Independent Subspace Analysis: Blind Source Separation

Juliette Achdou, Salma El Alaoui Talibi, and Camille Jandot

April 3, 2017

Abstract

Blind Source Separation (BSS) consists in recovering a set of source signals from a set of their mixtures. One way of performing BSS is to assume that the sources are mutually statistically independent: this formulation of the problem is called ICA and has been explored by many authors. But relaxing this hypothesis, we can also assume that not all sources are independent, but some groups of the sources are independent: this is the Independent Subspace Analysis problem. We present several algorithms to solve both ICA and ISA: in particular FastICA and JADE for the ICA problem, and MICA (using decompositions of JADE and FastICA) and FastISA for the ISA problem. As FastISA is only efficient in a neighborhood of the solution, we propose to initialize it with the mixing matrix output by FastICA, a fast algorithm which sometimes gives inaccurate solutions. Moreover, we propose to automate the projection step in MICA in the case of image data, using an appropriate feature descriptor and kernel.

1 Introduction

1.1 Blind Source Separation

The problem of *Blind Source Separation* (BSS) consists in retrieving the original sources from observed mixtures. One well-known example of application is the cocktail party problem. Its configuration is the following: several recordings of the same signal are gathered during a reception and we aim at separating the sources, that is we want to recover the conversations.

The problem of blind source separation can be generalized as follows: m records of sensor signals coming from a mixing and filtering system are observed, $\mathbf{x}(\mathbf{k}) = [x(k_1), \dots, x(k_T)] \in \mathbb{R}^{T\times 1}$ where k is a discrete variable (time, distance etc.). We then aim to find an inverse system in order to estimate the primary source signals $\mathbf{s}(\mathbf{k}) = [s(k_1), \dots, s(k_T)] \in \mathbb{R}^{T\times 1}$ up to certain indeterminacies, such as arbitrary scaling, permutation. In an important number of applications, these limitations are not essential, as we are more interested in the temporal waveforms or time-frequency patterns of the source signals, rather than their amplitudes or the order in which they appear.

Most of the time, we consider the following model where the sources were mixed linearly:

$$X = AS + V \tag{1}$$

where $X = [\mathbf{x_1}, \dots, \mathbf{x_m}]^\top \in \mathbb{R}^{m\times T}$ is the observation matrix, m is the number of available samples(mixtures), and T the number of observations. $A \in \mathbb{R}^{m\times n}$ denotes the unknown mixing matrix and n the number of sources; $V \in \mathbb{R}^{m\times T}$ is a matrix representing the noise and $S = [\mathbf{s_1}, \dots, \mathbf{s_n}]^\top \in \mathbb{R}^{n\times T}$ contains the unknown source signals.
BSS aims at finding the unmixing matrix W such that:

$$S(\mathbf{k}) = WX(\mathbf{k})$$

under different hypotheses on the sources which are the rows of the matrix S. For example, they should be statistically mutually independent for *Independent Component Analysis* (ICA) [1, 2, 3], or sparse for *Sparse Component Analysis* (SCA) [4, 5]. It can also be required that the estimated components take only non-negative values, in the case of *Non-Negative Matrix Factorization* (NMF). We will develop the ICA-based approach.

1.2 BSS using Independent Component Analysis

Independent Component Analysis for BSS consists in solving the Blind Signal Separation problem in the case where we have as many sources as observed mixtures of the original sources (eg. in the cocktail party problem, there are as many micros recording the conversations as independent conversations to be reconstructed) and where we assume that all sources are statistically independent.

Many authors have tackled the problem of Blind Source Separation using Independent Component Analysis. Several algorithms have been introduced to solve ICA, which corresponds to the problem posed in Equation (1) with the fundamental hypothesis that the sources $\mathbf{s}(\mathbf{k})$ are statistically mutually independent. Most of the associated algorithm tend to minimize a "measure of independence", also called contrast function of a quantity $w^\top \mathbf{x}$. Jutten and Herault, while introducing the INCA (now called ICA) problem [1], proposed a neuronal approach to solve it. Another approach, based on contrast maximization was introduced by Comon [2]. It consists in maximizing a contrast function by applying an algorithm that proceeds pairwise: it estimates pairwise cumulants and finds a plane rotation that maximizes the contrast in the plane defined by the pair, for each pair of indices. Bell and Sejnowski [6] implemented another algorithm based on the infomax principle, that maximizes the entropy of the outputs of a neural network with non linear scalar functions. Several authors proved that the infomax principle is equivalent to maximum likelihood estimation under certain circumstances. Hyvärinnen and Oja [3] introduced FastICA, an algorithm based on a fixed point algorithm for the maximization of different contrast functions that approximate negentropy, and are a posteriori measures of non-gaussianity. The JADE algorithm [7] introduced by Cardoso uses the fact that the whitened signal is a unitary mixture of the source signal, and that this unitary mixture matrix diagonalizes any cumulant matrix of the whitened signal, in order to identify the unitary matrix. In this algorithm the computation of the cumulants is followed by a joint diagonalization algorithm. Second-order blind identification (SOBI), proposed by Belouchrani *et al.* [8] also relies on joint diagonalization. The authors propose the joint diagonalization of a set of arbitrary covariance matrices (second order statistics) instead of the joint diagonalization of the cumulants as in JADE (fourth order statistics). In Hessian ICA [9], Theis proposes to joint diagonalize Hessians of the logarithmic density (or characteristic function).

1.3 Independent Subspace Analysis

In ICA algorithms that were introduced beforehand, all sources are assumed to be independent. Nevertheless, in some situations, not all sources are independent (for instance, two songs are being played by two different bands, and we want to separate the two songs): some sources are multidimensional. We consider M multidimensional sources, each of them being d-dimensional (we will assume that the dimensions of the independent group of sources are the same for the sake of simplicity).

This new problem has been first addressed by Cardoso [10] under the name of *Multidimensional Independent Component Analysis* (MICA). MICA is one way to solve *Independent Subspace Analysis* problem (ISA). MICA relies on first applying an ICA algorithm (JADE) to estimate one-dimensional sources, and then determining which of these sources are indeed independent.

Other approaches have been introduced to solve ISA. Theis proposes to extend the algorithms Hessian ICA and SOBI, to the multidimensional blind source separation problem. He introduces MHICA and MSOBI [11], that both rely on joint block diagonalization. Póczos *et al.* follow the approach of minimizing the entropy, which they estimate using k-nearest neighbors estimations [12], or geodesic spanning trees [13]. Hyvärinen *et al.* propose FastISA [14], a variant of FastICA that enables to solve the ISA problem.

2 Elements on ICA

Independent Component Analysis is the problem of recovering A and S given (1), under the hypothesis that the sources are statistically mutually independent. In this report, we will consider that the noise term, V, is zero. After having shown some generalities on the subject, we describe two algorithms, JADE (Cardoso) and FastICA (Hyvärinnen).

2.1 Generalities

Indeterminacies. In ICA, it is clear that there are some indeterminacies:

- Order indeterminacy: if $X = AS$, and A' is the result of a permutation of the columns of X and S' the result of a permutation of the rows of S, we also have: $X = A'S'$
- Scale indeterminacy: the columns of A and the rows of S are determined up to a scalar multiplication. To solve this, we could assume that each source signal has unit variance, this technique however leaves an indeterminacy regarding the sign.

Impossible separation of gaussian signals. ICA can not separate gaussian signals. In fact if the mixing matrix is orthogonal and the source signals are independent, their joint distribution is completely symmetric, which makes it impossible to extract information on A.

Non-gaussianity. The goal of ICA comes down to maximizing non-gaussianity. For simplicity, we assume that all $\mathbf{s_i}$ are iid. We assume that we multiply X by the transpose of a vector w: $w^\top X = z^\top S$ where $z = A^\top w$: $w^\top X$ is a linear combination of the s_i. It is thus always less gaussian than the original signals, unless it equals one of the $\mathbf{s}_i$. This gives the intuition that the problem we want to solve comes down to maximizing the non-gaussianity of a quantity $w^\top X$.
There are several measures of non-gaussianity

Definition 2.1 (Kurtosis)**.** *The kurtosis of the random vector y is defined as*

$$kurt(y) = \mathbb{E}(y^4) - 3\mathbb{E}(y^2)^2$$

In the case where y is of unit variance, the kurtosis is equal to the fourth order moment up to an additive constant.

Definition 2.2 (Negentropy)**.** *The negentropy of the random vector y is defined as*

$$J(y) = H(y_{gauss}) - H(y)$$

where $H(y) = -\int f(y)\log f(y)dy$ and y_{gauss} is a Gaussian random variable of the same covariance matrix as y.

The entropy is a measure of the information given by an observation. One can prove that the negentropy is zero if and only if y has a gaussian distribution. There exist several approximations of negentropy that are computationally stable, which will be detailed in the section on FastICA.

2.2 FastICA

Goal of FastICA. FastICA aims at minimizing an approximation of the negentropy of $w^\top x_i$ for all i, in order to maximize their non-gaussianity. The classical way of approximating negentropy is using this kind of estimation:

$$J(y) \approx \frac{1}{12}\mathbb{E}(y^3)^2 + \frac{1}{48}\text{kurt}(y)^2$$

Hyvärinnen developed other approximations of negentropy. In particular, we have:

$$J(y) \approx \big(\mathbb{E}(G(y)) - \mathbb{E}(G(v))\big)^2$$

for practically any non-quadratic function G, which generalizes the former approximation. Some choices of G have proved particularly efficient, as for example:

$$G_1(t) = \frac{1}{a_1}\log\cosh(a_1 t) \text{ and } G_2(t) = -\exp(-t^2/2)$$

where $1 < a_1 < 2$.

Preprocessing for FastICA. The preprocessing step that has to be applied before FastICA consists in a step of centering followed by a step of whitening. By whitening, we mean the fact of linearly transforming the centered $\mathbf{x_i}$ into new vectorss whose covariance matrix is equal to an identity matrix.

There are several ways to perform whitening on a vector z:

- **ZCA-Whitening** consists of performing an eigenvalue decomposition on the covariance: $\mathbb{E}(zz^\top) = EDE^\top$, and computing $z_{\text{whitened}} = ED^{-\frac{1}{2}}E^\top z$.
  It is easily verified that $\mathbb{E}(z_{\text{whitened}} z_{\text{whitened}}^\top) = I$.
- **PCA-Whitening** also starts by performing an eigenvalue decomposition on the covariance: $\mathbb{E}(zz^\top) = EDE^\top$, and ends by computing $z_{\text{whitened}} = D^{-\frac{1}{2}}E^\top z$.

Note that eigenvalue decomposition implies scale and order indeterminacy.

FastICA for one source. FastICA is a fixed point algorithm that minimizes an approximation of negentropy: $\mathbb{E}(G(y)) - \mathbb{E}(G(v))$, where G is chosen among $G_1(t) = \frac{1}{a_1}\log\cosh(a_1 t)$ $G_2(t) = -\exp(-t^2/2)$. g denotes the derivative of the chosen G, namely: $g_1(t) = \tanh(a_1 t)$ and $g_2(t) = \exp(-t^2/2)$.

Algorithm 1: FastICA algorithm

input : X, Initial (random) vector w
output : w
for t = 1,...,K **do**
  $w \leftarrow \mathbb{E}(Xg(w^\top X)) - \mathbb{E}(Xg'(w^\top X))w$;
  $w \leftarrow \frac{w}{\|w\|}$;
end

Complete FastICA. FastICA for one source only finds one direction w such that $w^\top X$ maximizes the chosen approximation of negentropy. In order to proceed to the complete FastICA , we need to run "FastICA for one source" n times. However, if we do so, there are some chance that all w_i will converge to the same solution. In order to avoid this, we can decorrelate $w_1^\top X \dots w_n^\top X$ One way to do this (symmetric decorrelation) is to update:

$$W \leftarrow \left(WW^\top\right)^{\frac{1}{2}} W$$

.

2.3 JADE

2.3.1 Theoretical Elements

Whitening As already described in the paragraph about preprocessing for FastICA, there exist ways to whiten a signal via a whitening matrix R such that RX is white ($R = ED^{-\frac{1}{2}}E^\top$ for ZCA-Whitening and $R = D^{-\frac{1}{2}}E^\top$ for PCA-Whitening).
Without loss of generality, as there is a scale indeterminacy, we can assume that the source signals have unit variance. This hypothesis yields:

$$RXX^\top R^\top = I_n = RAA^\top R^\top$$

Hence $RA = U$ where U is unitary and $A = R^{\#}U$.
The goal of JADE will hence be to find the unitary matrix U.

Two approaches are "mixed" to construct JADE.

Approach based on eigendecomposition. We define a "cumulant matrix associated to M", M being a $n \times n$ matrix, by a matrix $Q_{\widetilde{X}}(M)$ which components are:

$$Q_{\widetilde{X}}(M)_{i,j} = \sum_{1\leq(k,l)\leq n} \mathcal{C}\left(\tilde{\mathbf{x}}_{\mathbf{i}}, \tilde{\mathbf{x}}_{\mathbf{j}}^{\top}, \tilde{\mathbf{x}}_{\mathbf{k}}, \tilde{\mathbf{x}}_{\mathbf{l}}^{\top}\right) m_{l,k}$$

where $\mathcal{C}\left(\tilde{\mathbf{x}}_{\mathbf{i}}, \tilde{\mathbf{x}}_{\mathbf{j}}^{\top}, \tilde{\mathbf{x}}_{\mathbf{k}}, \tilde{\mathbf{x}}_{\mathbf{l}}^{\top}\right)$ is a fourth order joint cumulant. Using cumulant properties, one can prove that:

$$Q_{\widetilde{X}}(M) = \sum_{1\leq p\leq n} \left(k_p u_p^{\top} M u_p\right) u_p u_p^{\top} \tag{2}$$

where u_p are the rows of U, which is equivalent to

$$Q_{\widetilde{X}}(M) = U D_M U^{\top},$$

where $D_M = \mathrm{diag}\left(k_p u_p^{\top} M u_p\right)$.

The latter computation shows that any "cumulant matrix" is diagonalized by U. Hence a possible approach would be to compute any "cumulant matrix" and diagonalize it to find U(up to ICA's indeterminacies).
As there is a priori no right way to choose M, the proposed algorithm will joint-diagonalize a well-chosen selection of "cumulant matrices"

Approach based on optimization of a cumulant criterion. It can be proved that a valid criterion for non-gaussianity is the sum of the squared joint-cumulants with the same first and second indices: $\mathrm{crit}_2(V) = \sum_{i,k,l} \mathcal{C}\left((V^{\top}\widetilde{X})_i, (V^{\top}\widetilde{X})_i^{\top}, (V^{\top}\widetilde{X})_k, (V^{\top}\widetilde{X})_l^{\top}\right)$. One possible approach is hence to maximize this criterion on $V^{\top}X$ under the constraint that V is unitary, in order to find $U = \underset{(V\text{unitary})}{\arg\max}\ \mathrm{crit}_2(V)$.

Joint Diagonalization. The idea of JADE is to joint-diagonalize a well-chosen selection of "cumulant matrices". This implies the knowledge of a technique of joint diagonalization. Cardoso and Souloumiac give a clear description of one of the possible procedure for simultaneous diagonalization in [15]. One way to apply joint diagonalization to a set of normal commuting matrices A_k is to maximize the sum on k of the sums of the absolute values of all diagonal elements of $OA_kO^{\top}$ with O unitary: $\sum_{k\leq n}\sum_{i=j} |(OA_kO^{\top})_{i,j}|^2$. To do so, for each pair (u,v) in $(1,\dots,n)$, we look for the rotation $R_{u,v}$ in the plane defined by the the pair (u,v) that maximizes $\sum_{k=n}\sum_{i\neq j} |(R_{u,v}A_k{R_{u,v}}^{\top})_{i,j}|^2$.

Link between the latter approaches. Note that the criterion: $\sum_{k\leq n}\sum_{i=j} |(VA_kV^{\top})_{i,j}|^2$ has the nice property that if the A_k form the set of parallel slices: $\mathcal{N} = \{Q_{\widetilde{X}}(J_{k,l})\}$ where $J_{k,l}$ is the matrix with only one component equal to 1: the one at the (k,l) position, and otherwise all zeros, it is equal to the criterion of the second approach $\mathrm{crit}_2(V)$.

Reduction of the 4^{th}-order cumulants by eigenmatrices. For any d-dimensional random vector v, with 4^{th}-order cumulants, there exists d^2 real numbers $\lambda_1, ..., \lambda_{d^2}$ and d^2 matrices $M_1, ..., M_{d^2}$ called eigenmatrices verifying

$$Q_v(M_r) = \lambda_r M_r \ \mathrm{Tr}(M_r M_s^{\top}) = \delta(r,s)\ \forall r,s \in [1,d]$$

Thanks to equation (2) one finds that $Q_{\widetilde{X}}(u_pu_p^{\top}) = k_pu_pu_p^{\top}$ and $Q_{\widetilde{X}}(u_pu_q^{\top}) = 0\ \forall p \neq q^{\top}$: thus $\mathrm{Ker}(Q_{\widetilde{X}})$ has dimension $n\times(n-1)$ and we are only interested in the n highest eigenvalues: this allows to only compute n eigenmatrices instead of n^2 cumulant matrices.

2.3.2 JADE Algorithm

Algorithm 2: JADE algorithm

input : X
output : $\widetilde{A}$
for t = 1,...,K **do**
  Compute the whitening matrix R;
  Form the 4$^{\text{th}}$ order cumulants $Q_{\widetilde{X}}$ of the whitened process $\widetilde{X}$: compute the n most significant eigenpairs $\{\lambda_r, M_r\}$;
  Jointly diagonalize the set $\{\lambda_r M_r\}$ by a unitary matrix;
  Return $\widetilde{A} = R^{\#}U$;
end

3 Independent Subspace Analysis

Independent subspace analysis is the generalization of ICA to the case where not all sources are independent. We consider N groups of multidimensional sources. The goal is to separate signals coming from the independent groups. We describe two algorithms, MICA (Cardoso) and FastISA (Hyvärinen *et al.*).

3.1 Multidimensional Independent Component Analysis

Cardoso proposes an extension of Independent Component Analysis that enables to solve the ISA problem [10]. He first introduces a geometric parametrization that rephrases the ICA problem: we are not looking for the mixing matrix A, but for the "component spaces", or equivalently for the set of projectors $\{\Pi_p\}_{p=1,\dots,n}$ on these spaces. He introduces the projector on the p-th component space $\widetilde{\Pi}_p$ defined as:

$$\widetilde{\Pi}_p = \Pi_p \left(\sum_{q=1}^{n} \Pi_q \right)^{\#},$$

where $\Pi_p = \frac{a_p a_p^\top}{a_p^\top a_p}$ (a_p denotes the p-th a column of A), such that $x_p = \widetilde{\Pi}_p x$. The problem expressed in terms of the projectors is transposable to the multidimensional case.

We now try to find a solution to the multidimensional blind source separation problem. In the case we are not dealing with one-dimensional spaces, the projectors Π_p for $p \in [1, \dots, N]$, with N the number of subspaces, can be written

$$\Pi_p = A_p \left(A_p^\top A_p \right)^{-1} A_p^\top,$$

where the A_p for $p = 1, \dots, M$ are such that $A = [A_1, \dots, A_M]$ is full column rank.

MICA algorithm relies on two main steps. The first step is to use an ICA algorithm, eg. FastICA (Algorithm 1), JADE (Algorithm 2), to produce a "mixing matrix". The independence of the sources hypothesis not being fulfilled, there is still some work to be done, and this is where the second step comes: we need to group together the "sources" output by the first step that are not independent. Details on how to use MICA are given in Section 4.1, where we reproduce Cardoso's foetal ECG extraction experiment.

Cardoso proves that this process is justified by showing that under general assumptions, most ICA algorithms will at least separate the independent groups of dependent signals.

3.2 FastISA

FastISA [14] is a fixed-point algorithm that solves the ISA problem, which is similar to the algorithm FastICA introduced in Section 2.2. It is proven to converge quadratically to a local minimum.

Instead of asking the one-dimensional sources to be independent, we want the norms of the projections on the subspaces $\sqrt{\sum_{j \in S_j} s_j^2}$ to be independent, ie. we want to maximize the independence of the norms of the projections. The authors suggest to do so by maximizing

$$\log p(s_1, \ldots, s_n) = \sum_{j=1}^{n} \left(-\log Z_j - \frac{G(\sum_{i \in S_j} s_i^2)}{b} \right), \tag{3}$$

where G is a contrast function and $s_i^2 =$. The authors suggest to use $G(x) = \sqrt{x + \gamma}$.

We first apply the same preprocessing steps as in FastICA (centering and whitening the data). We maximize (3) with the following algorithm, where g denotes the derivative of the contrast function G:

Algorithm 3: FastISA algorithm

input : Matrix of signals X, maximum number of iterations K
output : Unmixing matrix W
i=0 ;
repeat
    i++;
    for j = 1,...,n **do**
        $w_j \leftarrow \mathbb{E}\left[X(w_j^\top X) g\left(\sum_{i \in S_j} \left(w_i^\top X\right)^2\right)\right] -$
        $\mathbb{E}\left[g\left(\sum_{i \in S_j} \left(w_i^\top X\right)^2\right) + 2\left(w_i^\top X\right)^2 g'\left(\sum_{i \in S_j} \left(w_i^\top X\right)^2\right)\right] w_j$;
    end
    orthogonalize W;
until *convergence or* ($i == K$);

Experiment on a toy dataset To validate this algorithm, we reproduce the experiment presented in the paper [14]: we generate mixtures of supergaussians with a subspace structure, mix them with a random mixture matrix, and separate them using FastISA algorithm. We generate the supergaussians by first generating $N = 20$ normal gaussian signals of $T = 10\,000$ samples. We divide these in subspaces of dimension 5 and multiply each "group" by a random signal drawn from a uniform distribution. We initialize the algorithm with the sum of the unmixing matrix and white noise.
To check that the algorithm converges to the expected solution, we compute WRA, and check that this matrix is block-diagonal. The matrix is represented in Fig. 1. For a more exact approach, we also computed its Amari index: its value was of 0.026.

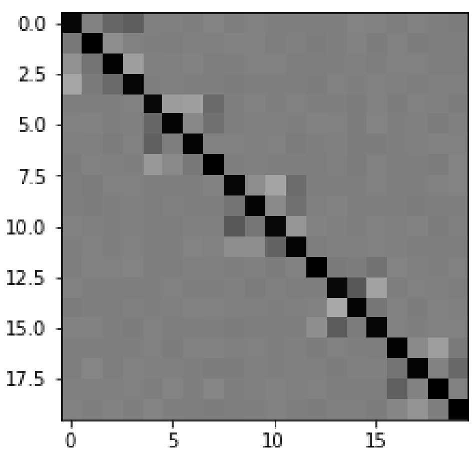

Figure 1: FastISA toy example: WRA

3.3 Amari Index

To validate the results of ISA, there exists an index that generalizes Amari's index for ICA [16] to ISA.

This Amari index is equal to:

$$I_{\text{Amari}}(M) = \frac{1}{2N(N-1)} \sum_{r=1}^{N} \left(\sum_{s=1}^{N} \frac{\|M_{r,s}^{(d)}\|}{\max_i \|M_{r,i}^{(d)}} \| - 1 \right) + \frac{1}{2N(N-1)} \sum_{s=1}^{N} \left(\sum_{r=1}^{N} \frac{\|M_{r,s}^{(d)}\|}{\max_i \|M_{i,s}^{(d)}\|} - 1 \right)$$

where d is the subspace dimension, N the number of subspaces, $M_{r,s}^{(d)}$ the $d \times d$ block matrix whose first component is at position (rd, sd).
This index measures the similarity of M with a permuted d-block-diagonal matrix. It is equal to zero when M is the permutation of a d-block-diagonal matrix, and one when all blocks have the same norm.

We compute this index on WRA to evaluate the performance of ISA-algorithms.
Note that in order to evaluate the performance of ICA-algorithms, we just have to set d to 1 and N to the shape of the matrix.

4 Experiments and Results

The code and instructions to reproduce the following experiments may be downloaded here.

4.1 Foetal ECG extraction using MICA

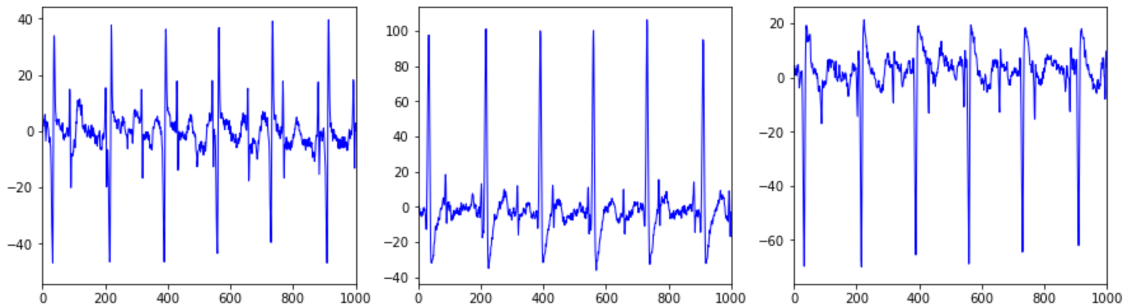

Figure 2: ECG signals coming from 3 sensors placed on the abdomen of a pregnant women

We use the MICA algorithm introduced in Section 3.1 on electro-cardiogram data. The goal is to extract foetal ECG from the ECG of a pregnant women. We use the same dataset as Cardoso for this experiment. It contains 8 signals; we only keep the first three recordings (Fig. 2), that were captured using electrodes placed on the abdomen of the patient. Each captor signal contains 2 500 measurements. The first step of MICA is performed using JADE algorithm, using Cardoso and Beckers's implementation [17] [1]. JADE outputs the unmixing matrix W, that once multiplied with the measured signals X, provides the estimations of supposedly independent sources. One can easily see (Fig. 3) that the frequency of one out of the three samples is higher than the other two: it is the foetal signal.

We then project the obtained signals on the mother space using $\widetilde{\Pi}_{\text{mother}}$, and on the foetus space using $\widetilde{\Pi}_{\text{foetus}}$. We obtain the estimations of the source signals displayed in Fig. 4.

We also tried to apply MICA using FastICA in the first step of the algorithm instead of JADE, but it did not extract the foetal ECG (there is no difference of frequency at the end of the ICA step). Some authors claim that FastICA indeed fails at extracting ECG signals without further work on the data [18, 19].

[1] http://perso.telecom-paristech.fr/c̃ardoso/Algo/Jade/jade.py

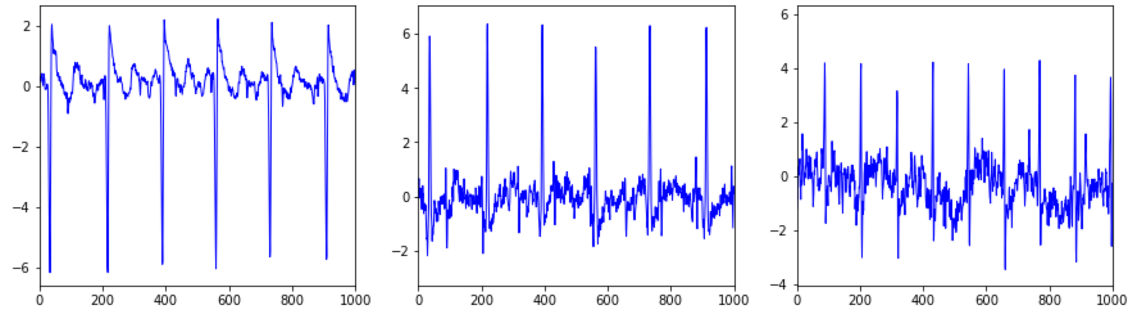

Figure 3: Components output by ICA (JADE). The right picture shows a signal coming from the foetus (higher frequency).

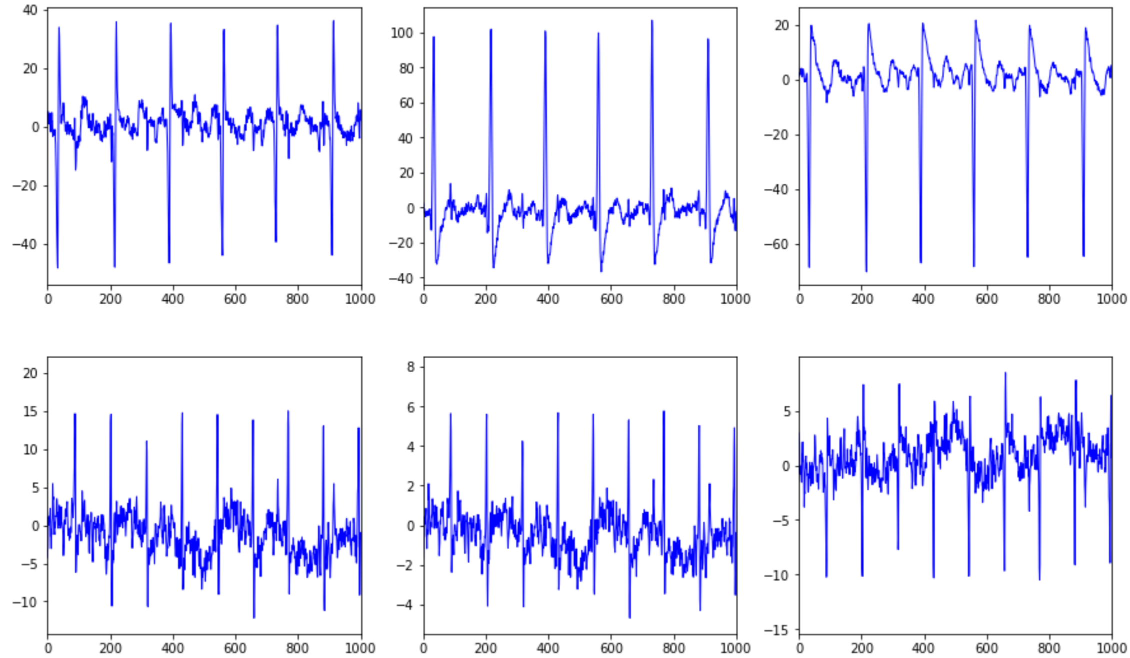

Figure 4: MICA components – Above: mother, below: foetus

4.2 Separating Mixtures of Images

In this experiment, we reproduce the classical ISA setting by attempting to recover independent subspaces of images from mixtures of dependent and independent images. The first subspace is composed of two source images of Emma Watson, one being a simple translation of the other, this way making sure that they are not statistically mutually independent. The second subspace is composed of two images of grass lawns with similar textures. Under the assumption that the two subspaces are independent, the objective is to recover each subspace without necessarily separating the dependent sources within each subspace.

MICA with Jade. Fig. 5 shows the intermediate steps and results of performing this experiment using the MICA method [10], with JADE (algorithm 2) as the underlying ICA algorithm. However, we streamline the method presented in the paper. Indeed, after performing the ICA step, and recovering the supposedly independent sources, the author compares the signals visually in order to deduce in which subspace to project them. Instead, we automate this step by extracting the Histogram of Oriented Gradients features [20] of each recovered source, and then compute a similarity matrix using a Chi-squared kernel. This allows us to directly assign each of the recovered sources to the right subspace, and then we proceed with the projections as previously explained in section 3.1.
The results on this data are very satisfactory, as JADE succeeds in recovering the sources (third

row of Fig. 5) almost flawlessly even before we perform the projections in the two subspaces.

MICA with FastICA. Fig. 6 shows the intermediate steps and results of performing the same experiment with MICA, this time using FastICA (Algorithm 1) as the underlying ICA algorithm. We can observe in the last two rows of Fig. 6 that the projections into the *Emma Watson-subspace* and *grass-subspace* are not as satisfying as they were with JADE. However, this experiment is useful in showing the importance of the projection step, since we can see that it has cleary improved the ICA decomposition which is visible in the third row.

FastISA. The main issue in applying the FastISA Algorithm 3 is that the author of [14] proved its convergence in a neighborhood of the solution (true unmixing matrix); which is why the algorithm is initialized with a sum of the unmixing matrix and white noise in the toy experiment presented in the paper.
We initialize the FastISA algorithm with the unmixing matrix obtained by FastICA decomposition in the previous experiment, that presented a poor separation of the sources (third row of Fig. 7), to which we added noise. We can see the last row of Fig. 7, in which the FastISA subspaces are displayed, presents a substantial improvement.
Performing FastISA after FastICA yields better results in terms of Amari index than FastICA alone. Indeed, the Amari index with FastICA is 0.44 with the mixing matrix of the experiment, while FastISA initialized with the unmixing matrix of FastICA, reaches an Amari index as low as 0.20, with an average of 0.42 (10 runs).
One can note that in terms of time complexity, FastICA followed by FastISA is significantly better than Jade-MICA.

4.3 Rock Bands' Cocktail Party

We chose to tackle the Cocktail Party problem, with two different variants. We first simulated the ICA variant: two songs are played at the same time. We then tested the ISA variant: two bands are playing in the same room, and we want to separate the instruments playing the first song from the instruments playing the second song. All links to the audio files that have been used for or generated by the experiment are recapitulated in Appendix A.

4.3.1 Two songs played simultaneously (ICA)

The data we used were two songs, *Let It Be* by The Beatles, and *Thunderstruck* by ACDC. We mixed the two songs, and tried to retrieve the original signals. We tried to do so using the JADE and FastICA algorithms. JADE's performance is highly satisfactory (listen to the mixtures beginning with "2" and to the separated songs, details about the file can be found in Appendix A), the two songs are very well separated FastICA totally fails at this task.

4.3.2 Two groups playing in the same room (ISA)

We want to simulate the situation in which two groups are playing in the same room, and several micros are recording the double concert. It is difficult to obtain such data, so we simulated it using digital transcriptions of the songs. For each instrument that we wanted to add to the mixture, we exported its transcription into separate files, creating as many tracks as instruments. For *Let It Be*, the tracks were the piano and the voice, and for *Thunderstruck*, the lead guitar and the rhythmic guitar. We mixed these 4 tracks and applied MICA, both with JADE and FastICA as first steps. The results can be heard here for JADE. The first step of MICA with JADE is already satisfying. After the projections of the second step of MICA, we cannot hear any hint of *Let It Be* in the recordings of ACDC's song, but if we are careful enough, we can notice that the melody of *Thunderstruck* is still here in the reconstructed Beatles's recordings (though almost unnoticeable). FastICA does not succeed at all at separating the sources: the results of the first step are not classifiable as from one song or the other one. One can still notice that one of the reconstructed recordings is the lead guitar from *Thunderstruck*.

We also applied FastISA on this data, initializing the unmixing matrix W with the solution given by JADE. This method yields a better Amari index than JADE alone (0.48 vs. 0.71). You may listen to the results here.

5 Conclusion

We introduced the problem of Blind Source Separation and presented algorithms to solve it in the case where sources (Independent Component Analysis) or group of sources (Independent Subspace Analysis) are independent. We proposed a way to automatically group the dependent components together in the case of images, by computing the Chi-squared similarity between the HoG (histogram of gradients) descriptor of the images obtained by the first step of MICA (using JADE or FastICA), instead of performing it by hand. We could have attempted a similar approach with time series signals (ECG and sound). It would have required to find a proper representation of the signal: we could have used fast Fourier transform for instance. We also tried to initialize FastISA with the solution given by FastICA (images) or JADE (sound) which seemed to lead to better separation.

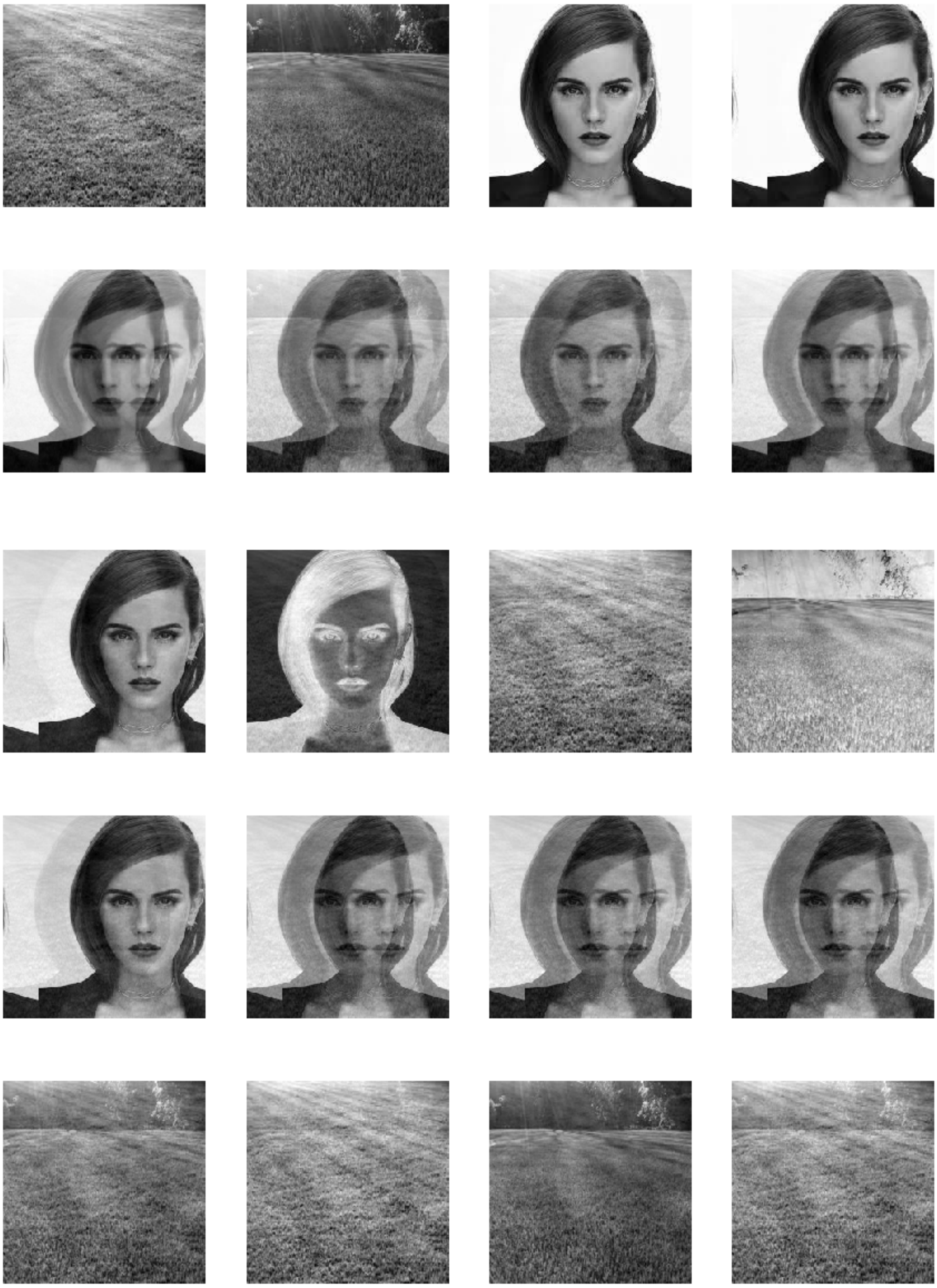

Figure 5: Results of MICA with JADE on a set of two Emma Watson and two grass lawn images. **First row:** source images. **Second row:** linearly mixed sources. **Third row:** recovered sources using JADE. **Fourth row:** projections in the *Emma Watson-subspace*. **Fifth row:** projections in the *grass-subspace*.

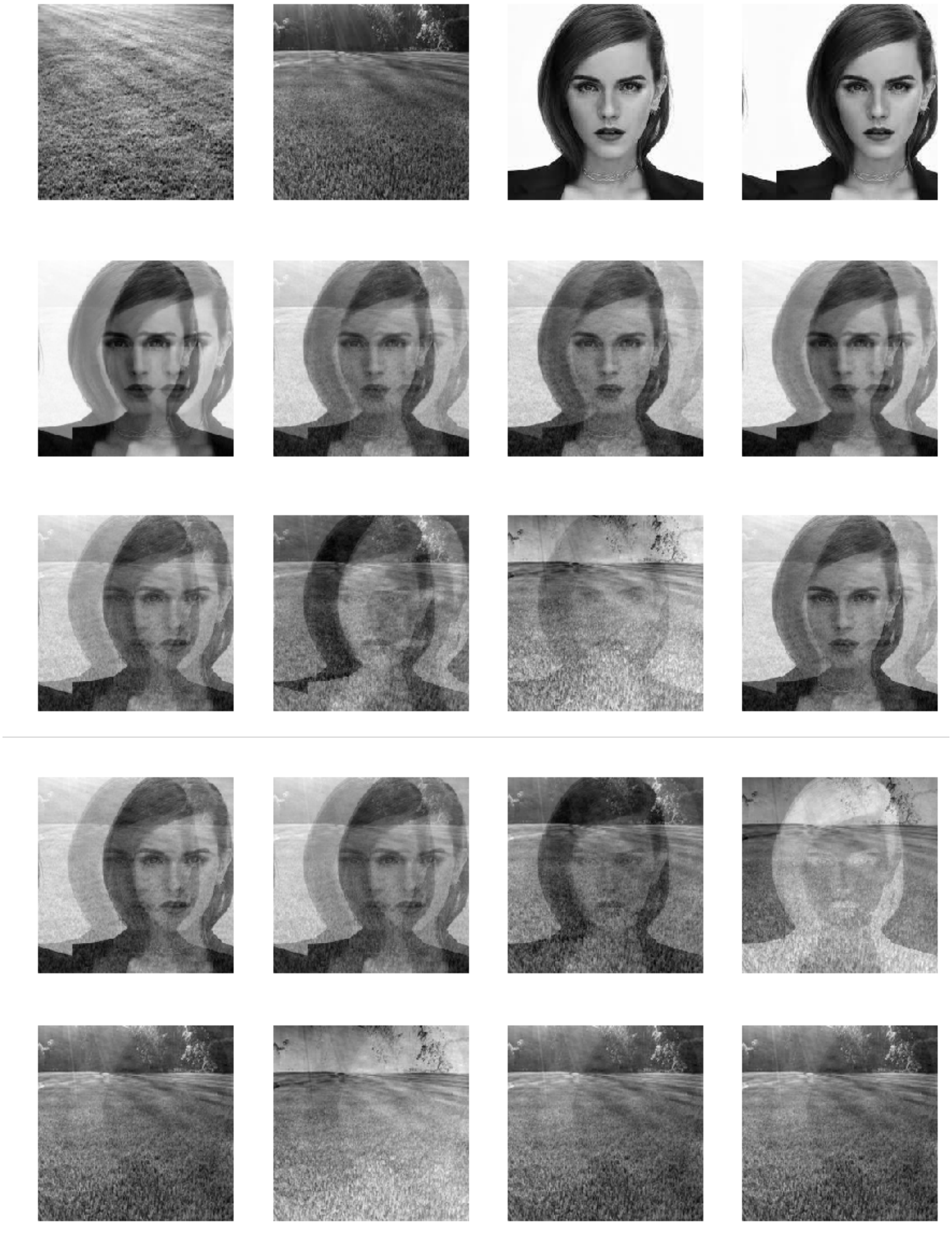

Figure 6: Results of MICA with FastICA on a set of two Emma Watson and two grass lawn images. **First row:** source images. **Second row:** linearly mixed sources. **Third row:** recovered sources using FastICA. **Fourth row:** projections in the *Emma Watson-subspace*. **Fifth row:** projections in the *grass-subspace*.

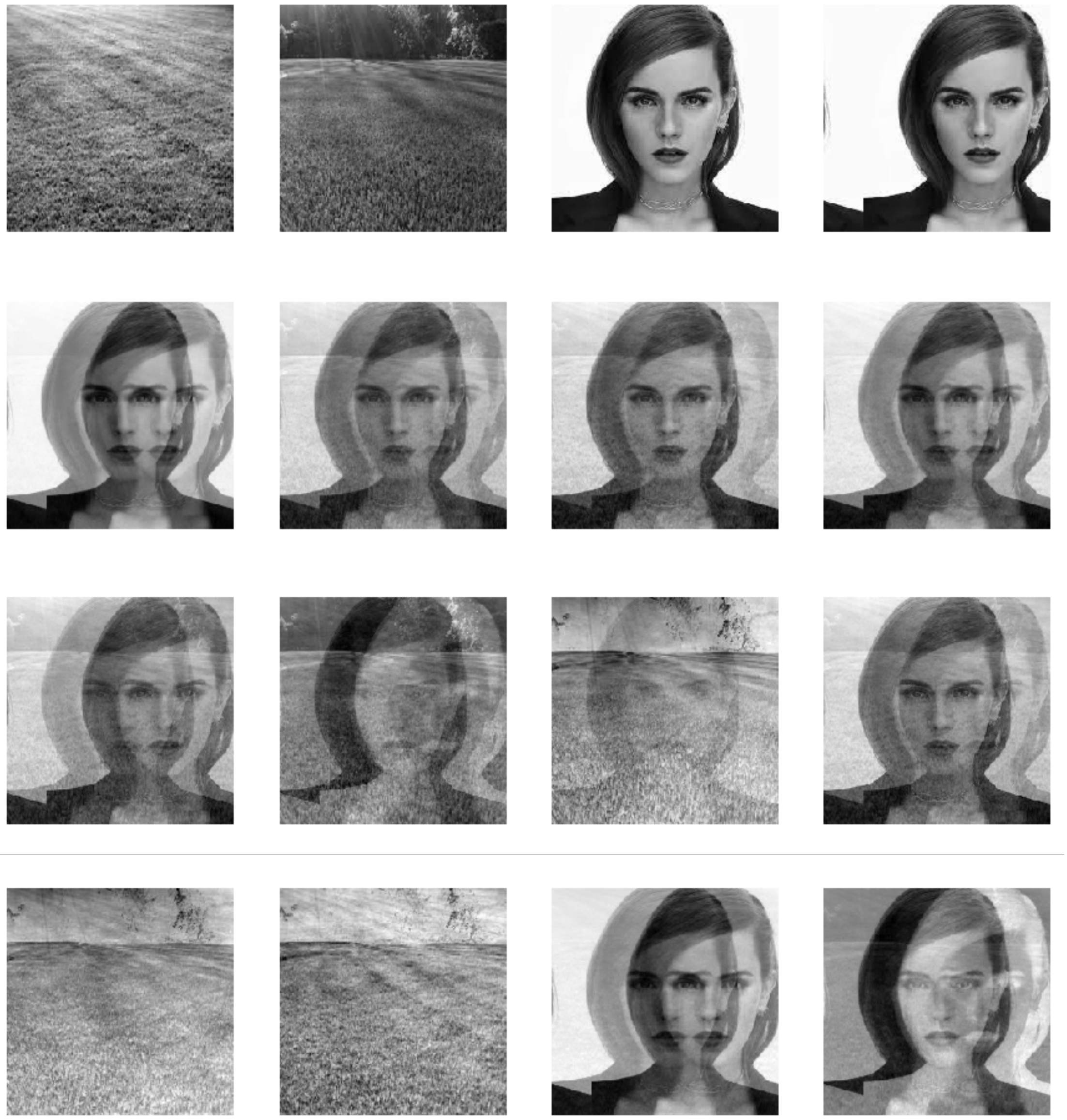

Figure 7: Results of FastISA on a set of two Emma Watson and two grass lawn images, initialized with an unmixing matrix obtained with FastICA decomposition. **First row:** source images. **Second row:** linearly mixed sources. **Third row:** recovered sources using FastICA. **Fourth row:** recovered *Emma Watson-subspace* and *grass-subspace*.

References

[1] Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10, 1991.

[2] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[3] Aapo Hyvärinen. Independent component analysis by minimization of mutual information, 1997.

[4] Michael Zibulevsky, Pavel Kisilev, Yehoshua Y Zeevi, and Barak A Pearlmutter. Blind source separation via multinode sparse representation. *Advances in neural information processing systems*, 2:1049–1056, 2002.

[5] Yuanqing Li, Andrzej Cichocki, Shun-ichi Amari, Sergei L Shishkin, Jianting Cao, and Fanji Gu. Sparse representation and its applications in blind source separation. In *NIPS*, pages 241–248, 2003.

[6] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[7] Jean-François Cardoso and Antoine Souloumiac. Blind beamforming for non-gaussian signals. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 362–370. IET, 1993.

[8] Adel Belouchrani, Karim Abed-Meraim, J-F Cardoso, and Eric Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444, 1997.

[9] Fabian J Theis. A new concept for separability problems in blind source separation. *Neural computation*, 16(9):1827–1850, 2004.

[10] J-F Cardoso. Multidimensional independent component analysis. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 4, pages 1941–1944. IEEE, 1998.

[11] Fabian J Theis. Blind signal separation into groups of dependent signals using joint block diagonalization. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 5878–5881. IEEE, 2005.

[12] Barnabás Póczos and András Lőrincz. Independent subspace analysis using k-nearest neighborhood distances. In *International Conference on Artificial Neural Networks*, pages 163–168. Springer, 2005.

[13] Barnabás Póczos and András Lõrincz. Independent subspace analysis using geodesic spanning trees. In *Proceedings of the 22nd international conference on Machine learning*, pages 673–680. ACM, 2005.

[14] Aapo Hyvärinen and Urs Köster. Fastisa: A fast fixed-point algorithm for independent subspace analysis. In *EsANN*, pages 371–376, 2006.

[15] Jean-Francois Cardoso and Antoine Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM journal on matrix analysis and applications*, 17(1):161–164, 1996.

[16] A Cichocki and HH Yang. A new learning algorithm for blind signal separation. *Advances in neural information processing systems*, 8:757–763, 1996.

[17] Jean-Francois Cardoso and Gabriel J.L. Beckers. Jade code (in matlab) and translation to python, 2007.

[18] Jarno MA Tanskanen and Jari J Viik. *Independent Component Analysis in ECG Signal Processing.* INTECH Open Access Publisher, 2012.

[19] Mohammad Niknazar. *Extraction et débruitage de signaux ECG du foetus.* PhD thesis, Université Grenoble Alpes, 2013.

[20] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005.*

Appendices

A Links to the Audio Files

This appendix summarizes all the links to audio files used or output by experiments from Section 4.3. The tracks can be downloaded, but cannot be played directly on the webpage. The link to the folder containing all the files described below is here. If you experience any issue with the links, please email us at achddou@enst.fr, salma.el-alaoui-talibi@polytechnique.fr or jandot@enst.fr .

Table 1: Links for ICA experiment (two songs played simultaneously) using JADE

Type	Links	Filename
Sources	Let It Be	Audio results/sources/LetItBe.wav
	Thunderstruck	Audio results/sources/Thunderstruck.wav
Mixtures	Mixture 1	Audio results/mixtures/2_mixtures_0.wav
	Mixture 2	Audio results/mixtures/2_mixtures_1.wav
Retrieved sources	Let It Be	Audio results/ica_jade/ica_jade_0.wav
	Thunderstruck	Audio results/ica_jade/ica_jade_1.wav

Table 2: Links for mixtures used in ISA experiment (two groups playing in the same room)

Type	Links	Filename
Sources	Let It Be, voice	Audio results/sources/LetItBe1.wav
	Let It Be, piano	Audio results/sources/LetItBe2.wav
	Thunderstruck, lead guitar	Audio results/sources/Thunder1.wav
	Thunderstruck, rhythmic guitar	Audio results/sources/Thunder2.wav
Mixtures	Mixture 1	Audio results/mixtures/4_mixtures_0.wav
	Mixture 2	Audio results/mixtures/4_mixtures_1.wav
	Mixture 3	Audio results/mixtures/4_mixtures_2.wav
	Mixture 4	Audio results/mixtures/4_mixtures_3.wav

Table 3: Links for MICA (JADE)

Type	Links	Filename
Estimated sources with JADE	Component 1	Audio results/MICA/y_source_0_jade.wav
	Component 2	Audio results/MICA/y_source_1_jade.wav
	Component 3	Audio results/MICA/y_source_2_jade.wav
	Component 4	Audio results/MICA/y_source_3_jade.wav
Let It Be subspace	Micro 1	Audio results/MICA/component_beatles_0_jade.wav
	Micro 2	Audio results/MICA/component_beatles_1_jade.wav
	Micro 3	Audio results/MICA/component_beatles_2_jade.wav
	Micro 4	Audio results/MICA/component_beatles_3_jade.wav
Thunderstruck subspace	Micro 1	Audio results/MICA/component_acdc_0_jade.wav
	Micro 2	Audio results/MICA/component_acdc_1_jade.wav
	Micro 3	Audio results/MICA/component_acdc_2_jade.wav
	Micro 4	Audio results/MICA/component_acdc_3_jade.wav

Table 4: Links for FastISA (initialized with JADE)

Type	Links	Filename
Let It Be subspace	Component 0	Audio results/fastISA/fastISA_0.wav
	Component 1	Audio results/fastISA/fastISA_1.wav
Thunderstruck subspace	Component 0	Audio results/fastISA/fastISA_2.wav
	Component 1	Audio results/fastISA/fastISA_3.wav