Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Facial Expression Detection in Videos using Convolutional Neural Networks

Seminar
Practical Video Analyses
Summer term 2015

Tom Bocklisch, Tom Herold, Norman Rzepka, Thomas Werkmeister

Supervisors:
Dr. Haojin Yang
Prof. Dr. Christoph Meinel

August 16, 2015

Contents

Automatic recognition of facial expressions in videos has a wide range of use cases. Given a video it can be used to find important segments in a video. Furthermore, it is possible to extract sentiments of individuals about statements or questions being posed. In this technical documentation we describe a two-stream neural network architecture for the recognition of facial expressions according to the Facial Action Coding System (FACS). The two-stream architecture leverages both temporal and spatial features of videos and joins them to predict the existence of facial expressions.

Keywords. Neural Networks. Machine Learning. Facial Expressions. Multimedia Retrieval.

1 Research Task

The goal of this research project is to apply state of the art deep neural networks to facial expression recognition for videos of human faces. To this end we make use of the Facial Action Coding System [2] as the primary means of identifying facial expressions.

1.1 FACS Label

The human face is home to a multitude of individual muscles or groups of muscles. Ekman et al. introduced the Facial Action Coding System to taxonomize human facial movements by their appearance on the face. The system describes anatomically possible facial expression in terms of independent action units (AU), e.g. "Outer Brow Raiser", "Lower Lip Depressor", "Jaw Clencher". The action units have a unique identifier and can be combined with an intensity attribute ranging from only a slight visibility to maximum pronunciation of a feature.

Action units can be used for higher order recognition tasks. Facial emotions are a combination of independent action units. Not only does this allow for automatic emotion identification but also for disambiguation of sincere and fake emotions. A prominent example is the smiling emotion. A sincere and involuntary smile raises the outer lip and contracts the muscles around the eye, a feature not present in an insincere and forced smile.

2 Data Set

An annotated data set can help to learn how to differentiate between distinct facial movements. For this purpose we make use of a FACS-labeled corpus of videos. In this section we introduce the underlying data set for our study and necessary pre-processing steps.

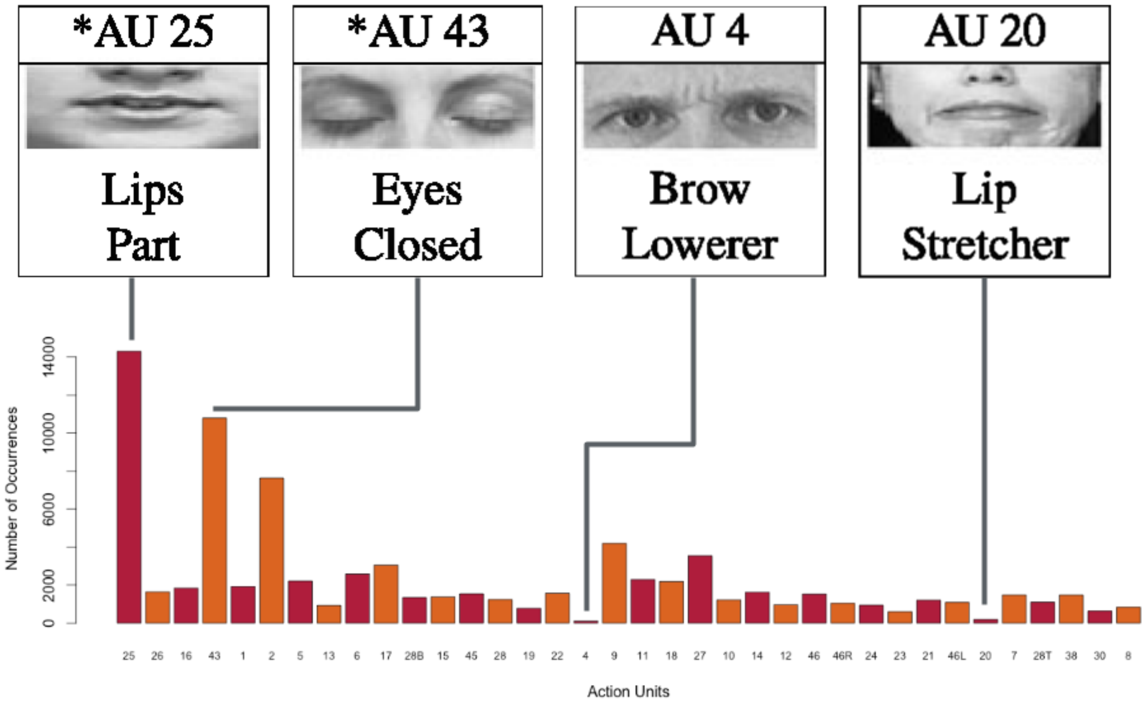
2.1 MMI

The MMI data set [11] consists of 2392 short video clips of 89 different subjects. It features nine different emotions, such as joy, anger and fear. It also contains 37 FACS-labels, such as "eyes closed", "brow lowerer" and "lip stretcher". For our study we focused on learning the FACS-labels, since emotions are simply a combination of multiple activated FACS. Out of the 2392 video clips only 328 are annotated on a per frame basis. For the others the annotations merely indicate the existence of a FACS-label at some point in the video clip. To correctly learn the FACS-labels we made use of the 328 videos that were annotated on a per frame basis.

2.2 Pre-processing

We introduced pre-processing steps on the original data to bootstrap more training data and to bring the videos in the right format for later consumption by the neural networks. Since

Figure 1: Distribution of the FACS-labels



we will be fine-tuning existing neural networks trained on ImageNet [1] we need to set up our data to have similar properties as the original network's training set.

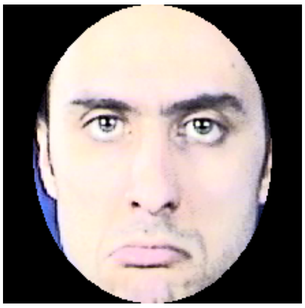
2.2.1 Bootstrapping

Given the small amount of available data we introduced two methods to generate more training data. First, we tilted the images in -6°, -3°, +3°, +6°angles, thus making the learned model more resilient to different head positions. Secondly, we duplicated images altering their brightness by fixed values. Finally, 56553 annotated frames were contained in the data set after augmentation. The distribution of the FACS-labels in the data set is non-uniform and heavily biased towards certain labels as shown in figure 1.

2.2.2 Recognizing faces

We used facial recognition to extract the relevant parts of the video frames. The facial recognition is performed using OpenCV's face detection algorithm that employs Haar feature-based cascades [14]. We cut out the front part of the face using an ellipse to mask the surroundings. Only the area of forehead, eyes, nose and mouth remain visible. Ears, hair and throat are concealed. This process is illustrated in figure 2.

Figure 2: An elliptic overlay is used to reduce noise around the face



2.2.3 Calculating means

To normalize the frames we use a two-pass approach. The first pass calculates the global means of all pixels for the different channels, such as RGB or the flow channels. The second pass subtracts the global mean from every frame per channel.

2.3 Bi-directional Optical Flow

A dense optical flow is calculated using the Gunnar Farneback's algorithm [3] from each frame to its successor. For each frame the five preceding and five succeeding flows are stacked to form a frame with ten channels. Those frames are fed into the flow network downstream which we will explain in section 3.5.

3 Network Architecture

In this section we propose a two-stream deep learning architecture for subject independent facial expression recognition on the MMI data set. For our implementation we used the caffe framework written in C++ [6]. In different experiments we trained networks that are based on one-vs-all classification and on direct multi-label classification.

We will analyze and explain the underlying network architectures that we used for the different experiments. Furthermore, we will describe the different network layouts and their advantages. We will start with an analysis of the multi-label classification problem.

3.1 Multi-Label classification

Multi-label learning is a form of supervised learning where the classification algorithm is required to learn from a set of instances. In contrast to multi-class learning an instance can hereby belong to multiple classes. Therefore, multi-label learning is a generalization of the multi-class problem. In our case each instance can be labeled by multiple FACS-labels. This is quite intuitive, since we can observe several muscle activities like "lowered brow" and "lips parting" at the same time. The advantage of multi-label classification compared to multiple single label classifications is the use of inter-dependencies between the labels as described by Zhang et al. [16].

Although there exist several applications of multi-label classification (e.g. automatic text-categorization [15]) the existing algorithms are not as established as the ones for multi-class classification. Therefore, many approaches of tackling the multi-label challenge reside on the approach of dividing the problem into multiple multi-class problems. In the following paragraphs we will give a short overview over the existing approaches and their application to facial expression recognition.

3.1.1 One-vs-One

In a *One-vs-One* classification $\frac{k\cdot(k-1)}{2}$ classifiers need to be trained (assuming symmetric models). During classification and evaluation a scoring system is used to combine the results of the different classifiers. Due to the amount of classifiers needed this is unfeasible for multi-class problems with a larger number of classes. Therefore, we decided to use a *One-vs-All* approach for the 37 FACS-label classification task.

3.1.2 One-vs-All

Training a model that learns to distinguish between one class and all the other classes is called *One-vs-All* classification. This reduces the classification task to a binary classification. It results in a total of k classifier which is still reasonable given our amount of classes. But it should be noted that training many complex models still leads to long training durations.

Both, One-vs-One and One-vs-All, need to be calibrated. A straightforward approach is to define 0.5 as a threshold for a label to belong to an example. More advanced approaches, e.g. using a ROC-AUC analysis, can be used to improve this initial guess. For our analysis we used a fixed threshold of 0.5.

3.1.3 BP-MLL

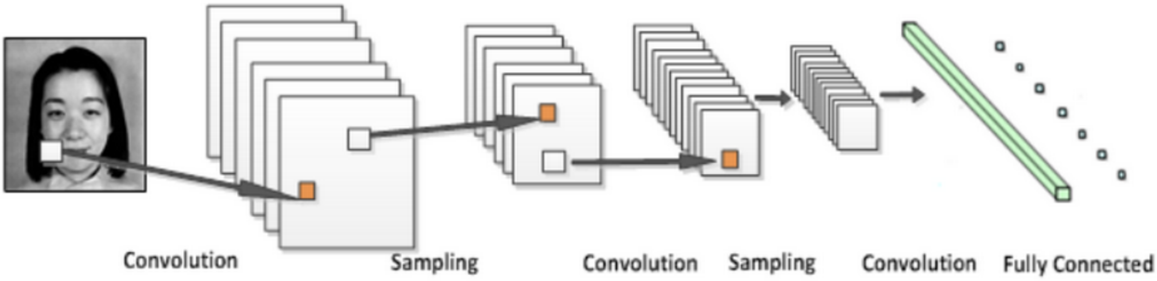
Instead of reducing the multi-label problem to a binary classification task there is also a variation of backpropagation for neural networks that supports multi-label classification (*BP-MLL*) proposed by Zhang [17]. Due to the nonexistent implementation in caffe we could not try this. Instead we relied on normal backpropagation with a sigmoid cross entropy loss function proposed by Nam et al. [9].

3.2 Convolutional Neutral Networks

Convolutional Neural Networks (CNN) are variants of Multilayer Perceptrons and are inspired from biology [5]. The network's main components, *Convolutional layers* and *Pooling layers*, are placed alternatingly in the hierarchical network architecture as seen in figure 3.

CNNs take advantage of local correlations in the data and employ local connectivity patterns between neighboring layers. For facial expression recognition this is useful, since the network is able to abstract over specific locations and variations of facial key points. Nevertheless, this comes with a downside which we will discuss further in section 4.

Figure 3: Convolution Neural Network Architecture based on Neagoe et al. [10]



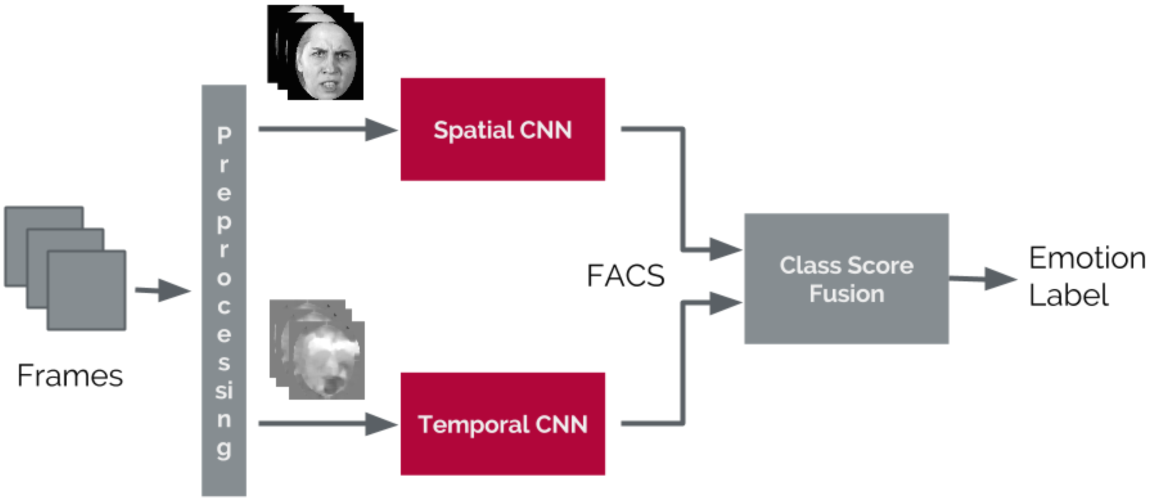
Convolutional layers provide the abstraction of learning features which are location independent, e.g. a neuron will produce the same activation for a learned feature independently of the location of the data in the input. It can be specified using the kernel size, the stride and the number of features. The feature map for a single feature k can be obtained by convolution between the linear filter W^i and the input data x augmented with a bias term b_k. To be able to represent non-linear functions an additional transformation is applied to the result. As a transformation we will employ a rectified linear unit (ReLU) as suggested by Nair and Hinton [8]. Therefore, for the k-th feature map M^k this results in:

$$M^k_{j,i} = ReLU((W^k \cdot x)_{i,j} + b_k)$$

3.3 Two-Stream Architecture

Simonyan et al. [12] introduced a two-stream network architecture to improve their prediction result in activity recognition. The approach is two-fold: One deep CNN is trained on a sequence of single, still frames to recognize spatial features. A second deep CNN is employed to recognize motion in the form of dense-optical flow and provides additional insights into action recognition. The resulting predictions are fused into a single overall output. The complete system can be seen in figure 4. Decoupling the spatial and temporal nets allows us to exploit the availability of large amounts of annotated image data for object recognition. One example is ImageNet [1], a huge collection of images suited for object recognition.

Figure 4: Two-stream architecture



3.4 Spatial Stream

The spatial network operates on individual video frames, effectively performing a still image recognition. The static information obtained by the spatial stream gives clues about which group of muscles are activated. Building on the success of other image recognition networks, we can base our work on other large image recognition data sets. We based our networks on the AlexNet [7] network from the ImageNet competition. During training we used the existing weights and performed fine-tuning to adapt the network to our use case. For One-vs-All classification we fine-tuned one AlexNet per label (we will call this setup *One vs all*). During our direct approach we fine-tuned a single AlexNet to recognize all our labels (*AN-Finetune*). The architecture of that net is shown in table 1. The One-vs-All networks are based on the same architecture except the last fully-connected (FC) layer having an output shape of $2x1x1$.

3.5 Flow Stream

After having described the spatial network we now turn to the temporal stream. To acknowledge a video's information change over time, the temporal stream is trained on stacked dense optical flow images. The construction of those flow stacks follows the explanations in section 2.3. Due to the special nature of the input data the network needs to be trained from scratch. Therefore we could neither use a One-vs-One nor a One-vs-All approach for this part of the network due to time constraints. We trained a single network on all labels as described in section 3.1.3.

Table 1: CNN architecture for AN-Finetune. Based on AlexNet. Only layers marked with * get trained during the fine-tuning.

Layer	Type	Output Neurons	Kernel / Pooling size	Additional notes
1	data	1 x 227 x 227	-	
2	conv + ReLU + LRN	96 x 55 x 55	11 x 11	
3	pool	96 x 27 x 27	3 x 3	max pooling
4	conv + ReLU + LRN	256 x 27 x 27	5 x 5	
5	pool	256 x 13 x 13	3 x 3	max pooling
6	conv + ReLU	384 x 5 x 5	3 x 3	
7	conv + ReLU	384 x 5 x 5	3 x 3	
8	conv + ReLU	256 x 5 x 5	3 x 3	
9	pool	256 x 2 x 2	3 x 3	max pooling
10	FC + ReLU + DO	4096 x 1 x 1		0.9 dropout
11 *	FC + ReLU + DO	512 x 1 x 1		0.9 dropout
12 *	FC	37 x 1 x 1		

4 Evaluation

In this section we will evaluate the performance of our prediction task. We measured several different system configurations. *(i)* A multi-label spatial network fine-tuned on AlexNet, *(ii)* 37 individual spatial binary-classification networks in a one-vs-all setting, *(iii)* a multi-label temporal network and *(iv)* the complete system with fusion of spatial and temporal network.

The resulting prediction results leave room to be improved upon. We have identified a number of reasons for this. First and foremost, the MMI data set with per-frame annotation is too small. Even with data augmentation our data set contains under 60000 images. Although a typical video spans 100 to 200 frames there are usually less than ten annotations per video further limiting the information density. Additionally, the label annotation distribution is very skewed. As seen in figure 1 a small number of labels are predominant.

Furthermore, there are architectures that are theoretically better suited to solve the classification tasks. As mentioned already, convolutional layers develop location independent features. However, for our task the location of the muscles is fixed and due to the preprocessing we know their location. Therefore, locally connected layers as described by Taigman et al. [13] would suite the task better. Using them would mean that we can not fine-tune anymore which would have resulted in unfeasible training durations for the scope of this project.

4.1 Performance

We evaluated our classifiers on a test split of the data set and calculated several multi-label performance metrics [4].

Figure 5 shows the precision of each classifier on all labels. We can see that labels which

Figure 5: Precision (per label)

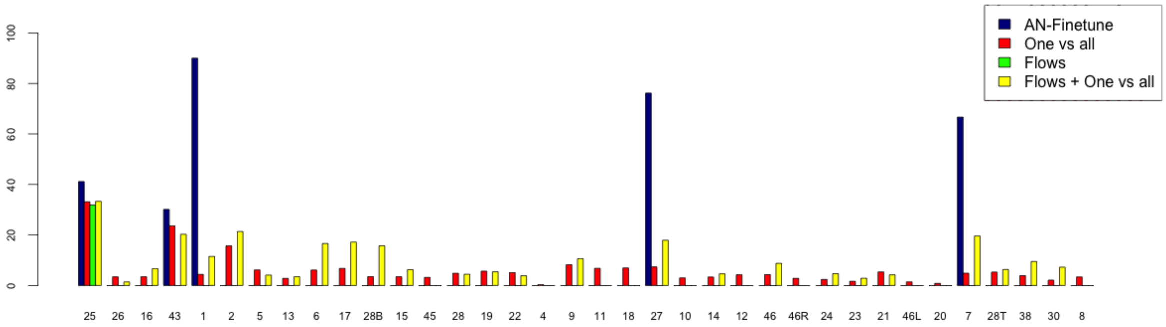
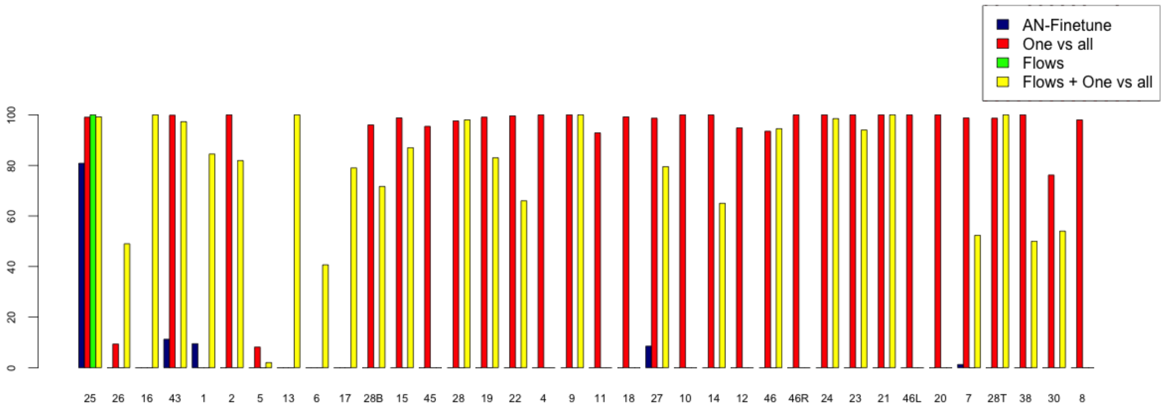


Figure 6: Recall (per label)



are more frequent in the training data set tend to have better precision metrics. Especially the *AN-Finetune* classifier has strong precision values for labels that are more frequent in the data set.

In figure 6 it is very obvious that our classifiers have been trained for high recall. Especially the two *One-vs-all* classifiers show high recall values for all labels. This is a result of them being trained on every label individually.

The summary statistic in figure 7 shows that the multi-label accuracy (i.e. Hamming score [4]) of all classifiers is rather poor. Again, we can see that the recall of the *One-vs-all* classifiers is very high. However, the other classifiers show a better average precision which comes as no surprise given the data observed in figure 5.

The unspectacular performance of the classifiers can be attributed to the small data set size. The data augmentation did not improve the performance significantly. Also, the classification task at hand is very difficult given that some labels only represent very subtle feature intensity differences in the images (see Figure 8).

Figure 7: Summary metrics: Average precision, average recall, multi-label accuracy

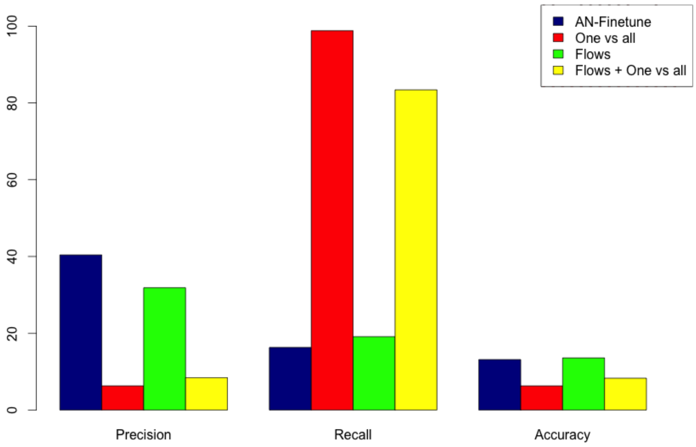
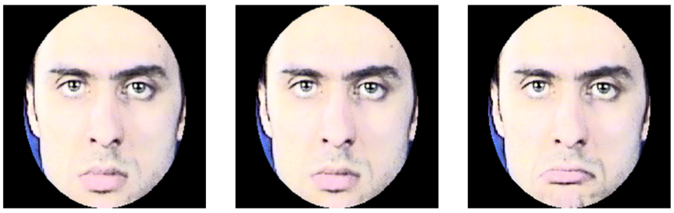


Figure 8: Three example images of the same label with different intensities

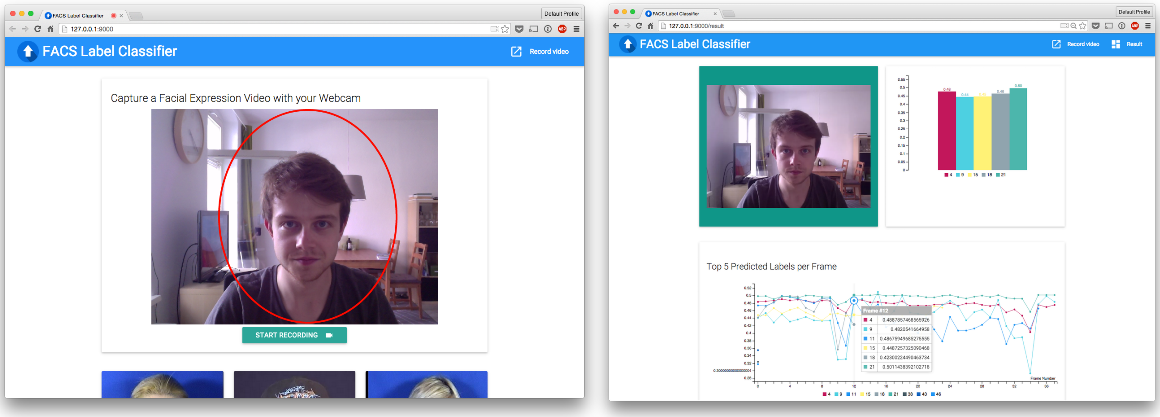


5 Web Demo

A web-based interface serves as an easily accessible interface to our system. Users are given the opportunity to record themselves using their computer's webcam as input for the classification. Given a modern browser, the demo utilizes WebRTC to record and ensemble a complete video file in the user's client. Alternatively, we provide example videos from the MMI dataset for quick access.
The uploaded video material undergoes the same preprocessing as outlined in section 2.2. The extracted frames and optical flows serve as the input for the spatial and temporal network, respectively.

The joint prediction results are returned by means of a REST interface and are presented two-fold as shown in figure 9: *(1)* Next to a preview of the video is the overall classification summary given by the top five label probabilities. *(2)* To reflect a video's temporal dimension the demo presents a per-frame based evaluation highlighting the top five probabilities for each frame. For easy verification a click on the graphical nodes synchronizes the preview video with the selected frame.

Figure 9: (left) Record yourself in web demo (right) Overview of prediction results



Frame #12	
4	0.4887857468565926
9	0.4820541664958
11	0.48675949685275555
15	0.4487257325090468
18	0.42300224490463734
21	0.5011438392102718

References

[1] A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge (ilsvrc), 2010. *URL http://www. image-net. org/challenges/LSVRC*, 2010.

[2] P. Ekman and W. V. Friesen. Facial action coding system. 1977.

[3] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. Springer, 2003.

[4] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Advances in Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

[5] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

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

[9] J. Nam, J. Kim, I. Gurevych, and J. Fürnkranz. Large-scale multi-label text classification - revisiting neural networks. *CoRR*, abs/1312.5419, 2013.

[10] V.-E. Neagoe, B. Andrei-Petru, N. Sebe, and P. Robitu. A deep learning approach for subject independent emotion recognition from facial expressions. *Recent Advances in Image, Audio and Signal Processing*, pages 93–98, 2013.

[11] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, page 5, 2005.

[12] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[13] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708. IEEE, 2014.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[15] B. Yang, J.-T. Sun, T. Wang, and Z. Chen. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 917–926. ACM, 2009.

[16] M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 999–1008, New York, NY, USA, 2010. ACM.

[17] M.-L. Zhang and Z.-H. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. on Knowl. and Data Eng.*, 18(10):1338–1351, 2006.