Recurrent Neural Networks for Emotion Recognition in Video

Samira Ebrahimi Kahou
École Polytechnique de Montréal, Canada
samira.ebrahimi-kahou@polymtl.ca

Vincent Michalski
Université de Montréal, Montréal, Canada
vincent.michalski@umontreal.ca

Kishore Konda
Goethe-Universität Frankfurt, Germany
konda.kishorereddy@gmail.com

Roland Memisevic
Université de Montréal, Montréal, Canada
roland.memisevic@umontreal.ca

Christopher Pal
École Polytechnique de Montréal, Canada
christopher.pal@polymtl.ca

ABSTRACT

Deep learning based approaches to facial analysis and video analysis have recently demonstrated high performance on a variety of key tasks such as face recognition, emotion recognition and activity recognition. In the case of video, information often must be aggregated across a variable length sequence of frames to produce a classification result. Prior work using convolutional neural networks (CNNs) for emotion recognition in video has relied on temporal averaging and pooling operations reminiscent of widely used approaches for the spatial aggregation of information. Recurrent neural networks (RNNs) have seen an explosion of recent interest as they yield state-of-the-art performance on a variety of sequence analysis tasks. RNNs provide an attractive framework for propagating information over a sequence using a continuous valued hidden layer representation. In this work we present a complete system for the 2015 Emotion Recognition in the Wild (EmotiW) Challenge. We focus our presentation and experimental analysis on a hybrid CNN-RNN architecture for facial expression analysis that can outperform a previously applied CNN approach using temporal averaging for aggregation.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Models, Applications

Keywords

emotion recognition; deep learning; multimodal learning; model combination; recurrent neural networks

ICMI 2015, November 9–13, 2015, Seattle, WA, USA.

ACM 978-1-4503-3912-4/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2818346.2830596 .

1. INTRODUCTION

Human emotion analysis is a challenging machine learning task with a wide range of applications in human-computer interaction, e-learning, health care, advertising and gaming. Emotion analysis is particularly challenging as multiple input modalities, both visual and auditory, play an important role in understanding it. Given a video sequence with a human subject, some of the important cues which help to understand the user's emotion are facial expressions, movements and activities. In some cases speech or high level scene context can also be useful to infer emotion. Most of the time there is a considerable overlap between emotion classes making it a challenging classification task. In this paper we present a deep learning based approach to modeling different input modalities and to combining them in order to infer emotion labels from a given video sequence.

The Emotion recognition in the wild (EmotiW 2015) challenge [9] is an extension of a similar challenge held in 2014 [8]. The task is to predict one of seven emotion labels: angry, disgust, fear, happy, sad, surprise and neutral. The dataset used in the challenge is the Acted Facial Expressions in the Wild (AFEW) 5.0 dataset, which contains short video clips extracted from Hollywood movies. The video clips present emotions with a high degree of variation, e.g. actor identity, age, pose and lighting conditions. The dataset contains 723 videos for training, 383 for validation and 539 test clips.

Traditional approaches to emotion recognition were based on hand-engineered features [17, 28]. With the availability of big datasets, deep learning has emerged as a general approach to machine learning yielding state-of-the-art results in many computer vision and natural language processing tasks [22, 19]. The basic principle of deep learning is to learn hierarchical representations of input data such that the learned representations improve classification performance.

The primary contribution of this work is to model the spatio-temporal evolution of facial expressions of a person in a video using a Recurrent Neural Network (RNN) combined with a Convolutional Neural Network (CNN) in an underlying CNN-RNN architecture. In addition to this, we also employed an Autoencoder based activity recognition pipeline for modelling user activity and a simple Support Vector Machine (SVM) based approach over energy and spectral features for audio. We also present a neural network-based

feature level fusion technique to combine different modalities for the final emotion prediction for a short video clip.

Previous work [18, 25] has achieved state-of-the-art results in the emotion recognition challenge using deep learning techniques which includes our work that won the 2013 EmotiW challenge. In contrast to [18, 16], which use an averaging-based aggregation method for visual features in video, here we employ an RNN to model the temporal evolution of facial features in video. We also explore feature-level fusion of our modality-specific models and show that this increases performance.

The remainder of this paper is organized as follows. In Section 2, 3 and 4 we describe each of the models used for different modalities followed by Section 5, which provides details on the fusion methods we applied. Section 6 presents our experimental results and provides a list of our submissions to the challenge. Finally, in Section 7 we draw some conclusions from our experiments.

2. SPATIO-TEMPORAL EVOLUTION OF FACIAL EXPRESSIONS

Modelling the spatio-temporal evolution of visual information plays an important role in understanding the behavior of objects and users in video. Emotion recognition is one of the tasks which involve modelling the behavior of a user. In this work, we use a two step approach to modelling emotion as the spatio-temporal evolution of image structure. In the first step, an CNN is trained to classify static images containing emotions. In the second step, we train an RNN on the higher layer representation of the CNN inferred from individual frames to predict a single emotion for the entire video. RNNs have undergone a resurgence of interest due in part to their impressive performance in handwriting and speech recognition [14, 13]. Much of this interest has been driven by the stability of learning achieved by the use of so-called long short term memory (LSTM) units [15]. RNNs have also proven to be powerful methods for other types of sequential data including video [1, 10] and natural language processing [2, 32]. As such we use an RNN structure for learning a model for video level representation and classification. The higher layer representation from the CNN provides structural information of a given frame and the RNN models the spatio-temporal evolution of the structure over time.

Unlike other work involving video and RNN techniques such as [1, 10], we do not use LSTMs. Here we use IRNNs [24] which are composed of rectified linear units (ReLUs) and employ a special initialization strategy based on scaled variations of the identity matrix. These elements of IRNNs are aimed at providing a much simpler mechanism for dealing with the vanishing and exploding gradient problem compared to the more complex LSTM framework. Recent work has compared IRNNs with LSTMs and found that IRNNs are able to yield comparable results in some tasks, including problems which involve long term dependencies [24].

We provide a detailed explanation of the CNN structure in Section 2.1 and of the RNN in Section 2.2. To compare with the non-sequential approach presented in [16], we also aggregated CNN features to a fixed-length feature vector and trained an SVM. This is described in Section 2.3.

2.1 Frame feature extraction using an CNN

The competition dataset has one emotion label per video which does not correspond to every frame. This introduces a lot of noise if the video labels are used as targets for training an CNN on individual frames. Our visual features are therefore provided by an CNN trained on a combination of two additional emotion datasets of static images. Moreover, using additional data covers a larger variety in age and identity in contrast to the challenge data where the same actor/actress might appear in multiple clips.

2.1.1 Datasets

The additional datasets used in the CNN training consists of two large emotion datasets, namely the Toronto Face Database (TFD) [31] with 4,178 images and the Facial Expression Recognition dataset (FER2013) [6] containing 35,887 images, both with seven basic expressions: angry, disgust, fear, happy, sad, surprise and neutral.

2.1.2 Pre-processing

To account for varying lighting conditions (in particular, across datasets) we applied histogram equalization. We used the aligned faces provided by the organizers to extract features from the CNN. The alignment involves a combined facial keypoints detection and tracking approach explained in [7]. We shall refer to this dataset as AFEW-faces. Different face detection and/or alignment techniques have been used for FER2013, TFD and AFEW-faces. In order to be able to leverage the additional datasets, we re-aligned all datasets to FER2013 using the following procedure:

1. We detected five facial keypoints for all images in the FER2013, TFD and AFEW-faces training set using the convolutional neural network cascade method in [30].

2. For each dataset we computed the mean shape by averaging the coordinates of keypoints.

3. Datasets have been mapped to FER2013 by using a similarity transformation between mean shapes. By computing one transformation per dataset we let the eyes, nose and mouth be roughly in the same location retaining a slight amount of variation. We added a noisy border for TFD and AFEW-faces as faces were cropped more tightly compared to FER2013.

4. AFEW-faces validation and test sets were mapped using the transformation inferred on the training set.

We also performed dataset normalization with the mean and standard deviation image from the combined FER2013 and TFD (FER+TFD).

2.1.3 CNN Architecture

We trained various CNN architectures on FER+TFD without using any challenge data for gradient computations. For early stopping we tried both leaving out 1000 samples of FER+TFD and the challenge data. We observed that the RNN yields slightly better performance when CNN early stopping was done on the challenge data as this avoids overfitting to FER+TFD. Therefore, for our best CNN structure, we trained on all FER+TFD and performed early stopping on AFEW-faces train+validation.

We have explored three main CNN structures:

- a very deep structure with small 3x3 filter size [26, 29],
- a three-layer CNN with 5x5 filters [21, 22] and
- a similar three-layer CNN with 9x9 filter size.

The CNN is trained mainly for feature extraction and we have only used the additional dataset for the training phase. Therefore, we searched for a structure that better generalizes to other datasets. Deep structures are known to learn representations that better generalize to other datasets [29]. However, we observed that the very deep structure quickly over-fitted to FER+TFD, and generalized badly to the challenge dataset. This could be due to the relatively small amount of labeled data available for the emotion recognition task here. For this reason we have tried a shallower network with three layers which appears to have moderately addressed the over-fitting problem. Finally, we increased the filter size from 5 to 9 and reduced the number of filters from 64-64-128 to 32-32-64. For all of the experiments we used data augmentation (horizontal flipping with probability of 0.5 and random cropping), as well as dropout (with rate 0.25).

2.2 Learning Sequences Using an RNN

We use an RNN to aggregate frame features for the following reasons:

- The temporal order of frames is respected in contrast to bag-of-features approaches.
- An RNN has the ability to learn to detect an event, such as the presence of a particular expression, irrespective of the time, at which it occurs in a sequence.
- RNNs naturally deal with a variable number of frames.

RNNs are a type of neural network which transforms a sequence of inputs into a sequence of outputs. At each time-step t, a hidden state $\mathbf{h}_t$ is computed based on the hidden state at time $t-1$ and the input $\mathbf{x}_t$ at time t

$$\mathbf{h}_t = \sigma(\mathbf{W}_{in}\mathbf{x}_t + \mathbf{W}_{rec}\mathbf{h}_{t-1}), \quad (1)$$

where $\mathbf{W}_{in}$ is the input weight matrix, $\mathbf{W}_{rec}$ is the recurrent matrix and σ is the hidden activation function. Each time-step also computes outputs, based on the current hidden state:

$$\mathbf{y}_t = f(\mathbf{W}_{out}\mathbf{h}_t), \quad (2)$$

where $\mathbf{W}_{out}$ is the output weight matrix and f is the output activation function. An example of an RNN in which only the last time-step produces an output is shown in Figure 1.

We use the IRNN, which as discussed above is a simple RNN with rectified linear hidden units (ReLUs) and with a recurrent matrix, that is initialized with scaled variations of the identity matrix [24]. The identity initialization trick ensures good gradient flow at the beginning of training and it allows us to train it on relatively long sequences.

We train the IRNN to classify a video by feeding the features for each frame from the CNN sequentially to the network and using the last time-step softmax output as class prediction. We used Stochastic Gradient Descent (SGD) with a learning rate of 0.005, gradient clipping at 1.0 and a batchsize of 64 sequences. We experimented by using different layers of the CNN as input features and chose the output of the second convolutional layer after max pooling, as this performed best on validation data.

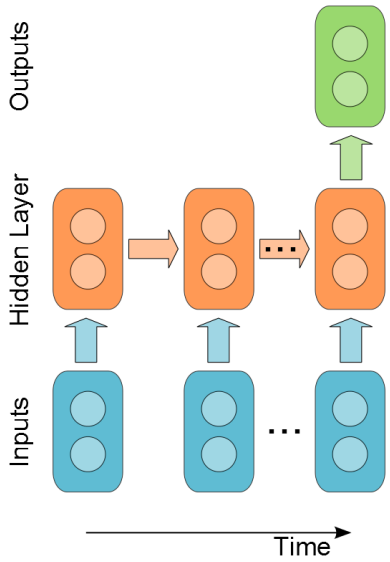


Figure 1: Structure of our recurrent neural network.

2.3 Aggregated CNN Features

As an alternative way of aggregating the frame level structural representations from the CNN, we employed k-average pooling together with an SVM for classification as in [16]. In this approach the per-frame CNN features are averaged into bins to generate a fixed length vector of size k as video representation. Heuristically, we selected $k = 15$ and we used the pre-softmax outputs of the CNN as per-frame features. For videos with a number of frames less than k the frames are locally repeated until sequence length is k.

The vector representations of videos together with corresponding emotion labels are used to train an RBF-kernel SVM. The hyper-parameters of the SVM are set via grid search. As shown in Table 1, the RNN achieves a validation accuracy of 39.6%, which is significantly higher than the aggregated CNN. Simple averaging of the per-frame probabilities yielded a validation accuracy of only 23.7%.

3. AUDIO

Given that the primary focus of this work is on vision based emotion recognition, we simply used the audio features employed in [7] for the audio channel of the video clips. These are based on the approach from [27]. It uses 1582 features extracted with the open-source Emotion and Affect Recognition (openEAR) [12] toolkit which uses openSMILE [11] as backend.

The toolkit encapsulates multiple low level audio feature descriptors (LLDs) and different functionals to apply on them. The feature set consists of 34 energy and spectral related LLDs and 21 functionals, 4 voicing related LLD $\times$ 19 functionals, 34 delta coefficients of energy and spectral LLD $\times$ 21 functionals, 4 delta coefficients of the voicing related LLD $\times$ 19 functionals and 2 voiced/unvoiced durational features.

In this work we used Principal Component Analysis (PCA) based dimensionality reduction as preprocessing on the 1582 dimensional input features and an RBF-kernel SVM for classification. The hyper-parameters for the SVM are set via grid search.

4. ACTIVITY

Spatio-temporal transformations of local image features, or activity, can be an important cue for emotion recognition. A subset of emotions can be represented as changes in facial expressions and in some cases the activity of the entire body of the person. Other approaches, based on vision, described in this work mainly deal with analyzing the emotion in a given video sequence based on static image features and different ways of aggregating them over time. The activity analysis pipeline is the only approach which relies on learning of local spatio-temporal transformations from video.

Our approach for activity analysis is based on the action recognition pipeline from [20, 23] which was also used for emotion recognition previously in [16]. The pipeline mainly consists of three different modules namely, local motion feature extraction, k-means quantization and SVM based classification. A Synchrony Autoencoder (SAE) [20] trained on cropped 3D video blocks of size $16 \times 16 \times 10$ ($space \times space \times time$) is used for local motion feature extraction. Figure 2 shows filters learned by the model on the AFEW 5.0 training set.

5. FUSION

In many discriminative tasks, the fusion of predictions or representations from models trained using different input modalities yields a significant improvement. We use two types of fusion approaches for combining the modality specific models described in previous sections, *feature level* and *decision level* fusion.

5.1 Feature Level

In this approach a combination of intermediate-level representations from the trained models is used as input for training an additional model on the classification task. For feature-level fusion we applied a variant of the regularized feature fusion network from [33]. The feature fusion network is a Multilayer Perceptron (MLP) with separate hidden layers for each modality as shown in Figure 3. The outputs of these layers are concatenated and fed to another hidden layer which is followed by a softmax layer whose number of units is equal to the number of emotion classes. The first layer of the fusion network, consisting of modality specific layers, is regularized to encourage a common representation by sharing similar subsets of hidden units between modalities, while still retaining the discriminative features present in some modalities.

The network is trained with SGD using a learning rate of 0.1 and gradient clipping using clipping threshold 10. The objective function is the categorical cross-entropy between target label and prediction. As input to the fusion network we used aggregated CNN per-frame features, the PCA-whitened audio features and the hidden layer activations of the last time-step of the RNN. We excluded the activity recognition model from the mix, as it tends to overfit its training set. We also explored adding dropout to the hidden layers to prevent over-fitting on the small challenge

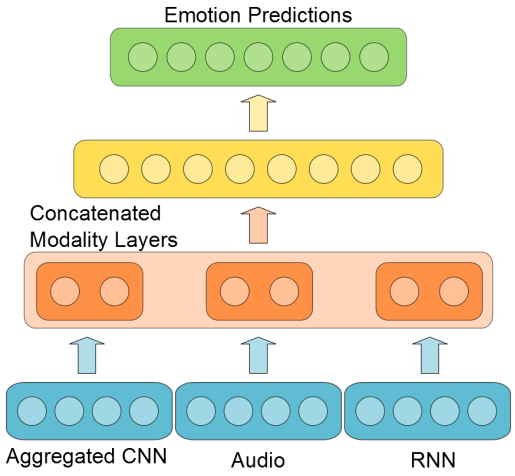


Figure 3: Structure of the feature fusion network.

dataset. The number of hidden layers and their sizes are selected using the validation set. Our best architecture has 100, 10 and 50 units in the aggregated CNN-, the audio- and the RNN-specific hidden layers, respectively. The common hidden layer has 70 units. The search space for determining the optimal size of the modality-specific layers was selected considering the input feature sizes and individual models' performances on the AFEW 5.0 validation set while training on the train set. More details are provided in Section 6.

5.2 Decision Level

For decision-level fusion, i.e. the combination of classifiers, we used a weighted sum of the class probabilities estimated by the modality-specific classifiers and the fusion network. The combined classifier has one weight per modality per class and the resulting score for each class is the weighted sum of all probabilities for the respective class. The combination weights are determined by random search [4], which was also used for model combination in the winning approach for the 2013 EmotiW challenge [18].

Weights are sampled uniformly from $[0.0, 1.0]$ followed by per class re-scaling, so that they sum up to 1. Then the best sampled weights are chosen based on the validation performance. Note that unless noted otherwise, we always use the dataset partition for the random search which was not used for model training, i.e. for models trained on the training set, we perform random search on the validation set and vice versa. After an initial random search with 100,000 iterations, we perform a local random search around the best set of weights found so far. This local random search consists of sampling weights from a Gaussian with mean set to the current best set of weights and standard deviation σ of 0.5. The current best $\tilde{w}$ is updated as soon as a new best is found. After every 100,000 iterations, the σ is decreased by a factor of 0.9 and the local search is stopped when σ is smaller than 0.0001. We also performed uniform local search from $[\tilde{w} - r, \tilde{w} + r]$, where $\tilde{w}$ is the current best set of weights and r is the range in which to search, however it roughly achieved the same performance. We explicitly tried all combinations of subsets of modalities and fusion. Consistently we found that decision level fusion benefited from including all models.

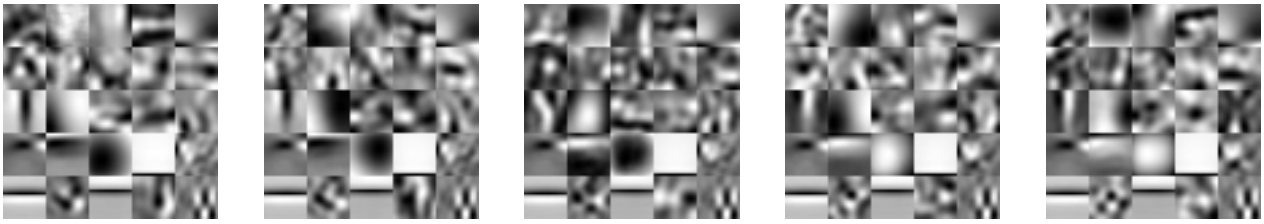


Figure 2: Subset of filters learned by SAE model on the AFEW5 training set. Left to right: Frames 1,3,5,7 and 9.

Table 1: Training and Validation Accuracies for All Modalities (Training on Train partition)

Model	Training	Validation
Activity	0.983	0.266
Audio	0.418	0.332
Aggregated CNN	0.505	0.350
RNN	0.848	**0.396**

6. RESULTS

In this section, we describe our submissions to the EmotiW 2015 challenge. We provide details on per model training strategies and variations of our fusion methods. We also present results and discuss the choices we made in each step.

6.1 Per-model Performance

This work mainly focuses on an RNN approach for visual features. However, given the challenge context we included three further models to achieve competitive performance. Table 1 shows each model's accuracy on the challenge validation set after training on the training set. The corresponding confusion matrices are presented in Figure 4. The matrices show different profiles and strengths for specific emotion classes which is beneficial for combination.

6.2 Feature Level Fusion

As mentioned before, we excluded the activity model from feature level fusion as it tends to over-fit on its training partition. This can be seen in Table 1 where activity has an extremely high discrepancy between training and validation accuracies. The input features to the fusion network are the following:

- The first ten components of the PCA whitened audio features (see Section 3).
- The aggregated CNN features, which are 105-dimensional (7×15 bins) vectors as described in Section 2.3.
- The RNN features, which are the hidden activations of the last time step. These are the only features which have been learned discriminatively on the video level and which therefore contribute strongly to the fusion network. The number of hidden units in the RNN is 200 (see Section 2.2).

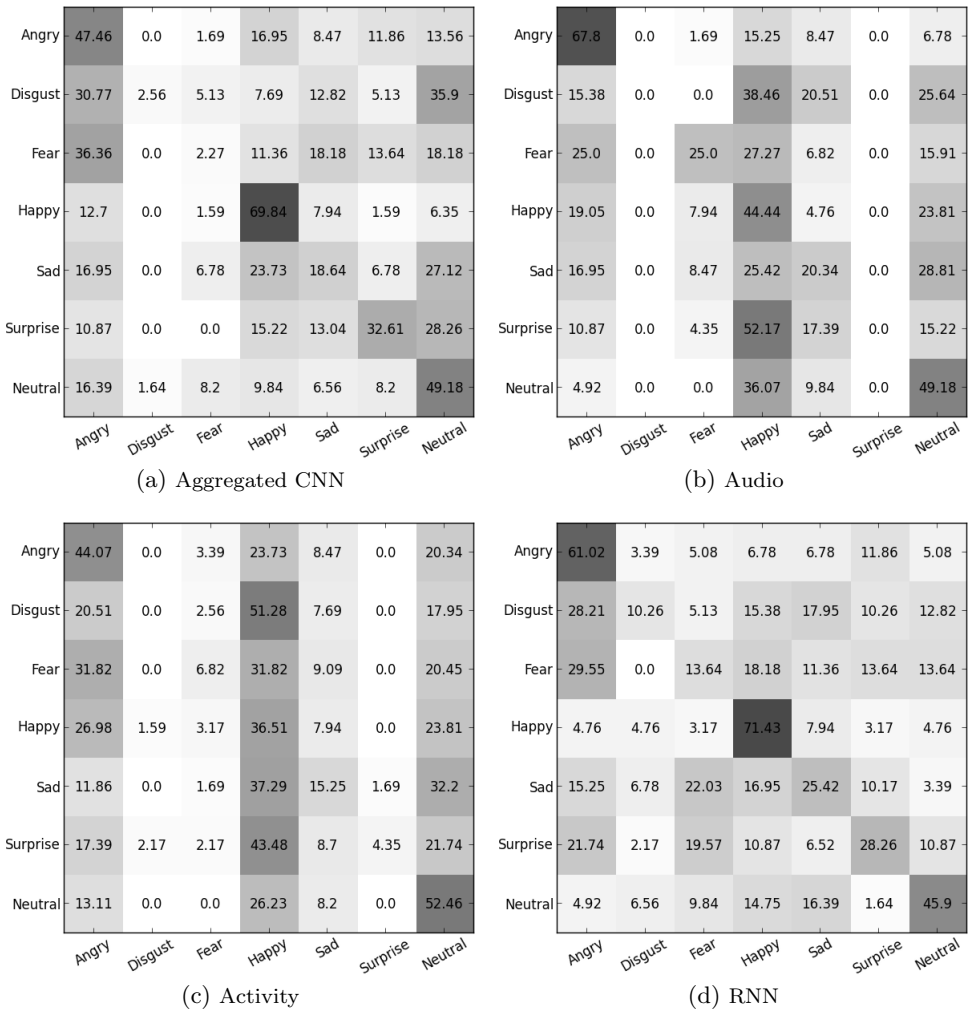
For training the fusion network, we tried replacing the sigmoid activation function of the hidden layers with rectified linear units $ReLU(x) = max(0, x)$ and rectified tanh units $RectTanh(x) = max(0, tanh(x))$. While this improved the validation performance by roughly 2%, it did not yield an improvement on the test performance. One observation during training was that the learning curves were oscillating which made the early stopping unreliable. To stabilize the learning, we lowered the learning rate to 0.001 from 0.1 and added momentum of 0.9. Figure 5 compares two learning curves before and after stabilization. The number of epochs in each sub-figure corresponds to the selected learning rate. Our fusion network achieves a validation accuracy of 43.7%, which is higher than any modality-specific classifier.

6.3 Submissions

Our submissions can be divided into two categories: those which use the training set for training and the validation set for early stopping and random search and those for which the training and validation sets were swapped. For both of these categories, we also submitted a version where models were retrained on the full training plus validation set, retaining all hyper-parameters including early stopping epoch number and combination weights. Note that the models that are CNN-based have also been retrained but not the underlying CNN as we used additional static emotion data for training. For all submissions, random search was done on the data partition that was not used for training the underlying models. For example, if models were trained on the training partition, random search was performed on the validation set. Searching on the same partition that the models were trained on was not an option, as random search would assign high weights to the over-fitters, which would result in poor generalization performance.

Table 2 lists our submissions with their training, validation and test accuracies. In the first category we trained modality-specific models and the fusion network on the challenge training data and validation data was used for early stopping. Then for the final predictions we performed random search on the validation set. This achieved a test set accuracy of 44.341%. With the stabilized fusion network the accuracy improved to 48.979%. Retraining the models with the combined training plus validation set, keeping the hyper-parameters of experiment 2, yielded a test accuracy of 50.463%.

In the second category, with swapped training and validation sets, our initial submission achieved a test accuracy of 50.092%. Here the stabilized fusion did not improve the performance, yielding a test accuracy of 47.680%. The retrained version achieved our best result of 52.875%. Note that for each category we picked the best submission for retraining. Random search as the last step in our pipeline has a big influence on the generalization potential of the whole



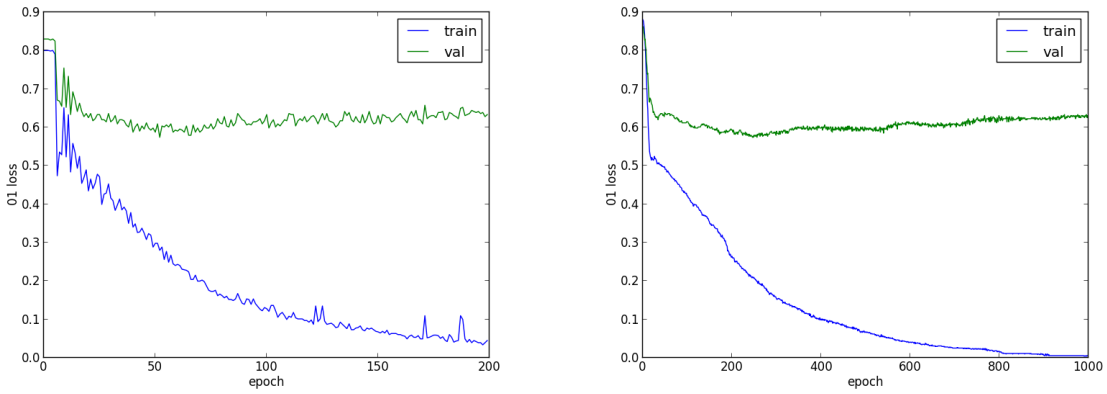
(a) Aggregated CNN (b) Audio

(c) Activity (d) RNN

Figure 4: Confusion matrices on the challenge validation set.

Table 2: Our submissions with training, validation and test accuracies (in percent) for the EmotiW 2015 competition (bold font shows the best accuracy)

Sub	Train	Valid	Test	Method
1	86.216	54.716	44.341	Training on Train, Validation on Valid
2	81.997	54.447	48.979	Training on Train, Validation on Valid, stable fusion
3	-	-	50.463	Training on Train+Val, hyperparams from submission 2, stable fusion
4	52.320	71.967	50.092	Training on Val, Validation on Train
5	52.742	68.463	47.680	Training on Val, Validation on Train, stable fusion
6	-	-	**52.875**	Training on Train+Val, hyperparams from submission 4
7	-	-	49.907	Random Search over combinations of submission 3 and 6 on Train+Val



(a) (b)

Figure 5: Comparison of the learning curves (a) before and (b) after stabilization.

model and likely benefits from the larger training set. This explains the higher performance of the swapped partitions.

Our last submission was an attempt to combine our two best submissions that were retrained on the training plus validation set. We combined those two using the same decision-level fusion strategy as before. The inputs to the random search were the probabilities predicted by the two models. A random search on these two models was performed on the full training plus validation set. The resulting test performance was only 49.907%. This might be explained by the fact that the whole data set has been seen which could have led to over-fitting.

7. CONCLUSIONS

We found that the spatio-temporal evolution of facial features is one of the strongest cues for emotion recognition. We presented the application of an RNN for modelling this spatio-temporal evolution via aggregation of facial features to perform emotion recognition in video. Our experiments in Section 2.3 have shown that this approach outperforms all other modalities, the averaging of per-frame vision-based classifications, and also the more sophisticated aggregation method employed by the 2013 challenge winners [18].

Furthermore, we explore two fusion methods, operating on the feature and on the decision level. Our feature-level fusion network combines features from different modalities and achieves a higher validation accuracy than any of the single-modality classifiers. Our experiments show that feature-level and decision-level fusion are complementary, and when combined they achieve a higher classification accuracy. However, care needs to be taken to prevent over-fitting, either by excluding strong over-fitters, as we did with the activity recognition model in the fusion network, or by using different dataset partitions for combination than for model training, as done in the random search.

We found it difficult to draw conclusions from some of our submission results. This might be caused by the large number of ambiguous cases that exist in this domain. We found that a fairly large number of training videos could be argued to show a mixture of two or more basic emotions (such as a mixture of surprise with fear or happy). This suggests that exploring the use of more than a single label for emotion recognition might be a useful direction for future research.

8. ACKNOWLEDGMENTS

The authors would like to thank the developers of Theano [3, 5]. This work was supported by an NSERC Discovery Award and the German BMBF, project 01GQ0841. We also thank the Canadian Foundation for Innovation (CFI) for support under the Leaders program.

9. REFERENCES

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In A. Salah and B. Lepri, editors, *Human Behavior Understanding*, volume 7065 of *Lecture Notes in Computer Science*, pages 29–39. Springer Berlin Heidelberg, 2011.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[3] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

[4] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305, 2012.

[5] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3. Austin, TX, 2010.

[6] P.-L. Carrier, A. Courville, I. J. Goodfellow, M. Mirza, and Y. Bengio. FER-2013 Face Database. Technical report, 1365, Université de Montréal, 2013.

[7] A. Dhall, R. Goecke, J. Joshi, K. Sikka, and T. Gedeon. Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *Proceedings of*

the 16th International Conference on Multimodal Interaction, ICMI '14, pages 461–466, New York, NY, USA, 2014. ACM.

[8] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Collecting large, richly annotated facial-expression databases from movies. *MultiMedia, IEEE*, 19(3):34–41, July 2012.

[9] A. Dhall, O. V. R. Murthy, R. Goecke, J. Joshi, and T. Gedeon. Video and image based emotion recognition challenges in the wild: Emotiw 2015. In *Proceedings of the 17th ACM on International Conference on Multimodal Interaction*, ICMI '15, 2015.

[10] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. 2014.

[11] F. Eyben, M. Wöllmer, and B. Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the international conference on Multimedia*, pages 1459–1462. ACM, 2010.

[12] F. Eyben, M. Wöllmer, and B. Schuller. openear - introducing the munich open-source emotion and affect recognition toolkit. In *ACII*, pages 576–581, 2009.

[13] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[14] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 545–552, 2009.

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

[16] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, R. Chandias Ferrari, M. Mirza, D. Warde-Farley, A. Courville, P. Vincent, R. Memisevic, C. Pal, and Y. Bengio. Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, pages 1–13, 2015.

[17] S. E. Kahou, P. Froumenty, and C. Pal. Facial expression analysis based on high dimensional binary features. In *ECCV Workshop on Computer Vision with Local Binary Patterns Variants*, Zurich, Switzerland, 2014.

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

[19] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv:1404.2188*, 2014.

[20] K. R. Konda, R. Memisevic, and V. Michalski. Learning to encode motion using spatio-temporal synchrony. In *Proceedings of ICLR*, April 2014.

[21] A. Krizhevsky. Cuda-convnet Google code home page. `https://code.google.com/p/cuda-convnet/`, Aug. 2012.

[22] 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.

[23] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

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

[25] 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*, ICMI '14, pages 494–501, New York, NY, USA, 2014. ACM.

[26] Nagadomi. Github: kaggle-cifar10-torch7. `https://github.com/nagadomi/kaggle-cifar10-torch7/`, 2014.

[27] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic. Avec 2011–the first international audio/visual emotion challenge. In *Affective Computing and Intelligent Interaction*, pages 415–424. Springer, 2011.

[28] C. Shan, S. Gong, and P. W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image Vision Comput.*, 27(6):803–816, May 2009.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[30] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 3476–3483, Washington, DC, USA, 2013. IEEE Computer Society.

[31] J. Susskind, A. Anderson, and G. Hinton. The toronto face database. Technical report, UTML TR 2010-001, University of Toronto, 2010.

[32] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[33] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. *arXiv preprint arXiv:1504.01561*, 2015.