

MIT Open Access Articles

*Multi-velocity neural networks for
facial expression recognition in videos*

The MIT Faculty has made this article openly available. **Please share**
how this access benefits you. Your story matters.

As Published: 10.1109/TAFFC.2017.2713355

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: <https://hdl.handle.net/1721.1/134853>

Version: Original manuscript: author's manuscript prior to formal peer review

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Multi-velocity neural networks for gesture recognition in videos

Otkrist Gupta

otkrist@mit.edu

Dan Raviv

raviv@mit.edu

Ramesh Raskar

raskar@media.mit.edu

Massachusetts Institute of Technology
Cambridge, MA

Abstract

We present a new action recognition deep neural network which adaptively learns the best action velocities in addition to the classification. While deep neural networks have reached maturity for image understanding tasks, we are still exploring network topologies and features to handle the richer environment of video clips. Here, we tackle the problem of multiple velocities in action recognition, and provide state-of-the-art results for gesture recognition, on known and new collected datasets. We further provide the training steps for our semi-supervised network, suited to learn from huge unlabeled datasets with only a fraction of labeled examples.

1. Introduction

Nonverbal communication is a key factor in the interaction between individuals, replacing or amplifying spoken words. Our body language, voice pitch, intonation and volume, movement of our pupils or our chronemics choices are just a few examples emphasizing the richness of human communication skills [32]. A special subset of non-verbal interactions, explored in this paper, is based on facial expressions. Their perception initiates rapid cognitive processes in the brain and have both communicative and reflexive components [12]. They assist in verbal communication by providing context to what we are saying, making their recognition important for studying social interactions.

We use computers on a daily basis, interacting with artificial agents in increasing number of tasks. Advances in speech and natural language processing have presented us with personalized smart agents [42, 11], while examining our facial gestures has provided a richer set of tools for improving human computer interaction [9]. In the last two decades we have seen several attempts to automate the way computers respond towards human emotions [23, 6, 2], where the ultimate goal is to create humanoid robots which can blend in the environment [53].

Unfortunately this task is hard to solve, and current state-

of-the-art results are far from satisfactory. Recent advances in machine learning have shown that if we provide a neural network with enough samples, it can learn very complex structures [15]. Today hard tasks in computer vision, such as labeling images, recognizing objects and faces or classifying videos have become a feasible task for computers which can now provide competitive results to humans and sometimes even outperform them [37, 45].

In this paper we focus on facial gesture recognition from videos using deep neural networks. We tackle the problem of a small size labeled dataset, and present a new layer which compensates for velocity changes in the time domain. We compare our methods to multiple techniques and datasets, as well as presenting our own collected data, and we report state-of-the-art results in almost every category.

We hope to empower researchers in this area by providing them with a huge dataset which can help them build even bigger and better deep neural networks, while eliminating the need to spend several months required to acquire a dataset of such proportions, and for some living under limited resources acquiring such a dataset can be prohibitively impossible. We summarize our contributions by:

1.1. Contributions

1. To the best of our knowledge, we have built **the largest face video dataset** to date comprising of **162 million** face images with facial landmark labels (**7.8 billion annotations**) contained in 6.5 million video clips, and 2777 videos which have been labeled for seven emotions. The dataset will be made public for research purposes.
2. We develop a multi-velocity autoencoder architecture using **new multi-velocity** layers for generating velocity-free deep motion features.
3. We report state of the art results for video gesture recognition using spatio-temporal convolutional neural networks.
4. We introduce a new topology and protocol for semi-supervised learning, where the number of labeled data

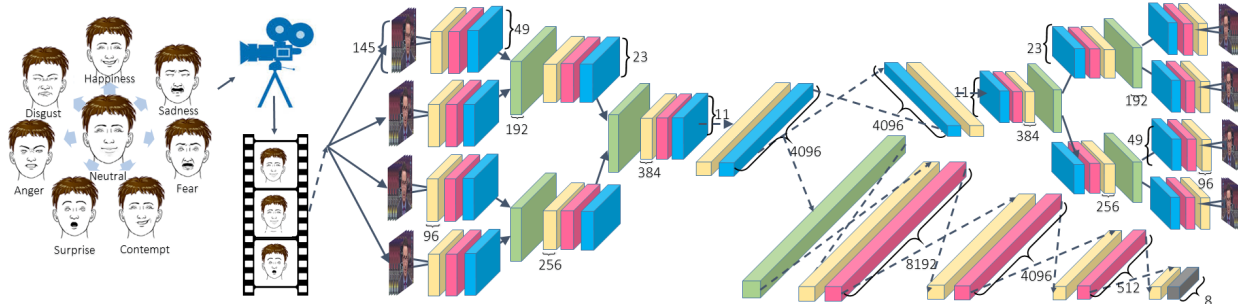


Figure 1. We learn 7 different facial emotions from short (about 1 sec, 25 frames) video clips. Our prediction system is based on slow temporal fusion neural network, trained by hybridization of autoencoding a huge collected dataset and a loss prediction on a small set of labeled gestures.

points is only a fraction of the entire dataset.

2. Related Work

Machine learning techniques such as Support Vector Machines have been used for facial expression recognition given the movement of facial fiducial points [24, 33, 41, 10] achieving real time performance [38]. Many of these techniques involve a pipeline with multiple phases - face detection and alignment, feature extraction/landmark localization and classification as the final step. Other interesting approaches [7, 50, 36, 48] we should mention are based on temporal features [28, 51], and multiple kernels [29], action units [54, 40], as well as emotion recognition from speech [34, 39]. We will compare our method against some of those approaches in section 5.

Recently, deep neural nets have been shown to perform well on classification tasks on images and videos, outperforming most traditional learning systems. One of the most interesting results was presented three years back on a large scale dataset (LSVRC 2011), where a deep convolutional net outperformed all other methods by far [25]. With advances in convolutional neural nets, we have seen neural nets applied to video classification [21, 46] and even facial expression recognition [1, 13] but these networks were not deep enough or used other feature extraction techniques like PCA or Fisherface.

Training a neural net normally requires a large labeled dataset which is hard to obtain using reasonable resources. Providing high quality results when only a small part of the data is labeled is an interesting problem referred to as semi-supervised learning. In [26] the authors pre-trained the system using pseudo labels, while in [52, 22] they embedded the data in a low dimensional space. Very recently superior results have been shown [30, 20, 18, 14, 19] using deep neural nets to combine labels and un-labeled data in the same package. In this paper we follow those guidelines and train from start-to-end a hybrid system composed

of autoencoders for unlabeled data and additional loss function for the classification tasks.

3. Method

We propose a semi-supervised approach using a deep neural network, by combining an autoencoder with a classification loss function, and training both of them in parallel. The input for the first layer is a short sequence of facial gestures composed of 9 frames cropped to 145×145 pixels window. The loss function is evaluated by combining a predictive loss from 7 different pre-labeled gestures (for the labeled part of the dataset), and autoencoder Euclidean loss for the entire (labeled and un-labeled) collection. The weights of each layer are dynamically altered such that the importance of the autoencoder loss decreases with relation to the predictive loss as the training progresses. While generating the data, we use Viola and Jones face detection [49] for cropping the faces. We use slow fusion based convolutional neural network with convolutions in both space and time (see figure 4 for a detailed overview).

3.1. Action autoencoder

Our action autoencoder consists of convolutional autoencoder for learning deep features and reducing the dimensionality of the data. We use convolutional filters with weight sharing in the first 6 layers followed by 2 fully connected layers. This network is similar to Imagenet [25] but accepts inputs of size $145 \times 145 \times 9$ as an input. Using shorthand notation the full architecture can be written as $C(96, 11, 3) - N - C(256, 5, 2) - N - C(384, 3, 2) - N - FC(4096) - FC(4096) - DC(96, 11, 3) - N - DC(256, 5, 2) - N - DC(384, 3, 2)$, where $C(n, f, s)$ stands for convolution layers with n filters of size f and stride s . $DC(n, f, s)$ stands for deconvolution layers with n deconvolving filters of size f and stride s . $FC(n)$ stands for fully connected layers with n nodes and N stands for local response normalization layers. We extend the convo-

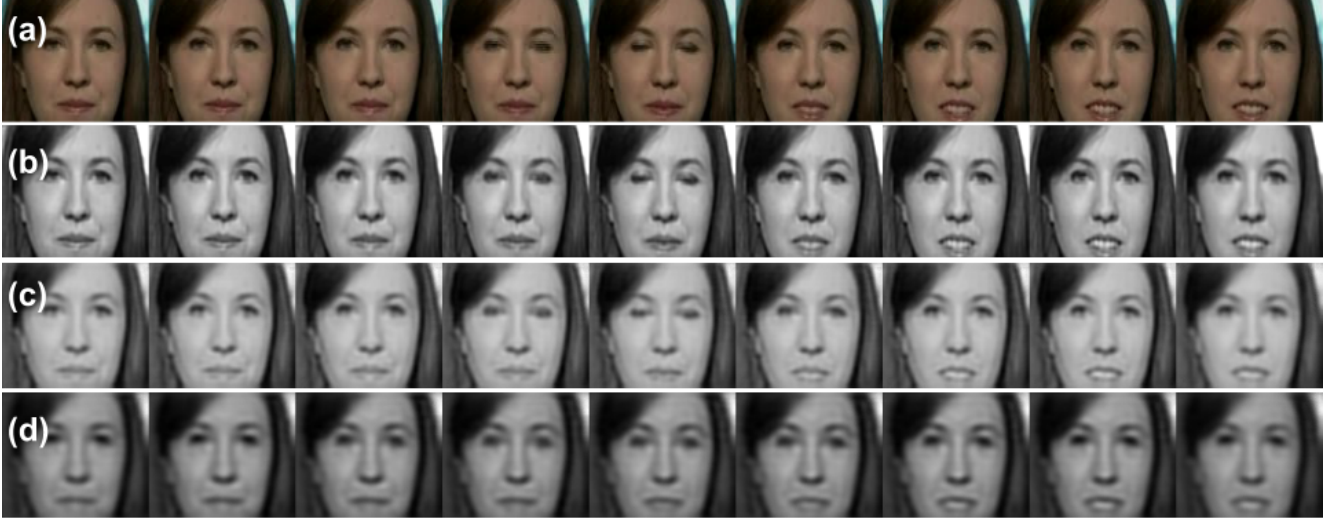


Figure 2. Results from reconstruction using temporal convolutional autoencoder on a face video. (a) Input video sequence. (b) Reconstruction after using 4 convolutional layers. (c) Reconstruction after using 8 layers. (d) Reconstruction after using 12 layers.

lution layers in time and use slow fusion model [21] which slowly combines temporal information in successive layers. The first convolution filters have size 3 and stride 2 in time domain, the next layer has size 2 and stride 2 and the third layer combines all temporal features. The deconvolution layers are extended in time as well and reverse the slow fusion generating temporal features successively (see figure 3).

3.2. Multi-Velocity Encoders

One of the main challenges in action recognition is related to assigning similar classification to objects at different velocities. In this work we propose to learn the velocity of the sequence in parallel to its classification by adaptive temporal interpolation. Our multi-velocity autoencoder consists of 3 action autoencoders combined together to access temporal features for different velocities. We achieve this by adding a convolution layer as the first layer which uses cubic b-spline interpolation to *slow down* the video and generate intermediate frames. Piece-wise cubic b-spline interpolation is preferred over polynomial techniques as it can minimize interpolation error for fewer points and lower degree polynomials [16]. For initialization a sampling factor of 1, 2/3 and 1/3 is chosen, which is later refined as a part of the learning.

Next we show how to generate the required weights for interpolation and encode them as a neural network layer. Cubic b-splines are continuous *piecewise-polynomial* functions containing polynomials of degree 3 or less. A cubic b-spline spanning $n + 1$ points comprises of n cubic polynomials ($\mathbf{S}_n(x)_{n=1}^N$) which can be uniquely defined using $4n$ coefficients. These coefficients can be recovered by ap-

plying linear constraints arising from continuity and differentiability of the function on the break points (or *knots*). We represent input video at each pixel as a function of time and use cubic b-splines to approximate intermediate values. We represent intermediate polynomials between $n + 1$ frames as a coefficient vector \bar{p} containing coefficients for all n polynomials.

Let \bar{x}, \bar{y} be the frame numbers and pixel values known to us at the different frames, these frames are obvious choices for break points as we try approximating space between frames using b-spline curves. Cubic b-spline coefficients \bar{p} for each pixel can be generated by solving a linear equation $\mathbf{A}\bar{p} = \mathbf{T}\bar{y}$ as shown in the appendix. Here both \mathbf{A} and \mathbf{T} depend only on frame numbers (\bar{x}) and are independent of pixel values (\bar{y}) or pixel coordinates.

Let u_o be one of new points in time where we want to interpolate a video frame, we can compute it now by selecting k^{th} consecutive frames containing u_o , choosing piecewise polynomial contained in-between these frames ($\mathbf{S}_k(x)$) and evaluating it at u_o . We can write this as a dot product between coefficients and input $\bar{r} \cdot \bar{p}$, where u_o lies between k^{th} consecutive frames and \bar{r} is as defined below in (1):

$$[\bar{r}_i] = \begin{cases} u_o^{i-4k} & \text{if } x_{\lfloor i/4 \rfloor} \leq u_o \leq x_{\lfloor i/4 \rfloor + 1} \\ 0 & \text{o/w} \end{cases} \quad (1)$$

Extending (1) to several temporal positions; Let $\bar{u} = [u_i]$ be locations in time where frames needs to be interpolated, we compute them as a matrix vector product $\mathbf{R}\bar{p}$. Here each row of $\mathbf{R} = [r_{j,i}]$ is computed using a shifted version of the

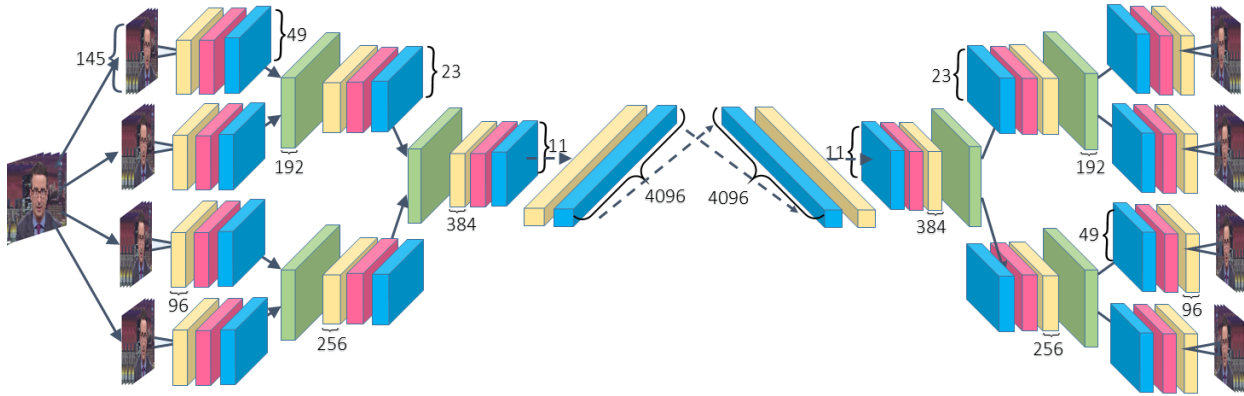


Figure 3. Convolution autoencoder using slow fusion technique combined with convolutions in time. Deconvolution layers *deconvolve* the temporal features and reconstruct frames which are compared to the original input.

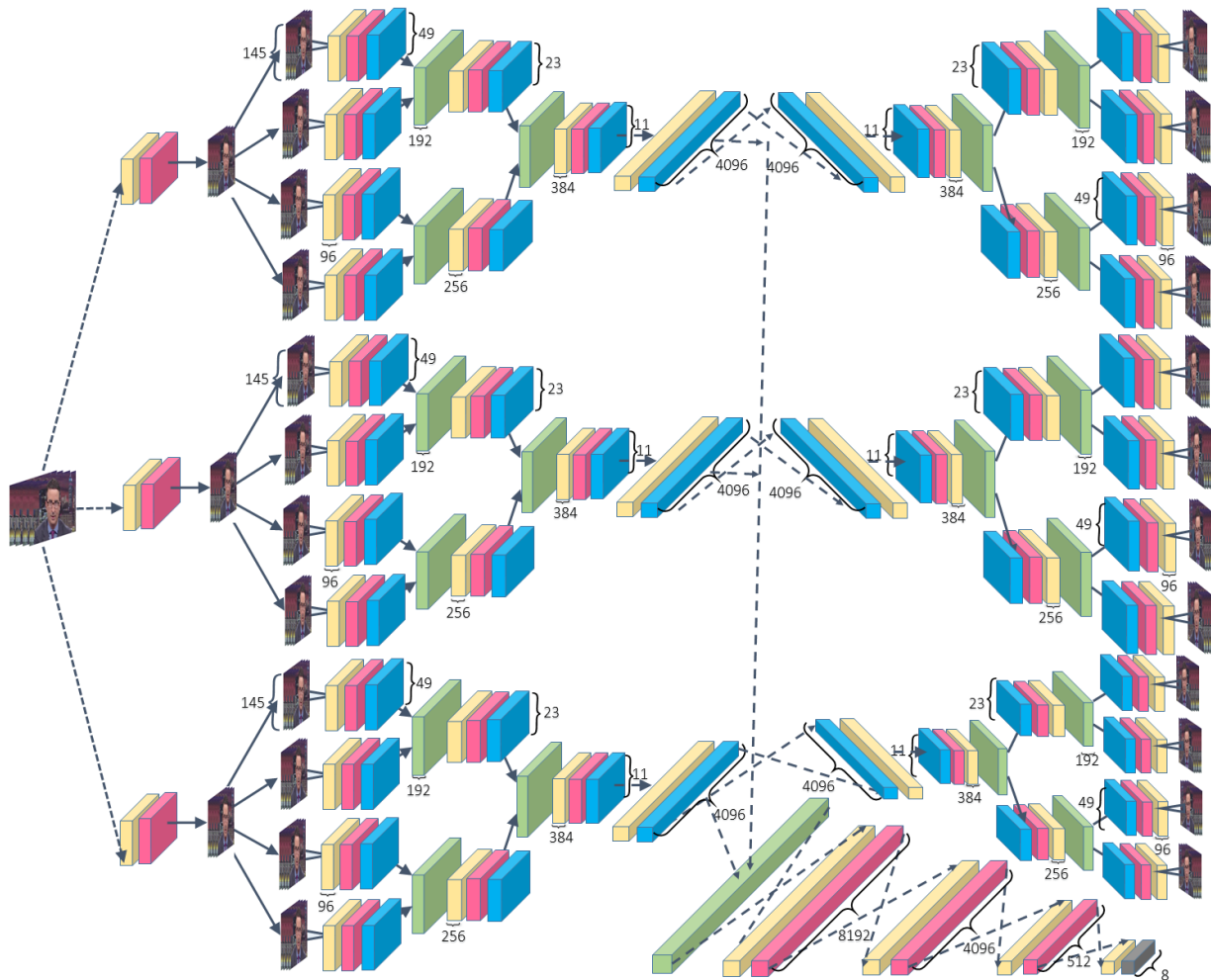


Figure 4. Complete architecture of multi-video semi supervised learner comprises of temporal filters for generating video frames at multiple velocities serving as input to 3 separate autoencoders. The predictor merges deep features from all 3 autoencoders and learns classification labels using deep neural net on top of these.

equation given above, specifically:

$$[r_{j,i}] = \begin{cases} (u_j - j)^{i-4k} & \text{if } x_{\lfloor i/4 \rfloor} < u_j < x_{\lfloor i/4 \rfloor + 1} \\ 0 & \text{o/w} \end{cases} \quad (2)$$

Equation (2) shows how interpolated pixel values $\mathbf{F}(\bar{u})$ are linearly related to b-spline coefficients $\mathbf{R}\bar{p}$. Solving for \bar{p} using $\bar{p} = \mathbf{A}^{-1}\mathbf{T}\bar{y}$ we infer that $\mathbf{F}(\bar{u}) = \mathbf{R}\mathbf{A}^{-1}\mathbf{T}\bar{y}$. The new velocity layer weights are initialized by computing matrix $\mathbf{R}\mathbf{A}^{-1}\mathbf{T}$ which is independent of pixel values \bar{y} and their spatial locations. We can represent this matrix as a *caffe* convolution layer with shared weights, which contains n filters of size $1 \times 1 \times n$ applied to all frames of video. We use algorithm 1 to create 3 different weight matrices which interpolate sampling factors of 1, 2/3 and 1/3.

Algorithm 1 Generate convolution layer spline weights.

Input: Frame numbers \bar{x} , new temporal locations \bar{u}

Output: Caffe Weight Matrix \mathbf{W}

```

1: function SPLINEWEIGHTS(c)
2:    $nSplines \leftarrow length(\bar{x}) - 1$ 
3:   for ( $i \leftarrow 0; i < nSplines; i++$ ) do
4:      $p \leftarrow 4i$ 
5:      $\mathbf{T}_{p,i} \leftarrow 1$ 
6:      $\mathbf{T}_{p+1,i+1} \leftarrow 1$ 
7:     for ( $h \leftarrow 0; h \leq 1; h++$ ) do
8:        $s \leftarrow p - 4h$ 
9:        $\mathbf{A}_{p+h,p:p+3} \leftarrow [h^3, h^2, h, 1]$ 
10:       $\mathbf{A}_{i+2,s+4:s+8} \leftarrow -1^{h+1}[3h^2, 2h, 1, 0]$ 
11:       $\mathbf{A}_{i+3,s+4:s+8} \leftarrow -1^{h+1}[6h, 2, 0, 0]$ 
12:       $\mathbf{A}_{i+3,i-4:i+3} \leftarrow [6, 0, 0, 0, -6, 0, 0, 0]$ 
13:       $\mathbf{A}_{i+4,0:7} \leftarrow [6, 0, 0, 0, -6, 0, 0, 0]$ 
14:      for ( $i \leftarrow 0; i < length(\bar{u}); i++$ ) do
15:         $p \leftarrow find(\bar{x}, \lfloor \bar{u}(i) \rfloor)$ 
16:        for ( $h \leftarrow 0; h < 4; h++$ ) do
17:           $\mathbf{R}_{i,4p+h} \leftarrow (\bar{u}(i) - \bar{x}(p))^h$ 
18:       $\mathbf{W} \leftarrow \mathbf{R}\mathbf{A}^{-1}\mathbf{T}$ 
19: return  $\mathbf{W}$ 

```

3.3. Semi-Supervised Learner

One of the main challenges we face today for training deep neural networks is the need for large labeled datasets. The richness of data is probably one of the main reasons why neural nets report such impressive predictive results in almost every field, but it is also extremely hard to collect and label such datasets. In Semi-supervised paradigm, we assume that only a part of the data is labeled, yet we wish to utilize the knowledge hidden within the entire set. Here we combine the action autoencoder convolution layers with a softmax loss function for the labeled set. The classifier neural net is inspired by Imagenet [25], with additional

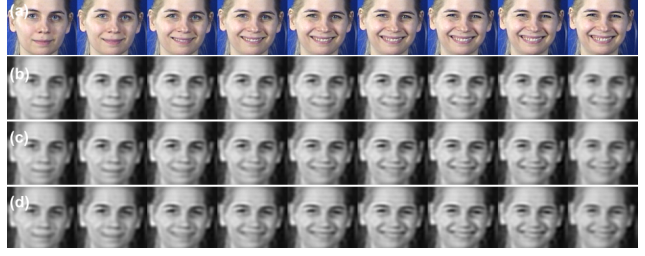


Figure 5. Results from reconstruction using multi velocity encoders, bottom 3 images are output from autoencoder ensemble. (a) Input video sequence. (b) Reconstruction using encoder with sampling factor of 1/3. (c) Reconstruction using sampling factor of 2/3. (d) Reconstruction at original velocity.

fully connected layers which are shared with the autoencoder, to generate deeper classification features from the latter. The full architecture of the predictor is $C(96, 11, 3) - N - C(256, 5, 2) - N - C(384, 3, 2) - N - FC(4096) - FC(8192) - FC(4096) - FC(512) - FC(8)$ with softmax layers in the end for label classification. Please refer to section 3.1 for explanation of the architecture shorthand.

The protocol we suggest for training the net is as important as the topology itself. We begin by training the autoencoder as a sole learner from the outer layer to the inner ones. Meaning, we adaptively add layers to the autoencoder, train the neural net, and use the produced weights as initialization for the next step. This is one of the traditional approaches used to train autoencoders [15, 5]. Next, we use the weights for initialization of the semi-supervised net, allowing the entire net to fine tune. A key factor in training is the learning rate of the two matched learners. We begin the training using a higher learning rate for the autoencoder (with predictor layers staying fixed using zero learning rate) and end the process with increased importance to the labeled loss function. While training on the labeled data, ratio between the two varies from a factor of 10^3 to a factor of 10^5 favoring the loss layer.

3.3.1 Multi-Velocity Semi-Supervised Learner

Finally we attach the new proposed Multi-Velocity layers as the first structure of the semi-supervised neural net. Each sub-structure (See Figure 4), has its own autoencoder, all of which are concatenated after the inner most convolution layer into a feature vector (size 12288), later used by the labeled loss function. The learner loss function can be expressed as a weighted sum of autoencoder and predictor loss given in equation 3 below.

$$L = \alpha \sum_v \|\bar{x} - \bar{x}_v\| - \beta \sum_j y_j \log \left(\frac{e^{o_j}}{\sum_k e^{o_k}} \right) \quad (3)$$

Confusion matrix using our methods on Cohn-Kanade								Confusion matrix using external methods							
	Anger	Contempt	Happy	Disgust	Fear	Sadness	Surprise		Anger	Contempt	Happy	Disgust	Fear	Sadness	Surprise
Anger	0.85	0.08	0	0	0	0.08	0	Anger	0.73	0	0.07	0	0	0.20	0
Contempt	0	0.60	0	0	0	0	0.40	Contempt	0	0.86	0	0	0	0.14	0
Happy	0	0	1.00	0	0	0	0	Happy	0	0	0.95	0	0.05	0	0
Disgust	0.06	0	0	0.94	0	0	0	Disgust	0.25	0.12	0	0.38	0	0.12	0.12
Fear	0	0	0.14	0	0.57	0	0.29	Fear	0	0	0	0	1.00	0	0
Sadness	0.25	0	0	0	0	0.75	0	Sadness	0.33	0	0	0.11	0	0.44	0.11
Surprise	0	0	0	0	0	0	1.00	Surprise	0	0	0	0.05	0	0	0.95

Table 1. Confusion matrices over test results for Cohn Kanade dataset using our methods and best performing external method which uses *Expressionlets* [28]. On the left we show results for the proposed multi-velocity semi-supervised approach across various facial expressions, while on the right we present confusion matrix from *Expressionlets* method. Highest accuracy in each category is marked using green color. We outperform competing method in 5 verticals by getting 100% accuracy on happiness, 100% on surprise, 94% on disgust, 85% in anger and 75% in sadness. For both methods misclassification occur when emotions like sadness get recognized as anger and *vice-versa*.

Here \bar{x}, \bar{x}_v are autoencoder inputs and outputs, y_j are the input labels and o_j is the outputs from predictor layer. $\sum_v \|\bar{x} - \bar{x}_v\|$ is the combined Euclidean loss across three multi-velocity encoders and $-\sum_j y_j \log\left(\frac{e^{o_j}}{\sum_k e^{o_k}}\right)$ is softmax loss [4]. While training using labeled data, the loss coefficient β is selected to keep softmax loss an order of magnitude higher than the Euclidean loss. Loss coefficient α is adjusted as softmax loss goes down to continue training predictor layers, without overfitting autoencoder layers. Notice that we use two coefficients for the energy function and not just controlling the ratio between the two since the back-propagation algorithm has its own additional parameters.

4. Datasets

In order to evaluate the proposed architecture we use two known datasets from literature as well as present two additional datasets collected by us; The first dataset contains more than 160 million images combined into 6.5 million short (25 frames) clips, used by us to train our autoencoders. The second dataset is comprised of 2777 short clips labeled for seven emotions. In the following section we elaborate on the four datasets.

4.1. Autoencoder dataset

In order to train very deep neural nets we must obtain a huge collection of data. Here we collected 6.5 million video clips containing 25 frames each, adding up to more than 162 million face images. We used viola-jones face detector to find and segment out the faces. Next, we localized landmarks for each frame using a deformable model for the face [3] and detected the facial pose by fitting a 3D model to the landmarks. This process allowed us to restrict the dataset to videos which contain faces tilted less than 30 degrees and remove any faces looking sideways.

In order to extract only meaningful video clips we re-

moved clips with static gestures or those where the faces were rapidly altering, either due to some high speed movement or simply due to appearance of a different face. We achieved this by blurring the clips and calculating the difference between consecutive frames.

The raw videos were taken from public sources such as CNN, MSNBC, FOX and CSPAN. To our knowledge this is the biggest facial dataset reported in literature, and we plan to make it public.

4.2. Asevo dataset

In order to collect and label our own gestures we developed a video recording and annotation tools. We developed the application using python based OpenCV and captured the clips using Logitech C920 HD camera. The database contains facial clips from 160 subjects both male and female, where gestures were artificially generated according to a specific request, or genuinely given due to a shown stimulus. We collected a total of 2777 clips out of which 1745 were captured after providing the stimulus while 1032 were generated artificially. To create natural facial expressions we selected a bank of YouTube videos for each facial expression and showed them to subjects, capturing their reaction to the visual stimulus. We quantitatively summarize this dataset in table 2, where posed clips refers to the artificially generated expressions and non-posed to the stimulus activation procedure.

4.3. Cohn Kanade Dataset

The Cohn Kanade Dataset [31] is one of the most popular datasets used for facial expression recognition. The dataset contains 593 sequences out of which 327 are labeled for 7 emotions. Along with posed facial expressions, the dataset also contains non-posed smile expressions. However the dataset lacks depth in having other non-posed expressions and is not extensive as Asevo dataset in capturing naturally expressed emotions. Each video clip contains facial expres-

Confusion matrix using our methods on Asevo Dataset							Confusion matrix using external methods								
	Anger	Contempt	Happy	Disgust	Fear	Sadness	Surprise		Anger	Contempt	Happy	Disgust	Fear	Sadness	Surprise
Anger	0.62	0.13	0.02	0.10	0.01	0.06	0.06	Anger	0.61	0.08	0.06	0.07	0.04	0.06	0.08
Contempt	0.16	0.33	0.22	0.04	0.04	0.05	0.16	Contempt	0.09	0.44	0.27	0.06	0.02	0.06	0.05
Happy	0.01	0.06	0.86	0.01	0.00	0.03	0.03	Happy	0.01	0.07	0.85	0.01	0	0.01	0.06
Disgust	0.25	0.13	0.16	0.31	0.02	0.06	0.07	Disgust	0.19	0.14	0.25	0.22	0.01	0.02	0.16
Fear	0.15	0.03	0.03	0.10	0.18	0.04	0.46	Fear	0.21	0.09	0.06	0.07	0.12	0.06	0.39
Sadness	0.23	0.10	0.05	0.04	0.06	0.18	0.34	Sadness	0.26	0.20	0.19	0.02	0.05	0.13	0.16
Surprise	0.06	0.05	0.02	0.04	0.08	0.09	0.66	Surprise	0.15	0.03	0.09	0.01	0.06	0.05	0.61

Table 3. Confusion matrix over test results for Asevo dataset using the proposed multi-velocity semi-supervised learner (left) and external competing method using *Covariance Riemann kernel* [29]. We outperform multiple kernel based methods in 6 out of 7 emotion categories. Similar to the Cohn-Kanade dataset (see table 1), both methods fail when disgust, fear or sadness get classified as anger.

Emotion	Posed	Non-Posed	Cumulative
Anger	132	318	450
Sadness	118	148	266
Contempt	153	301	454
Fear	137	96	233
Surprise	188	232	420
Joy	172	503	675
Disgust	132	147	279
Total	1032	1745	2777

Table 2. Data distribution for Asevo dataset for various emotions. Posed clips refer to the artificially generated clips, while non-posed refer to those captured using the stimulus activation procedure.

sion going from baseline neutral to peak of expressed emotion.

4.4. MMI Dataset

MMI facial expression dataset [35] is an ongoing effort for representing both posed and non posed facial expressions. The dataset has total 2894 video clips out of which 197 have been labeled for six basic emotions. MMI originally contained only posed facial expressions and recently was extended to contain induced happiness, disgust and surprise [47]. Each video clip in MMI contains people going from neutral to peak and then back to neutral facial expression.

5. Experiments and Results

5.1. Video autoencoder

Our first experiment shows qualitatively results of a single video autoencoder. We use $145 \times 145 \times 9$ clips as input, where the spatial resolution received by downsampling all clips to that single size using bspline interpolation, and 9 frames are extracted from the clip by using every third frame. We use *caffe* [17] to train the system. In practice we convert each video clip into a image strip containing con-

secutive frames placed horizontally and use *caffe* "image-data", "split" and "concat" layers for video data input.

We minimize contrastive divergence [5] to train autoencoder layers *successively*. We train the first 4 beginning and end layers by creating an intermediate neural network ($C(96, 11, 3) - N - C(256, 5, 2) - N - DC(256, 5, 2) - N - DC(384, 3, 2)$) and training it on facial video clips. We then train third convolution and deconvolution layer by initializing weights from previously trained neural net and fixing the weights for first 4 beginning and end layers. We fine tune all layers once the neural net weights have converged. We repeat the process for fourth fully connected layer to generate deep features.

Please refer to figure 2 to see results from neural net based reconstruction using different number of layers.

5.2. Multi-velocity video autoencoder

Multi velocity semi-supervised learner comprises of an array of three independent autoencoders and a predictor net. We initialize the autoencoders using the weights from the video autoencoder and add a convolution layer as described in section 3.2. We fine tune the multi-velocity layers by creating 3 datasets containing video clips at different velocities. We achieve that by selecting every third frame to create set 1 ($speed = 3x$), selecting every second frame to generate set 2 ($speed = 2x$) and taking first 9 frames for set 3 ($speed = 1x$). The weights from this step are used for initialization of our multi-velocity predictor which described next.

5.3. Multi-velocity predictor

For training, testing and validation we divide each dataset into 3 parts randomly. We select 50% inputs for training, 30% of dataset for testing and use 20% of dataset for validation. After the dataset was split, we further increased the size of the training dataset by shifting each video along both axes, rotating images and taking their mirror.

We train our proposed semi-supervised learner and the multi-velocity semi-supervised learner on the three datasets

Dataset	MMI	CKPLUS	Asevo
Technique			
External Methods			
<i>Covariance riemann kernel based multiple kernel methods</i>	40.9	79	51.05
<i>Multiple kernel methods with gaussian riemann kernel</i>	40.9	67	46.92
<i>Grassman kernel based multiple kernel approaches</i>	9.09	17.9	44.99
<i>Expressionlets based manifold learning techniques</i>	52.91	82.7	48.6
Our Methods			
<i>Semi-Supervised Learner for gesture classification</i>	59.01	87.36	51.11
<i>Multi-Velocity Encoder based learner for gesture classification</i>	58.7	89.47	52.59

Table 4. Comparison of results from various techniques on CKPLUS, MMI and Asevo datasets. The dataset was divided into 3 parts test, train and val randomly. Training set was 50%, test and validation were 30% and 20% respectively. Our method consistently won for both small and large datasets (winning method is shown in green and the leading method is showed using yellow).

(MMI, CK and Asevo), and compare our results against multiple kernel methods [29] and expression-lets base approaches [28]. We used sources downloaded from *Visual Information Processing and Learning Resources* [43] as a reference to compare to our methods. Note that we made the same data partitioning scheme (train, validation, test) for all methods to show a fair comparison.

We outperform all the methods compared on all the datasets used, by a substantial gap, in almost all cases. We summarize our findings in Table 4, and show confusion matrices per facial expression in Tables 1 and 3. For baseline comparison against other deep neural architectures, we compare our methods against [25] and GoogleNet [44]. We further verified our results against prior state of the art methods discussed in [27] by performing **10 fold cross validation**. On MMI we get 66.15 (vs 63.4) % and on CK+ we get 94.18 (vs 92.4) %, making our method **state of the art** for face expression recognition.

6. Discussion and Future Work

This paper presents a learning strategy for large datasets with a dramatically lower number of labeled points, in addition to new layers carefully designed to improve recognition in multi-velocity setup. We currently trim the videos to the facial window using Viola and Jones face detection, and focus solely on frontal views. Recognition in-the-wild still remains a challenge with a known low success rate. We believe that given a large and rich dataset this problem would be feasible to solve in our system, and we plan to explore that in the future.

We introduced a new layer, which adaptively resamples the videos, achieving a multi-velocity invariant learning procedure. Inserting invariants into a learning process is a research direction that we must push forward. Today training of deep neural network is still time consuming, where huge clusters are being heavily used on reasonably

large datasets. We are already reaching the time-space limit of this process, and better/smarter approaches need to be considered for advancement. Our multi-velocity setup is one approach for reducing the need for data in multiple velocities, while other invariants should be explored in future work.

7. Conclusions

In this paper we introduced a new topology and learning protocol for semi-supervised convolutional neural networks on video sequences. We further developed a multi-velocity layer based on temporal resampling which was tuned as part of the learning procedure on an enormous collected facial dataset. We report state-of-the-art results on our own data and on public available datasets.

Appendix A: Equations for B spline Interpolation

Let $\{x_k, f(x_k)\}_{k=0}^N$ be $N + 1$ observations of a function f . Cubic spline is defined as a set of polynomials $\mathbf{S}_n(x)_{n=0}^{N-1}$ with coefficients $p_{n,i}$ which approximate f as follows

$$\mathbf{S}(x) = \mathbf{S}_n(x) = p_{n,0} + p_{n,1}(x_n - x_k) + p_{n,2}(x_n - x_k)^2 + p_{n,3}(x_n - x_k)^3, \quad (4)$$

where $x_k < x_n < x_{k+1}$. We need at least $4N$ constraints to recover $p_{n,i}$ uniquely. We can generate $4N - 2$ constraints by fixing the values of polynomials at the boundaries and assuming first and second derivatives of adjacent polynomials coincide at the boundaries as well. We add additional constraints assuming that the curve is *natural* [8] and has zero derivative at boundaries. The coefficients $p_{k,n}$ are constrained by:

$$\mathbf{S}(x) = f(x_k) \quad \forall k \in \{0..N\} \quad (5a)$$

$$\mathbf{S}_k(x) - \mathbf{S}_{k+1}(x_{k+1}) = 0 \quad \forall k \in \{0..N-2\} \quad (5b)$$

$$\mathbf{S}'_k(x) - \mathbf{S}'_{k+1}(x_{k+1}) = 0 \quad \forall k \in \{0..N-2\} \quad (5c)$$

$$\mathbf{S}''_k(x) - \mathbf{S}''_{k+1}(x_{k+1}) = 0 \quad \forall k \in \{0..N-2\} \quad (5d)$$

$$\mathbf{S}''_0(x) = \mathbf{S}''_{N-1}(x) = 0 \quad (5e)$$

Let \mathbf{A} denote matrix representing the constraints on spline polynomial coefficients and \bar{p} represent coefficients as described in equation 5. Let \bar{y} be the vector of function values $f(x_k), k \in \{0..N\}$ known to us. Right side of 5 can be written as a product of matrix \mathbf{T} with vector \bar{y} , where \mathbf{T} is a binary matrix. Then $\mathbf{A}\bar{p} = \mathbf{T}\bar{y}$ or $\bar{p} = \mathbf{A}^{-1}\mathbf{T}\bar{y}$.

References

- [1] Z. Abidin and A. Harjoko. A neural network based facial expression recognition using fisherface. *International Journal of Computer Applications*, 59(3):30–34, 2012. 2
- [2] E. André, M. Klesen, P. Gebhard, S. Allen, and T. Rist. Exploiting models of personality and emotions to control the behavior of animated interactive agents. In *Workshop on Achieving Human-Like Behavior in Interactive Animated Agents*, pages 3–7, 2000. 1
- [3] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1859–1866. IEEE, 2014. 6
- [4] Y. Bengio, N. L. Roux, P. Vincent, O. Delalleau, and P. Marcotte. Convex neural networks. In *Advances in Neural Information Processing Systems*, pages 123–130, 2005. 6
- [5] M. A. Carreira-Perpinan and G. E. Hinton. On contrastive divergence learning. In *Proceedings of International Workshop on Artificial Intelligence and Statistics*, pages 33–40, 2005. 5, 7
- [6] E. Cerezo, S. Baldassarri, and F. Seron. Interactive agents for multimodal emotional user interaction. *Multi Conferences on Computer Science and Information Systems*, pages 35–42, 2007. 1
- [7] H. Chen, J. Li, F. Zhang, Y. Li, and H. Wang. 3d model-based continuous emotion recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1836–1845, 2015. 2
- [8] S. Chung-Tze. On spline finite element method. *Mathematica Numerica Sinica, (MNS)*, 1:50–72, 1979. 8
- [9] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE*, 18(1):32–80, 2001. 1
- [10] A. Dhall, A. Asthana, R. Goecke, and T. Gedeon. Emotion recognition using PHOG and LPQ features. In *International Conference on Automatic Face & Gesture Recognition*, pages 878–883. IEEE, 2011. 2
- [11] M. Freed, J. G. Carbonell, G. J. Gordon, J. Hayes, B. A. Myers, D. P. Siewiorek, S. F. Smith, A. Steinfeld, and A. Tomasic. Radar: A personal assistant that learns to reduce email overload. In *AAAI*, pages 1287–1293, 2008. 1
- [12] C. Frith. Role of facial expressions in social interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3453–3458, 2009. 1
- [13] M. Gargesha and P. Kuchi. Facial expression recognition using artificial neural networks. *Artificial Neural Computer Systems*, pages 1–6, 2002. 2
- [14] L. He, D. Jiang, L. Yang, E. Pei, P. Wu, and H. Sahli. Multimodal affective dimension prediction using deep bidirectional long short-term memory recurrent neural networks. In *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, pages 73–80. ACM, 2015. 2
- [15] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. 1, 5
- [16] H. S. Hou and H. Andrews. Cubic splines for image interpolation and digital filtering. *Transactions on Acoustics, Speech and Signal Processing*, 26(6):508–517, 1978. 3
- [17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 7
- [18] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim. Joint fine-tuning in deep neural networks for facial expression recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2983–2991, 2015. 2
- [19] 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. 2
- [20] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, Ç. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, and R. C. Ferrari. Combining modality specific deep neural networks for emotion recognition in video. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 543–550. ACM, 2013. 2
- [21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732. IEEE, 2014. 2, 3
- [22] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014. 2
- [23] T. Klein and W. Picard. Computer response to user frustration. *MIT Media Laboratory Vision and Modelling Group Technical Reports, TR 480*, 1999. 1
- [24] I. Kotsia and I. Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *Transactions on Image Processing*, 16(1):172–187, 2007. 2
- [25] 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. 2, 5, 8
- [26] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, 2013. 2
- [27] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen. Deeply learning deformable facial action parts model for dynamic expression analysis. In *Computer Vision—ACCV 2014*, pages 143–157. Springer, 2014. 8
- [28] M. Liu, S. Shan, R. Wang, and X. Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1749–1756. IEEE, 2014. 2, 6, 8
- [29] M. Liu, R. Wang, S. Li, S. Shan, Z. Huang, and X. Chen. Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 494–501. ACM, 2014. 2, 7, 8
- [30] P. Liu, S. Han, Z. Meng, and Y. Tong. Facial expression recognition via a boosted deep belief network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1805–1812. IEEE, 2014. 2
- [31] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 94–101. IEEE, 2010. 6
- [32] A. Mehrabian and J. A. Russell. *An approach to environmental psychology*. the MIT Press, 1974. 1
- [33] P. Michel and R. El Kaliouby. Real time facial expression recognition in video using support vector machines. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 258–264. ACM, 2003. 2
- [34] T. L. Nwe, S. W. Foo, and L. C. De Silva. Speech emotion recognition using Hidden Markov Models. *Speech communication*, 41(4):603–623, 2003. 2
- [35] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *International Conference on Multimedia and Expo*, pages 5–pp. IEEE, 2005. 7
- [36] L. Presti and M. Cascia. Using hankel matrices for dynamics-based facial emotion recognition and pain detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 26–33, 2015. 2
- [37] M. A. Ranzato, J. Susskind, V. Mnih, and G. Hinton. On deep generative models with applications to recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2857–2864. IEEE, 2011. 1
- [38] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1692. IEEE, 2014. 2
- [39] B. Schuller, G. Rigoll, and M. Lang. Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–577. IEEE, 2004. 2
- [40] T. Senechal, D. McDuff, and R. Kaliouby. Facial action unit detection using active learning and an efficient non-linear kernel approximation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18, 2015. 2
- [41] C. Shan, S. Gong, and P. W. McOwan. Robust facial expression recognition using local binary patterns. In *International Conference on Image Processing*, volume 2, pages II–370. IEEE, 2005. 2
- [42] D. Siewiorek, A. Smailagic, and T. Starner. Application design for wearable computing. *Synthesis Lectures on Mobile and Pervasive Computing*, 3(1):1–66, 2008. 1
- [43] Sources. Visual information processing and learning. [Online; accessed 10-July-2015]. 8
- [44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 8
- [45] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708. IEEE, 2014. 1
- [46] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767*, 2014. 2
- [47] M. Valstar and M. Pantic. Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In *Workshop on EMOTION: Corpora for Research on Emotion and Affect*, page 65, 2010. 7
- [48] R.-L. Vieriu, S. Tulyakov, S. Semeniuta, E. Sangineto, and N. Sebe. Facial expression recognition under a wide range of head poses. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–7. IEEE, 2015. 2
- [49] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 2
- [50] R. Walecki, O. Rudovic, V. Pavlovic, and M. Pantic. Variable-state latent conditional random fields for facial expression recognition and action unit detection. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–8. IEEE, 2015. 2
- [51] Z. Wang, S. Wang, and Q. Ji. Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3422–3429, 2013. 2
- [52] J. Weston, F. Ratle, H. Mobahi, and R. Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. 2
- [53] L. Yang, H. Cheng, J. Hao, Y. Ji, and Y. Kuang. A survey on media interaction in social robotics. In *Advances in*

Multimedia Information Processing (PCM), pages 181–190. Springer, 2015. [1](#)

- [54] K. Zhao, W.-S. Chu, F. De la Torre, J. F. Cohn, and H. Zhang. Joint patch and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2207–2216, 2015. [2](#)