

# Requirements and Software Framework for Adaptive Multimodal Affect Recognition

Elena Vildjiounaite, Vesa Kyllönen, Olli Vuorinen, Satu-Marja Mäkelä, Tommi Keränen,  
Markus Niiranen, Jouni Knuutinen, Johannes Peltola  
VTT Technical Research Centre of Finland  
Kaitoväylä 1, 90571 Oulu, Finland  
firstname.lastname@vtt.fi

## Abstract

*This work presents a software framework for real time multimodal affect recognition. The framework supports categorical emotional models and simultaneous classification of emotional states along different dimensions. The framework also allows to incorporate diverse approaches to multimodal fusion, proposed by the current state of the art, as well as to adapt to context-dependency of expressing emotions and to different application requirements. The results of using the framework in audio-video based emotion recognition of an audience of different shows (this is a useful information because emotions of co-located people affect each other) confirm the capability of the framework to provide desired functionalities conveniently and demonstrate that use of contextual information increases recognition accuracy.*

## 1. Introduction

Existing works on multimodal affect recognition largely fall into two categories: first, thorough offline studies (the work [1] presents a recent survey on audio-visual affect recognition); second, real-time interactive applications developing an affect recognition method for a particular task (as in the work [2]). We have not found works presenting multimodal fusion software for real time affect recognition, which would allow to adapt on the fly to changes in data availability, environments, users and application tasks (e.g., to take into account that same emotion may be expressed differently if a person is talking to a boss than if he/she is talking to a spouse) and in the same time to utilize diverse methods of increasing recognition accuracy. Some of the listed above functionalities are provided by generic machine learning libraries, but these libraries suit mainly for offline comparison of reasoning methods [3].

Context recognition and adaptation to users and contexts is an actively developing research area, but again adaptation is usually done in an application-specific manner. This research provided methods to recognize diverse situations automatically, for example, to use address book of a phone for distinguishing between calls to a boss and to a spouse; to acquire user

location via GPS and services providing coordinates of main points of interest such as museums, concert halls, stadiums etc; other information about user situation (such as a formal dinner with business partners vs. a party with friends) can be acquired from a personal calendar [4]. Consequently, it becomes possible to use context in affect recognition, but the survey [1] stated the need to take into account context of expressing emotions as an important, but rarely addressed issue. Dynamics of emotional states was listed as another important issue.

This work presents a software framework allowing to deal with these and other important issues with a little configuration effort, and the experiments with using the framework for audio-visual recognition of emotions of an audience in different contexts. Emotion recognition of an audience may be useful for interactive installations, for giving a prize of audience preferences, for memory aid tools and also because emotions of surrounding people affect emotions of an individual. For example, liking or dislike of others affect personal mood if a person watches TV in a company [5].

## 2. Current Trends in Affect Recognition

Approaches to emotion recognition differ, first, in choice of emotional models. Use of categorical models is a more common approach because such models are used by humans in daily life and thus labelling of collected emotional data with categorical models is quite natural. Categories used by different researches include some (or none) of basic Ekmanian emotions (joy, sadness, fear, anger, surprise and disgust) and/ or some other categories such as frustration [6] or boredom [7]. Choice and number of categories depend on the application goal, for example, the work [8] aims at distinguishing between two emotional categories only (fear and neutral) for surveillance purposes. Another approach is to use dimensional models, such as PAD (Pleasure/ Arousal/ Dominance) model, but labelling of emotional data with dimensional models is more difficult and thus either non-trainable fusion rules are used [2] or special training of annotators is required before labelling [1]. Labelling can be also simplified to classification into selected sectors (e.g., positive/negative, low/ mid/ high) at the expense of information loss [1].

Second, approaches to multimodal emotion recognition differ in choice of fusion methods: fusion can be performed on decision level [9], that is, each modality outputs a “final” category and these outputs are fused, for example, by voting. Majority of works on multimodal fusion uses lower-level fusion methods, so that each modality outputs one of modality-dependent classes and/or scores (for example, one modality outputs “sitting upright” posture class, and another one – a skin conductivity value [6]), and these outputs are further combined to produce a “final” emotional category. Fusion can be also done on even lower feature level, that is, sets of features of all modalities are concatenated into one vector, and this vector is fed into a classifier to obtain a “final” category [7]. Third, different reasoning methods can be used for fusion: SVM or decision tree [7], Gaussian process classification [6] and many others.

Forth, some works on emotion recognition proposed to detect transitions between emotional states, for example, to distinguish between onset, apex and offset of emotional states [10]. Thus, it is needed to reason along timeline (on the data at different time moments). Reasoning along timeline can be useful also for synchronizing data of different modalities [7] and for detecting long-lasting emotional states, for example, long-lasting frustration of a student [6].

Last but not least, recently it was proposed to use context in emotion recognition, such as a state of a tutoring dialogue [11] or situation of a person, because same emotions may be expressed differently under pressure to be formal, as in a court, and in relaxed state, as in a party [12]. Also cultural differences in perceiving and expressing emotions exist [13].

Furthermore, studies into multimodal fusion and machine learning in other domains suggested several other ways for improving recognition accuracy, for example, biometrics-based personal authentication was improved by employing user-dependent fusion models in the work [14]. A well-known method to increase accuracy is to employ classifier ensembles [15]. This method aims at overcoming the problem that none of existing machine learning algorithms outperforms the others with all data, and it works as follows: first a large set of algorithms, called classifier pool, is trained on one part of available data and validated on another part. Then, depending on the validation results, an appropriate subset of the trained algorithms is selected at the moment of fusion. Such a subset (called classifier ensemble) can include one or more members and can be selected in different ways.

### 3. Fusion Framework Implementation

The overview of functionalities, used in different multimodal fusion approaches, is presented in Fig. 1: some researchers employ fusion directly on feature vectors (that is, no optional grey blocks are used), while others employ first or second blocks (e.g., when input components are developed separately). The third option, reasoning along timeline, is used less commonly.

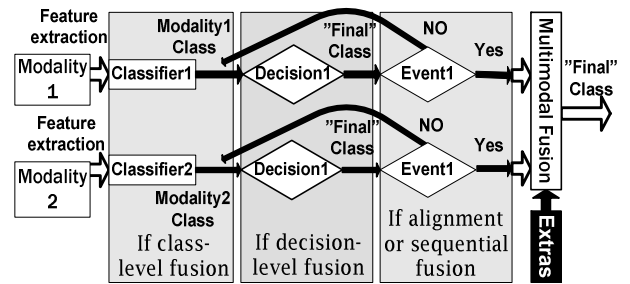


Figure 1: Overview of multimodal fusion for affect recognition. Grey boxes denote optional elements; each of these elements may or may not be included into processing. Grey box “Extras” denotes any additional information used, e.g., results of offline testing for selecting an appropriate classifier ensemble, user’s location, task, nationality, ID etc.

In order to provide possibility to flexibly combine functionalities, listed in the previous section and shown in Fig. 1, we implemented a software framework with the architecture presented in Fig. 2. Currently the framework supports only categorical emotional models because they provide intuitive labelling and a freedom to choose any application-specific set of categories and thus are most commonly used [1], but classification can be done along several dimensions simultaneously, thus providing support for PAD models annotated by discrete values. AND/ OR fusion rules, provided by the framework, are useful only for decision-level or class-level fusion for classification with categorical models, but other implemented fusion methods, Weighted Sum and SVM (Support Vector Machines, we use its implementation in TORCH library [16]), can be used at any level of fusion and can be extended to support continuous dimensional models.

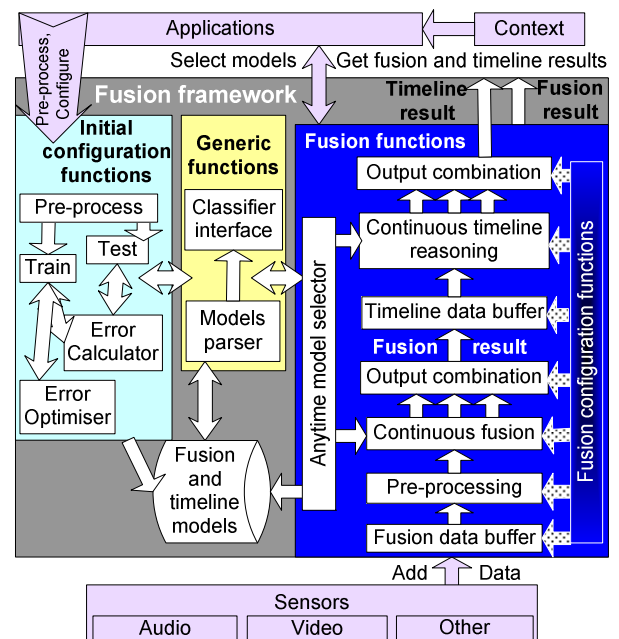


Figure 1: Architecture of the fusion framework.

### 3.1. Initial configuration and generic blocks

Initial configuration is essentially creation of models which will be used for fusion. All provided fusion methods (AND/ OR rules, Weighed Sum and SVM) can be trained by optimizing errors on training data. Except for SVM, all other fusion methods can be also used without training. Combining several trained and fixed methods together is also possible, for example, combining several SVM or SVM with AND/OR rules.

One of the main generic building blocks of Fusion Framework is Model Parser: it reads templates of trainable models from template files at training stage and ready (non-trainable and already trained) models from model files at fusion and test stages and chains the methods in the specified order depending on the found keywords (such as “IF”, “THEN”, “SVM”, “AND”, “>” etc). The actual functionality of the fusion methods and storage of all their parameters (weights and thresholds, for example) are implemented within the second generic building block, called Classifier Interface. Consequently, for specifying the desired fusion functionality it is needed either to create appropriate template files and to train them, or to write models by hand. In some cases pre-processing is required before training and testing, so it is needed to choose from the provided pre-processing functions for combining asynchronous data of different modalities and/ or for data normalization and to place these functions at the start of the methods chain.

For training and testing it is also needed to specify a function for calculating the error cost. This is done within one of the core building blocks of initial configuration, called Error Calculator. Training aims at minimizing the cost function (it can be a weighted sum of misclassifications of each class, for example, where weights represent class importance) while satisfying additional constraints, for example that number of misclassifications of some class should not exceed an application-dependent target value. Although there is no guaranteed way to achieve a desired trade-off between misclassification rates of different classes in multi-class classification problems (especially if number of training samples differs for different classes, which is often the case), using cost functions is a popular approach to this problem [17]. Choice of cost function affects the choice of another core building block of initial configuration, Error Optimizer, an algorithm for minimizing the chosen cost function. Finding SVM model requires solving quadratic optimization problem (implemented in TORCH [16]). Different trade-offs between misclassification rates of different classes can be achieved by varying penalties for misclassification of positive and negative examples. Finding optimal weights and thresholds for other methods is done with differential evolution method [18].

Templates of trainable fusion models include fusion methods, input modalities and output modality, where input modality can be any feature value or a class score:

```
IF speech1 > X
AND audio_volume2 > X
THEN FusionResult=excited
```

In the example above “speech1” denotes a score of recognising class “speech” by method number one (such as by a particular microphone or a particular audio processing algorithm – there may be many microphones or audio processing algorithms employed). Users can choose any string for naming input and output modalities. The keyword “X” denotes that this value should be searched during training; “X” is used always, although actual values will differ from each other after training, when this template will turn into a model:

```
IF speech1 > 0.7
AND audio_volume2 > 0.8
THEN FusionResult=excited
```

Such model can be also written by hand, for example, decision-level models don’t require training.

As emotion recognition with categorical models is a multi-class classification task, all models are trained in “one against all” fashion, and templates need to be written for recognizing each emotional state (e.g., “final result” in Fig. 1). Input modalities and fusion methods, used for recognizing different emotional states, may be different, for example, a template for recognizing “user approval” state can look as following:

```
IF SVM(clapping, audio_volume2) > 0
OR SVM(laughing, smiling) > 0
THEN FusionResult=approval
```

Training according to this template results in two sets of support vectors – one set for each SVM.

It is possible to create either a set of models (classifier pool) or just one model for recognizing each state. After models are created, they can be described by a list of descriptors including error rates (useful for selecting a classifier ensemble), user nationality or ID, context in which these models are valid (for example, user task or formal vs. informal situation) and anything else useful for model selection at the moment of fusion. Confusion matrix, test data size and modalities are automatically added to a model description during testing, and other descriptors can be added manually. An example of model description, stored together with each model, is:

```
Modalities: audio_volume2, clapping,
laughing, smiling
Test data size: 410
User Nationality: 11
User Situation: concert/theatre
Confusion Matrix:
      approval disapproval neutral total
approval    72         8         20    100
disapproval  5         95         10    110
neutral     10         10        200    220
```

In the example above only three emotional states are listed, but the framework supports any number of output classes, as well as any number of input modalities.

Models for reasoning along timeline are created and described in a same way as fusion models, with just one

difference: time intervals (in seconds) between events should be defined. For example, a primitive model for detecting onset of user anger can look as following:

```
IF neutral > 0.8
AND NEXT (0-10) angry > 0.7
THEN TimelineResult=anger_onset
```

Expression “angry > 0.7” denotes that confidence in recognizing “angry” state should exceed 0.7.

Additionally, reasoning along timeline can be done by voting among results within some (usually small) time window. We added this option after the first version of the framework was developed for two-class recognition problems and tested in a simulated task of continuous biometric verification [19]. After these first tests we significantly simplified chaining of different framework functionalities, and now specifying that reasoning along timeline should be done by voting requires only setting of two parameters in the framework configuration file, the interpretation method and the voting time window:

```
Interpretation: voting
InterpretationVotingInterval: 0.1
```

After the first tests the framework was also extended to multi-class multi-dimensional problems in order to provide simultaneous classification of inputs along several dimensions, for example, to estimate at the same time level of pleasure (e.g., neutral, approval or disapproval) and level of excitement (e.g., low, mid or high) – the option valuable for emotion recognition because often it is easier to detect excitement than to estimate level of pleasure, and thus in one-dimensional classification information regarding pleasure may be lost due to relatively low confidence in it. Specifying multi-dimensional classification is also an easy task: the dimension is the model term between the keywords “THEN” and “=” (the examples of fusion models above are all along the only dimension “FusionResult”, while two-dimensional classification would be performed if in some models the term “FusionResult” would be replaced by the term “Pleasure” and in other models – by for example the term “Arousal”).

After the first tests we also improved real-time fusion functionalities, described in the next section, provided easier to use options for on-the-fly adaptation of fusion and added pre-processing (such as combination of asynchronous data, normalization etc) functionalities for training, testing and real time fusion stages.

### 3.2. Fusion

As Fig. 2 shows, fusion framework interface towards individual modalities is very simple: each time when new data is available, it should be put into Fusion Buffer using Add Data function. Data is added in a format “modality name – value – confidence in this value”, which allows to have as many modalities as users want and to use confidence in reasoning. Interface towards applications is also simple: a function for selecting models for fusion and for reasoning along timeline,

called Anytime Model Selector, and functions Get Fusion Result and Get Timeline Result that return a corresponding result and a confidence in this result. It is also possible to configure diverse fusion parameters: to select pre-processing options; to specify confidence thresholds and time intervals for keeping the data in each buffer; to chose how to combine outputs of different models (e.g., by voting or by weighted sum).

Fusion is performed continuously (triggered by new data arrival) on the data stored in a Fusion Buffer by models selected by Anytime Model Selector. Results of continuous fusion are stored to Timeline Buffer, and reasoning along timeline is performed, also continuously, on the data in this buffer by models selected by Anytime Model Selector or by voting.

Anytime Model Selector provides a convenient way to adapt to diverse contexts and to select classifier ensembles for improving recognition accuracy or for satisfying specific application requirements. For example, if emotion recognition from speech is running on a user’s mobile phone, adaptation to context can be done by using a model for “informal situation” if a person calls to a spouse, and by switching to another model for “formal situation” at the next moment if the person answers a phone call from his/ her boss.

Anytime Model Selector also provides a convenient way to improve recognition accuracy and/or to adapt to application requirements by selecting a subset of models that have shown the best accuracy for currently available modalities, currently most confident modalities or current values of the modalities: selection of models according to specifics of each data sample (called dynamic classifier selection) is one common way to increase accuracy [15]. If by some reason a certain subset of emotional states is currently more important for an application than the other states, Anytime Model Selector allows to choose the models that have shown the best accuracy in recognizing these particular states. It is also possible to select models with a desired set of modalities if an application has higher trust in them.

When an application calls Anytime Model Selector, it submits a list of descriptors that are compared to model descriptors one by one, and at each step models matching next descriptor are selected from a group of previously selected models. Output Combination block combines outputs of all selected models by either voting or weighted sum of the normalized scores.

## 4. Experiments

We validated the fusion framework in the tests on audio-video based emotion recognition of an audience in three contexts: in a theatre, circus and a sport event. We tested, how much effort is required to get the desired functionalities (to change a parameter in a framework configuration file, to call some function or to write a piece of code) and to create models, and whether real-time processing of video and audio data, model selection and fusion of asynchronous data work together sufficiently fast. In the experiments we attempted to

differentiate between the following situations:

- audience waiting for a start of a show
- audience leaving a show (e.g., during a break)
- moderate approval of a show
- strong approval

Recognition of the last two situations is the main goal if an application is interested in evaluating degree of an interest of the audience, and it is needed to distinguish between these two situations and the first two situations that do not allow to evaluate, whether the audience liked or disliked the show (and thus we consider them as a neutral state of an audience). Naturally an interactive application needs also to recognize an audience' dislike of a show, but we were not able to find such data.

#### 4.1. Data and individual components

For this study we used shots of audience found in movies and TV programs. We found shots of three contexts: in a concert, in a circus and in a basketball match. For each context we were aiming at distinguishing between three emotional states: neutral, moderate approval and strong approval. We also found shots showing how an audience leaves a show, but only in a concert hall. As an audience is almost never shown for a long time, each shot lasted for few seconds. (During a match an audience can be shown for longer time periods, but most of the time a commentator is speaking and thus shots of audience' emotions without the commentator's voice are not long either.) We found 5-10 shots of each emotional state for each context, for examples see Fig. 3. The smallest number of shots was found for "strong approval" of a concert and "leaving a show" situations.



Figure 3: Examples of video shots in the database: "moderate approval" state is presented in the left, "strong approval" – in the right. For each emotional state the top example shows an audience in a concert hall; the middle one – in a sports match; the downmost one– in a circus.

In the tests we projected the shots to a screen and used a web camera facing the screen as video input and

a microphone as audio input. Data projection decreases quality of video input, but the problem of emotion recognition from faces in a crowd is not solved by the current state of the art algorithms anyway and thus from video data we used only the optical flow – amount of motion at some moment of time compared to the previous moment (all shots were taken by still cameras).

The video analysis component is based on the widely used OpenCV library [20] and performs several types of processing and analysis on the live video in parallel. The optical flow of the video stream is calculated as the average of motion vectors estimated for each image pixel over a small region around them, representing the average overall motion in the scene at any given time -- and not considering the movement of any one object in particular. The component also provides the face detection functionality, based on the Viola-Jones rapid object detection algorithm [21], so in a future we plan to acquire better quality data and to use a ratio between the number of faces detected and the optical flow as more reliable indicator of the audience' activities.

In our tests optical flow of "strong approval" during a basketball match was much greater than in any other situation and appeared to be a fairly reliable indicator of this situation. Optical flow of "leaving a show" situation was fairly similar to that of "moderate approval", but greater than that of a "waiting of a show start" situation.

Our audio processing component processes live audio and outputs the frame power and the classification result. It classifies the audio stream into eight classes: silence, speech, music, variable and constant noise, whistling (e.g., for recognition of disapproval in a sports match), applauding and clapping (applauding by a few persons only). Audio classification is based on HMM (Hidden Markov Models, we use the implementation in TORCH library [16]) because HMM is a trade-off between accuracy and computational cost and can be well applied to live audio analysis due to its short response time. 29 Mel-frequency cepstral coefficients are calculated using 20 ms time window and fed into HMM models of each class. Class models were trained beforehand on a separately recorded data (mono, 16 bits, 16 kHz sampling frequency) so that number of HMM states, number of Gaussian mixtures and parameters of each class model were optimised for recognising this class.

Power of an audio signal can be used as an indicator of level of excitement of an audience, but it does not allow distinguishing between positive and negative excitement, let alone occasional noises. The power of applauding is a reliable indicator of positive interest, though. Recognition of speech and music classes may allow to postpone emotion recognition until sounds from other sources than audience (such as commentator' speech in a sports match or music in a concert) cease.

Audio component classified "strong approval" and "moderate approval" states in a context of a sport match as "variable noise" (there were no applauding indeed), and "leaving a show" also as "variable noise". "Strong

approval” of a circus audience was classified mainly as applauding, but also as “variable noise” and “clapping”. “Moderate approval” of a circus audience was classified as “clapping” mainly, but also as “variable noise” and “speech”. Audio classification in a concert appeared to be the most difficult task as this audience is the least expressive one: “strong approval” state was classified partly as “clapping” and partly also as “speech”, while “moderate approval” – mainly as “speech”. “Waiting for a show start” situation was partly classified as “speech” (sometimes viewers talked to each other indeed) and partly as “silence” or “variable noise”.

## 4.2. Fusion

In order to test the functionalities of the fusion framework, we created models using half of the available data and logged results of real-time processing of the other half of the data. We attempted to distinguish between the four above-listed situations with context-independent models (SVM trained on the merged data of all three contexts: concert, circus and match and using a merged feature vector of all audio classes, power of audio signal and strength of optical flow) and with context-dependent models: AND rules created for each of three contexts separately, using own set of modalities for each context. We did not use SVM for context-dependent classification because even amount of merged data for context-independent classification was fairly small for SVM training, and we did not use AND rules for context-independent classification because accuracy of SVM is usually much higher than that of AND rules. Models were first trained and then tested, and after that ensembles of the best models were used in fusion. The capability of the framework to employ classifier ensembles was found to be important in these tests, while attempts to use the only model per situation resulted in very low classification accuracy because none of situations was associated with the only audio class and because of random delays in video processing.

As audio and video components are not fully synchronized, fusion is done each time when new data of any of modalities arrives, using most recent data of all modalities stored in a fusion buffer. (The term “modality” here denotes either one audio class or power of audio signal or optical flow; that is, altogether we had nine modalities). The overall classification of a shot is done by reasoning along timeline and depends on how many times fusion resulted in this class.

Table 1 presents the results of classifying the test shots and shows that “strong approval” in “sport” context was the easiest situation to classify for both generic and context-dependent fusion models. For the majority of other situations context-dependent models have shown higher recognition accuracy despite that AND rule is a fairly primitive fusion method.

Table 1: Confusion matrix presenting percent of correct recognition of four situations in three contexts: “SA” stands for “strong approval”; “MA” – for “moderate approval”, “WN” – for “neutral, waiting” and “L” – for “leaving, neutral” states. “G” stands for generic and “C” – for context-dependent models.

		SA		MA		WN		L		
		G	C	G	C	G	C	G	C	
S	concert	0	36	16	16	36	64	0	16	
	A	circus	36	67	16	16	0	16	16	16
		sport	88	88	12	12	0	0	0	0
M	concert	0	0	33	50	50	67	0	0	
	A	circus	0	0	33	50	33	50	17	17
		sport	0	0	33	50	17	50	17	33
W	concert	0	0	17	0	50	50	33	33	
	N	circus	0	0	17	0	50	50	33	33
		sport	0	0	0	0	50	67	33	33
L	concert	0	0	17	17	17	17	66	66	

## 5. Conclusion

This paper presented a framework for multimodal real time fusion and the experiments of using this framework for multimodal recognition of emotional states of an audience. Existing works on emotion recognition are not targeted at recognizing emotions of an audience despite that emotions of co-located people affect each other [5] – probably because current video processing algorithms do not recognize facial expressions even of one person in realistic settings [1], not to speak about many faces in a crowd. We found recognition of selected emotions of an audience to be an interesting task which allowed us to test diverse functionalities of the fusion framework. Although we were not particularly aiming at developing a method of recognizing emotional states of an audience, we think that the achieved recognition rates are not too bad considering that the video component provided only optical flow and that audio classification by the audio component was not always in agreement with a human perception.

Furthermore, as video data in our experiments was of low quality and as we used fairly lightweight video and audio analysis methods, as well as simple fusion methods, we think that the presented method can also work on mobile devices, exploiting their context recognition capabilities together with their capability to provide accelerometer data for affect recognition.

However, the main goal of the experiments was to test whether the framework provides all desired functionalities in a fairly convenient way. The experiments confirmed that the framework easily adapts to whatever input data is available; allows to flexibly combine feature-level, class-level and decision-level fusion methods; to reason on data along timeline and to select appropriate classifier ensembles in real time depending on contexts of emotional behaviour and test accuracy. The main goal of the framework development

was not to invent any new fusion or classifier ensemble selection methods, but to provide means to experiment with diverse methods in real time settings with no or very little programming effort and this way to facilitate research on affect recognition. The tests have shown that configuring the majority of desired functionalities is easy: it requires either to write a model template or to set a parameter in the framework configuration file.

Although currently adding a new pre-processing functionality requires writing a function, and choosing existing normalisation functionality requires writing several lines of code (currently there are no parameters for it in the framework configuration file), it is a fairly little effort and in a future will be further reduced. Integration of other reasoning methods is also simplified due to the modular framework structure.

The experiments have confirmed that adaptation to context increases recognition accuracy and that employing classifier ensembles for class-level fusion is essential when emotional states can not be associated with the only class. As use of context and dynamics of emotional behaviour are important issues in affect recognition [1], we consider the framework capability to provide these functionalities in a convenient and flexible way as an important advantage. Adaptation to application requirements (for example, choosing a trade-off between recognition errors of different emotional states), although not tested in this work, can be another useful feature. Although it may be needed to store a fairly large set of fusion models in case of adaptation to many different situations, search for fusion models is fairly fast.

Currently the framework capability to reason on dynamics of data is not so advanced, but the architecture of the framework allows integration of other methods of temporal (along timeline) and instant (parallel) fusion. Future work includes integration of Hidden Markov Models for reasoning along timeline, even though HMM can not always outperform simpler methods [7]. Apart from integrating more fusion methods, future work includes research on dealing with errors of input components. Currently fusion framework allows to use confidence in inputs in reasoning, but it does not significantly reduce the number of misclassifications because the confidence in them can be fairly high. How to deal with erroneous inputs is generally a challenging problem of multimodal fusion, and there are not so many solutions proposed.

Despite that the problem of creating appropriate fusion models for each task is the responsibility of application developers (and will remain so in a near future because this is a global problem in machine learning, mainly solved by trial and error approach), the tests presented in this work are encouraging enough to apply the framework also to other fusion tasks.

**Acknowledgement:** this work was funded by EU Callas project, contract IST – 034800.

## References

- [1] Zeng, Z., Pantic, M., Roisman, G., Huang, T., A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1): 39-58, 2009.
- [2] Gilroy, S. et al., An emotionally responsive AR art installation. *ISMAR 2007*.
- [3] <http://www.cs.waikato.ac.nz/ml/weka/>
- [4] Lahti, J., et al., Context-aware mobile capture and sharing of video clips, *Handbook of Research on Mobile Multimedia*, Ibrahim, I. K., Kepler, J. (eds.), 2006
- [5] Masthoff, J., Gatt, A., In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Model. User-Adapt. Interact.* 16(3-4): 281-319 (2006)
- [6] Kapoor, A., Bursleson, W., Picard, R., Automatic prediction of frustration, *International Journal of Human-Computer Studies*, 65(8): 724-736, 2007
- [7] Gunes, H., Piccardi, M., Automatic Temporal Segment Detection and Affect Recognition from Face and Body Display, *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 39(1): 64-84, 2009.
- [8] Clavel, C. et al. Fear-type emotion recognition for future audio-based surveillance systems. *Speech Communication*, 50(6): 487-503, 2008.
- [9] Caridakis, G., Karpouzis, K., Kollias, S., User and context adaptive neural networks for emotion recognition. *Neurocomputing*, 71(13-15), 2008.
- [10] Pantic, M., Patras, I., Dynamics of Facial Expression: Recognition of Facial Actions and Their Temporal Segments from Face Profile Image Sequences, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 36(2): 433-449, 2006.
- [11] Forbes-Riley, K., Litman, D., Predicting Emotion in Spoken Dialogue from Multiple Knowledge Sources, *HLT/NAACL 2004*: 201-208, 2004
- [12] Douglas-Cowie, E. et al. The HUMAINE Database: Addressing the Collection and Annotation of Naturalistic and Induced Emotional Data, *ACII 2007, LNCS 4738*
- [13] Kleinsmith, A., De Silva, R., Bianchi-Berthouze, N., Cross-cultural differences in recognizing affect from body posture, *Interacting With Computers*, 18(6), 2006.
- [14] Jain, A., Ross, A., Learning User-Specific Parameters in a multibiometric system, *ICIP 2002*
- [15] Ko, A., Sabourin, R., Britto, A.: From Dynamic Classifier Selection to Dynamic Ensemble Selection, *Pat. Recognition* 41, 1718-1731, 2008
- [16] TORCH, <http://www.torch.ch/>.
- [17] S. Suresh, N. Sundararajan, P. Saratchandra. Risk-sensitive loss functions for sparse multi-category classification problems. *Inf. Sciences*, 178(12), 2008
- [18] Storn, R., Price, K., Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization* (1997), 11 (4), pp. 341 – 359
- [19] Vildjiounaite, E., Kyllönen, V., Software Framework for Multimodal Fusion in Ubiquitous Computing Applications, *Network and Service Security Conf. 2009*
- [20] <http://sourceforge.net/projects/opencvlibrary/>
- [21] Viola, P., Jones, M., Rapid object detection using a boosted cascade of simple features, *IEEE Conference on Computer Vision and Pattern Recognition 2001*