BENCHMARKING METHODS FOR AUDIO-VISUAL RECOGNITION USING TINY TRAINING SETS

Xavier Alameda-Pineda[‡†], *Jordi Sanchez-Riera*[‡] *and Radu Horaud*[‡]

[‡]INRIA Grenoble Rhône-Alpes and [†]Université de Grenoble

ABSTRACT

The problem of choosing a classifier for audio-visual command recognition is addressed. Because such commands are culture- and user-dependant, methods need to learn new commands from a few examples. We benchmark three state-of-the-art discriminative classifiers based on bag of words and SVM. The comparison is made on monocular and monaural recordings of a publicly available dataset. We seek for the best trade off between speed, robustness and size of the training set. In the light of over 150,000 experiments, we conclude that this is a promising direction of work towards a flexible methodology that must be easily adaptable to a large variety of users.

Index Terms— Audio-visual classification, command recognition, tiny training sets.

1. INTRODUCTION

For the last decade, human-computer interaction methods have rapidly evolved towards flexible multimodal systems. There is a clear need to understand the users commands. In this context, we are interested in the recognition of human audio-visual commands, that is combinations of gestures and short phrases. For instance, a person asks his/her companion robot to perform a task. Because such commands are cultural-, language- and user-dependent, methods need to constantly adapt to a specific user and hence to learn from very few examples. In this paper we cast the audio-visual command recognition task into a multimodal discriminative classification problem. In order to seek out the most accurate AV classification method, we benchmarked three state-of-the-art approaches on *tiny datasets*, e.g., 10-15 instances per class.

Audio-visual discriminative classification approaches can be grouped depending on the way the audio-visual command is represented. *Early Fusion* applies when the representation is audio-visual, *i.e.*, one observation vector corresponds to *concatenated* audio-visual information. *Late Fusion* applies when two different observations vectors represent the modalities (auditory and visual). In the following, we present the existing literature on audio-visual discriminative classification.

Early Fusion: In [1] an audio-visual representation named short-term audio-visual atom is proposed. It is a concatenation of color/texture, motion and auditory features. Targeting semantic concept detection, the method is evaluated on a dataset of 3,000 sequences. A different way to combine audio-visual features at an early stage is proposed in [2], where a bipartite graph quantizes features coming from auditory and visual channels. The authors evaluate the audio-visual event detection performance on a dataset of about 9000 sequences. In [3] audio-visual video concept detection is targeted and the approach consists of concatenating the visual and auditory descriptors, thus forming an audio-visual representation. Tests are performed on a dataset of around 45,000 videos.

Late Fusion: Also in [3], the auditory and visual representation are fused through Multiple Kernel Learning (MKL). This technique is popular because the relative relevance of different kernels is learned from the data. A two-stage strategy is proposed in [4]. First, MKL is used to classify auditory and visual features separately. Second, the normalized scores are merged using a Bayesian model. This is tested in a dataset of about 45,000 videos. In [5] several auditory and visual features are computed. Afterward, they are classified separately and a convex combination of the uninomodal classification scores allows to choose the best audio-visual score. The method is tested on a dataset of 900 videos and 12 classes. Similarly, a convex function is used in [6] to combine the unimodal classification. In this case the dataset consists of more than 200 sequences from 9 different classes. The visual and auditory descriptors are low-dimensional scene-flow features combined with Mel frequency cepstral coefficients (MFCC). In [7] two methods based on feature selection are compared. The complete set of audio-visual features is a 3000-dimensional vector, from which 35 to 70 features are selected. Tests are performed on a data set with 15 training instances per class.

Up to the present, almost all existing approaches have been tested on large datasets, and trained with at least 50 instances per class. This quantity is prohibitive for user-adaptive methods whose discriminative power should be high when trained on tiny datasets (10-15 instances per class). Both [6, 7] deal with such datasets. Albeit, the work in [7] uses sequential forward feature selection, an iterative algorithms that slows down the training process. This is unsuitable for general purpose real-world applications, where methods

need to learn new commands very fast.

The focus of this paper is on the performance of different audio-visual discriminative classifiers using tiny training sets. More precisely, we would like to answer three research questions: (1) which is the best classification method? (2) how the methods' accuracy vary when reducing the size of the training set? (3) does the benchmark correspond to the ones obtained using larger training sets? To answer them, we conducted an extensive set of experiments on a publicly available dataset, thus assessing the quality of different approaches and setting a basis for method comparison. For the sake of generality, we ran the experiments with signals acquired using one color camera and one microphone, a minimal sensor configuration needed to perform audio-visual classification.

The rest of the paper is structured as follows. Section 2 briefly describes the benchmarked methods; Section 3 describes the details of the experiments; Section 4 shows and comments the accuracy of the classification scores, and section 5 draws conclusions and delineates future work.

2. METHOD

The methods we compare in this paper follow the Bag-of-Words (BoW) paradigm, which consists of five different steps: (i) extract local descriptors, (ii) cluster them to get a vocabulary, (iii) map each of the descriptors to the vocabulary, (iv) build a histogram of word occurrence and (v) feed these histogram-based representations to a classifier. During the first three steps, a codebook of size K is built. Subsequent steps are used to represent instances and learn a classifier from these representations. Later for recognition, an unlabeled audio-visual sequence is first represented as a histogram which is fed to the classifier to estimate the sequence's class.

The choice of BoW is justified by the vast literature proving efficiency and robustness. In addition, when the classifier is the Support Vector Machines (SVMs), both the training and testing have closed-form solutions, leading to very fast methods. The power of BoW rises from the quality and quantity of the descriptors as well as the discriminability of the classifier. The use of tiny training sets does not affect the descriptors, but the classifier, which is the precise focus of our study.

The auditory descriptors are based on the MFCC representation since it has been proved to be suitable for describing speech signals [8]. The visual descriptors use space-time interest point [9], which exhibit excellent descriptive capabilities for visual gesture recognition.

Once the descriptors are extracted, the vocabulary is constructed using K-means. The clustering is applied to a sampled set of descriptors. Afterward, the remaining descriptors are mapped into the clusters using the nearest neighbor algorithm. These steps are standard in the BoW framework and generate one histogram of visual words and one histogram of auditory words per training instance.

These histogram-based representations of the training instances are used to learn a multiclass classifier, *i.e.*, a discriminant function $f : \mathbb{R}^M \times \mathcal{C} \rightarrow \mathbb{R}$, where $\mathbb{R}^M$ is the observation space, $\mathcal{C} = \{1, \ldots, C\}$ is the set of labels, and C is the number of classes. A new unlabeled observation $\mathbf{x} \in \mathbb{R}^M$ is classified with:

$$c^*(\mathbf{x}) = \arg\max_{c \in \mathcal{C}} f(\mathbf{x}; c).$$

In the case of SVMs, f has the following expression:

$$f(\mathbf{x}; c) = \sum_{n=1}^{N} \beta_{n,c} k(\mathbf{x}, \mathbf{x}_n),$$

where $\{\mathbf{x}_n\}_{n=1}^N$ is the training set, $k(\cdot, \cdot)$ is the kernel function and $\beta_{n,c}$ are computed during the training phase. Since the auditory and visual representations (histograms) are K-dimensional, the observation space is either $\mathbb{R}^K$ for uniomodal methods (auditory or visual) or $\mathbb{R}^{2K}$ for audio-visual approaches. For clarity purposes, $\mathbf{x}^{AV} = (\mathbf{x}^A, \mathbf{x}^V)$ will denote the audio-visual observations.

We compared five SVM-based methods, with the following discriminant functions f:

AUD: Audio-only

$$f_{\mathbf{AUD}}(\mathbf{x}^A, c) = \sum_{n=1}^{N} \beta_{n,c}^{\mathbf{AUD}} \; k(\mathbf{x}^A, \mathbf{x}_n^A).$$

VID: Video-only

$$f_{\mathbf{VID}}(\mathbf{x}^V, c) = \sum_{n=1}^{N} \beta_{n,c}^{\mathbf{VID}} \; k(\mathbf{x}^V, \mathbf{x}_n^V).$$

CAT: Audio-visual concatenation

$$f_{\mathbf{CAT}}(\mathbf{x}^{AV}, c) = \sum_{n=1}^{N} \beta_{n,c}^{\mathbf{CAT}} \; k(\mathbf{x}^{AV}, \mathbf{x}_n^{AV}).$$

CWS: The convex weighting scheme described in [6]

$$f_{\mathbf{CWS}}(\mathbf{x}^{AV}, c) = \lambda f_{\mathbf{VID}}(\mathbf{x}^V, c) + (1 - \lambda) f_{\mathbf{AUD}}(\mathbf{x}^A, c)$$

MKL: The multiple kernel framework already used in [3]

$$f_{\mathbf{MKL}}(\mathbf{x}^{AV}, c) = \sum_{n=1}^{N} \beta_{n,c}^{\mathbf{MKL}} (\mu k_V(\mathbf{x}^V, \mathbf{x}_n^V) + (1-\mu) k_A(\mathbf{x}^A, \mathbf{x}_n^A)).$$

Notice that the **AUD** and **VID** use only auditory and visual data respectively. Thus, these two methods do not perform any fusion. On the contrary, **CAT** performs *early fusion*, and **MKL** and **CWS** perform *late fusion*. The difference between **CWS** and **MKL** is that, while the first one estimates the SVM coefficients and λ in two different stages, the second performs a joint optimization. A priory, **CWS** is faster but less accurate than **MKL**.

3. EXPERIMENTS

Recall that the aim of this work is to evaluate the performance of different audio-visual classifiers on a tiny dataset. For this, we selected the "Robot Gesture" scenario of the RAVEL dataset [10]. This scenario contains audio-visual recordings of eight different actors, performing nine commands (three times each). These gesture and [voice] commands are: wave ["Hello!"], walk towards the robot ["I am coming."], walk away from the robot ["Bye."], gesture for 'stop' ["Stop."], gesture to 'turn' ["Turn around."], gesture for 'come here' ["Come here."], point ["Look!"], affirmative head motion ["Yes"] and negative head motion ["No"]. Summarizing, the data set consists of nine classes and 24 observations per class, which is suitable for our study.

We evaluated the methods splitting actor-wise the dataset into a training subset and a testing subset several times, following a standard cross-validation strategy. We named the experiments **E**n, where n is the number of actors in the training set. Hence, **E**n is the average of $\binom{8}{n}$ different training sets, in which there are $3n$ observations per class. We conducted experiments for values of $n = 3, 4, 5, 6, 7$, so a total of 218 different training sets.

Since this is the first work (up to the authors' knowledge) that compares audio-visual command classification methods on tiny datasets, we believe necessary to test different possibilities regarding the kernels used and their parameters. The tested kernels are: **[L]** linear $k_L(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\intercal\mathbf{x}'$, **[P]** polynomial $k_P(\mathbf{x}, \mathbf{x}'; d) = (\mathbf{x}^\intercal\mathbf{x}' + 1)^d$, **[G]** Gaussian $k_G(\mathbf{x}, \mathbf{x}'; \sigma^2) = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2/2\sigma^2)$, **[C]** χ^2 $k_{\chi^2}(\mathbf{x}, \mathbf{x}'; \nu) = \exp\left(-\frac{1}{\nu}\sum_{k=1}^{K}\frac{(x_k - x'_k)^2}{x_k + x'_k}\right)$ (where $\mathbf{x} = (x_1, \ldots, x_K)$) and **[S]** sigmoid $k_S(\mathbf{x}, \mathbf{x}'; a, c) = \tanh(a\mathbf{x}^\intercal\mathbf{x}' + c)$. The kernel parameters are: $d \in \{2, 3, 4, 5, 6\}$, $\sigma^2, \nu \in \{10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^{0}, 10^{0.5}, 10^{1}\}$ and $a = 20$, $c \in \{-0.5, -0.25, 0, 0.25, 0.5\}$. The codebook size was set to $K = 500$.

For each sub-experiment (training set) and for all choices of kernel(s) and kernel parameter(s), the five methods were evaluated. Notice that for each sub-experiment there are 25 kernel choices for the methods **AUD**, **VID** and **CAT** and 625 for the methods **CWS** and **MKL**. In summary, we trained more than 150,000 SVMs[1] to present this study.

4. RESULTS

In order to compare different methods and kernels we compute the global accuracy of the classifiers, *i.e.*, the percentage of correct classifications. Tables 1 and 2 show the global accuracies for all the experiments we conducted. Before going into the numeric details we explain how these results are presented in there. First, **M** denotes the method (**AUD**, **VID**, **CAT**, **CWS** or **MKL**), k indicates the kernel used (**L**, **P**, **G**, **C** or **S**) and **E** refers to the experiment (**E3**, **E4**, **E5**, **E6** or **E7**). Second, each entry of the table corresponds to the best kernel parameter. Last, the numbers in bold denote the best kernel(s) choice given an experiment **E** and a method **M**.

Table 1. Accuracy results (%) of the methods **AUD**, **VID** and **CAT** on training sets of different sizes. Bold indicates the best kernel choice.

E	M\ k	L	P	G	C	S
E7	**AUD**	65.3	65.3	64.8	**71.3**	64.8
	VID	59.3	64.4	64.8	**69.0**	65.3
	CAT	74.1	78.2	78.2	**84.3**	77.3
E6	**AUD**	62.2	63.4	64.2	**68.2**	62.4
	VID	58.9	63.3	64.3	**68.5**	64.4
	CAT	73.4	75.9	75.9	**81.5**	75.9
E5	**AUD**	60.2	60.9	61.7	**66.0**	60.3
	VID	58.2	61.6	62.5	**65.9**	62.7
	CAT	72.0	73.5	73.5	**78.8**	73.8
E4	**AUD**	56.0	57.6	58.6	**63.2**	57.6
	VID	56.6	59.6	60.6	**63.7**	60.7
	CAT	69.9	71.5	71.5	**76.0**	71.7
E3	**AUD**	49.0	52.6	54.4	**58.8**	54.2
	VID	54.9	57.0	57.8	**61.0**	57.8
	CAT	66.7	67.9	67.9	**72.4**	69.0

Table 1 shows the accuracy results of three of the methods namely: **AUD**, **VID** (*no fusion*) and **CAT** (*early fusion*). We first notice that the audio-visual method performs systematically better than both monomodal approaches. Second, there is no significant difference between methods **AUD** and **VID**. It is also worth noticing how the accuracy of the classifiers decreases when the size of the training set decreases. Indeed, when there is not enough training data, the classifier does not capture the underlying structure of the data, thus causing an accuracy drop.

Table 2 shows the performance of the methods **CWS** and **MKL**. The columns and rows correspond to the kernel used on visual and auditory data respectively. We remark in the first place that **MKL** works better than any monomodal classifier. However, **CWS** does not: its accuracy is roughly the same as the monomodal classifiers on the smallest training sets. It is also worth to notice that the **MKL** and the **CAT** methods are comparable and both perform better than the **CWS** approach. This last statement is in disagreement with [3], where **MKL** outperforms **CAT**. Albeit, the experimental conditions are not the same. Indeed, both the size of the training set and the number of classes are smaller here. **CWS** shows bad accuracy for smaller datasets compared to **MKL** or **CAT** because **CWS** has to train twice the number of parameters than **MKL** and **CAT**. Moreover, the training of those parameters is performed in each modality independently, not allowing, for instance, the auditory information compensate for visual missrepresentations. Hence, when the size of the training set is reduced, the accuracy drop of **CWS**

[1] At this point we would like to mention the MKL C++ library SHOGUN [11] and thank the reactivity of its developers, specially Sergey Lisitsyn.

Table 2. Accuracy results (%) of the methods **CWS** and **MKL** on training sets of different sizes. Bold indicates the best kernel choice.

E	M k	CWS L	CWS P	CWS G	CWS C	CWS S	MKL L	MKL P	MKL G	MKL C	MKL S
	L	71.8	77.3	79.2	78.2	78.7	76.4	71.8	71.8	69.0	65.3
	P	65.3	66.7	68.1	68.1	68.5	65.3	75.9	75.9	75.9	73.1
E7	**G**	78.2	79.2	80.6	78.2	77.8	71.8	75.9	75.9	75.9	74.5
	C	71.3	71.3	71.3	71.3	71.3	71.3	75.9	80.6	**81.0**	77.3
	S	71.8	76.9	79.6	80.6	**81.5**	64.8	74.5	79.6	77.8	77.8
	L	66.0	68.8	69.8	70.4	70.3	74.2	71.0	70.8	68.5	64.4
	P	63.4	63.7	64.1	64.0	64.2	63.4	74.7	74.8	74.9	70.6
E6	**G**	74.9	**77.2**	76.2	77.0	72.2	70.6	74.5	74.7	75.5	74.3
	C	68.2	68.2	68.2	68.2	68.2	68.2	76.1	79.8	**79.8**	77.0
	S	69.8	72.4	72.4	72.4	72.7	62.4	72.8	76.4	76.9	75.6
	L	62.5	64.8	65.1	65.8	65.2	72.1	68.8	68.7	65.9	62.7
	P	60.9	60.9	60.9	60.9	60.9	60.9	73.2	73.3	72.8	69.5
E5	**G**	72.1	**73.1**	72.6	72.7	68.7	68.6	73.1	73.3	73.4	73.2
	C	66.0	66.0	66.0	66.0	66.0	66.0	74.7	78.4	**78.4**	75.7
	S	65.1	65.5	65.8	65.7	66.0	60.3	64.7	74.2	75.3	74.2
	L	57.6	59.5	59.7	60.4	59.9	69.6	66.5	66.6	63.7	60.7
	P	57.6	57.6	57.6	57.6	57.6	57.6	70.7	70.7	70.5	67.6
E4	**G**	65.5	**67.1**	66.3	66.7	64.2	66.4	70.6	71.1	71.4	71.2
	C	63.2	63.2	63.2	63.2	63.2	63.2	69.1	75.4	**76.3**	73.7
	S	59.9	60.2	60.4	60.8	60.6	57.6	57.8	71.7	72.7	71.7
	L	49.1	49.4	49.6	49.6	49.7	61.1	63.9	63.9	61.0	57.8
	P	52.6	52.6	52.6	52.6	52.6	52.6	66.9	67.0	67.2	65.1
E3	**G**	57.3	**59.9**	59.2	59.8	57.9	63.8	66.7	67.3	68.5	67.2
	C	58.8	58.8	58.8	58.8	58.8	58.8	59.4	72.2	**72.6**	70.7
	S	55.0	55.2	55.6	55.4	55.3	54.2	54.2	68.4	69.7	68.5

Table 3. Time per SVM

Method	AUD/VID	CAT	CWS	MKL
Time Spent [s]	0.79	0.86	1.59	3.27

is stressed.

We would also like to remark that the best kernel to use is the χ^2 with all the tested methods on all the different training set sizes. This statement goes accordingly with the existing literature, and there is a simple explanation for that. When using histograms, differences on full bins are less important that differences in almost empty bins. This kind of touch is exactly what the χ^2 kernel accounts for.

In order to present a final comparison, Figure 1 shows the accuracy results of the five methods using the χ^2 kernel on the different experiments. From this plot it is clear that (i) audio-visual fusion increases the accuracy, (ii) **MKL** and **CAT** perform equivalently, (iii) the **CWS** method outperforms the other fusion approaches and (iv) when the size of the training set decreases, the accuracy drops.

Table 3 shows the average time spent on the training and testing of one multiclass classifier for the benchmarked methods. As expected, monomodal classifiers are the fastest, closely followed by **CAT**. **MKL** is the slowest method, spending more than twice the time used by the **CWS** method.

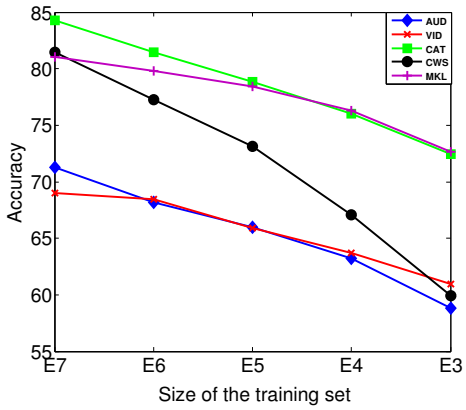

Fig. 1. Best kernel's accuracy for different methods as a function of the training set's size.

At the light of these results we answer noe the original researh questions. In our particular setup, the best trade off between speed, robustness and user-adaptivity is given by **CAT**. When the size of the training set is reduced, all methods experiment an accuracy drop, as expected. We remark that this drop is much more stressed in the case of **CWS**. Finally, the results show that **CAT** and **MKL** react similarly when reducing the size of the training set. This is in disagreement with the literature (see [3]). Notice, however, that the experimental conditions are not the same.

5. CONCLUSIONS & FUTURE WORK

In this paper we addressed the task of audio-visual command recognition, looking for methods with high robustness, speed and user-adaptivity. We present an extensive set of experiments providing for a solid benchmark framework. Since the speed and robustness are provided by the BoW+SVM paradigm, we focused on reducing the size of the training set, thus looking for the method yielding the highest user-adaptability.

Since the tests conducted do not clarify which of the compared methods is the most accurate, extra tests on datasets with more classes will be done in the future. In addition, we would like to perform other tests on audio-visual command datasets recorded in different languages and countries thus having a large variety of gesture and speech utterances, thus evaluating the cultural influence on the proposed approaches and on their results. This will throw the bases for future work towards a continuous audio-visual command recognition method.

6. ACKNOWLEDGMENTS

This research was supported by EC project FP7-ICT-247525-HUMAVIPS.

7. REFERENCES

[1] W. Jiang, C. Cotton, S.-F. Chang, D. Ellis, and A. Loui, "Short-term audio-visual atoms for generic video concept classification," in *Proceedings of the ACM International Conference on Multimedia*, 2009.

[2] G. Ye, I.-H. Jhuo, D. Liu, Y.-G. Jiang, D. Lee, and S.-F. Chang, "Joint audio-visual bi-modal codewords for video event detection," in *Proceedings of the ACM International Conference on Multimedia Retrieval*, 2012.

[3] M. Mühling, R. Ewerth, J. Zhou, and B. Freisleben, "Multimodal video concept detection via bag of auditory words and multiple kernel learning," in *Proceedings of the International Conference on Advances in Multimedia Modeling*, 2012.

[4] P. Natarajan, S. Wu, S. N. P. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan, "Multimodal feature fusion for robust event detection in web videos," in *Proceedings of the IEEE International Conference on Computer Vision an Pattern Recognition*, 2012.

[5] Q. Wu, Z. Wang, F. Deng, and D. D. Feng, "Realistic human action recognition with audio context," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications*, 2010.

[6] J. Sanchez-Riera, X. Alameda-Pineda, and R. Horaud, "Audio-visual robot command recognition: D-META'12 Grand Challenge," in *Proceedings of the International Conference on Multimodal Interaction*, 2012.

[7] J. Lopes and S. Singh, "Audio and video feature fusion for activity recognition in unconstrained videos," in *Intelligent Data Engineering and Automated Learning*, 2006.

[8] L. R. Rabiner and R. W. Schafer, *Theory and Applications of Digital Speech Processing*, Pearson, 2011.

[9] I. Laptev, "On space-time interest points," *International Journal on Computer Vision*, vol. 64, no. 2-3, 2005.

[10] X. Alameda-Pineda, J. Sanchez-Riera, V. Franc, J. Wienke, J. Čech, K. Kulkarni, A. Deleforge, and R. P. Horaud, "Ravel: An annotated corpus for training robots with audio visual abilities," *Journal of Multimodal User Interfaces*, 2012.

[11] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. d. Bona, A. Binder, C. Gehl, and V. Franc, "The shogun machine learning toolbox," *Journal of Machine Learning Research*, vol. 99, pp. 1799–1802, August 2010.