Fusing Multilabel Deep Networks for Facial Action Unit Detection

Mina Bishay and Ioannis Patras
School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

Abstract— **The automatic detection of the activation of facial muscles, i.e. the detection of the so called facial Action Units (AUs), has received significant attention due to the application of facial expression analysis/recognition in areas such as affect recognition or behavior analysis. However, the recognition of subtle expressions is a challenging task that requires a multi-modal approach where several sources of information are used. In this paper, we follow such an approach and propose a novel Deep Learning architecture that fuses information from several specialized Deep Neural Networks (DNNs) each of which models a different aspect of the problem in question. At the core of our approach is a novel dynamic adaptation of the Deep Network cost function so as to deal with the data imbalances that are inherent in multilabel classification problems - this allows cross-database training. We show the benefits of the proposed training approach and how different architectures are more suitable for particular AUs. Extensive experimental results show that our multi-modal approach outperform the state of the art by a considerable margin.**

I. INTRODUCTION

Human affect, emotions, and personality can be identified, mainly from the facial information [2]. Automatic facial analysis has been an active research area in computer vision in the last twenty years. Ekman and Friesen put the cornerstone for studying facial expressions, by proposing the Facial Action Coding System (FACS) [7]. FACS has different combinations of facial muscle movements, that result in different facial expressions. These muscle movements are represented by Action Units (AUs). Manual annotation of these AUs is a very hard task as it requires hours for annotating a minute of a video. Subsequently, building an automatic and reliable AUs detection system will offer a powerful tool for different fields (e.g. Psychological field [6], [16]), that use manual annotation for studying facial behaviour. Moreover, it will have a great impact on the current challenging facial-based applications (e.g. affect recognition [30]).

Automatic AUs annotation has been the focus of many researchers for a long time. However, there is still a gap between the state-of-the-art results and the performance needed in several face-analysis applications. This is due to head pose variation, appearance differences, and limitations of the available datasets, i.e. lack of sufficient positive samples for certain AUs, as well as recording in controlled conditions.

Inspired by the remarkable success of Deep learning in several Computer Vision problems [11], [13], [15], [18], several works apply this paradigm for AUs detection [10], [32]. Literature review shows that learned features give better performance than the hand-crafted ones. However, many works address the problem of AUs detection as a binary classification problem, where a different model is built for each AU, and ignoring in this way informative correlations between the different AUs. Other works that pose the problem as a multilabel classification problem are faced with the inherent imbalance of the data, since the number of positive examples for different AUs vary wildly. Moreover, combining different datasets in the training process becomes infeasible, as they being annotated with different AUs.

978-1-5090-4023-0/17/$31.00 ©2017 IEEE

In this paper, we propose a Deep Learning architecture that fuses information from several sources. The proposed architecture consists of eight Deep Networks, each one of which is specialized in different aspects of the problem, as depicted in fig. 1. At the first stage, two Convolutional Neural Networks (CNNs) are used for learning deep appearance features based on cropped face images, and two Multi-Layer Perceptrons (MLPs) for learning distinctive geometric features based on facial landmarks locations. A multilabel classifier is used in each CNN and MLP to extract the frame-based AUs correlations. At the second stage, a Recurrent Neural Network (RNN) is placed at the top of each of the four networks of the first stage so as to learn the spatio-temporal AUs correlations over consecutive video frames. At the third stage the predictions of those eight models are fused. Experimental results on well-known datasets show that the different networks perform significantly different in detecting diverse AUs and that the combined architecture performs considerably better than any single network.

In the heart of our architecture is a method for training each of the individual Deep Networks as a multilabel classifier that at test phase simultaneously detects all AUs. This is in contrast to other approaches that treat the problem as several binary classification problems [1], [10], [27], one for each AU, and fail, therefore, to extract features that are shared between the different AUs or utilize patterns of co-occurrence. Here we make two contributions. First, we address a common issue in multilabel problems, and in particular in AU-annotated datasets that is the inherent data imbalance - while this can be solved efficiently in binary classification problems, for example with over/under sampling, in multilabel problems balancing the data (typically the current batch) with respect to one class (AU in our case) will inevitably result in unbalancing it with respect to another class. In this paper, we address this problem by proposing to adjust the cost term associated with each AU positive example with the ratio of negative to positive examples in the current batch and therefore control the back-propagated error.

The second contribution is that we address the problem of threshold selection at the output neurons at test time. In our architecture, in order to avoid threshold selection, each

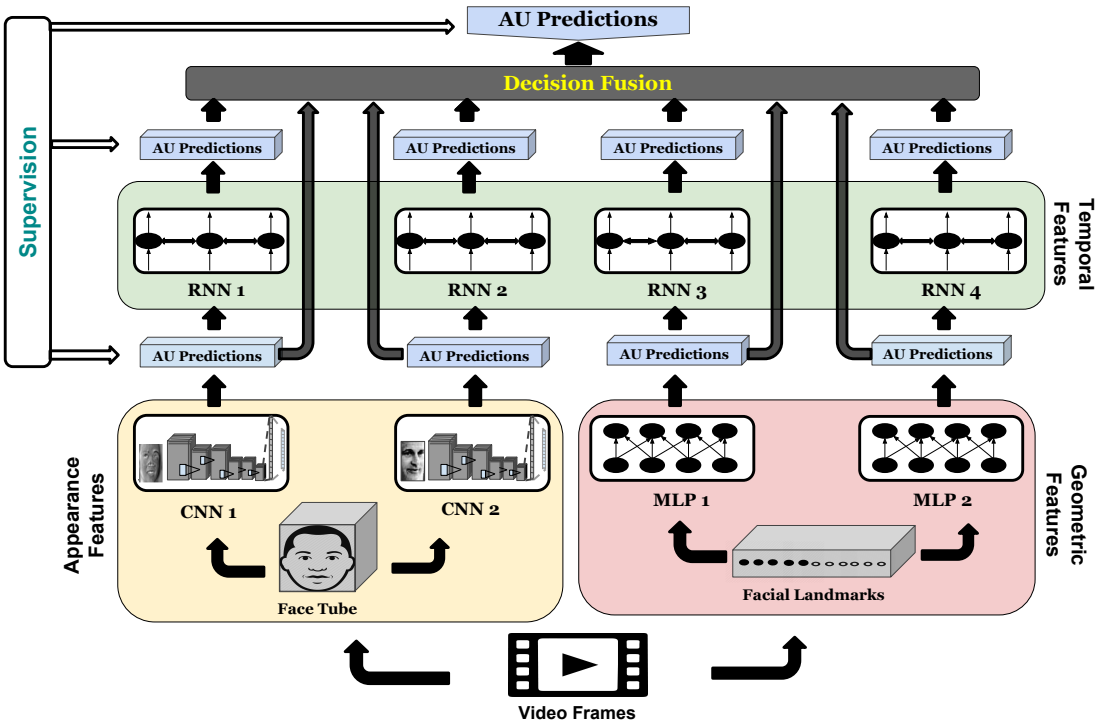

Fig. 1. The proposed architecture.

class is represented by 2 neurons, 1 for positive activation while other for negative activation. During training, those output neurons are supervised with complementary information, and during testing, the maximum of the 2 neurons is chosen to represent the activation. Those two contributions allow us to design a more general architecture that can be trained across several datasets and for the detection of many AUs. We have used in our analysis the four well-known spontaneous databases UNBC [21], DISFA [22], and FERA [27] (FERA includes two databases, BP4D and SEMAINE). The proposed architecture outperforms the state-of-the-art methods by a significant margin.

In section 2, we will review the related literature in AUs detection and highlight how the proposed method addresses their shortcomings. The proposed multilabel training scheme is described in section 3, and the proposed architecture with the fusion of several specialized Deep Networks is presented in section 4. Finally, experimental results and conclusion are given in section 5, and section 6, respectively.

II. Related Work

Automatic AUs recognition has been the focus of many researchers for a long time. Different methods have been proposed, that differ in many aspects. In this section, we present some of the state-of-the-art methods, to illustrate the main trends and highlight their shortcomings.

One of the critical steps in AUs detection is the feature extraction. Extracted features can be divided roughly into hand-crafted [1], [14], [29], [31], and learned features [8], [9], [10], [32]. Each of these features can be further split into appearance and geometric ones. In [1], the hand-crafted appearance and geometric features are fused together for better performance. In spite of the great performance achieved by the learned features, the fusion of the deep appearance and geometric features have not been discovered yet in AUs detection. Moreover, Jung et al. proved in [12] that better accuracy can be achieved in emotion recognition when both of them are used. In this paper, both deep appearance and geometric features are fused to improve the feature extraction step.

The extracted features are then used for training several binary classifiers or a multilabel classifier. In [1], [10], [27], a binary classifier is used for each AU, to learn AU-specific features. Using AU-specific classifier increases linearly the complexity, and the computational cost of the whole architecture. In [8], [9], a single multilabel classifier is used for different AUs, in order to learn general AU features, and the embedded spatial AUs correlations. In [32], a similar multilabel classifier is used, but replacing the conventional CNN general filters by a region-specific ones. The main drawback of this method is the large number of parameters (approx. 56 million), which easily make the network overfit when trained on limited data or subjects. Based on that, a single multilabel classifier is used in this paper to exploit the frame-based AUs correlations. Moreover, the multilabel classifier is modified to address the data imbalance and threshold selection problems.

Another aspect is the domain for extracting the features, that can either be spatial or spatio-temporal domain. Most of the existing works [8], [9], [32], focus on extracting features at the spatial domain. In [10], Jaiswal and Valstar proposed to

extract the short-term spatio-temporal features by using a 3D CNN, and the long-term ones by adding a bidirectional Long Short Term Memory (LSTM) to the 3D CNN. Although, the CNN-LSTM model shows a good performance in extracting spatio-temporal features, multiple single-label classifiers are trained, one for each AU, and therefore the AUs correlations are discarded. In the proposed architecture, a RNN is used on the top of a multilabel classifier to extract the spatio-temporal AUs correlations.

The normalization of the features or the face images using the subject's neutral face helps in extracting more discriminative features. In [8], Ghosh et al. proposed to normalize face images using the subject's mean (neutral) face. In [1], Baltrusaitis et al. proposed to normalize the extracted features by using features calculated from the subject's median (neutral) face. Although, subtracting the mean/median face can improve the performance significantly compared to using face images directly. The calculated mean/median face is not always the neutral face. Larger improvement can be achieved if an accurate neutral face image is fed to the network. In order to tackle the neutral face detection, two networks are used, one based on mean (neutral) face subtraction, while the other is based on original face images. The two networks complement each other for better AUs detection results.

Finally, different AU-annotated databases are available for the research community. The way that these databases are used for training and testing affects the AUs classification and reflects the generalization of the classifier. In [10], [29], models are trained and tested on the same database. In [8], Ghosh added a way of using one database for training, while other for testing. In [1], Baltrusaitis combined different databases for specific AU, in order to increase the number of training examples. Although combining databases can improve classifier performance by avoiding overfitting, it seems a hard task when a multilabel classifier is used since not all of the databases are annotated to the same AUs. In this paper, the proposed architecture is trained on all databases by back-propagating only the errors coming from the annotated AUs in each database.

III. Multilabel training scheme

AUs detection can be naturally seen as a multilabel classification problem in which, at each example, one or typically more AUs are activated. Several works address AUs detection as independent binary classification problems, where a different classifier is built for each of the AUs. Then, a ranking or thresholding technique is used to give final predictions. However, the complexity of such an approach increases as the number of classes increase. Moreover, label dependencies which are very strong in the AUs detection problem, are discarded in this way, and the networks are likely to overfit on AUs that have a few annotated examples. In this paper, we follow [24] that extends a common single-label multiclass classifier to a multilabel multiclass classifier, in order to exploit embedded correlations and reduce complexity. Such multilabel classifiers are employed in each of our eight specialized models that we will describe in the next section.

Deep learning techniques have many parameters that can easily overfit when a limited number of subjects, or data are used in training. In order to avoid overfitting and extract distinctive AU-related features, different databases should be used in the training process. The main impediment for using combined databases in training a multilabel classifier is the unequal number of AUs annotated in these databases. In order to solve this problem and exploit all the available datasets, each image in the used databases is annotated in terms of 18 AUs, with a ground truth label $q \in \{0, 1, NL\}$. The AU presence is labelled by 1, AU absence by 0, and NL if the image is not annotated for this AU. The computed cost for the NL-labelled AUs is discarded, and does not take part in the average back-propagated cost. Therefore, the computed cost is only for the annotated AUs in each batch.

The first contribution that we make in this field is addressing the problem of threshold selection for AUs classification. Typically, in order to make a binary decision on whether the AU in question is activated or not, the corresponding neuron output is thresholded either at 0.5 or, more commonly, by a threshold that is chosen based on the training set. However, different conditions (e.g. head motion, lighting effects) can affect the neuron output, and therefore using a certain threshold for all images is not the best choice. In order to overcome this problem and choose the threshold automatically, each AU i is represented by 2 neurons, one representing AU presence AU_1^i while the other representing AU absence AU_0^i. During training they are supervised by complementary information, and during testing the one with the highest output is selected. Doing so allows the network to choose the threshold automatically according to the given input conditions - for example blurring or a darker image will affect both neurons in the same manner.

The second contribution that we make in this field is a scheme that addresses the problem of data imbalance. Data imbalance is a common problem in many applications including AUs detection and results in biasing the classifier towards the class with the most samples. Typically, positive examples are limited - this can be tackled by duplicating the positive examples (named "Oversampling"), or removing some negative examples (named "Undersampling") [4]; however, this is only possible in a binary classification problem - in a multilabel classification problem balancing the data with respect to one AU will result in imbalance with respect to other AUs. In this paper, we propose a new method for balancing the data in a multilabel classifier. For each batch in the training set, let us denote by p_i the number of the positive examples and n_i the number of negative examples for the AU i. Then, the ratio r_i of the negative to positive examples is computed as:

$$r_i = \begin{cases} \dfrac{n_i}{p_i}, & \text{if } n_i \; and \; p_i \neq 0 \\ 1, & \text{otherwise,} \end{cases} \tag{1}$$

where the index $i \in \{1, 2, 3, ..., 2K\}$ where $2K$ is twice

the number of AUs, as we use 2 neurons for each AU. Then, we create a weight matrix M, having the same size of the output batch. In the weight matrix, we set the 0-labeled examples by 1, NL-labeled by 0, and 1-labeled by r_i, where r_i is given by eq. 1, and is different for each AU i. This weight matrix is multiplied elementwise by the output cost matrix. By doing so we adjust the misclassification cost of positive examples so as to prevent the biasing of the network towards the negative class when a few positive examples are available. We use the binary cross-entropy as a cost function. That is, the total batch cost is:

$$C = -\frac{1}{2K}\sum_{i=1}^{2K}\frac{1}{z_i}\sum_{j=1}^{bs} M_{ij}(t_{ij}\log q_{ij} + (1-t_{ij})\log(1-q_{ij})) \tag{2}$$

$$z_i = \sum_{j=1}^{bs} M_{ij}\ , \tag{3}$$

where z_i denotes the sum of the weights at AU i, bs the batch size, t the target value and q is the predicted value. Across the different databases, the multilabel classifier will learn general and discriminative features for AUs detection.

IV. Fusion Architecture

Our architecture consists of multiple deep networks for facial AUs detection. Two CNNs are used for extracting appearance features, and two MLPs are used for extracting geometric features. Then, a RNN is built on the top of each of the spatial models to learn the spatio-temporal AUs correlations. Finally, the eight models' predictions are fused in a linear layer, in order to pick the best weights for each AU, over the different models.

A. Preprocessing Step

Given any of the database spontaneous videos, preprocessing is a crucial step to ensure that a stream of aligned face images and landmarks are fed to our architecture. Face detection is an effective step, as all of the processing steps depend on it. Therefore, two face detectors were employed; OpenCV face detector, trained on frontal and profile faces, and Zhu-Ramanan face detector [33]. First, OpenCV is used for its robust and fast performance. Then, Zhu-Ramanan detector is used as a complementary model to process the failed frames, as OpenCV fails sometimes when the roll angle of the head pose increases. The extracted face images are passed for landmarks localization. The Supervised Descent Method [28] is used for extracting 49 landmarks. The detected landmarks are then aligned to a reference frame using Procrustes transform. The points that are invariant to facial expressions, like eye corners and nose tip are used for the alignment. Finally, the faces are cropped to 48x48, and converted to gray scale, and used with the landmarks locations as inputs to the CNNs and MLPs, respectively.

B. Convolutional Neural Networks

Following the great success achieved by AlexNet in image classification [15], CNNs have been extensively used for different Computer Vision problems in the last years. CNNs learn better appearance features than the designed ones. In this work, two CNNs are employed for extracting deep appearance features. The first CNN (called CNN1) is based on subtracting subject's mean face, which can learn meaningful features away from the appearance differences. The other CNN (called CNN2) is based on the original face images, this network can better learn AUs when the calculated neutral face is not accurate enough. More specifically, AUs are subtle and differ according to face appearance, shape, and dynamics, so subtracting the mean (neutral) face can avoid these differences by highlighting the AUs locations when its activated. The assumption of using the mean [8], or the median [1] of all subject's frames as the neutral face is not always accurate. Fig. 2 shows several subjects' mean faces, the first two are almost neutral, but in the last two the mouth is slightly opened (AU25). Subsequently, CNN1 works better for almost all AUs, but CNN2 is employed to better classify the mistaken AUs (e.g. AU25), due to the neutral face inaccuracy. CNN1 and CNN2 complement each other for better performance.

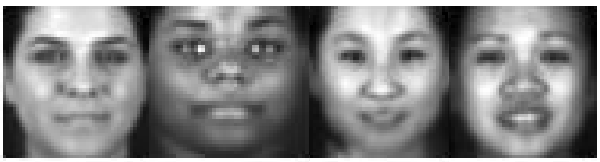

Fig. 2. Samples of subjects' mean face selected from BP4D dataset.

The CNNs take batches of 48x48 face images as input and randomly crop each into 44x44 smaller sub-image. Sub-images are randomly flipped horizontally with a probability of 0.5, in order to augment the data and avoid overfitting problems. At test time classification is done using 44x44 sub-images cropped from the center of the original face images. Each CNN consists of 3 convolutional layers and 1 fully connected layer, each convolutional layer is followed by a max-pooling layer [19]. For the first and second convolutional layers, 64 filters of size 9x9 and 5x5 respectively are used, and for the last convolutional layer, 128 filters of size 5x5 are used. The activation function used in the convolutional layers is the rectified linear unit (ReLU) function [23]. Dropout is used for regularization to avoid overfitting [26]. Max-pooling layers have filters of size 2x2 in order to make the network translation invariant. The last layer consists of K sigmoid units, where K is equal to the number of the detected AUs. The CNNs are trained using stochastic gradient descent, with the same learning parameters: 0.005 learning rate, 0.9 momentum, and 0.25 dropout. The learning rate decays with increasing epochs at a rate of 0.001 for CNN1 and 0.0005 for CNN2.

C. MultiLayer Perceptron

The inspiration of using MLP along with CNN comes from its success in emotion recognition in [12]. The normalized

landmarks' locations are used as input to the MLP. The main idea of using landmarks for AUs detection is that some AUs (e.g. AU25, AU26) are characterized by a large shift in their locations, that make the network easily learn better features for AUs detection. Our algorithm contains two MLPs trained on the extracted landmarks' positions. The first MLP (called MLP1) is trained using the 49 landmarks, normalized by subtracting subject's mean landmarks. The other MLP (called MLP2) is trained using the original 49 landmarks, normalized according to the method in [12]. The idea of using two MLPs is the same as with the CNNs, they complement each other when the subject's neutral landmarks are not accurately detected.

Each landmark location has two coordinates (x, y), so the length of the input feature vector is 49 x 2 = 98. Two hidden layers are used, each consisting of 600 neurons. The output layer consists of K sigmoid units, where K is the number of detected AUs. ReLU is used as the activation function after each hidden layer. Both MLPs are trained using stochastic gradient descent, and share the same learning parameters: 0.005 learning rate, 0.9 momentum, and 0.25 dropout.

D. Recurrent Neural Networks

Facial AUs are strongly correlated both in the spatial and temporal domains. Spatial AUs correlations are extracted by the multilabel classifier used in each of the MLPs and CNNs. In order to extract the temporal correlations, a Bi-directional Recurrent Neural Network (B-RNN) is employed [25]. The B-RNN transform a number of inputs (X) to a number of outputs (Y) based on the input values, and previous and future information. In [17], Hinton compared several RNNs over different tasks, and found that a simple RNN with ReLU and scaled identity initialization can give better performance compared to others like LSTM. Based on that, this RNN is used in our analysis. Outputs of the CNNs and MLPs are fed to the RNNs as inputs over different video frames. At frame t, the output y_t is calculated as follows:

$$y_t = a(W^f_{out} h^f_t + W^b_{out} h^b_t + b_{out}), \quad t \in \{1, 2, ..., T\} \quad (4)$$

$$h^f_t = a(W^f_{in} x_t + W^f_h h^f_{t-1} + b^f_h) \quad (5)$$

$$h^b_t = a(W^b_{in} x_t + W^b_h h^b_{t+1} + b^b_h)\,, \quad (6)$$

where W^f_{out}, and W^b_{out} are the output weight matrices connecting the forward and backward hidden states to the output layer, respectively. W^f_{in}, and W^b_{in} are the input weight matrices connecting the input layer to the forward and backward hidden states, respectively. W^f_h, and W^b_h are the forward and backward hidden weight matrices, respectively. b_{out}, b^f_h, and b^b_h are the output, forward and backward hidden bias vectors, respectively. T is the length of video sequence. The activation function a is the sigmoid function for the output layer, and ReLU function for the hidden layers.

The B-RNNs are trained on databases with different video lengths. In our analysis, videos are partitioned into segments of length 90 frames. The hidden layers weights are initialized by a scaled identity matrix, where 0.1 is chosen as the scale value. All the B-RNNs were trained by stochastic gradient descent, with a learning rate of 0.01, gradient clipping at 1.0, and batches of size 32 sequences.

E. Decision Fusion Stage

Appearance (CNNs) and geometric (MLPs) models have varying AUs detection performances. In order to exploit both models information/features, and tackle the defects in neutral face detection, decision fusion is employed for the eight deep models. Indeed, decision fusion will also fuse the spatial AUs correlations extracted by CNNs and MLPs in the multilabel classifiers, with the temporal ones extracted by the B-RNNs.

A linear model whose parameters are optimized with random search [3], [13], is used to combine the predictions from the different modality classifiers. In random search, one weight is given for each AU/class in each model, and the final AU prediction is the weighted sum of all models predictions. Random sampling of uniform distribution is used to get weights between 0 and 1. Then, each class weights are normalized to 1. The best sampled weights are chosen based on the best F1-score. In our architecture, 25,000 iterations are used initially, then a local random search is performed around the best weights chosen over different classes. The weights for the local search are sampled from a Gaussian distribution with a mean equal to the best chosen weights, and standard deviation std of 0.5. The local search is repeated around the best chosen weights after every 1000 iterations, and at each time the std is decreased by a factor of 0.8 and stopped when it is smaller than 0.001.

V. Experiments

Experimental setup. Four spontaneous databases are used in our experiments: UNBC, DISFA, and FERA (which includes two datasets, SEMAINE and BP4D). FERA is divided into training, validation, and testing sets. In our first experiment, the BP4D is used to show the effect of the proposed methods for data balancing and automatic threshold selection in a multilabel classifier. In the second experiment, UNBC, DISFA, and FERA training set are randomly combined for training our architecture. Then, the FERA validation set is used for testing in terms of 18 AUs, to show the AUs detection performance over the 8 deep models, as well as their fusion model. In the third experiment, the code of the trained model is submitted to the FERA organizers, in order to be tested on the FERA 2015 challenge. FERA challenge specifies 6 AUs on SEMAINE, and 11 AUs on BP4D, for challenging. Using the FERA platform, allows all participants to test their methods in similar conditions, for a better and fair evaluation. In the last experiment, 3-fold partitioning is implemented for the BP4D database, in order to compare the proposed method performance to the results reported in [5], [20], [31], [32].

Performance metrics. In this paper, accuracy and F1-score are used for evaluating the performance of the proposed algorithm. Accuracy is a widely used and powerful metric, but when the ratio of the negative to positive examples is large, the detection accuracy of the positive class is almost neglected. On the other hand, F1-score depends mainly on

TABLE I

F1-SCORE AND ACCURACY AT DIFFERENT STAGES OF THE PROPOSED MULTILABEL CLASSIFIER.

AU	MLC		ATS		DB		ATS + DB		ATS + DB + CDT	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
AU1	0.514	83.02	0.517	84.84	0.502	83.73	0.551	81.13	0.502	73.73
AU2	0.356	78.27	0.358	82.20	0.357	81.81	0.371	76.82	0.403	73.73
AU4	0.545	80.23	0.510	81.14	0.555	81.67	0.556	78.74	0.515	74.28
AU6	0.775	77.30	0.787	79.10	0.792	78.19	0.791	78.65	0.805	77.86
AU7	0.735	68.75	0.736	70.04	0.745	69.46	0.743	70.70	0.763	69.12
AU9	0.269	92.00	0.229	93.04	0.269	92.69	0.351	89.84	0.349	83.49
AU10	0.804	75.40	0.836	79.03	0.816	76.70	0.809	76.50	0.846	79.50
AU12	0.861	83.23	0.859	82.98	0.856	82.19	0.865	83.79	0.868	83.71
AU14	0.647	64.88	0.613	63.92	0.685	66.31	0.628	62.80	0.648	61.98
AU15	0.371	78.15	0.287	79.78	0.345	76.67	0.454	76.17	0.465	69.87
AU17	0.616	71.34	0.596	74.40	0.629	71.97	0.637	73.15	0.656	70.42
AU23	0.398	79.54	0.352	82.26	0.423	79.49	0.465	75.82	0.461	73.42
AU24	0.445	83.82	0.287	82.55	0.397	81.83	0.525	83.29	0.562	82.49
AU28	0.363	95.33	0.429	96.70	0.426	96.46	0.416	95.17	0.403	94.89
Avg	0.550	79.38	0.533	80.86	0.557	79.94	0.583	78.75	0.589	76.32

the detection performance of the positive class, but the number of the true negatives does not take a part in the computation. In what follows we report both metrics.

Results. In the first experiment, we show the effect of adding Automatic Threshold Selection (ATS), Data Balancing (DB), as well as Cross-Dataset Training (CDT) to the MultiLabel Classifier (MLC). As an illustrative case we show the performance of one of the eight models, that is CNN1, on the BP4D dataset using a 2:1 ratio of training and validation. Results on 14 out of 18 AUs that are annotated in BP4D are reported. We report results for various combinations of ATS, DB, and CDT. When ATS is not used, the AU threshold is chosen based on the best F1-score and when CDT is selected, several datasets, i.e. UNBC, DISFA, SEMAINE and BP4D are used for training.

Table I summarizes the obtained results for the different settings. Using only ATS seems to reduce the average F1-score compared to the simple MLC - a reason for that is the increase in false negative for AUs in which the positive/negative ratio is very low, where one output neuron is biased towards the more frequent class (i.e. 0) and the other output neuron is biased towards the most infrequent class (i.e. 1). Our cost adaptation method for data balancing improves the MLC performance by 0.7% in F1-score, but larger improvement is obtained when adding the ATS with DB - the F1-score is improved by approx 3.3%. Finally, using ATS and DB with the expansion of the training set adds an additional 0.6% to the F1-score, as better features are extracted. Based on that, ATS, DB, and CDT is used in the training of each of our 8 deep models.

In order to show the importance of fusing information from several sources, and how the different models perform on different AUs, the SEMAINE and BP4D validation sets are combined and used for testing our architecture. The F1-score and accuracy over the different stages of the architecture are shown in Table II. By comparing the performance of the appearance (CNNs) and geometric (MLPs) features, we found that on average, the appearance features perform better. However, AUs performance varies over the MLPs and CNNs - typically, CNNs detect better AUs that are characterized by subtle change in the appearance (e.g. AU2, AU6, AU10, AU17), while MLPs perform better for AUs characterized by large displacement in landmarks locations (e.g. AU25, AU26, AU28).

The effectiveness of the neutral face subtraction can be inferred from the good performance achieved by CNN1 in comparison to CNN2. On average, CNN1 outperforms CNN2 on both F1-score and accuracy. CNN1 works better for most of the AUs, except some of those related to the mouth area (e.g. AU15, AU17, AU24, AU25). That is the result of the inaccuracy in neutral face estimation at the mouth region. In the same way, comparing the performance of MLP1 and MLP2 gives similar conclusions. In order to address the problem of the neutral face estimation and utilize both the appearance and geometric features, decision fusion is implemented. Adding the fusion step to the four spatial models leads to the second best performing model, where the F1-score improves by approximately 3% and accuracy by 1% compared to the best spatial model CNN1.

The RNNs extract the spatio-temporal AUs correlations over different video frames. Table II, shows the effect of adding the RNN for each spatial model. The F1-score is not affected for CNN2 and MLP2, but for CNN1 and MLP1, the improvement gained is 1.5% and 2.5%, respectively. Adding the RNN to the CNN1 led to the third best model. The decision fusion of the 4 spatial models with the 4 temporal models led to the best performing model, where the F1-score improves by 0.28% and accuracy by 1.2% compared to the second best model. Fusion of the 8 models exploits the appearance and geometric features, spatio-temporal AUs correlations, and addresses the inaccuracy in neutral face detection, which helps in boosting the performance of the AUs detection.

The proposed architecture has been also tested on the FERA challenge, including the BP4D and SEMAINE testing sets. Table III and table IV show the obtained results on the BP4D and SEMAINE, respectively, with other results reported in the literature [1], [9], [10], [27], [29]. We achieved the best F1-score on BP4D dataset and the second best on the SEMAINE dataset.

TABLE II

F1-SCORE AND ACCURACY OVER DIFFERENT MODALITIES IN THE PROPOSED ARCHITECTURE.

AU	CNN1		CNN2		MLP1		MLP2		CNN1 -RNN		CNN2 -RNN		MLP1 -RNN		MLP2 -RNN		CNN-MLP Fusion		CNN-MLP -RNN Fusion	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
AU1	0.469	80.59	0.261	67.38	0.466	79.68	0.323	68.74	0.489	82.80	0.263	64.82	0.458	76.50	0.308	55.88	0.453	80.43	0.466	81.94
AU2	0.448	80.31	0.264	67.30	0.391	77.92	0.274	68.42	0.432	82.06	0.273	65.32	0.407	78.28	0.298	60.94	0.441	81.86	0.439	82.40
AU4	0.508	86.44	0.399	79.90	0.438	81.82	0.390	80.48	0.508	87.38	0.392	80.51	0.431	76.61	0.296	61.27	0.558	87.89	0.552	88.52
AU6	0.750	83.55	0.703	78.24	0.688	80.01	0.681	75.90	0.769	84.44	0.724	80.78	0.702	78.97	0.652	70.42	0.772	84.80	0.775	85.17
AU7	0.703	78.51	0.696	76.06	0.660	75.62	0.676	74.17	0.718	78.91	0.693	77.25	0.670	71.88	0.620	60.92	0.718	79.20	0.718	79.38
AU9	0.237	91.73	0.185	90.18	0.204	91.86	0.122	90.35	0.241	90.04	0.179	93.68	0.169	84.20	0.09	70.10	0.255	91.17	0.288	93.28
AU10	0.805	82.28	0.761	78.26	0.707	74.16	0.760	76.22	0.815	82.77	0.767	78.25	0.750	74.19	0.750	71.94	0.803	81.96	0.803	82.01
AU12	0.828	82.06	0.819	79.67	0.796	79.17	0.821	80.65	0.833	82.16	0.817	79.39	0.807	80.11	0.826	81.11	0.841	83.16	0.841	83.17
AU14	0.591	71.49	0.565	67.60	0.532	67.31	0.567	67.24	0.581	71.63	0.575	66.04	0.546	66.61	0.581	61.74	0.600	72.19	0.587	71.77
AU15	0.356	76.02	0.373	74.54	0.329	76.41	0.283	68.03	0.378	79.77	0.369	77.17	0.342	71.52	0.281	56.54	0.398	78.74	0.409	80.98
AU17	0.573	73.11	0.584	70.71	0.531	68.97	0.516	60.46	0.582	74.63	0.585	70.90	0.563	68.05	0.544	59.77	0.602	74.09	0.603	74.97
AU23	0.398	83.07	0.383	78.75	0.379	81.75	0.337	75.97	0.395	80.85	0.385	81.43	0.350	77.70	0.294	65.47	0.419	82.40	0.427	84.00
AU24	0.403	84.30	0.444	84.39	0.403	82.47	0.330	75.29	0.427	85.94	0.458	84.82	0.432	80.11	0.349	69.43	0.417	85.08	0.430	85.99
AU25	0.675	74.34	0.735	76.86	0.688	77.65	0.746	74.36	0.672	73.24	0.739	77.20	0.772	80.69	0.780	78.12	0.766	80.81	0.775	81.23
AU26	0.373	78.43	0.319	80.76	0.373	79.65	0.424	70.16	0.457	81.18	0.327	79.45	0.477	74.61	0.420	61.29	0.344	81.75	0.355	81.99
AU28	0.379	96.32	0.206	89.94	0.384	95.62	0.250	92.18	0.393	95.91	0.186	86.37	0.416	94.96	0.277	89.76	0.509	97.00	0.486	96.81
AU43	0.291	92.46	0.205	87.34	0.179	92.88	0.269	89.28	0.289	91.79	0.169	87.19	0.290	87.15	0.219	63.77	0.353	93.40	0.340	93.83
AU45	0.369	69.79	0.288	69.63	0.355	68.13	0.325	59.75	0.375	70.90	0.278	65.73	0.366	69.40	0.331	62.66	0.394	72.08	0.398	72.36
Avg	0.506	81.38	0.455	77.64	0.472	79.51	0.450	74.87	0.520	82.02	0.454	77.57	0.497	77.31	0.448	66.17	0.536	82.67	0.539	83.87

TABLE III

F1-SCORE ON THE BP4D TESTING SET.

AU	B-LGBP [27]	B-Geo [27]	BCNN [9]	CDPSL [1]	DLE [29]	CNN-LSTM [10]	Proposed
AU1	0.180	0.188	**0.399**	0.260	0.261	0.280	0.349
AU2	0.159	0.185	0.346	0.250	0.167	0.280	**0.370**
AU4	0.225	0.197	0.317	0.250	0.283	0.340	**0.345**
AU6	0.671	0.645	0.718	0.730	0.729	0.700	**0.756**
AU7	0.751	0.799	0.776	**0.800**	0.785	0.780	0.776
AU10	0.799	0.801	0.797	**0.840**	0.802	0.810	0.807
AU12	0.792	0.801	0.793	0.820	0.779	0.780	**0.836**
AU14	0.666	0.720	0.681	0.720	0.625	**0.750**	0.636
AU15	0.139	0.238	0.235	0.340	**0.348**	0.200	0.344
AU17	0.245	0.311	0.368	0.330	**0.380**	0.360	0.376
AU23	0.239	0.320	0.309	0.340	**0.441**	0.410	0.426
Avg	0.442	0.473	0.522	0.516	0.508	0.520	**0.547**

TABLE IV

F1-SCORE ON THE SEMAINE TESTING SET.

AU	B-LGBP [27]	B-Geo [27]	BCNN [9]	CDPSL [1]	DLE [29]	CNN-LSTM [10]	Proposed
AU2	0.755	0.569	0.372	0.410	0.655	**0.800**	0.505
AU12	0.517	0.595	0.707	0.570	**0.769**	0.740	0.702
AU17	0.066	0.091	0.067	0.200	0.215	**0.320**	0.108
AU25	0.400	0.445	0.602	0.690	0.623	**0.850**	0.810
AU28	0.009	0.250	0.040	0.260	0.251	0.330	**0.338**
AU45	0.209	0.396	0.257	0.420	0.325	**0.570**	0.451
Avg	0.326	0.391	0.341	0.425	0.481	**0.600**	0.486

Finally, in order to compare our work with other methods in the literature, in the last experiment, a 3-fold partitioning is adopted on the combined BP4D training and validation sets, where 2 partitions are mixed with UNBC, DISFA, and SEMAINE datasets for architecture training, while the remaining partition is used for testing. We report the average F1-score of 3 runs over the 12 AUs mentioned in [32]. The obtained results were compared with other state-of-the-art methods, namely JPML [31], DRML [32], CNN-RNN [5], and EAC [20], in table V. The proposed architecture outperforms the other methods on 9 AUs out of 12, and get the best average F1-score by a considerable margin.

VI. CONCLUSION

In this work, we proposed an architecture that fuse different deep models (CNN, MLP, B-RNN) together, in order to capture deep appearance, geometric, and temporal features. In the core of our architecture, a new method is proposed for data balancing without adding any extra computational cost, in addition to a novel way for selecting threshold automatically based on each image. Experimental results show that the proposed technique outperforms the state-of-the-art techniques by a significant margin.

TABLE V

F1_SCORE ON THE 3-FOLDED BP4D DATASET.

AU	JPML [31]	DRML [32]	CNN-RNN [5]	EAC [20]	Proposed
AU1	0.326	0.364	0.314	0.390	**0.563**
AU2	0.256	0.418	0.311	0.352	**0.471**
AU4	0.374	0.430	**0.714**	0.486	0.570
AU6	0.423	0.550	0.633	0.761	**0.791**
AU7	0.505	0.670	**0.771**	0.729	0.768
AU10	0.722	0.663	0.450	0.819	**0.843**
AU12	0.741	0.658	0.826	0.862	**0.878**
AU14	0.657	0.541	**0.729**	0.588	0.662
AU15	0.381	0.332	0.340	0.375	**0.431**
AU17	0.400	0.480	0.539	0.591	**0.602**
AU23	0.304	0.317	0.386	0.359	**0.435**
AU24	0.423	0.300	0.370	0.358	**0.512**
Avg	0.459	0.483	0.532	0.559	**0.627**

VII. ACKNOWLEDGMENTS

The work of Mina Bishay is a part of the Newton-Mosharafa PhD scholarship, which is jointly funded by the Egyptian Ministry of Higher Education and the British Council.

REFERENCES

[1] T. Baltrušaitis, M. Mahmoud, and P. Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 6, pages 1–6. IEEE, 2015.

[2] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 5, pages 53–53. IEEE, 2003.

[3] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[4] N. V. Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 853–867. Springer, 2005.

[5] 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.

[6] S. Dimic, C. Wildgrube, R. McCabe, I. Hassan, T. R. Barnes, and S. Priebe. Non-verbal behaviour of patients with schizophrenia in medical consultations–a comparison with depressed patients and association with symptom levels. *Psychopathology*, 43(4):216–222, 2010.

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

[8] 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.

[9] 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.

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

[11] K. Jarrett, K. Kavukcuoglu, Y. Lecun, et al. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153. IEEE, 2009.

[12] H. Jung, S. Lee, S. Park, I. Lee, C. Ahn, and J. Kim. Deep temporal appearance-geometry network for facial expression recognition. *arXiv preprint arXiv:1503.01532*, 2015.

[13] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, et al. Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, pages 1–13, 2015.

[14] 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.

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

[16] M. Lavelle, S. Dimic, C. Wildgrube, R. McCabe, and S. Priebe. Non-verbal communication in meetings of psychiatrists and patients with schizophrenia. *Acta Psychiatrica Scandinavica*, 131(3):197–205, 2015.

[17] Q. V. Le, N. Jaitly, and G. E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

[18] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE, 2011.

[19] Y. LeCun, K. Kavukcuoglu, C. Farabet, et al. Convolutional networks and applications in vision. In *ISCAS*, pages 253–256, 2010.

[20] W. Li, F. Abtahi, Z. Zhu, and L. Yin. Eac-net: A region-based deep enhancing and cropping approach for facial action unit detection. *arXiv preprint arXiv:1702.02925*, 2017.

[21] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews. Painful data: The unbc-mcmaster shoulder pain expression archive database. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 57–64. IEEE, 2011.

[22] 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.

[23] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[25] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[26] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[27] 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.

[28] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013.

[29] A. Yüce, H. Gao, and J.-P. Thiran. Discriminant multi-label manifold embedding for facial action unit detection. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 6, pages 1–6. IEEE, 2015.

[30] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):39–58, 2009.

[31] 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.

[32] 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.

[33] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.