Multimodal Emotion Classification using EEG and Facial Expressions from DEAP dataset

Francesco Stella

Abstract—Emotion Recognition (ER) is a central task in many fields, such as Affective Computing (AC) and Human-Computer Interaction (HCI). Typically, in a lab setting, short movies are employed to elicit emotions in viewers, while some kind of physiological signals acquisition is performed, e.g. Electroencephalography (EEG), Galvanic Skin Response (GSR), Electrooculogram (EOG) or face recording. This work aims at analysing the problem of ER, and more precisely the emotion classification task, through the fusion of two modalities, EEG and face recordings, both retrieved from the DEAP dataset. A simple framework is devised in order to experiment with different models and techniques. In the following work we present results from Random Forest (RF) classifiers and Support Vector Machines (SVMs), employed to predict Valence and Arousal given the multimodal data. Furthermore, a Bayesian Optimisation approach is used in order to optimise the hyperparameters of the models. A few experiments show that the proposed approach provides accuracy (and F1) scores in predicting Valence and Arousal respectively equal to 79.4% (0.839) and 74.9% (0.809), that are comparable to other similar works.

1 Introduction

Emotions can be seen as physiological/neurological changes experimented by a subject in response to the perception of an object or situation, they reflect the feelings of the subject and can be conveyed through verbal or non-verbal behavior (e.g. while talking, through facial expressions, posture, etc.). Emotion Recognition is a central topic in the fields of Affective Computing and Natural Interaction, providing us with useful information about the personality, interests and, of course, emotional state of a given subject. Moreover, it is more reliable than self-reports that can be dependent on other social or personality factors, e.g. males are less likely to report fear [1], and can be performed in an autonomous and continuous way. Other fields of application comprise Human-Robot Interaction (HRI), safe driving, social networking and distance education. [2]

In recent years, multimodal ER has been further explored. This approach entails the fusion of different modalities, such as EEG, voice or facial expressions and can provide us with complementary information with respect to the single modality approach. In this work, two types of models are devised, more precisely Random Forests and SVMs for the independent classification of EEG and facial features and for the *decision-level* fusion of the modalities, i.e. for the classification of the elicited emotion starting from the outputs of the first two models. A Bayesian Hyperparameter Optimisation technique is used in order to estimate the best hyperparameters for each model. DEAP dataset [1] is used to retrieve EEG signals and face recordings from 32 subjects.

2 State of the Art

Several studies showed that multimodal ER generally outperforms the single modality approach [2], [3], [4], [5]. The combination of different modalities for this task has been explored, for example Koelstra et al. [4] and Li et al. [2] used features extracted from EEG and facial landmarks in their experiments, Yin at al. [6] used EEG, EOG, Electromiogram (EMG), GSR, respiration, blood volume and pressure and skin temperature features with an ensemble of Stacked Autoencoders (SAEs) for the classification task. In general, in order to take full advantage of the complementary information provided by the different modalities, *decision-level* fusion is typically preferred to the *feature-level* fusion, since it provides a better prediction robustness achieved by using the classification results of previous stages [7].

Moreover, different types of models have been tested, with a recent preference toward Deep Learning techniques, which are promising for extracting useful features or latent patterns from large amounts of data and provide good generalisation capabilities. A recent review from Chaudary and Jaswal [8] on experiments involving DEAP dataset showed that LSTM-RNN achieves cutting-edge performance in predicting Valence and Arousal from raw EEG signals, outperforming models such as SVM, CNN and BiLSTM.

Finally, Parui et al. [9] perform the emotion classification task on DEAP dataset with a less studied Gradient Boosting algorithm, obtaining very good accuracy (75.97% for Valence, 74.20% for Arousal) with respect to other more common models.

F. Stella, Affective Computing & Natural Interaction, A/A 2020-2021, University of Milan, via Celoria 18, Milan, Italy
E-mail: francesco.stella@studenti.unimi.it

[1] DEAP: http://www.eecs.qmul.ac.uk/mmv/datasets/deap/

3 Theoretical Models

The proposed approach involves experiments with random forests and SVMs trained independently on the multimodal data. The more promising models are chosen and their predictions are merged using decision-level fusion to obtain the final result. This last step is performed by training an SVM on the outputs of the previous models. Random forests are chosen since they generally provide robustness against noise and overfitting, moreover they are robust in presence of redundant attributes (i.e. in presence of strong correlation between attributes) and they are also highly interpretable.

3.1 Random Forest classifier

The first model is a Random Forest, an ensemble method based on *bagging*, which considers a set of Decision Trees (DTs) of small complexity, the so called *weak learners*, in order to perform the classification task. In the following we briefly describe DTs and their construction, then we discuss bagging and random forests.

3.1.1 Decision Trees

Decision trees are highly interpretable models that can be applied for classification or regression tasks. They are composed by three types of nodes, i.e. a *root*, several *internal nodes* and *leaves*, each one representing a *predictor* on which to perform a decision, with the exception of the leaves, that represent the actual predictions.
In order to construct a tree, starting from the root a *recursive binary splitting* procedure is applied, involving the selection of a predictor X_j and the splitting of the predictor space into two regions based on a *cutpoint* s. The final goal is to minimise, for each j and s, the sum of the Residual Sum of Squares (RSS) of the two subregions of the predictor space, as shown in equation 1.

$$min\left[\sum_{i:x_i\in R_1(j,s)}\left(y_i-\hat{y}_{R_1}\right)^2+\sum_{i:x_i\in R_2(j,s)}\left(y_i-\hat{y}_{R_2}\right)^2\right] \quad (1)$$

where $R_1(j,s)=\{X|X_j<s\}$, $R_2(j,s)=\{X|X_j\geq s\}$ and $\hat{y}_{R_j}$ is the mean response for the observations in the j-th region. The tree construction stops if a stop criterion is reached, for example when a total of J subregions has been obtained and each region has at most a fixed number of observations. The prediction is then performed by using the mean of the training observations of the region in which a datapoint to predict falls. In the case of classification, an alternative to RSS is needed, in practise measures such as the *Gini index* (see equation 2) or *entropy* (see equation 3) are used. In both cases, $\hat{p}_{mk}$ represents the ratio of training observations in the m-th region that belong to the k-th class.

$$G=\sum_{k=1}^{K}\hat{p}_{mk}(1-\hat{p}_{mk}) \quad (2)$$

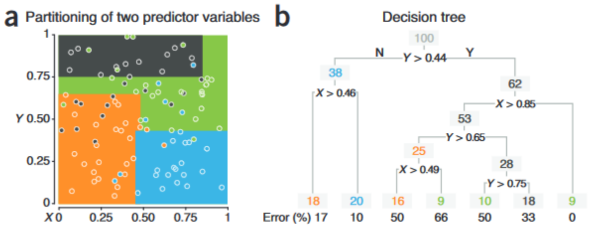


Fig. 1: An example of decision tree from [10].

$$D=-\sum_{k=1}^{K}\hat{p}_{mk}\log\hat{p}_{mk} \quad (3)$$

In *Fig. 1* an example of decision tree is shown together with the feature space.

3.1.2 Bagging

Decision trees are prone to overfitting and they typically have high variance and low bias. Bagging, also known as *bootstrap aggregation*, tries to solve this problem by training a set of weak learners and then computing the average of the predictions. This approach is based on the fact that, given n independent observations each with variance σ^2, we can retrieve the variance of their mean by $\frac{\sigma^2}{n}$. Practically, given a dataset, B samples are retrieved and used as training sets for B decision trees. Then, the average of the B predictions (for an observation x) is computed as in equation 4.

$$\hat{f}_{bag}(x)=\frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x) \quad (4)$$

In bagging, the B trees are deep and each one has high variance and low bias. By averaging the predictions, also variance decreases. In case of a classification task, in order to predict the label associated to a test datapoint, a *majority voting* approach can be used.

3.1.3 Random Forest

Random forests provide a further improvement. As in bagging, samples are retrieved from a dataset in order to train a set of trees, but these trees are now *decorrelated*. In particular, at each split, only a subset of m predictors is randomly sampled from the total set of p predictors, typically with $m\approx\sqrt{p}$. This is the main difference with bagging and it leads to even smaller variance when considering the average of the predictions, since thay are now uncorrelated.

3.2 Support Vector Machine (SVM)

The second model devised in this work is an SVM, a supervised learning method which uses high-dimensional *hyperplanes* and *kernels* in order to perform classification or regression. Since SVM generalises other simpler methods, we briefly discuss them in the following in order to derive and correctly understand SVM.

3.2.1 Classification through a Separating Hyperplane

Classification through a separating hyperplane is the simplest method that uses hyperplanes in order to perform a binary classification task. Given a p-dimensional space, an hyperplane is a $p-1$ *affine subspace* that can be defined through the equation 5:

$$\beta_0 + \beta_1 X_1 + ... + \beta_p X_p = 0 \tag{5}$$

where X = $(X_1, X_2, ..., X_p)$ is a point in the p-dimensional space. If X satisfies the equation, than it lies on the hyperplane, otherwise it will fall in one of the two sides accordingly to the sign. If we represent the sign through the labels {-1, 1}, then given an arbitrary number of examples in a p-dimensional space, with the i-th example represented by $(x_{i1}, x_{i2}, ..., x_{ip}, y_i)$, we can perform the classification for the positive class as shown in equation 6, the negative class case is straightforward:

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} > 0 \ \ if \ y_i = 1 \tag{6}$$

3.2.2 Maximal Margin Classifier

In the Maximal Margin Classifier (MMC), the goal is to find an hyperplane that maximises the distance, the so called *margin*, between itself and its closest points belonging to the different classes. The maximal margin can be constructed as shown in equation 7, in which the constraints guarantee that each point is correctly classified and its distance from the hyperplane is at least M:

$$\begin{aligned} &\max_{\beta_0,...,\beta_p,M} M \\ &subject \ to \ \sum_{j=1}^{p} \beta_j^2 = 1 \\ &y_i(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}) \geq M \end{aligned} \tag{7}$$

This approach is not suitable for *non-separable* cases, i.e. when a separating hyperplane does not exist.

3.2.3 Support Vector Classifier (SVC)

SVC is also known as *Soft Margin Classifier* and its main difference with MMC is the possibility to have a small amount of wrong predictions. SVC allows some errors in classification in order to provide greater robustness to individual observations, while still performing correct classification for most of the points. The soft margin can be constructed similarly to the case of MMC, as shown in equation 8, with the inclusion of a certain number of variables ϵ_i, also known as *slack variables*.

$$\begin{aligned} &\max_{\beta_0,...,\beta_p,\epsilon_1,...,\epsilon_n,M} M \\ &subject \ to \ \sum_{j=1}^{p} \beta_j^2 = 1 \\ &y_i(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}) \geq M(1-\epsilon_i) \\ &\sum_{i=1}^{n} \epsilon_i \leq C \\ &\epsilon_i \geq 0 \end{aligned} \tag{8}$$

The slack variables allow a certain degree of error and two cases can be outlined:

- $\epsilon_i > 0$: i-th observation is on the wrong side of the margin, i.e. between the margin and the hyperplane;
- $\epsilon_i > 1$: i-th observation is on the wrong side of the hyperplane, i.e. it is misclassified;

The parameter C is non-negative and represents the *tolerance*, i.e. a sort of amount of violations to the margin that one can tolerate. It provides an upper bound to the sum of the slack variables, hence if C=0 than no violations are admitted. A representation of a linearly separable case is provided in figure 2, in which no violation occurs. Finally, with the name *support vectors*, we refer to the observations that lie exactly on the margin or on the wrong side of the margin and they are those that affect the classifier.

3.2.4 SVM

Support Vector Machines are an extension of SVC that support non-linear boundaries. SVMs use *kernels* in order to map the features and then compute the decision boundary as seen for the SVC. Kernels are generalisations of the *inner product* and quantify the similarity between two points. There are several types of kernels, such as the linear kernel, polynomial or the *Radial Basis Function (RBF)* (also called *squared exponential*).
In this work, SVM is used with an RBF kernel, shown in equation 9, where γ is a positive constant.

$$K(x_i, x_{i'}) = e^{-\gamma \sum_{j=1}^{p}(x_{ij} - x_{i'j})^2} \tag{9}$$

3.3 Bayesian Hyperparameter Optimisation

In order to estimate hyperparameters that lead to possibly high accuracy for the models, Bayesian Hyperparameter Optimisation is performed for each model, searching in a predefined hyperparameter space and estimating the accuracy of the models through cross validation. This approach is typically chosen when time requirements for the training of the models are expensive, since it can provide a good approximate solution for the best hyperparameters in a reasonable time. Although promising, sometimes this approach can fail to find good hyperparameters, in particular in higher dimensional spaces, whereas *random search*

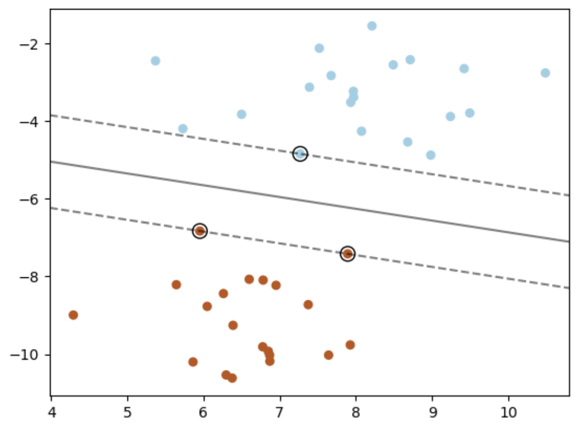


Fig. 2: Example of decision boundary and margins of an SVC (source)

may be more suitable [11]. Here the general idea behind Bayesian Optimisation (BO) is discussed, mainly referring to the material in [12], while in section 4.3.3 the specific details of the hyperparameter optimisation performed in this work are presented.

The core idea behind BO is to use the Bayes' theorem in order to define a posterior distribution, also known as *surrogate function*. The surrogate function is a simplified version of the objective function that we want to optimise, parameterised by a set of hyperparameters. Practically, we can infer some kind of parameters (hyperparameters or weights), by maximising the *expected utility* (or, equivalently, minimising the *expected risk*) of a utility function known, in this context, as *acquisition function*. This function allows us to sample the surrogate function and collect knowledge about the true objective function. Equation 10 shows the bayes rule, in which on the left of the equality there is the *posterior distribution*, on the numerator of the right side of the equation there is the *likelihood* multiplied by the *prior*, while on the denominator we have the *marginal likelihood* (also called *evidence*), used as a normalising constant.

$$P(w|y,X) = \frac{P(y|X,w)p(w)}{P(y|X)} \tag{10}$$

The prior distribution is typically a Gaussian Process (GP), an infinite-dimension stochastic process that represents an extension of the multivariate Gaussian distribution. Equation 11 represents a GP, where *f(x)* is a Gaussian.

$$f(x) \sim \mathcal{GP}(m(x), k(x,x')) \tag{11}$$

Fig. 3 shows a representation of a GP, in which two sampling steps are performed, starting from two observations. The acquisition function can be designed in different ways, although the final goal is always to guide the search toward the optimum of the objective function [12]. In order to achieve this goal, the activation function should be queried in such a way that it is possible to leverage on two aspects:

- *exploitation*: acquisition function has high values where the predicted value of the objective function is high.
- *exploration*: acquisition function has high values in positions corresponding to high uncertainty.

Ideally, the next sample should be chosen in positions that satisfy both the aspects of exploitation and exploration. The most common acquisition functions are:

- **Probability of Improvement (PI)**: chooses the next point x for which there is high confidence that it brings to improvements at least equal to a parameter $\xi \geq 0$ with respect to the best point found so far, x^+. It is an exploitative strategy:

$$PI(x) = \mathcal{P}(f(x) \geq f(x^+ + \xi)) \tag{12}$$

- **Expected Improvement (EI)**: it provides a balancing over exploitation and exploration and it can be generalised as in equation 13 if the variance of the GP is $\sigma(x) > 0$ in a given point x, otherwise we have $EI(x) = 0$:

$$EI(x) = (\mu(x) - f(x^+) - \xi)\Phi(Z) + \sigma(x)\phi(Z) \tag{13}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are, respectively, the *cumulative distribution function* and the *probability distribution function* of the standard normal distribution. Z is equal to 0 if $\sigma(x) = 0$, otherwise it is defined by:

$$Z = \frac{\mu(x) - f(x^+ - \xi)}{\sigma(x)} \tag{14}$$

- **Lower Confidence Bound (LCB)**: the next point is chosen based on the equation 15, in which k is a tunable parameter:

$$LCB(x) = \mu(x) - k\sigma(x) \tag{15}$$

The *Upper Confidence Bound (UCB)* can be defined by simply exchanging the minus sign with a plus sign.

4 Simulation and Experiments

A binary classification task is performed, following an approach similar to those suggested by [9] and [14]. Since the true labels to predict fall within the continuous interval [0, 9], a threshold equal to 4.5 is chosen in order to convert the problem into a binary classification one. Predictions are performed on the Russell's bidimensional Valence/Arousal (VA) emotion space, that can be represented as in figure 4, in which labels smaller than or equal to 4.5 correspond to low Arousal/negative Valence, while labels greater than 4.5 correspond to high Arousal/positive Valence. Models are then trained on

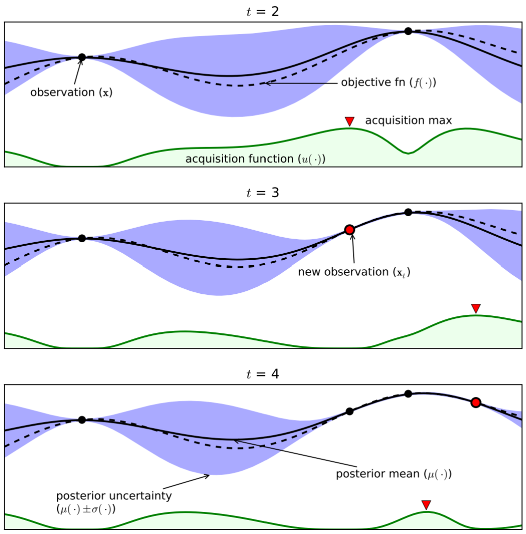


Fig. 3: Representation of a Gaussian Process [13]. The *blue-shaded* area represents the uncertainty related to the surrogate function in a specific point x. It is defined by the mean ± variance of the GP in x.

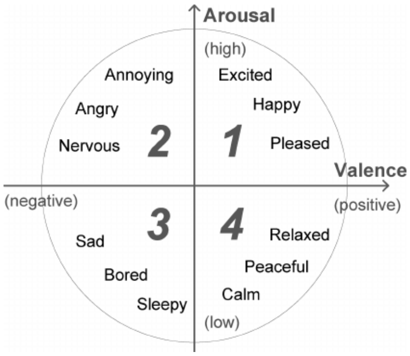


Fig. 4: A representation of Russell's 2D Valence/Arousal emotion space.

the dataset using a *binary logistic* objective function and *accuracy* and *F1 score* are collected in order to evaluate the models.

4.1 Dataset

Multimodal data for this experiment is retrieved from the DEAP dataset, a collection of physiological signals and face video recordings gathered from a total of 32 participants, each one involved in watching 40 one-minute long music videos. Video recordings are collected for 22 participants, with a few participants missing some of them. EEG and other peripheral signals, for a total of 48 channels, are collected for all participants during each trial. In this work 32 EEG-related channels are selected from the already preprocessed version of the DEAP dataset, which contains signals downsampled to 128Hz and involves other preprocessing steps, specifically EOG artefacts removal, bandpass filtering in the range 4.0-45.0Hz, Common Average Referencing (CAR) of the data and a few other operations.

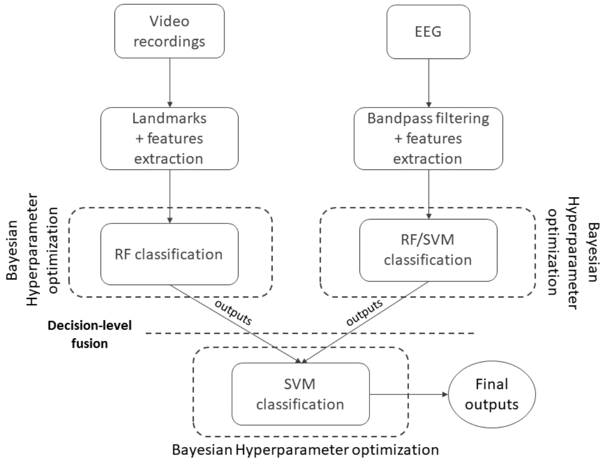

Video recordings
EEG
Landmarks + features extraction
Bandpass filtering + features extraction
Bayesian Hyperparameter optimization
RF classification
RF/SVM classification
Bayesian Hyperparameter optimization
outputs
outputs
Decision-level fusion
SVM classification
Final outputs
Bayesian Hyperparameter optimization

Fig. 5: Classification pipeline with decision-level fusion of the single-modality predictions

Furthermore, ratings in terms of levels of arousal, valence, like/dislike, dominance and familiarity are performed at the end of each trial by the participants by means of Self-Assessment Manikins (SAM) and the results are used as ground truth.

4.2 Architecture

The overall pipeline is composed by two separate models used independently in order to classify EEG and facial features into positive/negative Valence and high/low Arousal, followed by a third model that performs the decision-level fusion and, hence, the final classification. In general, the system can be represented as in figure 5, in which a preprocessing step is performed for both video recordings and EEG data in order to extract the features.

4.3 Implementation Details

As a first step, feature extraction from the single modalities is needed. EEG is a *non-stationary random* signal and, typically, three different approaches are considered for feature extraction. In the first approach *time-domain* features are considered, with windowing performed on the original signal. The second approach entails *frequency-domain* feature extraction and typically bandpass filtering is performed on the original signal in order to retrieve Alpha, Beta, Gamma and Theta sub-bands, followed by Power Spectral Density (PSD) estimation of each sub-band through, e.g., Welch method or FFT. The last approach involves *time-frequency domain* features extraction, with Short-Time Fourier Transform (STFT) representing one of the most used approaches. Several studies report the presence of correlations between PSDs in the four bands and Valence/Arousal [15], [16], [17], hence in this work PSD estimation is performed for each of the four sub-bands. Please refer to appendix A for a more detailed description.

Task	Model	Folds	Repetitions
EEG	RF	5	1
EEG	SVM	5	3
Facial Features	RF	3	1
Decision-level	SVM	5	1

TABLE 1: Number of folds and repetitions for the *Stratified K-Fold cross validation* procedure, for each tested model and task.

The second step involves facial landmarks extraction from video recordings. An attempt with Viola-Jones algorithm has been performed in order to retrieve 68 landmarks from faces, however, due to the irregular and unpredictable movements of some subjects (e.g. see subject 12, trial 09), the algorithm failed to accurately detect the landmarks in some video recordings. Hence, a deep learning approach to facial landmark detection is chosen, specifically Mediapipe FaceMesh model from Google [18] is employed in order to retrieve 468 landmarks in 3D coordinates and then filter them by considering only those of our interest and converting them into a 2D space. Then, feature extraction is performed from the selected landmarks (for further details, please refer to appendix B.

A *subject-independent* approach is followed for the training, hence the data from the subjects is collected as a single dataset. Then, in the case of EEG features, some of them are removed based on their *Pearson correlation coefficients*. More precisely, those having a correlation ≥ 0.85 are removed. Before training the models, the datasets can be *normalised* or *standardised* and two partitions are retrieved after *shuffling* the data. Specifically, *train* and *test* sets are retrieved with proportions with respect to the entire dataset respectively equal to 0.8 and 0.2. On the train set, a *Repeated Stratified K-Fold cross validation* technique is used, in which the train set is divided into K non-overlapping subsets each having approximately the same representativeness for both classes. Then, K-1 are used to train the model while the K-th is used as the *validation set*, with the procedure iterating until each subset has been chosen exactly once as the validation set. It is also possible to repeat the entire procedure *m* times, each time shuffling the dataset. The number of folds and repetitions used in this work are shown in table 1.

4.3.1 Random Forest implementation

Random forests are implemented through *xgboost*[2], a distributed library providing implementations and GPU support for several tree-based methods, such as tree boosting and random forests. Since we use Bayesian Hyperparameter Optimisation, a *search space* is defined over a set of hyperparameters by specifying boundaries and the uniform sampling distribution. Relevant hyperparameters tuned through the BO technique are:

- **reg_alpha**: L1 regularisation on weights.
- **reg_lambda**: L2 regularisation on weights.
- **subsample**: subsample ratio of the training instance.
- **colsample_bynode**: subsample ratio of features after each split.

4.3.2 SVM implementation

SVMs are implemented by using the *scikit-learn library*[3]. Relevant hyperparameters tuned through the BO technique are:

- **C**: it is a regularisation parameter, more precisely it represents the L2 penalty. The entity of regularisation is inversely proportional to its value, that must be strictly positive.
- **tol**: tolerance parameter used in order to define the stopping criterion.

4.3.3 Hyperparameter optimisation

As stated in section 3.3, Bayesian Optimisation is chosen for hyperparameter estimation. The library scikit-optimize [4] is used in order to perform a search over the hyperparameter space (using the *BayesSearchCV()* function) and cross-validation is performed to evaluate the generalisation performance of the models using the current set of hyperparameters. *Algorithm 1* shows the general steps followed by the Bayesian Optimisation procedure.

Algorithm 1 Bayesian Optimisation

```
for t=1,2,... do
    find $x_{t+1}$ by optimising the acquisition function $\alpha$ over the objective function f:
        $x_{t+1} = \underset{x}{\operatorname{argmax}}\ \alpha(x; D_t)$
    query f in order to obtain $y_{t+1}$
    augment data $D_{t+1} = \{D_t, (x_{t+1}, y_{t+1})\}$
    update statistical model
end for
```

The specific configuration of the BO loop used in this work comprises a GP as the prior (*base_estimator* in scikit-optimize jargon) and a mixture of PI, EI and LCB as acquisition function, where the best between the three is chosen at each step.

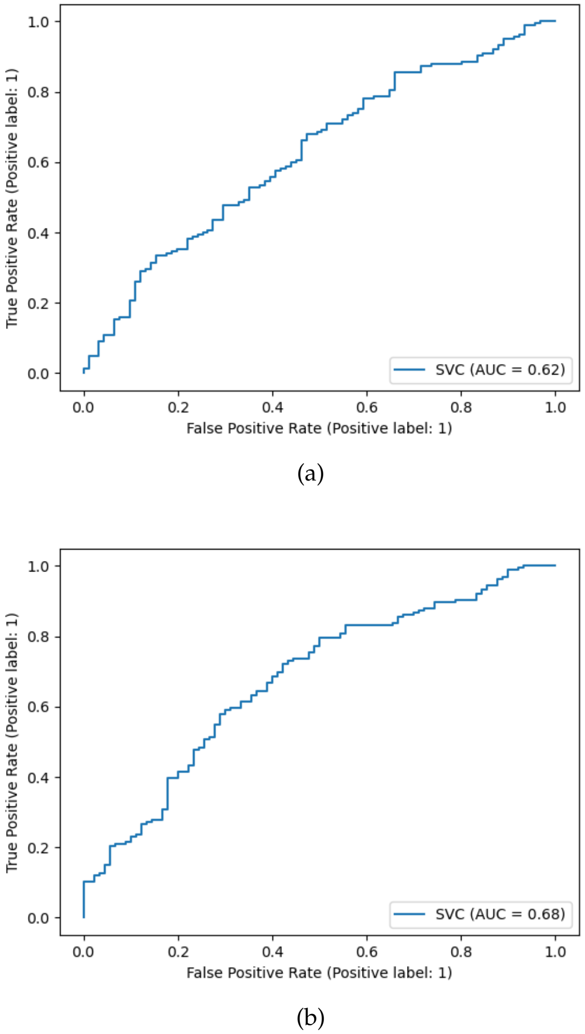
5 Results

After performing the training of the models, accuracy and *F1-score* are retrieved for each model and for each emotion dimension (Valence and Arousal). The results for EEG, facial features and decision-level predictions are reported, respectively, in tables 2, 3 and 4, where F1-score is in brackets and the abbreviations *norm* and *std* represent normalisation and standardisation. Moreover, ROC_AUC curves are computed on the test datasets and

[2] link to xgboost

[3] link to scikit-learn

[4] link to scikit-optimize



(a)

(b)

Fig. 6: ROC curves for Valence (a) and Arousal (b) obtained using SVMs on EEG test datasets.

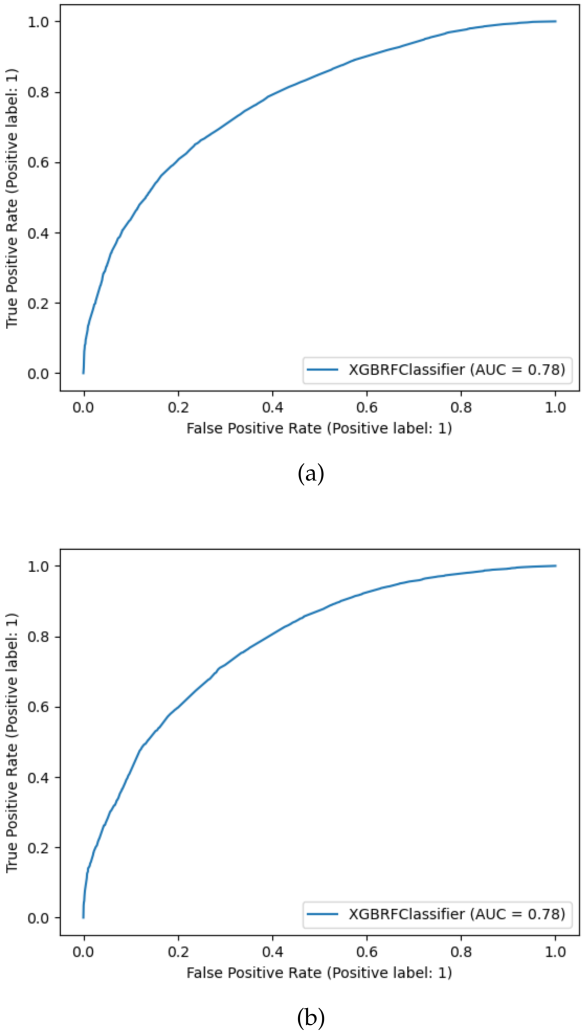
the results are shown in figures 6, 7 and 8. ROC_AUC curves for EEG models are slightly smaller than those presented in [9] as validation metrics.

Model Type	Preprocessing	Valence	Arousal
SVM	std	0.660 (0.760)	0.668 (0.773)
RF	-	0.629 (0.738)	0.609 (0.702)

TABLE 2: Tested models for EEG predictions.

Model Type	Preprocessing	Valence	Arousal
RF	-	0.720 (0.787)	0.740 (0.809)
RF	norm	0.712 (0.789)	0.716 (0.795)

TABLE 3: Tested models for facial features predictions.



(a)

(b)

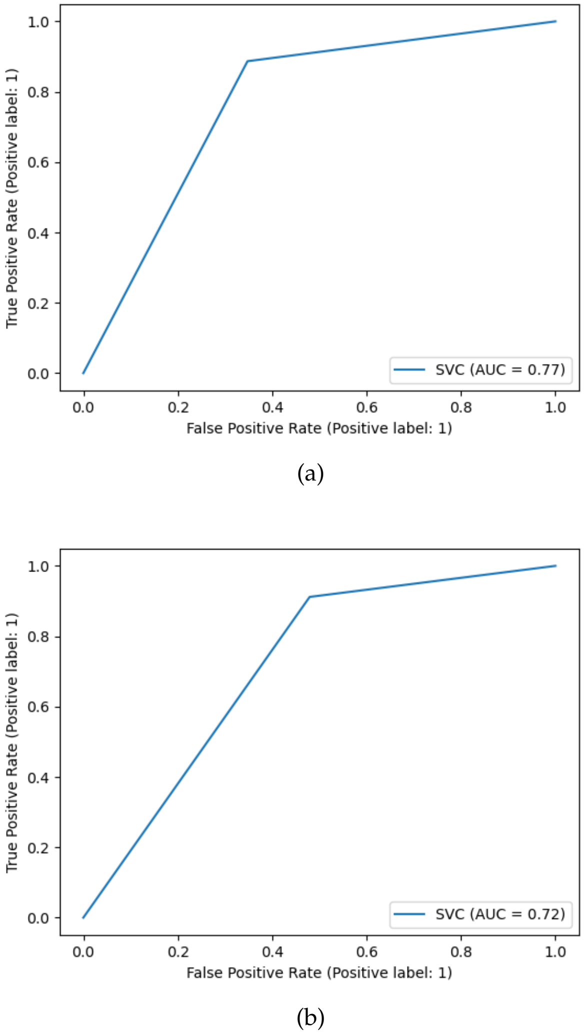
Fig. 7: ROC curves for Valence (a) and Arousal (b) obtained using random forests on facial features test dataset.

Model Type	Preprocessing	Valence	Arousal
SVM (SVM + RF)	-	0.794 (0.839)	0.749 (0.809)

TABLE 4: Tested models for final multimodal predictions.

6 Conclusions

We presented an approach to classify emotions in a lab setting using the VA emotion space and selecting useful features, especially the RPSD that should mitigate the variability of EEG among subjects. Moreover, the Bayesian Optimisation approach used in this work may result useful to gain a new insight on the optimisation of the models used in this kind of tasks, since at our knowledge there are not many related works leveraging on this technique. Although the results appear to be promising, the chosen approach has evident limitations, since it treats emotions as discrete values, providing us with a sort of low granularity of the emotion itself. Furthermore, the controlled lab environment is a constraint that cannot provide us with useful data to draw conclusions on the performance of the models in real-



(a)

(b)

Fig. 8: ROC curves for Valence (a) and Arousal (b) obtained using SVMs for decision-level fusion.

world contexts, hence limiting the range of applicability of this approach. Finally, more extensive tests may be performed in order to retrieve further information about the suggested approach, for example testing other types of models, using different types of EEG/facial features, using three classes for VA predictions or performing a more exhaustive search for the optimum in the Bayesian Optimisation task.

Appendix A
EEG filtering and feature extraction

EEG signals provided by the DEAP dataset are collected into arrays of 8064 samples, with a sampling frequency of 128 Hz. A baseline of 3 seconds is recorded for each trial and it is removed in our preprocessing step, leading to an actual number of 7680 samples per trial. Bandpass filtering is performed through an eight-order Butterworth filter, leading to four EEG bands, i.e. Theta band (4-8 Hz), Alpha band (8-12 Hz), Beta band (12-30 Hz) and Gamma band (30-45 Hz). *Fig 9* shows an example of extracted bands from an EEG signal.

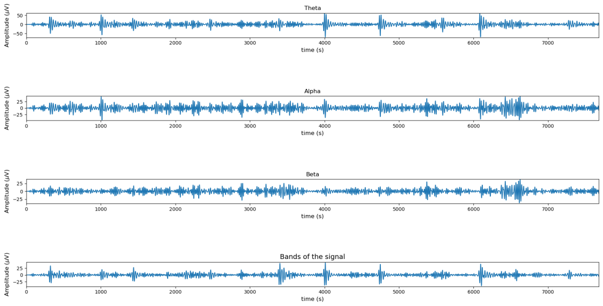


Fig. 9: Example of extracted bands from an EEG signal.

Over the different possible approaches, the computation of Relative Power Spectral Density (RPSD) is preferred, since PSD tends to vary across subjects and at different times even with a fixed stimuli, as suggested in [19]. RPSD should mitigate this phenomenon, taking into account the relative variation of the concerned band (PSD_{BOI}) with respect to the total power spectrum (PSD_{TOT}), as shown in equation 16. Hence, after the filtering step, the average PSD is first extracted for each channel and each band, leading to 32x4=128 features. Welch's method, which is a direct method for the PSD estimation, is used with the *scipy signal* library[5]. More precisely, Hann is used as window, the segments of the signal are fixed to a length equal to the sampling frequency (128) and the overlapping between segments is set at 64 samples.

$$RPSD = \frac{PSD_{BOI}}{PSD_{TOT}} \tag{16}$$

Appendix B
Facial features extraction

Each trial in DEAP has a corresponding video recording of the face of the subject (with some missing trial for a few subjects). Each video is recorded with 50 fps. As a first step of our facial feature extraction procedure, a subsampling is performed on the video recordings, considering only 2 frames per second. These frames are used as inputs for the Mediapipe FaceMesh model, which allows us to detect landmakrs with high accuracy. Detected landmarks are then filtered in order to select only those of our interest, using the specific mapping provided by the mediapipe repository [6]. Moreover, only 2D coordinates are considered for the selected landmarks. An example of detected landmarks can be seen in *Fig. 10*. Then, feature extraction is performed by following the approach suggested in [3].

[5] link to scipy Welch's method
[6] link to Mediapipe landmarks mapping

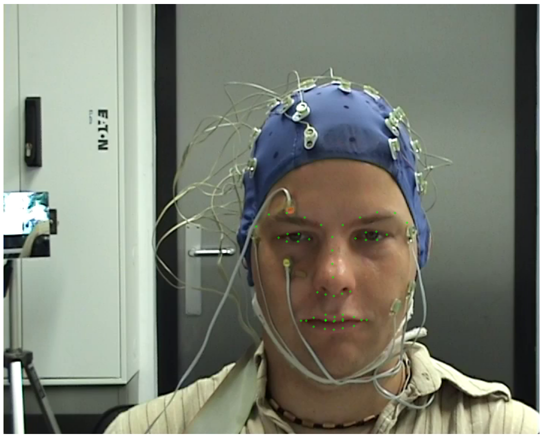


Fig. 10: Example of detected and selected landmarks from a video frame.

- distance between upper and lower lips:

$$d_1 = \sqrt{(x_{58} - x_{52})^2 + (y_{58} - y_{52})^2} \tag{17}$$

- left and right eye openings:

$$d_2 = \frac{1}{2}[(y_{38} - y_{42}) + (y_{39} - y_{41}] \tag{18}$$

$$d_3 = \frac{1}{2}[(y_{44} - y_{48}) + (y_{45} - y_{47})] \tag{19}$$

- left and right angles of the corner of the mouth:

$$\theta_1 = \tan^{-1} \frac{y_{49} - y_{58}}{x_{58} - x_{49}} \tag{20}$$

$$\theta_2 = \tan^{-1} \frac{y_{55} - y_{58}}{x_{55} - x_{58}} \tag{21}$$

- slope of the brows through polynomial fitting:

$$\begin{aligned} k_1 = polyfit[&(x_{18}, x_{19}, x_{20}, x_{21}, x_{22}), \\ &(y_{18}, y_{19}, y_{20}, y_{21}, y_{22}, 1)] \end{aligned} \tag{22}$$

$$\begin{aligned} k_2 = polyfit[&(x_{23}, x_{24}, x_{25}, x_{26}, x_{27}), \\ &(y_{23}, y_{24}, y_{25}, y_{26}, y_{27}, 1)] \end{aligned} \tag{23}$$

Hence, a collection of seven-dimensional vectors (d_1, d_2, d_3, θ_1, θ_2, k_1, k_2) represents the extracted facial features for each frame.

References

[1] M. Soleymani, S. Asghari-Esfeden, Y. Fu, and M. Pantic, "Analysis of eeg signals and facial expressions for continuous emotion detection," *IEEE Transactions on Affective Computing*, vol. 7, no. 1, pp. 17–28, 2016.

[2] D. Li, Z. Wang, C. Wang, S. Liu, W. Chi, E. Dong, X. Song, Q. Gao, and Y. Song, "The fusion of electroencephalography and facial expression for continuous emotion recognition," *IEEE Access*, vol. 7, pp. 155 724–155 736, 2019.

[3] Y. Yang, Q. Gao, X. Song, Y. Song, Z. Mao, and J. Liu, "Facial expression and eeg fusion for investigating continuous emotions of deaf subjects," *IEEE Sensors Journal*, vol. 21, no. 15, pp. 16 894–16 903, 2021.

[4] S. Koelstra and I. Patras, "Fusion of facial expressions and eeg for implicit affective tagging," *Image and Vision Computing*, vol. 31, no. 2, pp. 164–174, 2013, affect Analysis In Continuous Input.

[5] Z. Yin, M. Zhao, Y. Wang, J. Yang, and J. Zhang, "Recognition of emotions using multimodal physiological signals and an ensemble deep learning model," *Computer Methods and Programs in Biomedicine*, vol. 140, pp. 93–110, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260716305090

[6] ——, "Recognition of emotions using multimodal physiological signals and an ensemble deep learning model," *Computer Methods and Programs in Biomedicine*, vol. 140, pp. 93–110, 03 2017.

[7] S. Planet and I. Iriondo, "Comparison between decision-level and feature-level fusion of acoustic and linguistic features for spontaneous emotion recognition," in *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)*, 2012, pp. 1–6.

[8] S. D. Rama Chaudhary, Ram Avtar Jaswal, "Emotion recognition based on eeg using deap dataset," *European Journal of Molecular & Clinical Medicine*, vol. 8, no. 3, pp. 3509–3517, 2021.

[9] S. Parui, A. Bajiya, D. Samanta, and N. Chakravorty, "Emotion recognition from eeg signal using xgboost algorithm," 12 2019, pp. 1–4.

[10] M. Krzywinski and N. Altman, "Classification and regression trees," *Nature Methods*, vol. 14, no. 8, pp. 757–758, Jul. 2017.

[11] S. Rana, C. Li, S. Gupta, V. Nguyen, and S. Venkatesh, "High dimensional bayesian optimization with elastic gaussian process," in *ICML*, 2017.

[12] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010.

[13] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[14] J. Cheng, M. Chen, C. Li, Y. Liu, R. Song, A. Liu, and X. Chen, "Emotion recognition from multi-channel eeg via deep forest," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 2, pp. 453–464, 2021.

[15] R. J. Davidson, "Affective neuroscience and psychophysiology: Toward a synthesis," *Psychophysiology*, vol. 40, no. 5, pp. 655–665.

[16] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap: A database for emotion analysis ;using physiological signals," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 18–31, 2012.

[17] J. Onton and S. Makeig, "High-frequency broadband modulations of electroencephalographic spectra," *Frontiers in human neuroscience*, vol. 3, p. 61, 12 2009.

[18] I. Grishchenko, A. Ablavatski, Y. Kartynnik, K. Raveendran, and M. Grundmann, "Attention mesh: High-fidelity face mesh prediction in real-time," 2020.

[19] M. A. Rahman, A. Anjum, M. M. H. Milu, F. Khanam, M. S. Uddin, and M. N. Mollah, "Emotion recognition from eeg-based relative power spectral topography using convolutional neural network," *Array*, vol. 11, p. 100072, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590005621000205