layer to predict the occurrence and intensity of Facial AUs. A similar architecture was used by Tang [21], but replacing the softmax objective function with L2-SVM.

Liu et al.[15], used 3-dimensional CNN for dynamic learning of facial expressions in a video. They proposed a CNN architecture which can jointly localise certain dynamic parts of a face (action parts) and encode them for facial expression classification. In their CNN architecture, the resulting feature maps from the first convolutional layer were convolved with a bank of class (of facial expression) specific part filters to compute part detection maps. These part detection maps and a set of deformation maps are summed together (with learned weights), to enforce spatial constraints on the part detections. The resulting feature maps are passed through a partially connected layer before obtaining the decision values for each basic emotion.

Kahaou et al.[12] combine face models learnt using a deep CNN architecture with various other models for facial expression classification. They combined a CNN face model with a bag of words model for the mouth region, a deep belief network for audio information and deep autoencoder for modelling spatio-temporal information. A weighted average of the predictions from all the models was used to classify the emotion expressed.

Jung et al.[11] used a deep CNN to learn temporal appearance features for recognising facial expressions. Additionally, they also employed a deep Neural network to learn temporal geometric features from detected facial landmarks. Both the networks were learned independently to predict facial expression. The output decision values from each of the networks were combined (linear combination) to compute the final score for any example face image.

Liu et al.[14] learn receptor fields which simulates AU combinations on convolutional feature maps. From each receptor field, they learn features using multilayer Restricted Boltzmann Machines (RBM), which are concatenated to learn a linear Support Vector Machine (SVM).

Another notable work in this field is Rifai et al.[19], who propose a method to disentangle features which are discriminative for facial expressions, from all other features. They use features from a CNN (1 layered) whose filters were pre-trained using Contractive Auto-Encoders (CAE). The feature output from the CNN serves as input for a semi-supervised feature learning framework called Contractive Discriminative Analysis (CDA). CDA is a semi-supervised version of CAE, in which the input is mapped onto 2 distinct blocks of features. One of the blocks learns features which are discriminative for facial expressions, while the other block learns all other features. Both the blocks are

learnt so as to jointly reconstruct the input. The discriminative features from the first block are then used to train a SVM for facial expression classification.

Most of the above methods (e.g. [5, 9, 15, 14], do not utilise all the key facial features i.e. face shape, appearance and dynamics. Those which do utilise all three of them (e.g. [11]), do not learn them jointly. We believe that in order to find the most optimum combination of these features, it is necessary to model them jointly. Also, almost all the above CNN based approaches use a fixed time window to learn the temporal information. This limits the access to temporal information within a specific time window only. On the other hand, different facial AUs can occur over different time scales. For e.g., a smile (AU12) can last much longer compared to a blink (AU45) ,which occurs only for a period. Even within a specific AU, the variation in its duration of occurrence can be quite large. In contrast, our method uses all key features (shape, appearance and dynamics) and attempts to jointly learn them in a single CNN. Additionally, we also overcome the problem of a fixed time window in CNN by employing BLSTM for learning long term temporal dependencies.

3. Methodology

Our system uses small rectangular image regions and corresponding binary image masks to learn the relevant appearance and shape features respectively. We use a sequence of consecutive images in order to model the dynamics. A transformed sequence of image regions and binary image masks are used as input to train a CNN. The dynamic features learnt from this CNN are further used for training a BLSTM neural network. The output from this BLSTM neural network serves as the final decision value for the occurrence of an AU.

3.1. Image Regions

Learning facial action unit models can be a difficult task because only a small part of the face is responsible for the occurrence of a specific Action Unit. Automatically learning the facial regions which are responsible for a particular Action Unit is as difficult task due to the high dimensionality of the input image data and the limited amount of training examples available. To solve this problem, the relevant image image region for particular Action Unit is pre-selected according to domain knowledge.

In order to define image regions, the face images are first pre-processed by automatically tracking facial landmarks [27, 22] and aligning the face images with a reference shape. The alignment is done using a Procrustes transform of the shape defined by facial landmarks on the eyes corners and the nose. The locations of these facial landmarks are invariant to facial expressions and hence are suitable for the purpose of face alignment. A small set of facial points in the aligned face images are used to define a rectangular image region. The facial points that define these regions are selected by an expert according to the target AU. The mean of the selected facial points is taken as the center around which a rectangular image region of a fixed width w and height h, is defined. The rectangular image region is cropped from the original image and is denoted as f (see Fig. 1).

3.2. Binary masks

In order to obtain better alignment between different training examples and to automatically encode the shape of different parts of a face, we compute a binary mask b_i, each corresponding to an image region f_i. To compute the binary masks, the facial points selected for defining the image regions (see Section 3.1) are joined together in a predetermined order to form a set of polygons. The binary mask image b_i corresponding to the image region f_i is computed by setting all the pixel values of f_i which lie inside a polygon to 1 and all the pixel values which lie outside a polygon to 0 (see Fig. 1).

3.3. Dynamic encoding

Temporal information can provide vital features for recognition of any facial action unit [24, 1]. We encode temporal information in our features by extracting image regions $\{f_{t-n}, ..., f_{t+n}\}$ and their corresponding binary masks images $\{b_{t-n}, ..., b_{t+n}\}$ from a sequence of $2n+1$ consecutive frames of a video centered at the current frame t. The resulting sequence of image regions are transformed to a sequence $A = \{A_i\}$ where

$$A_i = \begin{cases} f_i, & \text{if } i = t \\ f_i - f_t, & \text{otherwise} \end{cases} \quad (1)$$

Similarly, the sequence of binary mask images are transformed to a sequence $S = \{S_i\}$ where

$$S_i = \begin{cases} b_i, & \text{if } i = t \\ b_i - b_t, & \text{otherwise} \end{cases} \quad (2)$$

The resulting image sequences A and S are used as input to a deep Convolutional Neural Network (CNN). This is in contrast to the previous approaches [15, 11] which directly use the images within a time window around the current frame. Our method of transforming the image sequences by taking difference from the current frame, makes it easier to learn the dynamics in a CNN framework.

3.4. CNN architecture

The CNN architecture that we use in this work is shown in Fig. 1. It has 2 input streams, one for the sequence of image regions A and another for the sequence of binary shape masks S. In both streams, the inputs are first passed through

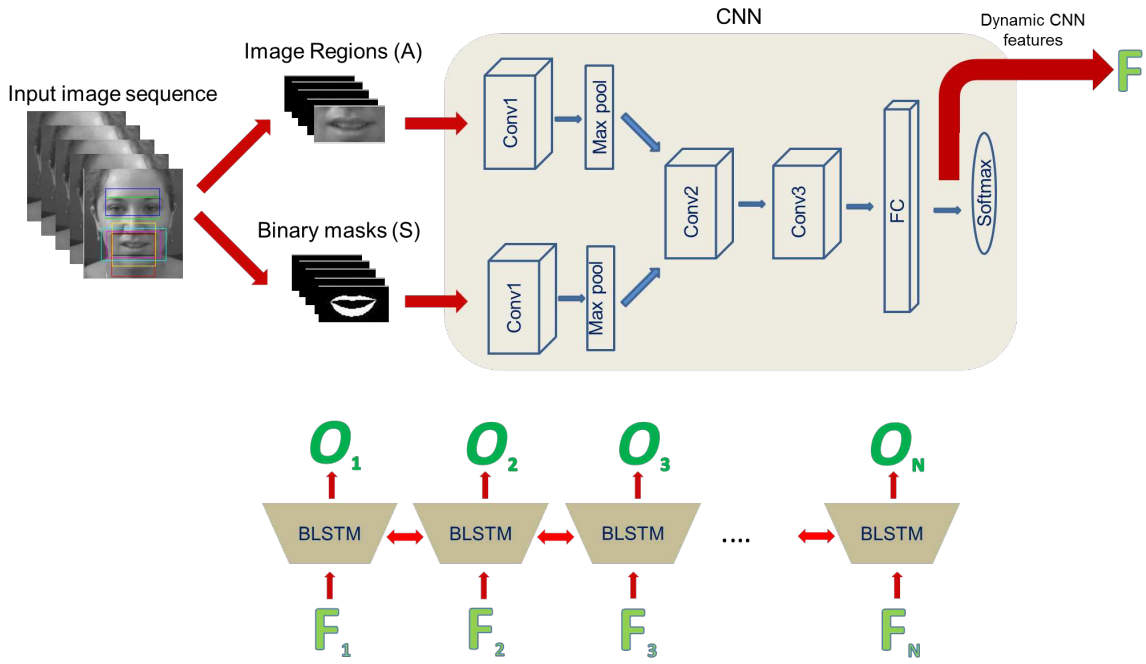

Figure 1. A graphical overview of our training pipeline: The colored rectangles in the input image sequence shows the different image regions selected for different AUs. Here we show the extraction of image regions (A) and binary masks (S) for AU 12. These are used as input to the train the CNN. Features extracted from the trained CNN (at the fully connected layer) denoted here as F, are used to train a BLSTM network to get final output prediction values O.

a convolutional layer ($Conv1$) which consists of 32 filters of size 5x5x(2n+1). This convolutional layer is followed by a max-pooling layer of size 3x3x1.

The outputs from both streams are merged after max-pooling into a single stream. This merged stream passes through 2 more convolutional layers and 1 fully connected layer. The first convolutional layer in the merged stream ($Conv2$) consists of 64 filters of size 5x5x64. The second convolutional layer in this stream ($Conv3$) consists of 128 filters of size 4x4x64. The fully connected layer (FC) has 3072 units and it uses dropout with a probability of 0.2. The output layer consists of 2 units, one for the positive class and another for negative class. In this network, we use Rectified Linear Unit (ReLU) as the activation function. We experimented by increasing the dropout probability parameter and adding max-pooling layers after the $Conv2$ and $Conv3$ layers, but observed no significant change in performance.

3.5. CNN training

The sequence of image regions A and the sequence of binary shape masks S for each training example are fed as input to the CNN described in section 3.4. The $2n+1$ images of both sequences A and S are stacked along the temporal dimension and fed into two input streams of the CNN as $2n+1$ channel inputs. Since the size of the $Conv1$ filters in the temporal direction is equal to size of sequence A and S, the network is fully connected in the temporal direction at the first convolutional layer and only one feature map is computed per filter at $Conv1$ for each training example. The network is trained with logarithmic loss function using mini-batch gradient descent method. The training data is normalised so as to have a zero mean and one standard deviation for each pixel across all training examples.

3.6. Training BLSTM with CNN features

The dynamic encoding described in section 3.3, enables us to learn only very short term ($2n+1$ frame window) temporal information. In order to learn temporal features over longer and variable time windows, we use a recurrent neural network architecture known as Bi-directional Long Short-Term Memory [8]. We train the CNNs for each AU as described in the previous sections and extracted the output of the CNN after the fully connected layer (with 3072 units). This gives us a 3072 dimensional CNN feature vector for each training example. We used these CNN features to train a BLSTM network with a single hidden layer of 300 units. The output from this BLSTM network serves as the final decision value for the occurrence of an AU.

4. Evaluation

Databases: We evaluated our proposed method on the FERA-2015 Challenge dataset [23]. The FERA 2015 challenge dataset consists of 2 separate databases: SEMAINE and BP4D. Both databases consist of videos in which the facial Action Units of the subjects are labeled. The SEMAINE database was recorded to study social signals that occur when people interact with virtual humans. It consists of videos in which users are interacting with emotionally stereotyped characters. A total of 6 Facial Action Units are labeled for each frame in the videos. The dataset is divided into a fixed training, development and test set. The partitioning is subject independent, i.e. the subjects present in the training set are not present in the test set and vice versa. The training partition consists of 16 sessions, the development partition has 15 sessions, and the test partition has 12 sessions. There are a total of approximately 48,000 images in the training partition, 45,000 in the development and 37,695 in the test partition.

The BP4D dataset consists of recorded videos in which the subjects are responding to emotion elicitation tasks. Like SEMAINE, BP4D dataset is also divided into a fixed set of training, development and test data. The training set consists of 21 subjects while the development and the test set consists of 20 subjects each. There are 8 sessions for each subject. In total, the training partition contains 75,586 images, the development contains 71,261 images and the test contains 75,726 images. Each of these images are annotated with 11 Action Units. For 6 of these Action Units, only occurrence labels are available. For the other 5 Action Units occurrence as well as intensity levels are available. It should be noted that in our experiments, we used the intensities (wherever available) instead of occurrence labels to train our models by switching to Mean-squared error as loss function rather than log-loss.

We chose these 2 databases for our experimental evaluations because, firstly, it contains large number of annotated images which benefits deep learning algorithms. Secondly, since these databases are divided into a fixed training and test set, they provide a good platform for a fair evaluation and benchmarking of different AU detection algorithms.

Experiments: A number of experiments were carried out to evaluate the effect of various aspects of our method and to compare the performance of our approach to that of other existing approaches. We conducted 4 sets of experiments: One for evaluating the effectiveness of various features of our CNN training method, another for exploring the effect of CNN architecture parameters and a third for evaluating the effect of adding BLSTM. The final set of experiments serves to compare the performance of our models with other existing methods reporting on the same dataset.

For evaluating the effectiveness of various aspects of our CNN training method, we performed a number of experiments on the SEMAINE dataset. We trained a number of baseline models each having some features of our proposed approach. Our first baseline model consists of a CNN similar to the CNN in our approach except that the input to this CNN is the full image of a face (aligned) defined by the face bounding box and the temporal window parameter $n = 0$. There are no image regions (defined by facial landmarks) or binary masks as input to this CNN. Since $n = 0$ for this model, it does not use any temporal information. We denote this baseline method as $\text{CF}_{n=0}$. Our second baseline method ($\text{CF}_{n=2}$) uses the same CNN but with temporal window parameter $n = 2$. Hence this baseline uses temporal information within a time window of $2n + 1 = 5$ frames. Our third baseline ($\text{CR}_{n=2}$) uses the same CNN architecture but uses image regions (see section 3.1) as input and the temporal window parameter $n = 2$ for this baseline. Our fourth and final method ($\text{CRM}_{n=2}$) is our proposed approach which takes as input the image regions and binary shape masks in 2 different streams of the CNN. The temporal window size $n = 2$ for this method as well. It should be noted that we do not use BLSTM training for this set of experiments and the performances are evaluated based on CNN output only.

For this set of experiments, we used the 2 Alternative Forced Choice (2AFC) scores as performance measure. The 2AFC score is a good approximation of the area under the receiver operator characteristic curve (AUC). It is defined as follows:

$$2AFC(\hat{Y}) = \sum_{i=0}^{n}\sum_{j=0}^{p}\sigma(P_j, N_i)\frac{1}{n \times p}, \tag{3}$$

$$\sigma(X, Y) = \begin{cases} 1, & \text{if } X > Y \\ 0.5, & \text{if } X == Y \\ 0, & \text{if } X < Y \end{cases}$$

where $\hat{Y}$ is a vector of output decision values from a classifier, P and N are subsets of $\hat{Y}$ corresponding to all positive and negative instances, respectively. n is the total number of true negatives and p is the total number of true positives.

Fig. 2 shows a comparison of the performance of our method ($\text{CRM}_{n=2}$) and the other baseline methods, on the SEMAINE dataset. In this plot we can observe that the performance of $\text{CF}_{n=2}$ is higher than that of $\text{CF}_{n=0}$ which shows that our method of encoding the temporal information works leading to a significantly improved performance. The difference in the performance of $\text{CR}_{n=2}$ and $\text{CF}_{n=2}$ also shows that using image regions as input instead of the entire face image, improves the performance considerably. Similarly, the improved performance of our method $\text{CRM}_{n=2}$ over $\text{CR}_{n=2}$ shows that our method of encoding shape using binary masks, also results in a significant improvement in performance. Our method $\text{CRM}_{n=2}$ which in-

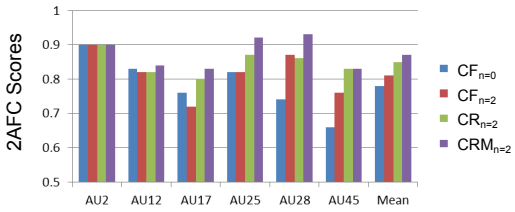

Figure 2. 2AFC scores on the SEMAINE test set, showing a comparison of the performance of the baseline methods and our proposed method CRM$_{n=2}$.

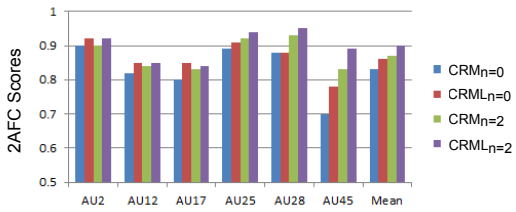

Figure 3. Performance (2AFC scores) comparison on the SEMAINE test set when using BLSTMs with CNNs.

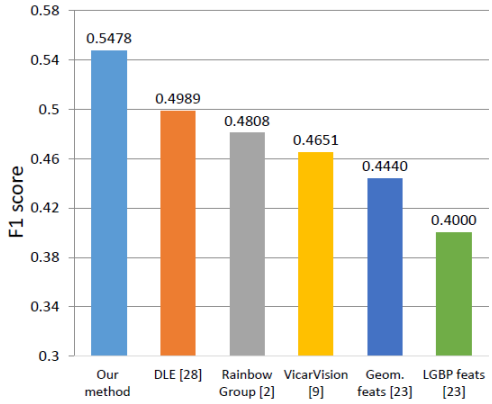

Figure 4. Weighted average performance on BP4D and SEMAINE for AU occurrence.

corporates all our proposed features achieves the best overall performance among all baseline models.

In our second set of experiments, we experimented with the architectural parameters of the CNN. More specifically, we evaluated the effect of adding more max-pooling (mp) layers (adding one after each convolutional layer) and varying the dropout factor (dp) at the fully connected layer. We took our proposed approach CRM$_{n=2}$ and evaluated its performance under different combination of the parameters (mp,dp). For this set of experiments we again used the SEMAINE dataset for performance evaluation and 2AFC as the performance measure. We observed that the average performance on the SEMAINE test set, does not changes significantly on adding max-pooling layers after the second and third covolutional layers. Increasing the dropout probability factor from 0.2 to 0.5 also did not make any significant change in the average performance. Hence we can conclude that our proposed method is more or less invariant to these architecture parameters, at least for this amount of data.

For evaluating the effect of adding BLSTM to our training pipeline, we conducted a series of experiments on the SEMAINE database. We computed the performance of our approach with BLSTM (CRML$_{n=2}$) along with 3 other methods. Our first baseline is the CRM$_{n=0}$, which does not use any temporal window and hence does not use dynamics. Our second baseline is CRML$_{n=0}$ which does not use any temporal window during CNN training but employs BLSTM after CNN training. Our third baseline is CRM$_{n=2}$ which uses a $2n + 1 = 5$ frame temporal window during CNN training, but does not use BLSTM.

Fig. 3 shows the relative performance of our final approach CRML$_{n=2}$ and the other 3 baseline methods. We observe that the performance of CRML$_{n=0}$ is higher than that of CRM$_{n=0}$, indicating that BLSTM is able to learn the dynamics resulting in an improved performance even without using a temporal window during CNN training. The performance of CRM$_{n=2}$ over that of CRM$_{n=0}$ indicates the extent to which, using a temporal window of 5 frames while training the CNN, can improve the performance even without using BLSTMs. However, the best performance is achieved with CRML$_{n=2}$, which uses a 5 frame temporal window during CNN training and also employs BLSTM for training with features extracted from CNN. This indicates that BLSTMs perform better with dynamic features (extracted using a fixed temporal window) compared to non-dynamic features (extracted using a single image).

In our final set of experiments, we compare the performance of our method (CRML$_{n=2}$) with 3 other existing methods on the SEMAINE and BP4D dataset. For this set of experiments, we used the F1 scores as the performance measure in order to directly compare with the performance reported in the literature. We compared the performance of our method with the Local Gabor Binary Pattern (LGBP, [26]) and SVM based approach described in [23]. Another method that we compare against is a geometric feature based approach which uses a deep neural network (GDNN). For computing the performance of this method we trained a deep neural network with 4 hidden layers (all fully connected). The input to this network were the locations of 49 facial landmarks within a time window of 5 frames. We also compared our method with the multi-label Discriminant Laplacian Embedding (DLE) approach proposed by Yüce et al.[28] (FERA-2015 Challenge winner).

Table 1 and 2 shows the performance comparison on the SEMAINE and BP4D dataset respectively. In table 1, we

AU	LGBP [23]	GDNN	DLE [28]	CRML$_{n=2}$
2	0.75	0.67	0.66	0.80
12	0.52	0.63	0.76	0.74
17	0.07	0.14	0.25	0.32
25	0.40	0.77	0.61	0.85
28	0.01	0.31	0.26	0.33
45	0.21	0.55	0.35	0.57
Mean	0.33	0.51	0.48	0.60

Table 1. Performance (F1 scores) comparison on SEMAINE test set.

AU	LGBP [23]	GDNN	DLE [28]	CRML$_{n=2}$
1	0.18	0.33	0.25	0.28
2	0.16	0.25	0.17	0.28
4	0.22	0.21	0.28	0.34
6	0.67	0.64	0.73	0.70
7	0.75	0.79	0.78	0.78
10	0.80	0.80	0.80	0.81
12	0.79	0.78	0.78	0.78
14	0.67	0.68	0.62	0.75
15	0.14	0.19	0.35	0.20
17	0.24	0.28	0.38	0.36
23	0.24	0.33	0.44	0.41
Mean	0.44	0.48	0.51	0.52

Table 2. Performance (F1 scores) comparison on BP4D Test set.

can see that the performance from our approach is significantly higher on the SEMAINE dataset, as compared to other approaches. Similarly, we outperform the other 3 approaches, on the BP4D dataset as well (see table 2). Fig. 4 shows the weighted average performance on SEMAINE and BP4D dataset. In this Fig. we compare the performance of our method with all other approaches ([28],[2],[9], [23]) of the partcipants of FERA-2015 Challenge. In this Fig. we can see that our method significantly outperforms other approaches on the FERA-2015 Challenge dataset.

5. Conclusions

We presented a novel CNN-BLSTM based approach which learns the dynamic appearance and shape of facial regions for Action Unit detection. The appearance and shape are learnt through local image regions and corresponding binary masks respectively. The dynamics are learnt through a combination of dynamic features (extracted from a time-windowed CNN) and BLSTM. We show that each component of our system contributes towards an improvement in performance and achieves new state-of-the-art performance on the FERA-2015 Challenge databases.

6. Acknowledgments

The work of Valstar and Jaiswal is supported by MindTech Healthcare Technology Co-operative. The work of Valstar is also funded by European Union Horizon 2020 research and innovation programme under grant agreement No 645378. We are also grateful for access to the University of Nottingham High Performance Computing Facility.

References

[1] T. R. Almaev and M. F. Valstar. Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 356–361. IEEE, 2013. 3

[2] T. Baltrusaitis, M. Mahmoud, and P. Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *Facial Expression Recognition and Analysis Challenge, in conjunction with IEEE Int'l Conf. on Face and Gesture Recognition*, 2015. 6

[3] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014. 2

[4] P. Ekman, W. Friesen, and J. Hager. *Facial Action Coding System (FACS): Manual*. A Human Face, Salt Lake City (USA), 2002. 1

[5] B. Fasel. Head-pose invariant facial expression recognition using convolutional neural networks. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 529–534, 2002. 2, 3

[6] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *PATTERN RECOGNITION*, 36(1):259–275, 1999. 1

[7] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587, 2014. 1

[8] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, pages 5–6, 2005. 2, 4

[9] A. Gudi, H. E. Tasli, T. M. den Uyl, and A. Maroulis. Deep learning based facs action unit occurrence and intensity estimation. In *Facial Expression Recognition and Analysis Challenge, in conjunction with IEEE Int'l Conf. on Face and Gesture Recognition*, 2015. 2, 3, 6

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. 2

[11] H. Jung, S. Lee, S. Park, I. Lee, C. Ahn, and J. Kim. Deep temporal appearance-geometry network for facial expression recognition. http://arxiv.org/abs/1503.01532. 2, 3

[12] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, c. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari, M. Mirza, S. Jean, P.-L. Carrier, Y. Dauphin, N. Boulanger-Lewandowski, A. Aggarwal, J. Zumer, P. Lamblin, J.-P. Raymond, G. Desjardins, R. Pascanu, D. Warde-Farley, A. Torabi, A. Sharma, E. Bengio, M. Côté, K. R. Konda, and Z. Wu. Combining modality specific deep neural networks for emotion recognition in video. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 543–550, New York, NY, USA, 2013. ACM. 2

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In

F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 1

[14] 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. IEEE*, pages 1–6. IEEE, 2013. 2, 3

[15] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen. Deeply learning deformable facial action parts model for dynamic expression analysis. In D. Cremers, I. Reid, H. Saito, and M.-H. Yang, editors, *Computer Vision – ACCV 2014*, volume 9006 of *Lecture Notes in Computer Science*, pages 143–157. Springer International Publishing, 2015. 2, 3

[16] S. Mavadati, M. Mahoor, K. Bartlett, P. Trinh, and J. Cohn. Disfa: A spontaneous facial action intensity database. *Affective Computing, IEEE Transactions on*, 4(2):151–160, April 2013. 1

[17] M. A. Nicolaou, H. Gunes, and M. Pantic. Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space. *IEEE Transactions on Affective Computing*, 2(2):92–105, 2011. 2

[18] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1424–1445, December 2000. 1

[19] S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza. Disentangling factors of variation for facial expression recognition. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ?ECCV 2012*, volume 7577 of *Lecture Notes in Computer Science*, pages 808–822. Springer Berlin Heidelberg, 2012. 2

[20] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin. Static and dynamic 3d facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683 – 697, 2012. 3D Facial Behaviour Analysis and Understanding. 1

[21] Y. Tang. Deep learning using linear support vector machines. In *Workshop on Challenges in Representation Learning, ICML*, 2013. 2

[22] G. Tzimiropoulos and M. Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *CVPR 2014*, 2014. 3

[23] M. Valstar, J. Girard, T. Almaev, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. Cohn. Fera 2015 - second facial expression recognition and analysis challenge. In *Facial Expression Recognition and Analysis Challenge, in conjunction with IEEE Int'l Conf. on Face and Gesture Recognition*, 2015. 1, 2, 5, 6, 7

[24] M. F. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(1):28–43, 2012. 3

[25] M. Wllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll. Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 31(2):153 – 163, 2013. Affect Analysis In Continuous Input. 2

[26] T. Wu, N. J. Butko, P. Ruvolo, J. Whitehill, M. S.Bartlett, and J. R. Movellan. Multi-layer architectures of facial action unit recognition. *IEEE Trans. Systems, Man and Cybernetics, Part B*, 2012. In print. 6

[27] X. Xiong and F. De la Torre Frade. Supervised descent method and its applications to face alignment. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, May 2013. 3

[28] A. Yüce, H. Gao, and J.-P. Thiran. Discriminant multi-label manifold embedding for facial action unit detection. In *Facial Expression Recognition and Analysis Challenge, in conjunction with IEEE Int'l Conf. on Face and Gesture Recognition*, 2015. 6, 7

[29] 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. Best of Automatic Face and Gesture Recognition 2013. 1