Near sensor decision making via compressed measurements for highly constrained hardware

Wissam Benjilali, William Guicquero, Laurent Jacques, Gilles Sicard

Near Sensor Decision Making via Compressed Measurements for Highly Constrained Hardware

Wissam Benjilali[1], William Guicquero[1], Laurent Jacques[2] and Gilles Sicard[1]
[1]Univ. Grenoble Alpes, CEA, LETI, F-38000 Grenoble, France
[2]ISPGroup, ICTEAM/ELEN, UCLouvain, Louvain-la-Neuve, Belgium
Email: {wissam.benjilali, william.guicquero, gilles.sicard}@cea.fr
laurent.jacques@uclouvain.be

***Abstract*—This work presents and compare three realistic scenarios to perform near sensor decision making based on Dimensionality Reduction (DR) techniques of high dimensional signals in the context of highly constrained hardware. The studied DR techniques are learned according to two alternative strategies: one whose parameters are learned in a compressed signal representation, as being achieved by random projections in a compressive sensing device, the others being performed in the original signal domain. For both strategies, the inference is yet indifferently performed in the compressed domain with dedicated algorithm depending on the selected learning technique. Our results, based on two common datasets, show that performing the inference in the compressed domain represents a competitive approach compared to the classical classification strategy (inference in the original signal domain) regarding memory and computational requirements. We also exhibit the fact that it is especially well suited for embedded applications in the context of hardware implementations with limited resources even with specific hardware design and limitations.**

I. Introduction

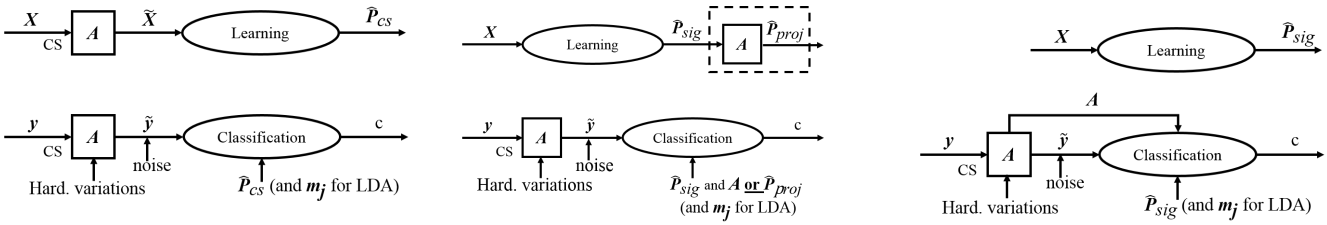
THE last few years have testified a widespread of data specific processing units [1], [2]. Considering any kind of computer vision problem (*e.g.*, ADAS), image descriptors (*e.g.*, HOG, LBP) combined with a proper classification algorithm (*e.g.*, Artificial Neural Networks) are used to enable objects detection and/or classification [3]. Embedding such system with high dimensional and complex data requires considerable memory and computational resources.

Recent advances in signal processing and pattern recognition tend to deal with high-dimensional problems by introducing new and efficient techniques in terms of computing and storage resources. For example, Dimensionality Reduction [4] (DR) relies on the projection of a high-dimensional data into a relevant low dimensional feature domain that preserves data intrinsic properties (*e.g.*, statistical or geometrical properties). Various DR techniques can be found in the literature, they can be linear or nonlinear, supervised or unsupervised [5]–[8]. One can identify two distinct approaches to achieve dimensionality reduction. First, DR can be performed by a *learned projection* that optimizes a regularized objective function. These techniques are basically introduced as machine learning algorithms to perform decision making in low dimensional domains. However, embedding such techniques involves dedicated hardware ressources to store ex-situ learned patterns. Alternatively, DR can be signal independent and achieved by *random projections* to acquire compressed features as performed in a Compressed Sensing (CS) [9] based system with remote signal reconstruction. In this case, the design of related sensing matrices has to satisfy Restricted Isometry Property (RIP) [10] to guarantee a stable embedding property and preserve geometrical properties [11]. In particular, it was shown that a wide variety of randomly generated matrices satisfies this property [12]. This well studied matrices can advantageously be generated by *e.g.*, pseudo-random generators (*e.g.*, LFSR [13], cellular automata [14]), relaxing as a consequence design constraints, in particular memory requirements. For the sake of clarity, we will call ML-DR DR performed by learned projections, and CS-DR DR via random projections.

Related work: In the CS state-of-the-art, several CS-based image sensor architectures have emerged each proposing an alternative scheme to the traditional image acquisition [13]–[15]. Indeed, one major limitation of CS based systems is the processing complexity related to the signal reconstruction. This consideration highly restricts the use of CS to niche applications. When combined with machine learning, CS might not involve this costly operation. On the other hand, [16]–[18] provide theoretical guarantees to perform compressed inference when dealing with CS measurements [19]–[22]. This is possible thanks to the RIP property which preserves the geometrical properties of the projected measurements in the compressed domain (*e.g.*, Euclidean distance) [17], [18].

Contribution: The motivation of this letter is to show, in the context of highly constrained hardware (*e.g.*, always-on ultra low power vision systems), the interest of using ML-DR and CS-DR for basic embedded image inference. Two processing stages have to be considered: *learning* ML-DR in an off-line system on labeled data, and then performing *embedded inference* on compressed data whose related classes is unknown (*e.g.*, considering embedded inference in a CS image sensor). As a comparative study we propose various learning and inference strategies for two ML-DR methods known as Linear Discriminant Analysis (LDA) [5] and Support Vector Machine (SVM) [5]. For each technique we present and compare three approaches to perform near-sensor decision making in the context of hardware limited systems. The first

Laurent Jacques is funded by the Belgian F.R.S.-FNRS and by the project AlterSense (MIS-FNRS).



(a) Approach A (b) Approach B (c) Approach C

Fig. 1: Schematic description of "inference learned in CS domain" (approach A), "projection based inference" and "inference in the reconstructed signal domain" (approach C). $\boldsymbol{X}$ represents the training set, $\tilde{\boldsymbol{X}}$ is the compressed training set, $\boldsymbol{y}$ is an unknown sample, $\tilde{\boldsymbol{y}}$ is an unknown compressed sample, $\boldsymbol{A}$ represents the CS matrix, $\hat{\boldsymbol{P}}_{cs}$ is the ML-DR transformation learned on CS measurements, $\hat{\boldsymbol{P}}_{proj}$ is the ML-DR transformation learned in the signal domain and projted in the CS one, $\hat{\boldsymbol{P}}_{sig}$ is the ML-DR transformation learned in the signal domain, $\boldsymbol{m}_j$ are classes centers, and c is the predicted class of an unknown sample.

approach (Fig. 1(a)) consists in performing ML-DR learning and embedded inference on compressed measurements taking advantage on CS-DR to reduce embedded resources requirements. In the second and third ones (Fig. 1(b) and Fig. 1(c) resp.), dedicated inference solutions are presented to deal with compressed measurements extracted using a CS device whose sensing scheme is not necessarily a priori known (*e.g.*, for security purposes [23] or to manage sensor non-idealities [24]). Performance of ML-DR methods is evaluated based on the inference accuracy regarding the learning approach, as well as general considerations on memory resources, computational complexity and robustness to some hardware variations for two object recognition applications.

II. ML-DR LEARNING BACKGROUND

Let us consider a database of N-length "vectors" in $\mathbb{R}^N$ (*e.g.*, images with N pixels) composed of C classes. This database is separated into two subsets: a "train" set $\boldsymbol{X} \in \mathbb{R}^{N\times n_1 C}$, where each class is composed of n_1 samples, associated with labels $\boldsymbol{l} \in \{1,\cdots,C\}^{n_1 C}$; and a "test" set $\boldsymbol{Y} \in \mathbb{R}^{N\times n_2 C}$ with unknown labels and composed of n_2 samples per class. We refer to $\boldsymbol{X}^j = (\boldsymbol{X}_1^j,\cdots,\boldsymbol{X}_{n_1}^j) \in \mathbb{R}^{N\times n_1}$ and $\boldsymbol{Y}^j = (\boldsymbol{Y}_1^j,\cdots,\boldsymbol{Y}_{n_2}^j) \in \mathbb{R}^{N\times n_2}$ for the train and the test sets restricted to the j^{th} class, respectively. When we write $\boldsymbol{x} \in \boldsymbol{X}$ or $\boldsymbol{x} \in \boldsymbol{X}^j$, we mean that the sample $\boldsymbol{x}$ is an arbitrary column of $\boldsymbol{X}$ or $\boldsymbol{X}^j$, respectively (and similarly for $\boldsymbol{Y}$). In the following, we first describe how to learn the considered ML-DR classifiers denoted $\hat{\boldsymbol{P}}(\boldsymbol{x}) = \hat{\boldsymbol{D}}\boldsymbol{x} + \hat{\boldsymbol{\delta}}$, and then present the corresponding inference algorithms for each approach. Here, $\hat{\boldsymbol{P}}_i(\boldsymbol{x})$ represent the projection of $\boldsymbol{x}$ on the i^{th} axis (line) of $\hat{\boldsymbol{P}}$, the mean vector of each class is expressed as $\boldsymbol{m}_j = \frac{1}{n_1}\sum_{i=1}^{n_1}\boldsymbol{X}_i^j$, for $1 \le j \le C$, and $\boldsymbol{m} = \frac{1}{C}\sum_{j=1}^{C}\boldsymbol{m}_j$ the mean vector of all samples. Greek letter λ represents a scalar regularization parameter. Depending on the technique, the matrix $\hat{\boldsymbol{D}}$ and the offset $\hat{\boldsymbol{\delta}}$ are computed using one of the following optimization problems.

A. LDA (Linear Discriminant Analysis)

The LDA [5] aims at finding the best projection minimizing the within-class variance while maximizing between class variance (Fisher's criterion [5]). This can be expressed using between-classes ($\boldsymbol{S}_B$) and within-classes scatter matrices ($\boldsymbol{S}_W$):

$$\hat{\boldsymbol{D}}_{\text{LDA}} = \underset{\boldsymbol{D}\in\mathbb{R}^{(C-1)\times N}}{\arg\max} \frac{|\,\boldsymbol{D}\boldsymbol{S}_B\boldsymbol{D}^\top\,|}{|\,\boldsymbol{D}\boldsymbol{S}_W\boldsymbol{D}^\top\,|}, \tag{1}$$

where $|\,.\,|$ denotes the determinant operation, $\boldsymbol{S}_B = \sum_{j=1}^{C}(\boldsymbol{m}_j-\boldsymbol{m})(\boldsymbol{m}_j-\boldsymbol{m})^\top$ and $\boldsymbol{S}_W = \sum_{j=1}^{C}\sum_{i=1}^{n_1}(\boldsymbol{X}_i^j - \boldsymbol{m}_j)(\boldsymbol{X}_i^j-\boldsymbol{m}_j)^\top$. Here, $\hat{\boldsymbol{\delta}}_{\text{LDA}} = \hat{\boldsymbol{D}}_{\text{LDA}}\boldsymbol{m} \in \mathbb{R}^{C-1}$ represents the projected training set centroid of all samples.

B. SVM (Support Vector Machine)

Another way to learn a discriminant classifier is to learn a multiclass SVM using a one-vs-all strategy [25], this allows to learn C binary soft margin classifiers to construct a boundary decision for each class versus the others. It consists in assigning a positive margin to the sample's class and a negative margin to the others (*i.e.*, $l_k^j = 1$ if the k^{th} sample belongs to class j, and -1 otherwise). Mathematically speaking:

$$\{\hat{\boldsymbol{D}}_{\text{SVM},j}, \hat{\delta}_{\text{SVM},j}, \hat{\boldsymbol{\xi}}_j\} = \underset{\boldsymbol{D}\in\mathbb{R}^N,\delta,\boldsymbol{\xi}\in\mathbb{R}^{n_1}}{\arg\min} \left(\left\|\boldsymbol{D}\right\|_2^2 + \lambda\sum_{k=1}^{n_1}\xi_k\right)$$
$$\text{s.t.} \quad l_k^j(\boldsymbol{D}\boldsymbol{x}+\delta)_j \ge 1-\xi_k,\ \xi_k \ge 0,\ \forall 1 \le k \le n_1. \tag{2}$$

We define $\hat{\boldsymbol{P}}_{\text{SVM},j}(\boldsymbol{x}) := \hat{\boldsymbol{D}}_{\text{SVM},j}\boldsymbol{x} + \hat{\delta}_{\text{SVM},j}$, for $1 \le j \le C$. Above, the matrix $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{\xi}}_1,\cdots,\hat{\boldsymbol{\xi}}_C)$ is made of $n_1 C$ slack variables that allow to deal with outliers, each variable is associated to one training sample. Here $\hat{\boldsymbol{D}}_{\text{SVM}} \in \mathbb{R}^{C\times N}$ and $\hat{\boldsymbol{\delta}}_{\text{SVM}} = (\hat{\delta}_{\text{SVM},1},\cdots,\hat{\delta}_{\text{SVM},C}) \in \mathbb{R}^C$.

III. CLASSIFICATION COMBINING ML-DR AND CS-DR

We here describe three approaches to perform the inference on CS measurements. In section *A* we present the first approach, where the ML-DR projection is learned on data compressed by a pseudo-random CS-DR (Fig. 1(a)). In this case, classification inference is performed in the compressed domain based on the learned ML-DR affine projection. The second scheme is presented in section *B*. In this case, the ML-DR is learned in the signal domain without the knowledge of the sensing matrix (Fig. 1(b)) and then projected in the CS domain using a sensing matrix $\boldsymbol{A}$. In contrast to the first and second approaches, the third one introduces a dedicated DSP allowing to implement a reconstruction-like algorithm

Embedded required resources	DR	Inference learned in CS domain	Projection based inference	Inference in the reconstructed signal domain
Memory needs	LDA	$\mathcal{O}(C^2)+\mathcal{O}(CM)$	$\mathcal{O}(C^2)+\mathcal{O}(CM)$	$\mathcal{O}(C^2)+\mathcal{O}(CM)$
	SVM	$\mathcal{O}(CM)$	$\mathcal{O}(CM)$	$\mathcal{O}(CM)$
Algorithmic complexity	LDA	$\mathcal{O}(C^2)+\mathcal{O}(CM)$	$\mathcal{O}(C^2)+\mathcal{O}(CM)$	$\mathcal{O}(qC^3)+\mathcal{O}(qMC^2)$
	SVM	$\mathcal{O}(CM)$	$\mathcal{O}(CM)$	$\mathcal{O}(qC^2)+\mathcal{O}(qMC)$

TABLE I: Embedded resources requirements to perform near sensor decision making.

to perform the inference using a ML-DR learned in the signal domain and compressed observations (Fig. 1(c)). In the following, studied DR techniques will be denoted $\hat{\boldsymbol{P}}_{\mathrm{cs-LDA}}$, $\hat{\boldsymbol{P}}_{\mathrm{proj-LDA}}$ and $\hat{\boldsymbol{P}}_{\mathrm{sig-LDA}}$ (for LDA using approach A, B and C resp.) and $\hat{\boldsymbol{P}}_{\mathrm{cs-SVM}}$, $\hat{\boldsymbol{P}}_{\mathrm{proj-SVM}}$ and $\hat{\boldsymbol{P}}_{\mathrm{sig-SVM}}$ (SVM).

A. Approach A: Inference learned in CS domain

A desirable CS sensor property is the capability to acquire a compact signal with a sparse representation that allows to extract its inherent information. To formulate this problem, let $\tilde{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{X} \in \mathbb{R}^{M\times n_1 C}$ and $\tilde{\boldsymbol{Y}} = \boldsymbol{A}\boldsymbol{Y} \in \mathbb{R}^{M\times n_2 C}$ the training and test sets observed in the CS domain using the CS matrix $\boldsymbol{A} \in \mathbb{R}^{M\ll N}$. This implies that all the training samples are of lowered dimensionality, which, in addition, reduces the computational complexity of the training. Here, the equations (1) and (2) are solved in the compressed domain and the training set $\boldsymbol{X}$ is replaced by its projection $\tilde{\boldsymbol{X}}$.

1) Classification for LDA: As the LDA clusters samples of the same class, an Euclidean distance metric is used to assign the test sample $\tilde{\boldsymbol{y}}$ to the nearest class represented by its center, *i.e.*, we estimate the class c of this sample with:

$$c = \arg\min_{1\leq i\leq C} \|\hat{\boldsymbol{P}}_{\mathrm{cs-LDA}}(\tilde{\boldsymbol{y}}) - \hat{\boldsymbol{P}}_{\mathrm{cs-LDA}}(\tilde{\boldsymbol{m}}_i)\|_2^2, \quad (3)$$

where $\tilde{\boldsymbol{m}}_i = \boldsymbol{A}\boldsymbol{m}_i$ is the mean class vector in CS domain and $\hat{\boldsymbol{P}}_{\mathrm{cs-LDA}}(\boldsymbol{x}) = \hat{\boldsymbol{D}}_{\mathrm{cs-LDA}}\boldsymbol{x} + \hat{\boldsymbol{\delta}}_{\mathrm{cs-LDA}}$.

2) Classification for SVM: In the SVM case, a one-vs-all strategy is used to learn C binary decision functions between one class and the rest. Then, with respect to the geometric criterion, a winner-take-all strategy assigns the test sample to the class with the highest margin, *i.e.*, the class label c is now:

$$c = \arg\max_{1\leq i\leq C} \ \hat{\boldsymbol{P}}_{\mathrm{cs-SVM},i}(\tilde{\boldsymbol{y}}). \quad (4)$$

B. Approach B: Projection based inference

This approach is based on a ML-DR transformation trained in the signal domain (*i.e.*, using $\boldsymbol{X}$ as training set). It has the advantage of not requiring the knowledge of the sensing scheme at the training stage. Moreover, this allows the acquisition system to generate various sensing matrices (on-the-fly), for example for data-encryption purposes and improve robustness against hardware attaks. In this approach, a signal $\boldsymbol{y} \in \mathbb{R}^N$ is observed in the compressed domain $\mathbb{R}^M$ (using the measurement matrix $\boldsymbol{A}$) and the decision is done in the inference domain $\mathbb{R}^C$ (thanks to $\hat{\boldsymbol{P}}$). A straightforward way is to reconstruct the signal using a sparsity-promoting prior (*e.g.*, ℓ_1-norm, **TV**), and then project it in the inference domain. However, this two-step scheme shows a dramatically high algorithm-hardware complexity. To overcome this cost, we propose to project the classifier of the signal domain (*i.e.*, $\hat{\boldsymbol{P}}_{\mathrm{sig-LDA}}$ and $\hat{\boldsymbol{P}}_{\mathrm{sig-SVM}}$) in the compressed domain using $\boldsymbol{A}$ [16], *i.e.*, project their N-dimensional axis in $\mathbb{R}^M$. This allows to classify a compressed test sample $\hat{\boldsymbol{y}}$ using either $\hat{\boldsymbol{P}}_{\mathrm{proj-LDA}}$ or $\hat{\boldsymbol{P}}_{\mathrm{proj-SVM}}$ defined respectively as: $\hat{\boldsymbol{P}}_{\mathrm{proj-LDA}}(\boldsymbol{x}) = \hat{\boldsymbol{D}}_{\mathrm{sig-LDA}}\boldsymbol{A}^\top\boldsymbol{x} + \hat{\boldsymbol{\delta}}_{\mathrm{sig-LDA}}$ and $\hat{\boldsymbol{P}}_{\mathrm{proj-SVM}}(\boldsymbol{x}) = \hat{\boldsymbol{D}}_{\mathrm{sig-SVM}}\boldsymbol{A}^\top\boldsymbol{x} + \hat{\boldsymbol{\delta}}_{\mathrm{sig-SVM}}$. The inference can be then performed as follows:

1) Classification for LDA:

$$c = \arg\min_{1\leq i\leq C} \|\hat{\boldsymbol{P}}_{\mathrm{proj-LDA}}(\tilde{\boldsymbol{y}}) - \hat{\boldsymbol{P}}_{\mathrm{proj-LDA}}(\tilde{\boldsymbol{m}}_i)\|_2^2. \quad (5)$$

2) Classification for SVM:

$$c = \arg\max_{1\leq i\leq C} \ \hat{\boldsymbol{P}}_{\mathrm{proj-SVM},i}(\tilde{\boldsymbol{y}}). \quad (6)$$

C. Approach C: Inference in the reconstructed signal domain

As in Approach B, to perform the inference independently of the training acquisition scheme, we propose to directly reconstruct a vector in the inference domain from its compressed observation. Thus, given $\tilde{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{y} \in \tilde{\boldsymbol{Y}}$ in the CS domain, the main goal is to reconstruct a signal $\hat{\boldsymbol{\alpha}}$ in the inference domain (*i.e.*, $\mathbb{R}^C$) such that the projection of its inverse mapping (*i.e.*, projection in the signal domain) in the CS domain minimizes the Euclidean distance to the compressed signal $\tilde{\boldsymbol{y}}$. Indeed, for a signal $\boldsymbol{\alpha} \in \mathbb{R}^C$ in the inference domain, we define its minimum energy inverse mapping as:

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u}\in\mathbb{R}^N}{\arg\min} \|\boldsymbol{u}\|_2^2 \text{ subject to } \hat{\boldsymbol{P}}(\boldsymbol{u}) = \boldsymbol{\alpha}. \quad (7)$$

Thus, the solution of (7) can be expressed as: $\hat{\boldsymbol{u}} = \hat{\boldsymbol{P}}^\dagger(\boldsymbol{\alpha})$, where $\dagger$ denotes the Moore-Penrose pseudo-inverse operator. Finally, under a regularization term promoting classes separability in the inference domain, the reconstructed signal $\hat{\boldsymbol{\alpha}}$ minimizing the energy $\|\tilde{\boldsymbol{y}} - \boldsymbol{A}\hat{\boldsymbol{P}}^\dagger(\boldsymbol{\alpha})\|_2^2$ will correspond to the projection of $\tilde{\boldsymbol{y}}$ in the inference domain. It allows as a consequence the classification of a compressed sample using a classifier learned in the signal domain. Moreover, the regularization function typically takes advantages on intrinsic properties of each method (*e.g.*, statistical and geometrical properties). In the following, this framework is applied for both LDA and SVM to perform embedded inference on CS measurements with dedicated reconstruction-like algorithms.

1) Classification for LDA: Given the CS matrix and the $\hat{\boldsymbol{P}}_{\mathrm{sig-LDA}}$ transformation, we find for each new sample its coefficients $\hat{\boldsymbol{\alpha}}$ with a constraint to encourage them to be close to the classes centers (*i.e.*, regularizing by $\boldsymbol{R}_{\mathrm{LDA},i}(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha} - \hat{\boldsymbol{P}}_{\mathrm{sig-LDA}}(\boldsymbol{m}_i)\|_2^2$). These coefficients are then used to find the

sample's class c which minimizes the cost function, *i.e.*, we estimate:

$$\hat{\boldsymbol{\alpha}}_i = \underset{\boldsymbol{\alpha}\in\mathbb{R}^{C-1}}{\arg\min} \|\tilde{\boldsymbol{y}} - \boldsymbol{A}\hat{\boldsymbol{P}}^{\dagger}_{\text{sig-LDA}}(\boldsymbol{\alpha})\|_2^2 + \lambda \boldsymbol{R}_{\text{LDA},i}(\boldsymbol{\alpha}), \quad (8)$$

$$c = \arg\min_{1\leq i\leq C} \|\hat{\boldsymbol{\alpha}}_i - \hat{\boldsymbol{P}}_{\text{sig-LDA}}(\boldsymbol{m}_i)\|_2^2. \quad (9)$$

2) Classification for SVM: The one-vs-all strategy assigns a positive margin to the sample's class and a negative one to the others. For this reason, the exponential function is chosen to reinforce sparsity of positive margins (*i.e.*, regularizing by $\boldsymbol{R}_{\text{SVM}}(\boldsymbol{\alpha}) = \|\exp(\boldsymbol{\alpha})\|_2^2$) allowing a better reconstruction of the $\boldsymbol{\alpha}$ with highest margin. Thus,

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}\in\mathbb{R}^{C}}{\arg\min} \|\tilde{\boldsymbol{y}} - \boldsymbol{A}\hat{\boldsymbol{P}}^{\dagger}_{\text{sig-SVM}}(\boldsymbol{\alpha})\|_2^2 + \lambda \boldsymbol{R}_{\text{SVM}}(\boldsymbol{\alpha}), \quad (10)$$

$$c = \arg\max_{1\leq i\leq C} \hat{\alpha}_i. \quad (11)$$

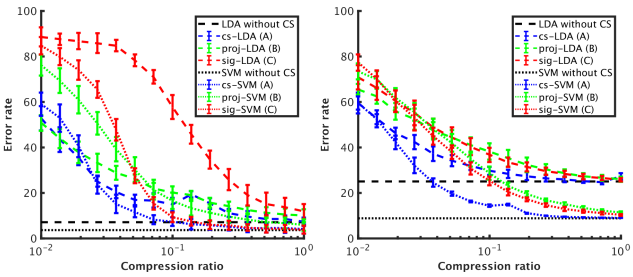
IV. Embedded Resources Requirements Study

Table I stands for the study of embedded resources requirements to implement our studied inference solutions. In the quest for the "most" hardware-friendly solution and depending on the targetted application specifications, we evaluate memory needs in terms of the number of coefficients to store and computational complexity in terms of the total number of operations (MACs) for each solution. Two main inference schemes are studied: affine projection (*i.e.*, (3) to (6)) and regularized based scheme (*i.e.*, (8) and (10)). Let us consider the affine projection based scheme (*i.e.*, approaches A and B), in this case, memory requirements are limited to the ex-situ learned ML-DR matrix for the SVM, and ML-DR matrix and the mean vectors for the LDA. In addition, to evaluate computational complexity, two subproblems have to be considered: the affine projection and the min/max operation. On the other hand, regularized based approach can take advantage on the commonly used iterative algorithms to solve problems in (8) and (10). In this case, one has to consider memory and computational requirements of multiple gradient calculations [26]. Finally, one can observe that performing the inference in the compressed domain allows to reduce memory requirements for the three approaches. However, focusing on the computational complexity, Approach A and B combined with an affine projection for the inference, need far less embedded operations compared to Approach C (we set q number of iterations to 100).

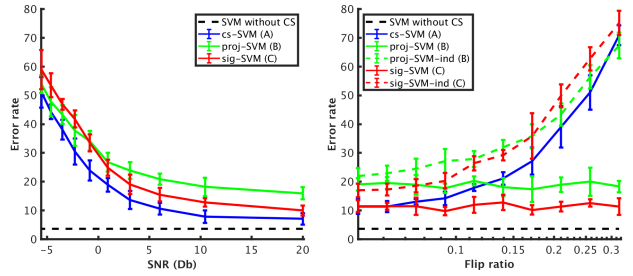
V. Experimental Results

Fig. 2 shows inference results of two databases: AT&T face database (40 classes, resized 32×32 pixels) [27] and MNIST digits database (10 classes, 28×28 pixels) [28]. We evaluate the error rate (ratio of incorrect predictions to the total number of test samples) and its standard deviation over random batches using two different types of CS matrices (Rademacher and Gaussian). We also explore the robustness of the proposed approaches in the presence of some hardware variations, *i.e.*, noise and CS matrix alterations. First and foremost, Fig. 2(a) and 2(b) attest that Approach A (blue lines) outperforms Approach B and C (green and red lines). Indeed, they analytically demonstrate that learning in the CS domain allows to achieve higher compression ratios thanks to the intrinsic properties of the sensing matrix [11]. Despite of learning the ML-DR on original data combined with a proper regularization term, Approach C still exhibits a lower inference accuracy for high compression ratios while being better than approach B for low compression ratios. Regarding considered ML-DR techniques, Fig. 2(a) shows that the LDA classifier slightly underperforms the SVM for the face recognition task (AT&T) even if all required assumptions are not met.

On the other hand, one can consider impact of hardware variations on the inference robustness. For example, in the presence of Additive White Gaussian Noise (AWGN) implying a certain SNR, Fig. (2(c)) reports that approach A still exhibits the best error rate while sharing the general behavior with approaches B and C, *i.e.*, the error rate massively increases for low SNR (*i.e.*, below 10 Db). In a second setting, binary alterations of a Rademacher sensing matrix are considered, *i.e.*, random bit flips due to nonideal hardware behaviour occurred during matrix generation. Unsurprisingly, Fig. 2(d) shows that Approach B and C (green and red plain lines) are more robust to such variations when there are known as a prior for the inference stage (green and red solid lines). Indeed, in the sensing device the actual "on-the-fly" generated sensing matrix can be provided to the hardware component performing the inference in order to be taken into account. However, when not considered because of its hardware cost (dashed lines) these approaches (B and C) are still less robust than A.



(a) AT&T - Rademacher (b) MNIST - Gaussian



(c) Robustness to additive noise (d) Acquisition scheme variations

Fig. 2: (a) and (b) Inference accuracy for the AT&T and MNIST databases. We set $n_1 = n_2 = 5$ for AT&T; and $n_1 = 5000, n_2 = 1000$ for MNIST. (c) Robustness to additive noise. (d) Robustness to hardware variations. Blue, green and red lines refer approaches A, B and C respectivelly.

VI. Conclusion

In the context of highly constrained hardware, three algorithmic approaches for near-sensor decision making were investigated. Our experimental results (based on AT&T and MNIST databases) show that a compression ratio of 10% can be reached while performing an equivalent inference accuracy as traditional linear classifiers. Moreover, to design an embedded decision making hardware, we show that performing the inference in the CS domain needs far less resources and MACs compared to an inference in the signal domain. However, when dealing with specific design constraints (*e.g.*, for privacy purposes), one can take advantage on dedicated algorithms to perform the inference on CS measurements while preserving a good trade-off between accuracy and robustness to unexpected hardware variations. Finally, to provide a more hardware-oriented study, future works will take into account the impact of quantizing both CS measurements [18] and ML-DR learned coefficients, as well as the advantage of using a hardware-friendly CS matrix [29] to fit with constraints of ultra-low power analog-to-information sensing devices.

References

[1] W. Kuzmicz. The future of cmos: More moore or the next big thing? In *2017 MIXDES - 24th International Conference "Mixed Design of Integrated Circuits and Systems*, pages 21–26, June 2017.

[2] T. Islam, S. C. Mukhopadhyay, and N. K. Suryadevara. Smart sensors and internet of things: A postgraduate paper. *IEEE Sensors Journal*, 17(3):577–584, Feb 2017.

[3] F. An, X. Zhang, A. Luo, L. Chen, and H. J. Mattausch. A hardware architecture for cell-based feature-extraction and classification using dual-feature space. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–1, 2017.

[4] L. Maaten, E. Postma, and J. Herik. Dimensionality reduction: A comparative review, 2008.

[5] C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, 2006.

[6] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb 2009.

[7] T. Verma and R. K. Sahu. PCA-LDA based face recognition system amp;amp; results comparison by various classification techniques. In *2013 International Conference on Green High Performance Computing (ICGHPC)*, pages 1–7, March 2013.

[8] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *2011 International Conference on Computer Vision*, pages 543–550, Nov 2011.

[9] L. Jacques and P. Vandergheynst. Compressed sensing: When sparsity meets sampling. Technical report, Wiley-Blackwell, 2010.

[10] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, Dec 2005.

[11] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk. Signal processing with compressive measurements. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):445–460, April 2010.

[12] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, Dec 2008.

[13] V. Majidzadeh, L. Jacques, A. Schmid, P. Vandergheynst, and Y. Leblebici. A (256 x 256) pixel 76.7mw cmos imager/ compressor based on real-time in-pixel compressive sensing. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2956–2959, May 2010.

[14] W. Guicquero, A. Dupret, and P. Vandergheynst. An algorithm architecture co-design for cmos compressive high dynamic range imaging. *IEEE Transactions on Computational Imaging*, 2(3):190–203, Sept 2016.

[15] Y. Oike and A. El Gamal. Cmos image sensor with per-column sigma delta adc and programmable compressed sensing. *IEEE Journal of Solid-State Circuits*, 48(1):318–328, Jan 2013.

[16] R. Calderbank, S. Jafarpour, and R. Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Technical report, 2009.

[17] A. S. Bandeira, D. G. Mixon, and B. Recht. Compressive classification and the rare eclipse problem. *CoRR*, abs/1404.3203, 2014.

[18] V. Cambareri, C. Xu, and L. Jacques. The rare eclipse problem on tiles: Quantised embeddings of disjoint convex sets. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 241–245, July 2017.

[19] H. Yu, H. Lu, T. Ouyang, H. Liu, and B. L. Lu. Vigilance detection based on sparse representation of eeg. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 2439–2442, Aug 2010.

[20] M. Zhang, C. Cai, and J. Zhu. Sparse representation for weed seeds classification. In *The 2010 International Conference on Green Circuits and Systems*, pages 626–631, June 2010.

[21] G. Bull, J. Gao, and M. Antolovich. Delineation of rock fragments by classification of image patches using compressed random features. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 1, pages 394–401, Jan 2014.

[22] J. Xia, N. Yokoya, and A. Iwasaki. A novel ensemble classifier of hyperspectral and lidar data using morphological features. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6185–6189, March 2017.

[23] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He. A review of compressive sensing in information security field. *IEEE Access*, 4:2507–2519, 2016.

[24] F. Qian, Y. Gong, G. Huang, M. Anwar, and L. Wang. Exploiting memristors for compressive sampling of sensory signals. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(12):2737–2748, Dec 2018.

[25] J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. *Proc of the 7th European Sympo-sium On Artificial Neural Networks*, pages 219–224, 01 1999.

[26] P. L. Combettes. H. H. Bauschke. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer.

[27] AT&T database. Available [online]: *http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html*.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[29] W. Benjilali, W. Guicquero, L. Jacques, and G. Sicard. A low-memory compressive image sensor architecture for embedded object recognition. In *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 881–884, Aug 2018.