DESIGNING EFFICIENT MULTIMODAL CLASSIFICATION SYSTEMS BASED ON FEATURES AND SVM KERNELS SELECTION

Anca APATEAN
Technical University of Cluj-Napoca, Communications Department

Abstract: An efficient classification system uses only the most representative features extracted from images in order to reach a decision. A multimodal system may consider multiple sources of such information. Selecting those features is not a simple task due to the fact that multiple features selection (FS) methods exists, with multiple setup possibilities and multiple possible feature vectors to be applied on. Moreover, by applying the FS, the new vector may comprise too few features and the recognition accuracy to significantly drop. This paper proposes solutions to compensate that accuracy loss by the SVM kernel selection.

Keywords: *object classification system, features selection, features extraction, SVM kernel selection, multimodal.*

I. INTRODUCTION

Like humans use their senses to relate to the world around them, today machines have to interpret the environmental data, and this is generally accomplished by different signal processing techniques. One of them, i.e. *computer vision* implies acquiring, processing, analyzing, and understanding images to produce numerical or symbolic information, e.g., in the form of decisions.

Computer vision has many applications already in use today in the intelligent vehicle field: road detection and following, scene understanding, pedestrian or vehicle detection, obstacles recognition, tracking, etc. In such applications, machine learning plays a central role: to build computer systems that learn from experience or data. These systems require a learning process, just like humans do; this will specify how they should respond (as a result of the experiences or examples they have been exposed to) to new examples, unknown.

In order to obtain the numerical representation of the data, which have to be recognized by a classification system, some features could be preferred (to encode this information), according to the application domain. To process images, for example, features like wavelets, statistical moments, coefficients of some transforms may be used. In an obstacle detection and recognition (ODR) system, these features were obtained in the features extraction (FE) module, being extracted from the obstacle corresponding images from the visible and infrared domains. Generally, the more varied these features are, the better, as they can retain much complementary information. Still, the main purpose is to extract a compact and pertinent numeric signature of obstacle image, followed by an efficient and as fast as possible classification of it.

Few questions are foreseen here in order to explain the title of the paper: *What means a multimodal classification system? Why is it considered efficient? What designing aspects are envisioned? Why are needed both features and classifiers selection?* Next, essential answers to these questions are envisioned.

A ***classification system*** or classifier is able (in certain conditions, e.g. after a proper learning stage) to identify to which of a set of categories a new observation belongs. To accomplish this, it generally uses a training set of data with examples or observations whose category membership is known. Being a ***multimodal*** one means that in taking the final decision it considers not only one type of data, but multiple ones. These could arise from the same source, from different sources of the same type or even from different sources of different type. For example, in computer vision applications one system may use different information from the same image provided by a visible spectrum camera (e.g. from the RGB channels, the data corresponding to red and to blue channels), or two images from two different visible spectrum cameras (e.g. in the case of stereo images), or even two images from two different spectrum cameras, like visible and infrared (e.g. in the case presented here).

In order to obtain a fast and accurate and thus ***efficient*** recognition system, only the most representative features for each modality used in the system or for the multimodal one (depending on the system type) should be retained. *How to decide which are the most representative features to describe the data in a classification system?* The answer is quite simple: by a ***features selection*** operation. Still, it is not as simple to accomplish due to the fact that multiple features selection methods exists, with multiple setup possibilities (thus also wondering *How to apply these features selection methods?*) . In the presented case these also may be applied to multiple possible feature vectors. Another possible drawback is that by applying the features selection, the new vector may comprise too few features and the new accuracy to significantly drop. The proposed solutions envisioned to compensate that accuracy loss by the ***SVM model selection*** (i.e. the kernel type and the hyper-parameters). A bi-optimization criteria was used and the designing aspects refers to how the features selection (FS) and kernels selection (KS) were accomplished to speed up the system.

Although these methods were applied in the frame of an ODR system to process VIS and IR images, other possible usage may be inferred.

The proposed solutions may function with two possible

Manuscript received May 3, 2016; revised September 21, 2016

databases, one working with monomodal data and the other working with multimodal data, as presented in the following sections. Even a bimodal situation was used, the adopted algorithms and methods may be easily adapted to a multimodal condition. The main purpose of this paper is to present multiple investigations, as concerns the system optimization by two different criteria: the accuracy of the recognition but also the computational time.

II. DESCRIBING THE CLASSIFICATION SYSTEM

In most classification systems, the FE module is followed by a features selection (FS) one, as shown in Figure 1. In the FS stage, the importance of the features previously extracted in the FE module and combined (in a features fusion step) in a single feature vector denoted FV, is estimated. Within the FS stage, only the features that are most relevant will be chosen to further represent the information. The resulted vector will be one containing only the selected features and it will be denoted sFV. In Figure 1, the features fusion module refers to the fact that different families of features were combined in a single FV.

One important task to consider, when trying to develop a robust model for an object classification system, was to use features and classifiers (i.e. SVM kernels) selection to better adapt the information specificity to the aimed classification problem.

For the classification task, an extensive number of combinations between types of features and classification algorithms have been tested during the last years. For example, in order to solve the road obstacle categorization problem, motion and appearance information was used with an AdaBoost cascade approach [1], HOG features were processed by an SVM [2], multiple cues were used within a neural network [3], HOG and Haar wavelets within an AdaBoost classifier [4] among others. These systems or the methods within them are generally difficult to compare because they are rarely tested on a common data set and with the same experimental setup. Just recently some authors compared the results previously obtained by them or by other research teams on the same database [5,6,7].

The road obstacle classification system uses different global texture features organized as feature families (FF) which have been extracted from visible (VIS) and infrared (IR) images. The features corresponding to 8 different FFs were obtained in the FE module and a FV was thus constructed. Then, the FS process followed, where only the most relevant features were retained (comprised in a sFV) for time reduction reasons.

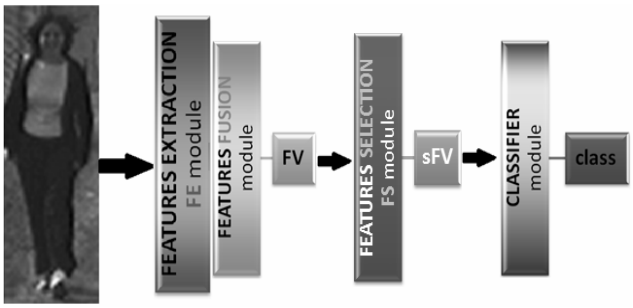

Figure 1. The FS step in the frame of a classic computer vision-based system.

A. Details about the FE module

For the ODR system, different types of features for VIS and IR images were investigated in order to find the best features combination to assure an efficient classification. Thus, in the FE module more features were extracted; these were organised in families: the first FF, denoted FF1, comprises only 2 geometrical features, as illustrated in Table 1, i.e. the width and the height of the original bounding box comprising the obstacle image (for both modalities); the other 7 FFs were separately extracted from each modality, VIS and IR in our case, and they were: 7 statistical moments, 64 Haar wavelets, 32 Gabor wavelets, 8 Discrete Cosine Transform (DCT) coefficients, 16 grey level coocurence matrix coefficients, 14 run length encoding features and 28 Laws features.

Table 1. Features Families (FF) used in the classification system to compose the Features Vector (FV)

Geometric features	FF1	2
Statistical moments	FF2	7
Haar wavelet	FF3	64
Gabor wavelet	FF4	32
DCT coefficients	FF5	8
Coocurence matrix	FF6	16
Run Length Encoding	FF7	14
Laws features	FF8	28

The number of features inside each FF is known, and it is the number of features decided to be extracted, so it may vary from one FE method to another. In this way, there are 2 common features for FF1, and for each modality the next modality-specific 7, 64, 32, 8, 16, 14 and respectively 28 features, as shown in Table 1.

Next, these FFs were combined in a way to obtain the monomodal vectors [FF1,FF2(M1), FF3(M1), …, FF8(M1)] or [FF1,FF2(M2),FF3(M2), …,FF8(M2)] and respectively the bimodal vector [FF1,FF2(M1),FF3(M1), …,FF8(M1), FF2(M2),FF3(M2), …,FF8(M2)], in the parenthesis being specified the modality from which those features were considered. The monomodal vectors have a length of 171 features, while the bimodal one has 340 features. For the general case, these values can be simply denoted by n or n1 and n2 when modality is also suggested, respectively n12 for the bimodal vector length.

In order to perform the FS task, Weka [8] was used and Figure 2 shows how a monomodal vector (with first and last features observable) looks like. Such FV is extracted for each of the 321 pedestrians, 329 vehicles, 45 cyclists and 237 background image objects, these being the training set.

Figure 2. An arff file with a monomodal vector in Weka

B. The problem to be solved

The problem to be solved when approaching the FS and the KS was to assure a real time processing for the proposed system but still providing a high accuracy if possible. Therefore, the reduction of the computational time corresponding to the FE step and to the classification mechanism was quite necessary. By reducing the FV dimension, the classification time will diminish too, this being generally accomplished in any classification problem. Moreover, if this reduction of the FV is achieved with a slight decrease of the accuracy rate, the system can be considered quite robust.

Standard classifier accuracy (Acc) is obtained based on all four combinations of true/ false and positive/ negative factors like, equation (1) shows:

$$Acc = [TP/(TP+FN) + TN/(TN+FP)]/2 \qquad (1)$$

Instead of equation (1), the arithmetic average of class accuracies, which is called a balanced accuracy (bAcc) was used; it is a particularly useful evaluation measure for unbalanced datasets, defined by the equation (2):

$$bAcc = \frac{1}{K}\sum_{i}^{K} F(y_i \mid y_i) = \frac{1}{K}\sum_{i}^{K} TPR(i) \qquad (2)$$

where K is the number of objects classes, and TPR is the true possitive rate computed for each class.

Since FE is desired to be fast, the performances of the entire system depends heavily on the choosen features. If the features extracted from different modalities are quite different as concerns their representing scales, in order to be properly combined in a multimodal vector, they should all be normalised in the same domain, thus also prior to their selection.

The experiments developed in the FS module were organized only as tests, performed on different scenarios. In addition, in the first attempts to approach the FS algorithms, either a proper database for the final goal was available, i.e. to perform the fusion between VIS and IR images. Still, we used the most proper one we have at that time, the Robin database [9], which contained VIS and IR images, but not-correlated eachother. Thus, the obstacles from VIS database do not have a corresponding obstacle in IR (same obstacle, same pose). The first set of developed experiments were performed on this Robin database and we did select features on each modality VIS and IR, but the results were not further used (for the last module, thus the fusion task). Only later, a proper database for performing the VIS-IR fusion was received (this was the last database we used, i.e. the Tetravision, provided by VisLab from Parma).

C. Details about the employed databases

Several companies and research centers have engaged in the past in the Robin project [9]. This competition was for the evaluation of object detection, object recognition and image categorization algorithms based on VIS and IR images not correlated each other. We subscribed for the dataset produced by Bertin – Cybernetix, where the proposed dataset was made of colour and infrared images of vehicles and pedestrians. The task was to discriminate humans and vehicles, so the goal was to assign the correct label to a patch which may contain an element of a class or some backgrounds. The experiments performed on the Robin database implied classifying with 5 classes (Standing Person, Unknown Posture, Motor Bike, Tourism Car and Utility Car). There were 1406 objects (train) and 691 objects (test) in the visible domain, and 1659 objects (train) and 1050 objects (test) in the IR domain.

The second database which was used comprises VIS and IR images (i.e. the Tetravision database), being designed to recognize the type of obstacles previously determined as regions of interest by a stereo-vision obstacle detection module. Very few systems based only on passive sensors and performing VIS - IR information fusion exist. Among them, the Tetra-Vision system proposed at VisLab [10], [11] at the University of Parma, was designed for pedestrian detection from four stereo correlated VIS–IR images. As a first attempt, there were only 486 objects, divided in 389 (train) and 97 (test), but for the final setup, there were 1164 objects. Even the database is small, it is a very difficult one, because of the high intra-class variability. The database was randomly divided into a training set (80%) and a testing set (20%), the class instances being well balanced between the training and testing sets. For more details and information, please consult [12].

The first database comprises two sets of data which cannot be interpreted in the same way, while the second one allows it. Thus, in the first case the database comprises more monomodal subsets of data, while the second database has multimodal data.

As previously mentioned, the main reason to perform the FS task was the reduction of the time needed by the system to classify a new test object. Having this in mind, few questions were quite obvious to be addressed: *Will the retained features, organized as sFVs, lead to some improvements regarding the classification time, compared to the original feature vectors (FVs)? Would the sFVs maintain the high accuracy rates? Or, if these will degrade, how much will be lost? Can we compensate this, somehow? Can a classifier selection task help in this case?* Thus, in the experiments we developed, we tried to find answers to these questions; which are possible solutions, and which one we considered best will be presented in the following sections.

III. APROACHING THE FS TASK

For the FS operations, there are multiple variants, concentrated on two fundamental directions: filters and wrappers. These differ mostly by their evaluation method. In the presented experiments, only filters were used, as they are generally faster. For any filter method, *an attribute evaluator* should be mentioned; this evaluator could be applied: to *individual features*, as for the *Ranker* methods or to *subset of features*, as for the *Search* methods.

Search methods get through the attribute space to find a good subset, and the quality of the respective subset is measured by an attribute subset evaluator.

The most utilized search methods in Weka are detailed presented in [13]. One of the search methods is the *Best First* one; this searches the features space by greedy hill climbing combined with a backtracking facility. The method is implemented in such a way that it does not use a stopping criterion based on the performance reduction. Instead, it has a parameter that specify how many consecutive nonimproving nodes must be encountered before the system backtracks, as presented in [13]. In this way, the exploration

of the entire search space is assured, in a *forward* (starting from the empty set of attributes), *backward* (starting from the full set), or in *both directions (*starts at an intermediate point) [14].

Cross-validation is generally used for model selection (e.g. find the best classifier or kernel, find the best regularization hyperparameters), but it could be also applied for FS, since cross-validation provides an estimate of the generalization ability of models. The ability of models to predict depends both on the used features and on the complexity of the used model. We have considered the use of the cross-validation process also in the FS stage, and indeed the time needed for the algorithm to perform the selection process is much more increased compared to the situation in which no cross-validation was applied. Still, the possible benefits are from both sides: the accuracy of the recognition and the classification time.

It is essential to estimate the computational burden of algorithms for FS problems, the computational time being generally strongly influenced by the *search strategy* and by the *evaluation criteria*. The evaluation criteria may also be expensive as it may involve training a classifier or comparing every pairs of examples or features. The fact that this processing of FS methods performed in the cross-validation loop takes much more time than another method not using the cross-validation process, is not critical for our system, it is not even affecting our system. No real-time operation is required in this stage, because the FS operation is performed off-line, when the system is not running on the road. Here, the system is just preparing for the real situation, thus for the online functioning.

In Weka, the FS methods could be applied directly, *on the full training set of data* when the FS method is applied only once on all the data from the training set or *by a cross-validation technique* when the FS method is applied on each individual fold of data; in the latter case, there is a number equal to the number of folds for the application of the respective FS method; here, a number of 10 folds have been chosen and this situation is denoted *10f-CV*.

Two methods were used for features evaluation, both combined with a *Best First search algorithm:*

The *CfsSubsetEval* evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Generally, subsets of features with low inter-correlation, but highly correlated with the class are preferred by this algorithm. The forward selection was used and the search stoped if 5 consecutive fully expanded subsets showed no improvement over the current (best) subset. [13]

The *Consistency Subset Evaluation* (or simply denoted Consistency) evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of features. It consider out of 10% from the total number of instances (the training data), run the algorithm and check the inconsistency criterion based on its selected features on the remaining 90% of the data. Then, add those patterns causing inconsistencies to the training data and run again the algorithm. This process continues until the number of inconsistencies is below a tolerable value [13].

In the first experiments, only one of these two *feature evaluation methods* were intended to be retained for further investigations.

IV. APROACHING THE KS TASK

All the proposed solutions implied to compensate the accuracy loss by the SVM model selection. The SVM classifier is a complex classifier [15,16,17], which according to the combination of its hyper-parameters, but also considering the complexity value, may offer better or worse performance on a specific dataset. Even the smallest variation may influence its behavior.

Suppose the training data as a set of instance-label pairs

(x_i,y_i), i=1,...,m where $x_i \in R^n$ represents the input vector and

$y_i \in \{-1,+1\}$ the output label associated to the corresponding

item x_i. The parameter n represents the input vector dimension, where x_i corresponds to $(x_i^1, x_i^2,..., x_i^n)$. These vectors will be mapped into a feature space using a kernel function K, which defines similarities between pairs of data. The kernel functions from the two modalities VIS and IR could be of different types, and could work with different hyper-parameters.

A first possible method to perform the KS was to use a grid search for every type of classical SVM kernel, the SK

type we denoted, SK ∈ {RBF, POL}, the linear one being a

particular case of the polynomial one. The effect is that a classifier will be used to process the first modality information and a possible different classifier will be used on the second one. Different types of SKs were tested: RBF, linear and polynomial, with the kernel parameters and the penalty parameter, denoted *C,* representing the values to be optimized (these have less than two parameters to optimize).

Another possible solution considered in order to compensate the accuracy loss by a KS, is to more deeply intervene on the classifier side: to use an improved kernel (i.e. a multiple kernel, denoted MK) for the SVM, as suggested in [18]. By performing the fusion between the extracted and selected features with different MKs, for the SVM classifier, we generally obtained an accuracy higher than the ones obtained with different classic types of kernels (i.e. SKs).

For the VIS-IR fusion case, a MK learned as a linear combination of two kernels, was proposed:

$$MK(x_i, x_j) = \alpha \cdot SK_{VIS}\left(x_i^{1,k}, x_j^{1,k}\right) + (1-\alpha) \cdot SK_{IR}\left(x_i^{k+1,n}, x_j^{k+1,n}\right), \quad (3)$$

where the single kernels SK_{VIS} and SK_{IR} could be any type of kernels (with similar or different hyperparameters).

Each simple kernel is involved with a weight that represents its relative importance for classification. The weighted value α allows the system adjustment to the VIS or IR domain according to the context. If an object from VIS domain is quite difficult to be detected or classified only from the VIS image, we can consider its counterpart from the IR domain, where the object intensities are higher and

uniform. Thus, the role of the α weighting parameter is to reinforce the importance of the domain which is more significant for the obstacle classification in a specific situation. The kernel selection process, with the optimization of the hyper-parameters, but also of the weighting value α has been described in [19].

The difference between SK and MK can be explained in a common scenario of a bimodal system. One can either use two SVM classifiers, thus with different SKs on the two different modalities, either use a single SVM classifier with a MK. The parameter C can be different in the first case, but in the second one, C is the same on both modalities. The entry vectors are also different: there are *two unimodal vectors* in the first case, while in the second one there is *a single bimodal vector*.

If there is more than one combination of parameters (so we have different kernels which act the same regarding the classification accuracy) the selection of the winner kernel was imposed to be made according to the shorter mean classification time from the cross-validation step. This was called ***a bi-objective optimization***: the accuracy must be as higher as possible, while the classification time, the lower, the better.

V. DATABASE WITH MONOMODAL DATA

All the investigations developed as first attempts for performing the FS will be presented in this section, as accomplished for the first setup of the experiments, thus on the Robin database.

A. The tested FS methods on monomodal vectors

The first tested method, i.e. *CfsSubsetEval* is based on *Correlation* and it is denoted FS_R1 in what follows. The second method was the *Consistency Subset Evaluation* (*Consistency*), denoted FS_R2 in what follows. Based on the obtained results, one single FS method should be retained in order to implement the final system. Thus, two of the most used FS methods from the specialty literature were tested and the sFVs presented in Table 1, in the left half of the table were obtained. The sFVs obtained by the 2 methods FS_R1 and FS_R2 comprise 42 VIS features and 30 IR features, respectively 10 VIS features and 9 IR features.

The features selected with FS_R1 represented only 25% and 18% from the corresponding 171 VIS features and respectively 171 IR features from the initial FVs. Similar, the selected features with FS_R2 are only 6% respectively 5% of them (the selected features from each corresponding sFV differed also by their ordinal number relative to the family they belongs).

For the general case, the number of features selected will be denoted by x and will be reported to the entire number of features from the respective monomodal vector. It also worth mentioning that there may be some features appearing in one modality, but not in the other modality. For example, with the Cfs Subset Evaluation FS_R1 method, the FF5 and FF8 (i.e. *dct* and *laws* here) features were missing in the first modality (i.e. VIS in the presented case), while they appeared in the second one (i.e. IR). Similarly, using the Consistency Subset Evaluation FS_R2 method, FF2 and FF8 (i.e. *statistical* and *laws*) features appeared just in the second modality.

The same remark can be done for features appearing in both domains, but with different ordinal numbers. For example, in the FS_R1 case, FF2, FF3, FF4 and FF6 (i.e. *wavelet*, *statistical*, *cooc* and *gabor*) features appeared in both domains, but with different orders for the respective features; this means they are different features even they belong to the same family; to conclude in this stage: multiple quite different sFVs resulted.

B. Solutions proposed with monomodal vectors

For the first possible solution, different combinations of the SVM hyperparameters were tested on the original vectors, on each domain. In this manner, the best SKs were found on each modality: for the RBF kernel, we obtained $C = 50$, and $gamma = 0.2$, for both domains VIS and IR, while for the polynomial and linear case we found $C = 25$ (for VIS) and $C = 100$ (for IR), and the polynomial degree $d = 5$ for VIS and d=4 for IR as presented in [20].

In all these cases, the input vector was the one containing all the 171 features, but Weka also considered the class of the object, so that appeared 172 in the GUI (as presented in Figure 2). The best accuracy values obtained were with a POL kernel on the first modality, i.e. 94.80% and an RBF kernel on the second modality, i.e. 94.27% as presented in Table 2. Using the same previously obtained kernels, SK type, also the sFVs (resulted after the application of the FS process on the initial FVs) were evaluated.

Table 1. Selected features in all the four developed experiments

	Search method + Cfs (FS_R1)		**Search method + Consistency (FS_R2)**		**Search method + Cfs (FS_T1)**		**Search method + Cfs by 10f-CV (FS_T2)**	
	First database				**Second database**			
First modality	42 25%	2 of 2 - FF1 4 of 7 - FF2 27 of 64 - FF3 6 of 32 - FF4 0 of 8 - FF5 1 of 16 - FF6 2 of 14 - FF7 0 of 28 – FF8	10 6%	2 of 2 - FF1 0 of 7 - FF2 8 of 64 - FF3 0 of 32 - FF4 0 of 8 - FF5 0 of 16 - FF6 0 of 14 - FF7 0 of 28 – FF8	17 10%	0 of 2 - FF1 0 of 7 - FF2 7 of 64 - FF3 1 of 32 - FF4 0 of 8 - FF5 0 of 16 - FF6 5 of 14 - FF7 4 of 28 – FF8	8 5%	0 of 2 - FF1 0 of 7 - FF2 5 of 64 - FF3 0 of 32 - FF4 0 of 8 - FF5 0 of 16 - FF6 2 of 14 - FF7 1 of 28 – FF8

Second modality	30 18%	2 of 2 - FF1 1 of 7 - FF2 16 of 64 - FF3 1 of 32 - FF4 1 of 8 - FF5 5 of 16 - FF6 0 of 14 - FF7 4 of 28 – FF8	9 5%	2 of 2 - FF1 1 of 7 - FF2 5 of 64 - FF3 0 of 32 - FF4 0 of 8 - FF5 0 of 16 - FF6 0 of 14 - FF7 1 of 28 – FF8	25 15%	0 of 2 - FF1 0 of 7 - FF2 12 of 64 - FF3 4 of 32 - FF4 1 of 8 - FF5 0 of 16 - FF6 7 of 14 - FF7 1 of 28 – FF8	12 7%	0 of 2 - FF1 0 of 7 - FF2 4 of 64 - FF3 2 of 32 - FF4 1 of 8 - FF5 0 of 16 - FF6 5 of 14 - FF7 0 of 28 – FF8

As presented in Figure 3, given at the input the initial FV comprising all initial monomodal features, i.e. 171 in the present case, or n in a general situation, the SVM kernel selection is performed and a selected SVM kernel, SK type results. Next, by using this sSK for each corresponding modality, also on the sFVs this was applied. The obtained accuracy values were smaller than those obtained with the initial FVs, as presented in Table 2; these differences were computed in percentage, to be more noticeable when reported to the initial FVs (last two columns). Smaller percentages were aimed if they are negative, or by contrast higher ones if they are positive. The second column illustrates the size of the new FV, so sFV, which is better if it is smaller.

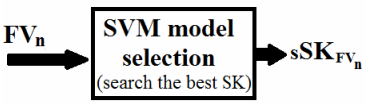

Figure 3. SK model selection based on the initial FV.

Table 2. Results obtained on 1st database with sSK_{FVn}.

FV	FV size	Acc on M1[%]	Acc on M2[%]
Initial FVs	***FVn***	***94.80***	***94.27***
sFVwith FS_R1	sFV_M1_{42}=25%; sFV_M2_{30}=18%	86.19 -9%	87.90 -7%
sFVwith FS_R2	sFV_M1_{10}=6%; sFV_M2_{9}=5%	78.38 -17%	82.47 -13%

With the same kernels previously used on VIS and on IR, as for the initial FV with 171 features, also the evaluation of the sFVs were accomplished and the following were obtained: on VIS an accuracy of 86.19% for FS_R1, respectively 78.38% for FS_R2 and on IR an accuracy of 87.90% for FS_R1 and respectively 82.47% for FS_R2. To deeply analyze this aspect, the lost in percentages was computed for the accuracy: 9% and 7% for the FS_R1, respectively 17% and 13% for FS_R2, as compared to the accuracy of the corresponding initial FVs. The analysis could be transposed on a multi-modality classification problem, in a similar manner.

If the percentage values from Table 2 are positive or they meet the requirements established in the first, designing phases of the system, then this first solution may be adopted. If not, as it was in the presented case, proceed further for another possible compensation of the accuracy loss.

The accuracy decreased after the FS step, and thus the performances obtained with the sFVs were degraded. But, in compensation, the processing time was lower (which is very important for the real time request) than using a large number of features. Next, the implications of a smaller FV were compared from the time perspective: the time needed for the FE process shows that the computation time in the case of the initial FVs is greater than the time needed to compute the sFVs. The *feature extraction time* but also the *classification time* for one object were analysed. The *feature extraction time* (FE time) is used to extract all the needed features for an object after it was previously identified as a possible obstacle by the detection module. Then, the corresponding object is classified by the recognition module, and this is accomplished in a specific amount of time, denoted the *classification time*.

In order to compare these two time indicators, the ones from only one modality were aimed, i.e. the VIS one, as for the other modality is quite a similar interpretation.

Considering the case for the vectors from the VIS domain, and as reported to the corresponding time obtained when using the initial vector with all features, the specific amount of time were reduced with 23% at FS_R1 and 51% at FS_R2 for the FE time; respectively reduced with 78% at FS_R1 and 84% at FS_R2 for the classification time. Thus, the most significant reduction of the processing time was for FS_R2: from the FE time ½ was saved, respectively a little more over 4/5 saved from the classification time, but with the inconvenient to loose 17% of the accuracy. In the FS_R1 case, the time savings were not so effective: only almost 1/2 of the amount was saved with FS_R2, which is approximately 1/4 from the FE side, respectively almost 4/5 on the classification, but as concerns the accuracy only 9% was lost, so almost a half from the loss recorded with FS_R2.

The first decision, to choose between the two features evaluation methods was taken: quite obvious, the interest was concentrated more on the FS_R1 method as it presented the smallest compromise between time and accuracy reduction. The FS_R2 was too radical, and implied too much loss from the accuracy side. Thus, in the investigations which followed, the Best First search combined with the Correlation based evaluation of features, was considered best for the presented system.

Another variant, which may constitute ***the second solution*** is to perform the SVM model selection for each individual sFV as suggested in Figure 4. The sFVs were considered different if the number of features was not equal, or if the modality was not the same, or even if a single feature was different in two sFVs having the same number of features.

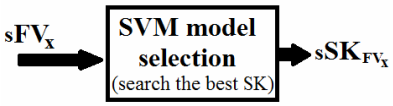

Figure 4. SK model selection based on the selected features, thus using sFV.

To conclude, the *first possible solution* was to compensate the accuracy loss: to perform a modality-based SVM model

selection, so to select the best SVM kernel, denoted SK, on each modality. This selection could be performed based on the initial monomodal FVs, comprising all the extracted features (n) or directly on the sFVs, thus comprising only the selected ones (x). By the application of these selected classifiers on the vectors containing the selected features, the accuracy values may be analysed. Based on this analysis, the solution may be adopted or discarded.

VI. DATABASE WITH MULTIMODAL DATA

First, we also applied the FS technique to the 2nd database, to obtain the sFVs with fewer coefficients than the entire VIS_{171} and IR_{171} or $VISIR_{340}$ feature vectors, and thus to reduce the processing time. Therefore, as presented in [21], we computed the mean of the accuracy and of the classification time (corresponding to one object) like in the previous experiment we developed (on the Robin database).

A. The tested FS methods on a bimodal vector

The Cfs Subset Evaluation method based on correlation was also chosen, combined with the Best First search as shown in Table 1, but here the algorithm was applied to the ***bimodal VISIR vector***. The FS method was thus applied on the ***fused feature vector*** $VisIr_{340}$ or $M1M2_{n12}$ in the general case. In the same way, the selection process considered the learning set. After applying the FS method, a number of 42 features were selected, with 17 features from VIS and 25 features from IR. The selected features represent only 10% and respectively 15% from the initial corresponding FVs, as shown in the right part of the Table 1. Using these selected features, 3 sFVs were thus computed: the first, a bimodal one, comprised all the 42 features ($VisIr_{42}$), the second one contained only the visible ones (VIS_{17}) and the last one with only the corresponding infrared features (IR_{25}). This FS process will be referred in what follows as FS_T1. Next, the same FS method was applied once again, but this time using a 10-folds crossvalidation technique, to evaluate the features individually. As a result, a number of 20 selected features (8 VIS and 12 IR) were obtained, these being also presented in Table 1. The selected features represent only 5% and respectively 7% from the initial corresponding FVs. In what follows, this processing will be referred as FS_T2 and the corresponding sFVs as $VisIr_{20}$, VIS_8 and IR_{12} or sFV_Mi_x (with the modality *i* and x selected features) in the general case as shown in Table 4.

To conclude, a total number of six sFVs were tested in this second experiment as presented in [21]. These computed sFVs have fewer features than their monomodal or bimodal corresponding initial FVs.

B. Solutions proposed with multimodal vectors

The first possible solution proposed to compensate the accuracy loss, was also tried on the Tetravision database, but this time the VIS and IR images were correlated eachother.

Like Figure 3 shows, given at the input the initial monomodal FV comprising all features, the SVM kernel selection was performed and a selected SK results. Next, by using this sSK for each corresponding modality, the obtained accuracy values were smaller than those obtained with the initial FVs, as presented in Table 3; these differences were also computed in percentage, as reported to the initial FVs. Once again smaller percentages were aimed if they were negative ones. The results presented in Table 3 are only for the RBF kernels, as they were better than their polynomial counterparts.

A second variant of this possibility was to test also with the bimodal vector VisIr; not surprisingly, the results were best (last column in Table 3). Still, all the obtained accuracy differences are negative, which means that by applying the FS task, the performance (as concerns the recognition rate) was diminished.

Table 3. Results obtained on the 2nd database with sSK_{FVn}

Initial	**FV size**	**Acc on modality [%]**		
		M1	**M2**	**M1M2**
FVs	***FV_n***	***97.1***	***97.1***	***97.1***
sFV with FS_T1	sFV_M1_{17}=10% sFV_M2_{25}=15%	86.19 -8.6%	87.90 -2.7%	95.30 -1.9%
sFV with FS_T2	sFV_M1_{8}=5% sFV_M2_{12}=7%	78.38 -11.6%	82.47 -6.2%	92.10 -5.1%

Considering that the selected features comprised only 12.5% for FS_T1, respectively 6% for FS_T2, we considered the obtained results quite good. Accuracy has increased with 1% only by the application on the bimodal vector.

The second solution we proposed was also applied in this case, so the SVM model selection was performed for each individual sFV. The obtained results are presented next, in Table 4.

Table 4. Results obtained on the 2nd database with $sSK_{FVx.}$

Initial FVs	**FV size**	**Acc on modality [%]**		
		M1	**M2**	**M1M2**
	FVn	***97.1***	***97.1***	***97.1***
sFVwith FS_T1	sFV_M1_{17}=10% sFV_M2_{25}=15%	89.7 -7.6%	95.9 -1.2%	96.9 -0.2%
sFVwith FS_T2	sFV_M1_{8}=5% sFV_M2_{12}=7%	87.5 -9.9%	94.9 -2.3%	94.9 -2.3%

As it was expected, all the accuracy differences obtained with the second proposed solution are smaller than the ones obtained with the first solution previously presented. This is due to the fact that the SK was optimized on that specific sFV. Even so, they are still negative.

The third solution can be applied only on bimodal data and implies the use of a MK instead of a SK, as shown in Figure 5. Thus, for an MK of type RbfRbf, as presented in [21] a 96.9% accuracy was obtained for FS_T1, respectively 95.9% for FS_T2 (as presented in Table 5). The accuracy values assure only -0.2%, respectively a -1.2% difference related to the initial FV (the ones comprising 340 features). Most important, with only 12% for FS_T1 and even a half, i.e. only 6% features for FS_T2, the accuracy is only a little smaller than that obtained with the original vector. The conclusion was that the reduction of the number of features was too significant.

The third proposed solution implies the optimization of the MK's parameters set (kernel, α , p_1 , p_2 ,C) on a learning set using the 10 fold-crossvalidation method. This means we were looking for the MK parameters for which the best mean recognition rate was acquired after the cross-validation process. The input vector x_i from equation (3) will be divided in $x_i^{1,k}$ for VIS domain and $x_i^{k+1,n}$ for IR, with $k \in \{1,2,...,n\}$. In the presented case, for the vectors obtained with FS_T1 and FS_T2, k is 17 and respective 8,

with the corresponding input vector dimension: n=42 and respectively n=20. A grid search was performed for every type of SK or MK, with the kernel parameter and the penalty parameter C representing the values to be optimized.

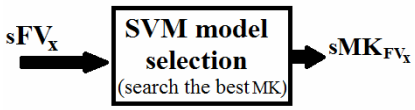

Figure 5. MK model selection based on a specific sFV

Table 5. Results obtained on 2nd database with sMK_{FVx}

Initial FVs	**FV size** FV_{340}	**Acc on M1M2[%]** ***97.1***
sFVs with FS_T1	$M1_{17}M2_{25}$=12%	96.9 => -0.2%
sFVs with FS_T2	$M1_8M2_{12}$=6%	95.9 => -1.2%

Considering the highest accuracy and the lowest classification time, RBF(IR_{12}) was the best processing from all the monomodal SKs and RBF($VisIr_{42}$) the best one for the bimodal SK; for the MK, the best combination was obtained with Rbf_1Rbf_2 ($VisIr_{42}$). Also, the accuracy was higher in the case of using concatenated features than in the case of separate feature-vectors, so bimodal vectors act better than monomodal ones when classified by a SK, i.e. using a classical SVM. In addition, the classification time for the SK is lower as compared to a MK. Still, the bimodal SK does not allow the system adaptation to the VIS-IR context (so the adaptation of the system to different environmental conditions), like MK does through the weighted parameter α .

The fourth solution was considered to reinforce the system based on fusion. This could be accomplished at different levels, so we tested them in order to decide which one is best for our system. By now, as presented in [21] a **feature fusion approach** was proposed, by the construction of a bimodal vector, but also a **kernel fusion** approach by the use of the MK inside the SVM. In [22] a relatively new type of fusion was considered, i.e. the **fusion at the classifier matching-scores level**. The results obtained by all types of fusion have been compared, but also we provided the results for a simple classifier, KNN with K=1 and K=3. The best results were for the matching-scores fusion: only by these approaches, the difference for the accuracy values, as compared to the initial monomodal FVs, was positive. In this way, for VIS_{17} and IR_{25}, thus with only 12% features from the entire number on both domains, we obtained 97.4%, overcoming with +0.3% the value obtained with the initial FV. In [23] another possibility to perform the fusion was proposed, this time being realized at the raw data level, the FVs needed in this **data-fusion case** being obtained from the combined VIS+IR image. The data-fusion situation, i.e. *maxDataFusion* seems to be the best solution to our problem, followed by the *FeatureFusion*. The experiments showed that the fusion at different levels can be considered to provide better results for our problem than the monomodal systems. If either solution proposed in this paper not function as one expect, it may be considered that the initial setup of the system should be changed. For example, more training data should be added, or the desired level of accuracy should be diminished in a first attempt; also, other interventions, in the latter stages of the entire system should be considered.

VII. CONCLUDING REMARKS

A consistent part of the work was dedicated to the experiments on the Tetravision database, as presented in [12], mostly due to the fact that they were even from the beginning, more organized and the results were validated in more complex scenarios, including fusion at different levels. But the way in which these were organized and how the setup parametrization was applied, were obtained by observations and repeated experiments. The receipt discovered in this way, thus to cover as many setup possibilities as one may imagine, was presented in this paper. First, the setup of the database, together with the aimed FS method and test mode should be established. Next, on the initial monomodal vectors or directly on the selected features vectors, the SVM model selection could be applied, so selected SKs resulted on each modality. If this solution is not good enough or does not fit the system type (being a bimodal one for example), the procedure could be repeated using a MK instead of a SK, and this could be successfully adapted to any bimodal system or even to a multimodal one.

VIII. ACKNOWLEDGES

The author would like to express the gratitude for having the possibility to use the Tetravision dataset provided by the VisLab and wants to address many thanks to Mr. Prof. Alberto Broggi. Also, the author expresses the gratitude for all the collaborations with INSA in developing some of the presented results.

REFERENCES

[1] P.Viola, et al., "Detecting pedestrians using patterns of motion and appearance", *Int. J. Computer Vision*, vol.**63**(2), pp. 153–161, 2005.
[2] P. Viola, M. Jones, "Robust real-time face detection". *Int. J. Computer Vision,* vol.**57**(2), pp.137–154, 2004.
[3] D.M. Gavrila, S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle", *Int. J. Computer Vision*, vol. 73(1), pp.41–59, 2007.
[4] C. Wojek, B. Schiele, "A performance evaluation of single and multi-feature people detection", *In Proceedings of the 30th DAGM Symposium on Pattern Recognition*, pp. 82–91, Springer, Berlin, 2008.
[5] D. Geronimo, et al., "Survey of pedestrian detection for advanced driver assistance systems", *TPAMI*, vol.32(7), pp. 1239–1258, 2010.
[6] P. Dollar, et al., "Pedestrian detection: an evaluation of the state of the art", *TPAMI,* vol. 34(4), pp.743–761, 2012.
[7] A. Miron, et al., "An evaluation of the pedestrian classif. in a multi domain multi-modality setup", *Sensors,*vol.15(6),pp.13851-73, 2015.
[8] Weka – Data mining software in Java, [Online] http://www.cs.waikato.ac.nz/ml/weka/, [Accessed: May 1, 2016].
[9] Robin dataset, [Online] http://robin.inrialpes.fr/datasets.php, [Accessed: May 1, 2016].
[10] Bertozzi, M., et al., "Low level pedestrian detection by means of visible and far infra-red tetravision",*IEEE IV Symp.*, pp.231–236, 2006.
[11] Bertozzi,M., et al., "Multi stereo-based pedestrian detection by daylight and far-infrared camera, *Hammoud, R., Augmented Vision Perception in Infrared:Algorithms and Applied Systems*, Springer Inc., pp. 371–401 (Ch16), 2009.
[12] Apatean, A., Rogozan, A., Bensrhair, A., 'Visible-Infrared Fusion Schemes for Road Obstacle Classification", *Journal of Transportation Research Part C: Emerging Technologies,* Vol. 35, pp. 180-192, 2013.
[13] I. Witten, et al. *Data Mining: Practical Machine Learning Tools and Machines*, 3rd ed., Elsevier, MK, 2011.
[14] I. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., San Francisco: MK, 2005.
[15] B.E. Boser, et al., "A training algorithm for optimal margin classifiers", *In proc.5th Annual ACM Workshop on COLT. ACM Press*, pp. 144–152, In Haussler, D. (ed.), 1992.
[16] V.N. Vapnik, *Statistical learning theory*. New York, USA: „Adaptive and Learning Systems for Signal Processing, Communications, and Control". John Wiley and Sons, 1998.

[17] N. Cristianini, J. Shawe-Taylor, *An introduction to support vector machines*. Cambridge,UK: 1st ed Cambridge Univ.Press, 2000.
[18] L. Dioşan, et al., "Optimising multiple kernels for SVM by genetic programming." *Evolutionary Computation in Combinatorial Optimization*. Springer Berlin Heidelberg, 2008. 230-241.
[19] Apatean, A., Rogozan, A., Bensrhair A., *Contributions to the Information Fusion. Application to Obstacle Recognition in Visible and Infrared Images*, editura UTPress, 2015.
[20] A. Apatean, A. Rogozan, A. Bensrhair, "Kernel and Feature Selection for Visible and Infrared based Obstacle Recognition", *11th IEEE Conf. Intelligent Transportation Systems*, pp.1130-1135, 2008.
[21] A. Apatean, A. Rogozan, A. Bensrhair, "Obstacle recognition using multiple kernel in visible and infrared images", *IEEE Intelligent Vehicle Symposium*, pp. 370-375, 2009.
[22] A. Apatean, A. Rogozan, A. Bensrhair, "SVM-based Obstacle Classification in Visible and Infrared Images", *Proceedings of the 17th European Signal Processing Conference*, 2009.
[23] A. Apatean, A. Rogozan, A. Bensrhair, "Information Fusion for Obstacle Recognition in Visible and Infrared Images", *IEEE Int. Symposium on Signal, Circuits and Systems*, pp. 1-4, 2009.