MULTIPLE DICTIONARY LEARNING FOR BLOCKING ARTIFACTS REDUCTION

Yi Wang[⋆]
University of Minnesota

Fatih Porikli
Mitsubishi Electric Research Labs

ABSTRACT

We present a structured dictionary learning method to remove blocking artifacts without blurring edges or making any assumption over image gradients. Instead of a single overcomplete dictionary, we build multiple subspaces and impose sparsity on nonzero reconstruction coefficients when we project a given texture sample on each subspace separately. In case the texture matches to the dataset with which the subspace is trained, the corresponding response will be stronger and that subspace will be chosen to represent the texture. In this manner we compute the representations of all patches in the image and aggregate these to obtain the final image. Since the block artifacts are small in magnitude in comparison to actual image edges, aggregation efficiently removes the artifacts but keep the image gradients. We discuss the choices of subspace parameterizations and adaptation to given data. Our results on a large dataset of benchmark images demonstrate that the presented method provides superior results in terms of pixel-wise (PSNR) and perceptual (SSIM) measures.

Index Terms— Deblocking, Dictionary Learning, JPEG

1. INTRODUCTION

The blocking effect is considered as the most disturbing artifact of JPEG decoded images and can dramatically degrade the visual quality especially when the images are encoded at high compression rates, where JPEG compression introduces high frequency quantization error to each individual block separately, resulting discontinuity across block boundaries.

Several techniques have been proposed to postprocess JPEG images aiming at improving the visual quality. Some methods analyze spatial domain manifestations [1], while others mainly work infrequency domain [2]. There are more sophisticated iterative methods based on projections onto convex sets [3, 4]. In addition, processing images adaptively to reduce artifacts while preserving edges simultaneously are proposed many times [5, 6, 7, 8]. Another noteworthy approach is reapplication of DCT on shifted images [9] and averaging results at each pixel. This approach is simple but produces nice results. Among the main drawbacks of these methods their computational complexity, explicit dependency on the correct image gradient information, sensitivity to preset parameters come.

Dictionary learning is to learn an over-complete basis and represent image patches sparsely under this basis. More precisely, it solves for dictionary $\mathbf{D}$ and reconstruction coefficients $\mathbf{A}$ by minimizing

$$\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda \|\mathbf{A}\|_1$$

where

$$\|\mathbf{A}\|_1 := \sum_{i,j} |A_{ij}|$$

and columns of $\mathbf{X}$ represent patches of an image. It has been shown that dictionary learning delivers efficient solutions in compression, denoising and other inverse problems in image processing [10, 11]. Furthermore, many methods are developed to explore group structures for dictionary learning. For instance, Yu et al. [12] propose a method based on Structured Sparse Model Selection (SSMS) which enforces reconstruction coefficients $\mathbf{A}$ to be block diagonal in the above formulation. SSMS behaves more stable than traditional dictionary learning methods and achieves remarkable performance in above problems. It is demonstrated in [13] that SSMS can restore compressed noisy images.

Inspired by both the idea of reapplication of DCT and recently popular dictionary learning methods, we note that they share a common point of processing of all overlapping patches in the given image. However, dictionary learning has the advantage of providing an adaptive and often overcomplete projection rather than spanning with a fixed orthonormal bases. When sparsity is imposed, spanning onto a data and task driven basis is shown to produce better reconstruction results. Thus, instead of using DCT, we consider incorporating multiple subspaces when we reassemble and aggregate patch responses. Each subspace is obtained offline by under-complete dictionary learning on particular edge orientations. In online processing, for each image patch we select the best dictionary, use it to update the corresponding subspaces. We call our method as Multiple Dictionary Learning (MDL). We examine the validity of such subspace projection for blocking artifact removal when the image is heavily compressed. Unlike [12] we analyze the affect of the reduced number of subspaces on the artifact removal. We show that it suffices to use only a few undercomplete subspaces sometimes even without additional update stage. Our results on a large dataset of benchmark images demonstrate that the the presented method provides superior results in terms of pixel-wise (PSNR) and perceptual (SSIM) measures while improving the JPEG Quality Score (JQS) from 2 (heavily distorted) to above 9 (almost no perceivable artifacts).

2. MULTIPLE DICTIONARY LEARNING

In an offline step, we first learn K orthogonal subspaces $\{\mathbf{D}_k\}_{k=1}^K$ from synthetic images each depicting a wedgelet at a particular orientation equally sampled from $[0, 2\pi)$. Previously, up to 18 such subspaces are utilized [12] with an additional full-rank DCT basis as the initial structured dictionary. However, it is not clear that using a large number of subspaces would actually brings substantial improvement despite its high computational load.

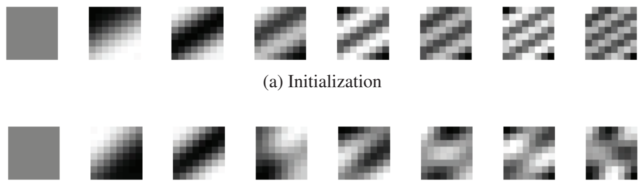
On the online step, MDL constructs multiple subspaces by iterating between a clustering stage and a dictionary learning stage:

1. Clustering: A $m \times m$ patch $\mathbf{x}$ is assigned to cluster $\hat{k}(\mathbf{x})$ by:

$$\hat{k}(\mathbf{x}) = \underset{k}{\operatorname{argmax}} \|(\mathbf{D}_k^d)^T \mathbf{x}\|_2^2 \tag{1}$$

where $\mathbf{D}_k$ is the orthonormal basis of the k^{th} cluster and $\mathbf{D}_k^d$

[⋆] This work was performed at Mitsubishi Electric Research Labs.



(a) Initialization

(b) After 5 Iterations

Fig. 1. First 8 components of the dictionary for $\pi/6$. Initial patterns (basis vectors of the subspace) are changed adaptive to the given image.

are the first d components of the basis. $k = 1, \dots, K$. $\mathbf{D}_k$ spans the k^{th} subspace for the corresponding cluster.

2. Dictionary Learning. $\mathbf{D}_k$ is updated by applying the Singular Value Decomposition (SVD) on the matrix $\mathbf{X}_k$ whose columns are the patches assigned to the k^{th} cluster, i.e.,

$$\mathbf{X}_k := [\mathbf{x} : \hat{k}(\mathbf{x}) = k].$$

Denote the result of SVD by $\mathbf{X}_k = \mathbf{USV}^T$, then we form $\mathbf{D}_k = \mathbf{U}$ and the coefficient matrix $\mathbf{A} = \mathbf{SV}^T$.

We show in Figure 1a the first 8 components of the dictionary for $\pi/6$ and in Figure 1b those after 5 iterations. As visible, the pattern change (note that sign changes are captured by the reconstruction coefficients) is more eminent for the less important basis vectors.

After the iterations converge, we threshold patch coefficients. Our intuition is that blocking artifacts do not establish themselves as strongly as the real edges in natural images, in consequence, a thresholding of smaller coefficients will remove the blocking artifacts. In other words, magnitudes of the artifacts when projected on the basis we learned offline and adapted online are not as large as those of the real edges (e.g., Figure 1). Applying a conservative threshold performs like a low-pass filter on blocking artifacts.

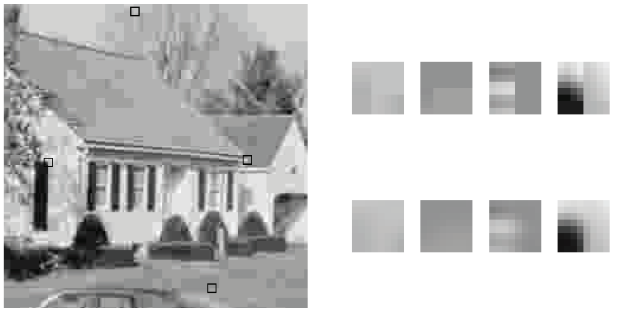
A patch x is then approximated by $\hat{x}$ as follows:

$$\hat{x} = \mathbf{D}^r_{\hat{k}} \delta_\tau((\mathbf{D}^r_{\hat{k}})^T \mathbf{x}) \tag{2}$$

where r is larger than the number of coefficients d used for clustering, and the thresholding δ_τ is applied on each element in the vector. More precisely, $\delta_\tau(a) = a$, if $|a| > \tau$; $\delta_\tau(a) = 0$ otherwise. This whole process is applied for all overlapping patches and the thresholded approximations are averaged at each pixel to reconstruct the image. As an example, Figure 2a shows 4 original blocks from the compressed image: 2 with blocking artifacts and other 2 containing real edges). In Figure 2b, the reassembling results after the thresholding are given.

3. PARAMETER SELECTION

We set the size of the blocks $m = 8$ and the number of the basis vectors in the subspaces as $d = 8$. We observed that after $r > 15$ for 8×8 blocks the reconstruction hardly improves. We thus let $r = 20$ to reduce computational load. Another important parameter is the threshold level τ which should be adaptive to the compression rate. We suggest specific values for τ as in Table 1. *Quality* is the parameter of MATLAB function *imwrite* which is an integer scaling from 0 to 100. Higher numbers mean higher quality i.e. less image degradation due to compression. This quantity can be determined by the compressed image. Any number around the recommended value works sufficiently well. One can even adjust these values for the image, e.g. if the image contains many textures then choose a smaller value for τ; if the image is rather uniform then a larger number for more aggressive thresholding.



(a) Compressed *House*

(b) Smooth (left two) and edge (right two) blocks. Top row is before MDL, bottom row is after MDL.

Fig. 2. Response for smooth and edge patches (marked in black squares in the image). MDL based thresholding removes the blocking artifacts yet preserves real edges (best viewed in high-contrast zoomed in display).

Table 1. Suggestions for choosing τ.

Quality	34	30	25	20	16	12	9	6
τ	9	12	15	18	21	30	36	45

We investigated two problems about the selection of parameters. One is how many iterations we need for convergence (i.e. until no more improvements happen) and the other is how many clusters are sufficient (i.e the number of subspaces). Our extensive experimental evaluation shows that MDL with 5 iterations and a large number of clusters ($K = 18$ orientations + DCT) does not help much on removing the blocking artifacts. Combined with the choice that $r = 20$, we conclude that using only a few ($K = 2$ orientations + DCT) under-complete subspaces (unlike multiple complete spaces in [12]) is the optimal trade off between accuracy and computation complexity. Sample results of Lena are given in Figure 3 for different scenarios.

4. EXPERIMENTS

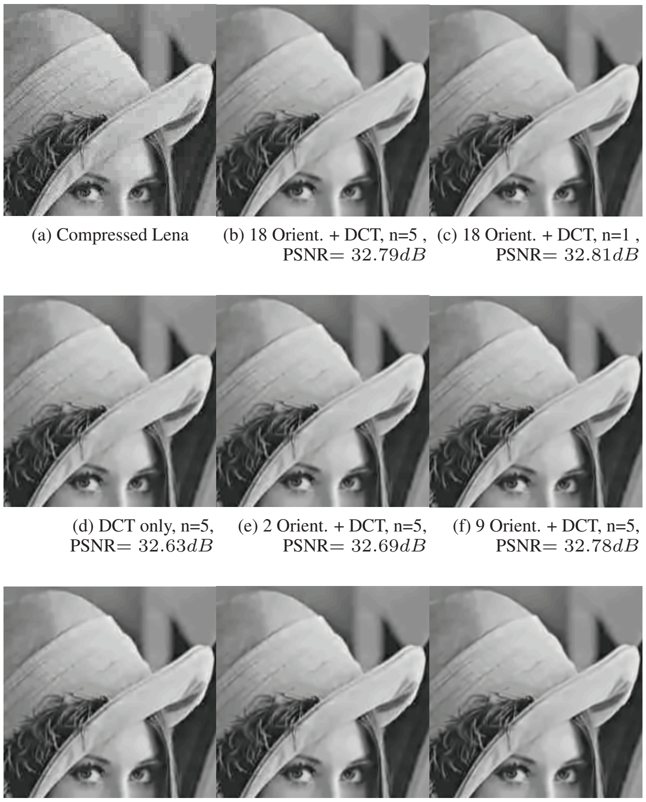
We used three measures of image quality, Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) index [14] and JPEG quality score (JQS) [15]. Among these measures, PSNR and SSIM evaluate the pixel-wise quality referring to original image and JQS assesses perceptual quality without any reference image.

Structural SIMilarity (SSIM) considers luminance, contrast and structure and aims to evaluate the perceptual quality of an approximate image comparing to the original one.

JPEG Quality Score (JQS): is a no-reference perceptual quality assessment of JPEG compressed images. The model is given by:

$$S = \alpha + \beta B^{\gamma_1} A^{\gamma_2} Z^{\gamma_3} \tag{3}$$

where B, A, Z are values computed from the compressed image. B estimates the blockiness and A and Z approximate the activity. The remaining parameters are determined by regression in subjective experiments (see more details in [15]).



(a) Compressed Lena (b) 18 Orient. + DCT, n=5 , PSNR= $32.79dB$ (c) 18 Orient. + DCT, n=1 , PSNR= $32.81dB$

(d) DCT only, n=5, PSNR= $32.63dB$ (e) 2 Orient. + DCT, n=5, PSNR= $32.69dB$ (f) 9 Orient. + DCT, n=5, PSNR= $32.78dB$

(g) DCT only, n=1, PSNR= $32.63dB$ (h) 2 Orient. + DCT, n=1, PSNR= $32.65dB$ (i) 9 Orient. + DCT, n=1, PSNR= $32.77dB$

Fig. 3. Blocking artifacts removal results with different parameters. Several iterations (n) and a large number of clusters (orientations) do not help much on this task (best viewed in zoomed in display).

To evaluate the performance of the proposed method, we apply it on 50 examples (10 color images, each has 5 compressed images of different compression rates) of IVC database [16] and compare with other two methods. We transform these images to gray scale and work on gray images. We detect *Quality* value for each compressed image and use it for reapplication of JPEG. The threshold value t is also chosen according to *Quality* as in Table 1. Furthermore, we evaluate our methods on several color images for which we obtain the data matrix in three channels. We apply MDL in each channel of the YUV space.

We compute PSNR, SSIM [14] of MDL, pointwise SA-DCT (SA) [6] and reapplication of JPEG (ReJPG) method [9]. Results are shown in Table 2 and Fig. 4. On average, MDL improves about $1dB$ over the input compressed images.

As for the complexity, the major computation of MDL is due to the orthogonal projections of the clustering stage and the SVD of the learning stage. Let N be the patch size (in MDL, $N = m^2$ and normally $m = 8$). Then, for n patches and $d < N < n$, MDL requires $\mathcal{O}(N^2 n) + \mathcal{O}(KdNn)$ operations, where $K = 3$ and $d = 10$ can achieve satisfying results. Moreover, the complexity of SA-DCT is $\mathcal{O}(N^3 n)$ on average and can reduce to $\mathcal{O}(nN^2 \log(N))$ by fast algorithms. The Re-JPEG method requires $\mathcal{O}(N^2 n)$ operations plus $\mathcal{O}(Nn)$ for Huffman coding.

Table 2. The average values for each image in different measures

	PSNR			SSIM		
images	MDL	ReJPG	SADCT	MDL	ReJPG	SADCT
avion	32.42	32.30	**32.44**	**0.902**	0.899	0.901
barba	**28.60**	28.17	28.01	**0.841**	0.826	0.837
boats	31.08	30.90	**31.12**	**0.873**	0.869	0.872
clown	31.70	31.54	**31.72**	**0.873**	0.869	**0.873**
fruit	**33.93**	33.72	33.61	**0.910**	0.901	0.904
house	**30.15**	**30.15**	30.14	**0.862**	0.861	0.860
isabe	**33.03**	32.80	32.95	**0.864**	0.861	0.858
lenat	**33.00**	32.70	32.97	**0.875**	0.870	0.875
mandr	**24.17**	24.14	-	**0.682**	**0.682**	-
pimen	**31.81**	31.59	31.95	0.834	0.828	**0.835**

From the results shown above, we see that in terms of both pixel-wise and perceptual metrics, MDL works better than JPEG reapplication (ReJPG) method in almost all the cases. MDL can more effectively remove the blocking artifacts as well as preserving clean edges and textures in images. MDL also obtains better results in the uniform regions (e.g. the arm on the left of *Barbara* in Figure 4). However, ReJPG produces favorable JQS [15] values although we see from the perceptual examples that MDL clearly outperforms ReJPG. This is because of the fact that JQS is designed with help of JPEG images, thus JQS measures blocking artifacts but lacks the power to evaluate overall quality. Furthermore, MDL is comparable to but sometimes better than point-wise SA-DCT. In comparison to point-wise SA-DCT, we notice that MDL is more effective recovering textures (e.g. *Barbara*) while SA-DCT oversmoothens (e.g. the right low region of the color image *Isabel* in Figure 4). This also explains why SA-DCT gives slightly better results for uniform images (e.g. *Peppers*). This is consistent with our argument that the method MDL is able to remove blocking artifacts without blurring edges.

5. CONCLUSION

In this paper, we apply multiple dictionary learning on removing blocking artifacts. We discuss its implementations and choices of parameters for this purpose and conclude that only 2 subspaces can be used. We provide insights of the reason why MDL performs well. Our experiments on a benchmark dataset of 50 images confirm that MDL not only improves SSIM but also PSNR too.

6. REFERENCES

[1] T. Goto, Y. Kato, S.Hirano, M. Sakurai, and T. Q. Nguyen, "Compression artifact reduction based on total variation regularization method for mpeg-2," *IEEE Trans. on Consumer Electronics*, vol. 57, no. 1, pp. 253–259, 2011.

[2] S. Liu and A. C. Bovik, "Efficient dct-domain blind measurement and reduction of blocking artifacts," *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 12, no. 12, 2002.

[3] Y. Yang and N. P. Galatsanos, "Removal of compression artifacts using projections onto convex sets and line process modeling," *IEEE Trans. Image Process.*, vol. 6, 1997.

[4] J. J. Zou and H. Yan, "A deblocking method for bdct compressed images based on adaptive projections," *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 15, no. 3, pp. 430–435, 2005.

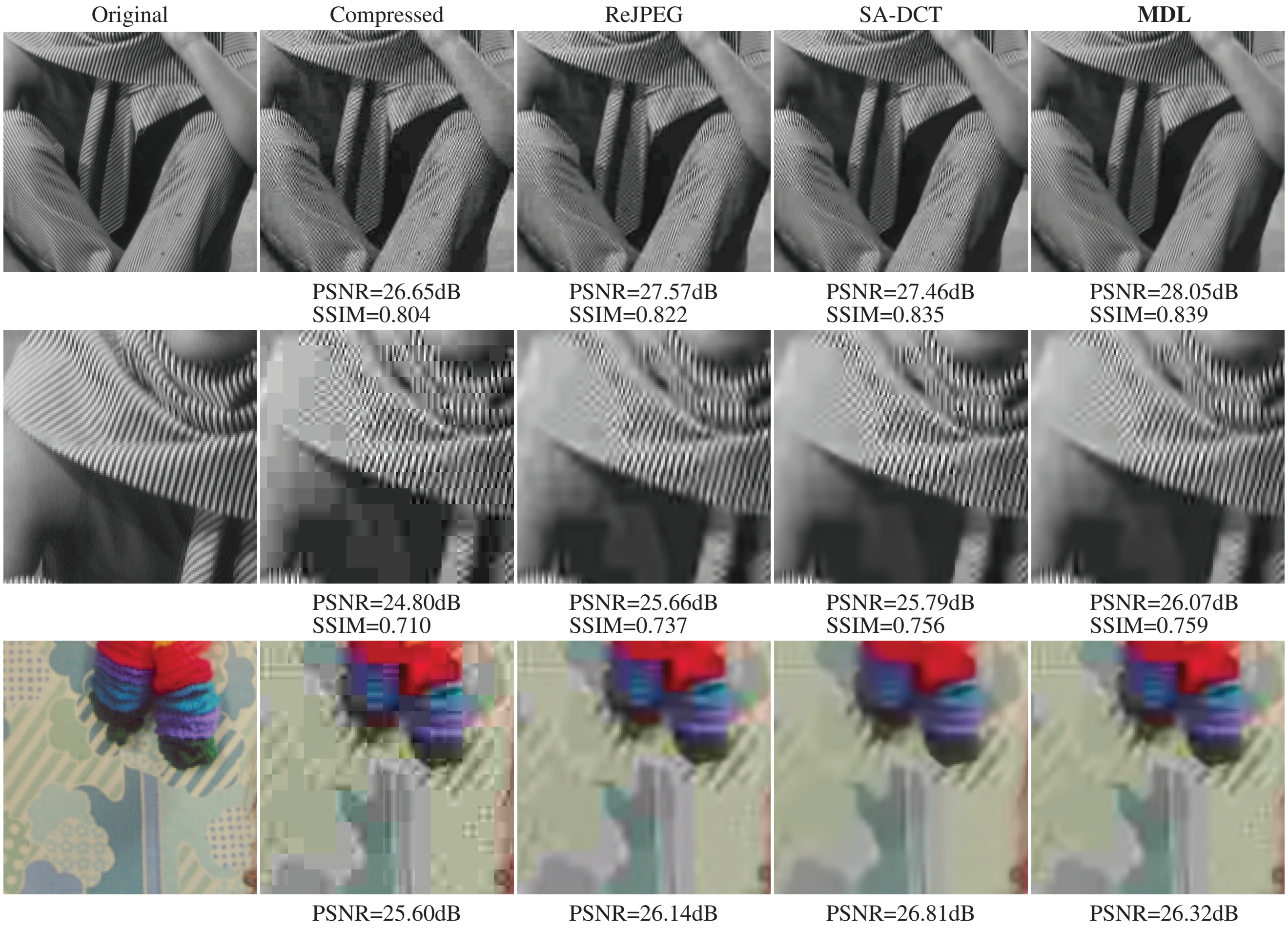


Fig. 4. Perceptual results on some examples. MDL obtains balanced performance (improves $1dB$ in PSNR on average) in both textures and uniform regions and is especially advantageous in textures. On the other hand, SA-DCT tends to oversmoothen, e.g. the lower right region of the color image *Isabel* above. ReJPG is less effective in uniform regions, e.g. the arm of *Barbara* in the above examples.

[5] H. Kong, A. Vetro, and H. Sun, "Edge map guided adaptive post-filter for blocking and ringing artifacts removal," in *ISCAS*, 2004, pp. 929–932.

[6] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, 2007.

[7] G. Zhai, W. Lin, J. Cai, X. Yang, and W. Zhang, "Efficient quadtree based block-shift filtering for deblocking and deringing," *J. Visual Commun. and Image Represent.*, vol. 20, no. 8, pp. 595–607, 2009.

[8] A. Chetouani, G. Mostafaoui, and A. Beghdadi, "Deblocking method using a percpetual recursive filter," in *ICIP*, 2009, pp. 3881–3884.

[9] A. Nosratinia, "Enhancement of jpeg-compressed images by re-application of jpeg," *J. VLSI Signal Process.*, vol. 27, pp. 69–79, 2001.

[10] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *CVPR*, 2006, pp. 17–22.

[11] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *Siam Review*, vol. 51, 2009.

[12] G. Yu, G. Sapiro, and S. Mallat, "Image modeling and enhancement via structured sparse model selection," in *ICIP*, 2010.

[13] G. M. Farinella and S. Battiato, "On the application of structured sparse model selection to jpeg compressed images," in *Proceedings of the Third international conference on computational color imaging*, 2011, CCIW'11, pp. 137–151.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[15] Z. Wang, H. R. Sheikh, and A. C. Bovik, "No-reference perceptual quality assessment of jpeg compressed images," in *ICIP*, 2002, pp. 477–480.

[16] P. Le Callet and F. Autrusseau, "Subjective quality assessment irccyn/ivc database," 2005, http://www.irccyn.ec-nantes.fr/ivcdb/.