Real Time Emotion Detection System

Gowtham Kannan B (gkbandep@iu.edu)

***Abstract*—Facial expression is one of the key aspect of human communication. It is a medium through which people express their emotional states that are important for social survival. In this paper , we develop a system that recognizes 10 facial expressions which are universally expressed and recognized across all cultures. What makes our project unique is that our system predicts human emotions in real-time. We developed a deep Convolutional network architecture combined with a robust face detection algorithm that extracts features from a live video application and instantly returns the emotion of the user.Our model, when trained using AffectNet Database [1] ,achieved 90% accuracy and during the test phase,it achieved 59.1% accuracy.**

***Index Terms*—Face detection, Neural networks, Convolutional neural networks, Batch normalizations, Stochastic gradient, Adagrad and Max Pooling**

I. Introduction

Humans use different forms of communications such as speech, hand gestures and emotions. Being able to understand ones emotions and the encoded feelings is an important factor for an appropriate and correct understanding.With the ongoing research in the field of robotics, especially in the field of humanoid robots, it becomes interesting to integrate these capabilities into machines allowing for a more diverse and natural way of communication. Besides Human Computer Interaction other fields like surveillance or driver safety could also profit from it. Being able to detect the mood of the driver could help to detect the level of attention, so that automatic systems can adapt.

Affective (psychological term for emotion) computing seeks to develop systems and devices that can recognize, interpret, and simulate human affects through various channels such as face, voice, and biological signals [2]. Face and facial expressions are undoubtedly one of the most important nonverbal channels used by the human being to convey internal emotion. Face and facial expressions are undoubtedly one of the most important nonverbal channels used by the human being to convey internal emotion. Many methods rely on extraction of the facial region. This can be realized through manual inference [3] or an automatic detection approach [4]. Methods often involve the Facial Action Coding System (FACS) which describes the facial expression using Action Units (AU). An Action Unit is a facial action like raising the Inner Brow. Multiple activations of AUs describe the facial expression [5]. Being able to correctly detect AUs is a helpful step, since it allows making a statement about the activation level of the corresponding emotion. Handcrafted facial landmarks can be used such as done by Kotsia et al. [3]. Detecting such landmarks can be hard, as the distance between them differs depending on the person. Not only AUs can be used to detect emotions, but also texture. When a face shows an emotion the structure changes and different filters can be applied to detect this [6].

In this paper, we propose a real time emotion detection system built on top of a deep convolutional network.The deep network proposed is a little variant of the one proposed in [7] with little changes in normalization and learning mechanisms. But the basic components of the model are same as the one proposed in [7]. The proposed model has similarities with [8] in the sense, there are convolutional layers that are being learned in parallel as we go deeper into the network. The proposed model is trained to predict the following emotion descriptors :-

1) Neutral
2) Happy
3) Surprise
4) Sad
5) Fear
6) Disgust
7) Anger
8) Contempt
9) None
10) Uncertain

This paper is arranged as follows: After this introduction, Related Work (II) is presented which focuses on Emotion/Expression recognition and the various approaches scientists have taken. III dicusses , Background, which focuses on the main components of the architecture proposed in this article. IV contains a summary of the used Datasets. In V the architecture is presented. This is followed by the experiments and its results (VI) . Finally, VII summarizes and concludes the article and VIII discusses future work.

II. Related Work

A detailed overview for expression recognition was given by Caleanu [9] and Bettadapura [5]. In this Section mainly work which similar to the proposed method is presented as well as few selected articles which give a broad overview over the different methodologies.

Szegedy et al.[8] have proposed an architecture called GoogLeNet. This is a 27 layer deep network, mostly composed of CNNs. The network is trained using stochastic gradient descent. In the ILSVRC 2014 Classification Challenge this network achieved a top-5 error rate of 6.67% winning the first place. Using the the Extended Cohn-Kanade Dataset Happy and Routray [6] classify between six basic emotions using dimensional reduction of features such as the eyes, lips and other interest points and then these reduced features are given to SVM and an average acuracy of 94.09% was obtained using

10 fold cross validation.

Video based emotion recognition has been proposed by Byeon and Kwak [10]. They have developed a three dimensional CNN which uses groups of 5 consecutive frames as input. A database containing 10 persons has been used to achieve an accuracy of 95%. Song et al. [11] have used a deep convolutional neural network for learning facial expressions. The created network consists of five layers with a total of 65k neurons. Convolutional, pooling, local filter layers and one fully connected layer are used to achieve an accuracy of 99.2% on the CKP set. To avoid over fitting the dropout method was used.

Kotsia and Pitas [3] detect emotions by mapping a Candide grid, a face mask with a low number of polygons, onto a persons face. The grid is initially placed randomly on the image, then it has to be manually placed on the persons face. Throughout the emotion, the grid is tracked using a KanadeLucasTomasi tracker. The geometric displacement information provided by the grid is used as feature vector for multiclass SVMs. The emotions are anger, disgust, fear, happiness, sadness, and surprise. They evaluate the model on the Cohn-Kanade dataset and an accuracy of 99.7% has been achieved. Shan et al. [12] have created an emotion recognition 3 system based on Local Binary Patterns (LBP). The LBPs are calculated over the facial region. From the extracted LBPs a feature vector is derived. The features depend on the position and size of the sub-regions over witch the LBP is calculated. AdaBoost is used to find the sub-regions of the images which contain the most discriminative information. Different classification algorithms have been evaluated of which an SVM with Boosted-LBP features performs the best with a recognition accuracy of 95.1% on the CKP set.

In 2013 Zafar et al. [13] proposed an emotion recognition system using Robust Normalized Cross Correlation (NCC). The used NCC is the Correlation as a Rescaled Variance of the Difference between Standardized Scores. Outlier pixels which influence the template matching too strong or too weak are excluded and not considered. This approach has been evaluated on different databases including AR FaceDB (85% Recognition Accuracy) and the Extended Cohn Kanade Database (100% Recognition Accuracy).

III. Background

A. Convolutional Layer

Convolutional Layers perform a convolution over the input. Let f_k be the filter with a kernel size $n \times m$ applied to the input x. $n \times m$ is the number of input connections each CNN neuron has. The resulting output of the layer calculates as follows:

$$C(x_{u,v}) = \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} f_k(i,j) x_{u-i,v-j} \tag{1}$$

To calculate a more rich and diverse representation of the input, multiple filters f_k with $k \in N$ can be applied on the input. The filters f_k are realized by sharing weights of neighboring neurons. This has the positive effect that lesser weights have to be trained in contrast to standard multilayer perceptrons, since multiple weights are bound together.

B. Max pooling

Max Pooling reduces the input by applying the maximum function over the input x_i. Let m be the size of the filter, then the output calculates as follows:

$$M(x_i) = max\{x_{i+k,i+l} \| |k| \leq \frac{m}{2}, |l| \leq \frac{m}{2} \;\; k,l \in N\} \tag{2}$$

This layer features translational invariance with respect to the filter size.

C. Rectified Linear Unit

A Rectified Linear Unit (ReLU) is a cell of a neural network which uses the following activation function to calculate its output given x:

$$R(x) = max(0, x) \tag{3}$$

Using these cells is more efficient than sigmoid and still forwards more information compared to binary units. When initializing the weights uniformly, half of the weights are negative. This helps creating a sparse feature representation.

D. Fully connected layer

The fully connected layer also known as Multilayer Perceptron connects all neurons of the prior layer to every neuron of its own layer. Let the input be x with size k and l be the number of neurons in the fully connected layer. This results in a matrix $W_{l \times k}$.

$$F(x) = \sigma(W * x) \tag{4}$$

σ is the so called activation function. In our network σ is the identity function.

E. Output Layer

The output layer is a one hot vector representing the class of the given input image. It therefore has the dimensionality of the number of classes. The resulting class for the output vector x is:

$$C(x) = \{i | \exists i \forall j \neq i : x_j \leq x_i\} \tag{5}$$

F. Softmax Layer

The error is propagated back over a Softmax layer. Let N be the dimension of the input vector, then Softmax calculates a mapping such that: $S(x) : R^N \rightarrow [0,1]^N$. For each component $1 \leq j \leq N$, the output is calculated as follows:

$$S(x)_j = \frac{e^{x_j}}{\sum_{i=1}^{N} e^{x_i}} \tag{6}$$

G. Batch normalization

Batch Normalization is a technique to provide any layer in a Neural Network with inputs that are zero mean/unit variance. Figure 1 shows various steps involved in batch normalization.

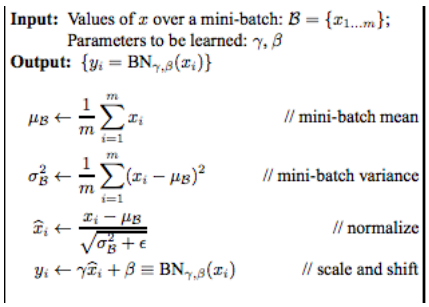

Figure 1: Step by step illustration of batch normalization

H. Stochastic gradient descent

Batch gradient descent use the entire training set to compute the next update to parameters at each iteration and this makes it to converge very well to local optima. However, often in practice computing the cost and gradient for the entire training set can be very slow and sometimes intractable on a single machine if the dataset is too big to fit. Another issue with batch optimization is there is no way to incorporate new data in an online setting. Stochastic gradient descent (SGD) addresses both of these issues by following the negative gradient of the objective after seeing only a single of a few training examples. THe use of SGD in the neural network setting is motivated by the high cost of running back propagation over the full training set. SGD can overcome this cost and still lead to fast convergence.

The standard gradient descent algorithm updates the parameters θ of the objective $J(\theta)$ as,

$$\theta = \theta - \alpha.\Delta_\theta E[J(\theta)] \tag{7}$$

where the expectation in the above equation is approximated by evaluating the cost and gradient over the full training set. Stochastic Gradient Descent (SGD) simply does away with the expectation in the update and computes the gradient of the parameters using only a single or a few training examples. The new update is given by,

$$\theta = \theta - \alpha.\Delta_\theta E[J(\theta; x^{(i)}, y^{(i)})] \tag{8}$$

with a pair $(x^{(i)}, y^{(i)})$ from the training set. But usually, each parameter update in SGD is computed w.r.t a few training examples or a mini batch as opposed to a single example.

I. Adagrad

Adagrad is an algorithm for gradient based optimization that adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters.For this reason, it is well suited for dealing with sparse data.Adagrad modifies the general learning rate α at each time step t for every parameter θ_i based on the past gradients that have been computed for θ_i:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,ii} + \epsilon}}.g_{t,i} \tag{9}$$

$G_t \in R^{d\times d}$ is a diagonal matrix where each diagonal element i, i is the sum of the squares of the gradient w.r.t θ_i up to time step t, while ϵ is a smoothing term that avoids division by zero. One of Adagrad's main benefits is that it eliminates the need to manually tune the learning rate.

IV. Dataset

A. AffectNet

The Affectnet database was introduced by Mollahosseini, Hassani and Mahoor [1]. It contains $\approx$ 367500 images with 10 different emotions. The distribution of images across each category in shown in the Table I

Expression	Number
Neutral	80,276
Happy	146,198
Sad	29,487
Surprise	16,288
Fear	8,191
Disgust	5,264
Anger	28,130
Contempt	5,135
None	35,322
Uncertain	13,163

Table I: Categorical distribution of images.

AffectNet (Affect from the InterNet) is the largest database of the categorical and dimensional models of affect in the wild (as shown in Table 1). The database is created by querying emotion related keywords from three search engines and annotated by expert human labelers.Emotion-related keywords were combined with words related to gender, age, or ethnicity, to obtain nearly 362 strings in the English language such as joyful girl, blissful Spanish man, furious young lady, astonished senior. These keywords are then translated into five other languages: Spanish, Portuguese, German, Arabic and Farsi. The direct translation of queries in English to other languages did not accurately result in the intended emotions since each language and culture has differing words and expressions for different emotions. A total of 1250 search queries were compiled and used to crawl the search engines for the images. Three search engines (Google, Bing, and Yahoo) were queried with these 1250 emotion related tags. Other search engines such as Baidu and Yandex were considered. However, they either did not produce a large number of facial images with intended expressions or they did not have available APIs for automatically querying and pulling image URLs into the database. Additionally, queries were combined with negative terms (e.g., drawing, cartoon, animation, birthday, etc.) to

Figure 2: Samples images from Affectnet

avoid non-human objects as much as possible.

The average image resolution of faces in AffectNet is 425×425. With the help of MS Face API it was found that 49% of the faces are men and average estimated age of the faces is 33.0 years with the standard deviation of 16.96 years. In particular, 10.85, 3.9, 30.19, 26.86, 14.46, and 13.75 percent of the faces are in age ranges [0, 10), [10, 20), [20, 30), [30, 40), [40, 50) and [50, -), respectively. Figure 2 shows a sample image in each category.

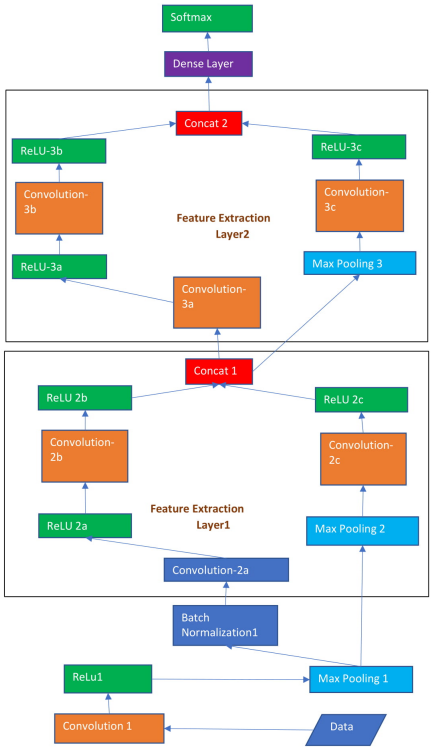

Figure 3: Proposed architecture. Edited from [7]

V. Architecture

The proposed deep Convolutional Neural Network architecture (depicted in Figure 3) consists of four parts. The first part automatically preprocesses the data. This begins with Convolution 1, which applies 64 different filters. The next layer is Pooling 1, which down-samples the images and then they are normalized by batch normalizaiton. The next steps are the two FeatEx (Parallel Feature Extraction Block) blocks, highlighted in Figure 3. They are the core of the proposed architecture and described later in this section. The features extracted by theses blocks are forwarded to a fully connected layer, which uses them to classify the input into the different emotions. The described architecture is compact, which makes it not only fast to train, but also suitable for real-time applications.

1) Feature Extractor: The key structure in our architecture is the Parallel Feature Extraction Block (FeatEx). It is inspired by the success of GoogleNet. The block consists of Convolutional, Pooling, and ReLU Layers. The first Convolutional layer in FeatEx reduces the dimension since it convolves with a filter of size 1×1. It is enhanced by a ReLU layer, which creates the desired sparseness. The output is then convoluted with a filter of size 3×3. In the parallel path a Max Pooling layer is used to reduce information before applying a CNN of size 1×1. This application of differently sized filters reflects the various scales at which faces can appear. The paths are concatenated for a more diverse representation of the input. Using this block twice yields good results.

Layer	Output Size
Data	300×300
Convolution 1	$64 \times 147 \times 147$
Pooling 1	$64 \times 73 \times 73$
Batch Normalization	$64 \times 73 \times 73$
Convolution 2a	$96 \times 73 \times 73$
Convolution 2b	$208 \times 71 \times 71$
Pooling 2a	$64 \times 71 \times 71$
Convolution 2c	$64 \times 71 \times 71$
Concat 2	$271 \times 71 \times 71$
Convolution 3a	$96 \times 71 \times 71$
Convolution 3b	$208 \times 69 \times 69$
Pooling 3a	$272 \times 69 \times 69$
Convolution 3c	$64 \times 69 \times 69$
Concat 3	$272 \times 69 \times 69$
Pooling 3b	$272 \times 34 \times 34$
Classifier	10×1

Table II: This Table lists the different output sizes produced by each layer.

2) Visualization: The different layers of the architecture produce feature vectors as can be seen in Fig 4. The first part until batch normalization preprocesses the data and creates

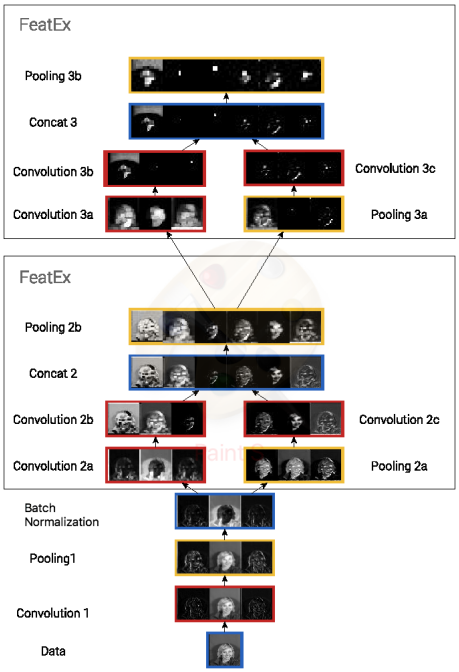

Figure 4: Feature visualization at various levels. Taken from [7]

multiple modified instances of the input. These show mostly edges with a low level of abstraction. The first FeatEx block creates two parallel paths of features with different scales, which are combined in Concat 2. The second FeatEx block refines the representation of the features. It also decreases the dimensionality. This visualization shows that the concatenation of FeatEx blocks is a valid approach to create an abstract feature representation. The output dimensionality of each layer can be seen in Table II.

VI. Experimentation & Results

The model was executed for 500 iterations on 5 GPU nodes in Bigred2. Total training time was ≈ 8 hrs. To ease the process of training, a variant of stochastic gradient namely AdaGrad (as proposed in [14]) is used.

We divided images into 80-20 ratio with 80% of the data for training and remaining 20% of the data for testing. Figure 5 shows the confusion matrix for the testing phase. From this it is clear that "Fear" is recognized with highest percentage and "Disgust" is recognized with lowest percentage. We got an average accuracy of 59.2% across all the different categories while testing. This is close to the one obtained in Affectnet using human annotators mentioned in [1]. The accuracy can be improved by tuning the hyper parameters like the learning rate, momentum for SGD etc. Also the model trained with expression specific images might yield higher accuracies.

VII. Conclusion

In this paper, we have classified human emotions from their facial expressions using real-time video streaming. Our model

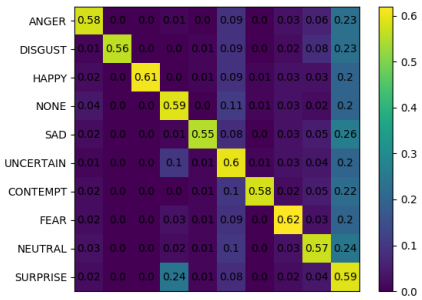

Figure 5: The confusion matrix on the test images of Affectnet. The lowest accuracy is achieved by the emotion Disgust with 56% while Fear is recognized with 62%

is fully automatic and can classify the ten universal emotions: Happiness, Sadness, Anger, Disgust, Surprise, Fear, Uncertainty, Neutral and Contempt . Our proposed model is a Deep Learning Network which comprises of several convolutional layers for feature extraction. The project is still in it's growing phase and is expected to produce better outcomes. In addition to that, it is our intention to extend the project to recognize emotions in domesticated animals like dogs and cats.

VIII. Future Work

Our model currently detects 10 kinds of emotions in real-time. The model can detect upto 3-4 faces and return their corresponding emotions. However, the accuracy of our current model is moderate and can be improved. Our next step towards improvising the model will be to train our model using other datasets. We have also planned to extend our model to detect emotions of animals as well. Due to lack of proper dataset for training our model, we have included it in our future work. Emotion detection can be helpful in many real life scenarios.We realized that our model can be used to detect violent activities in a crowded area. Hence, we have planned to extend our model to detect emotion in such areas.

Another drawback of our model is the problem of localization while predicting the emotions of many individuals in a live feed. So we plan to incorporate localization techniques proposed in [15] for image classification to ease the efficiency of our model.Future work also involves using capsule networks mentioned in [16] which has achieved state of art accuracy in [17] for emotion recognition using datasets proposed in [18] and [1].

IX. Code Base

The entire code base for the project can be found in EmoNet. The code base consists of a python wrapper for real time emotion detection, saved models and the model code.

Listing 1: How to execute

```
git clone https://github.iu.edu/gkbandep/↩
    CV_PROJ.git
cd CV_PROJ.git
pip install -r requirements.txt
python live_demo.py
```

After these steps, test the code with your most emphatic expressions 🙂.
Below are few screen shots from sample runs over our real time emotion detection system.

Acknowledgment

We would like to thank Dr. David Crandall for motivating us throughout the process. A special thanks to the Assistant Instructors for their patience and guidance during the crucial moments of our development. We also extend our sincere gratitude to Dr. Behzad Hasani of University of Denver, Iran University of Science Technology for granting us access to [1] database at free of cost. Lastly, we would like to thank Indiana University and School of Informatics,Computing and Engineering for providing us with resources without which our project wouldnt have been possible.

References

[1] Ali Mollahosseini, Behzad Hassani, and Mohammad H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *CoRR*, abs/1708.03985, 2017.

[2] Jianhua Tao and Tieniu Tan. Affective computing: A review. In Jianhua Tao, Tieniu Tan, and Rosalind W. Picard, editors, *Affective Computing and Intelligent Interaction*, pages 981–995, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[3] Jingying Chen, Dan Chen, Yujiao Gong, Meng Yu, Kun Zhang, and Lizhe Wang. Facial expression recognition using geometric and appearance features. In *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service*, ICIMCS '12, pages 29–33, New York, NY, USA, 2012. ACM.

[4] Philipp Michel and Rana El Kaliouby. Real time facial expression recognition in video using support vector machines. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, pages 258–264, New York, NY, USA, 2003. ACM.

[5] Vinay Bettadapura. Face expression recognition and analysis: The state of the art. *CoRR*, abs/1203.6722, 2012.

[6] S. L. Happy and Aurobinda Routray. Automatic facial expression recognition using features of salient facial patches. *CoRR*, abs/1505.04026, 2015.

[7] Peter Burkert, Felix Trier, Muhammad Zeshan Afzal, Andreas Dengel, and Marcus Liwicki. Dexpression: Deep convolutional neural network for expression recognition. *CoRR*, abs/1509.05371, 2015.

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] Ctlin Cleanu. Face expression recognition: A brief overview of the last decade, 05 2013.

[10] Behzad Hassani and Mohammad H. Mahoor. Facial expression recognition using enhanced deep 3d convolutional neural networks. *CoRR*, abs/1705.07871, 2017.

[11] Diego Mushfieldt, Mehrdad Ghaziasgar, and James Connan. Robust facial expression recognition in the presence of rotation and partial occlusion. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, SAICSIT '13, pages 186–193, New York, NY, USA, 2013. ACM.

[12] Shiqing Zhang, Xiaoming Zhao, and Bicheng Lei. Facial expression recognition based on local binary patterns and local fisher discriminant analysis. *WSEAS Trans. Sig. Proc.*, 8(1):21–31, January 2012.

[13] Aliya Zafar. Face recognition with expression variation via robust ncc, 12 2013.

[14] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011.

[15] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 237–244, 2009.

[16] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. *CoRR*, abs/1710.09829, 2017.

[17] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[18] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *Proceedings of IEEE Int'l Conf. Multimedia and Expo (ICME'05)*, pages 317–321, Amsterdam, The Netherlands, July 2005.