



HAL
open science

Importance order ranking for texture extraction. A more efficient pooling operator than max pooling?

Sofia Vargas Ibarra, Vincent Vigneron, Jean-Philippe Congé, Hichem Maaref

► **To cite this version:**

Sofia Vargas Ibarra, Vincent Vigneron, Jean-Philippe Congé, Hichem Maaref. Importance order ranking for texture extraction. A more efficient pooling operator than max pooling?. 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2022), Jul 2022, Lisbon, Portugal. pp.585–594, 10.5220/0011142200003271 . hal-04367045

HAL Id: hal-04367045

<https://hal.science/hal-04367045v1>

Submitted on 29 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Importance order ranking for texture extraction. A more efficient pooling operator than max pooling?

S. Vargas Ibarra¹ ^a and V. Vigneron¹ ^b and J.-Ph. Conge¹ ^c and H. Maaref¹ ^d

¹Univ Evry, Université Paris-Saclay, IBISC EA 4526, Evry, France.

{vincent.vigneron,hichem.maaref,sofia.vargasibarra,jean-philippe.conge}@univ-evry.fr

Keywords: Deep Learning, pooling function, rank aggregation, LBP, segmentation, contour extraction.

Abstract: Much of convolutional neural network (CNN)'s success lies in translation invariance. The other part resides in the fact that thanks to a judicious choice of architecture, the network is able to make decisions taking into account the whole image. This work provides an alternative way to extend the pooling function, we named rank-order pooling, capable of extracting texture descriptors from images. The rank-order pooling layers are non parametric, independent of the geometric arrangement or sizes of the image regions, and can therefore better tolerate rotations. Rank-order pooling functions produce images capable of emphasizing low/high frequencies, contours, etc. We shows rank-order pooling leads to CNN models which can optimally exploit information from their receptive field.

1 Introduction

Convolutional neural network (CNN) architecture is augmented by multi-resolution (*pyramidal*) structures which come from the idea that the network needs to see different levels of (*resolutions*) to produce good results. A CNN stacks four different processing layers: convolution, pooling, ReLU and fully-connected [Goodfellow et al., 2016].

Placed between two convolutional layers, the pooling layer receives several input feature maps. Pooling (i) reduces the number of parameters in the model (*subsampling*) and computations in the network while preserving their important characteristics (ii) improves the efficiency of the network (iii) avoids over-learning.

Thus, the pooling layer makes the network less sensitive to the position of features: the fact that an object is a little higher or lower, or even that it has a slightly different orientation should not cause a radical change in the classification of the image.

The max-pooling function, for example, down-samples the input representation (image, hidden layer output matrix, etc.), reducing its dimensionality.


Weaknesses of pooling functions are well iden-


tified [Yu et al., 2014]: (i) they do not preserve all spatial information (ii) the maximum chosen by the max-pooling in the pixel grid is not the true maximum (iii) average pooling assumes a single mode with a single centroid. The question is how (optimally) to take into account the characteristics of the (input image) regions being pooled into the pooling operation? Part of the answer lies in the work of Lazebnik's who demonstrated the importance of the spatial structure of pooling neighborhoods [Lazebnik et al., 2006]: indeed, local spatial variations of image pixel intensities (called textures in popular image processing) characterize an "organized area phenomenon" [Haralick, 1979] which cannot be captured in pooling layers.


This paper proposes a new pooling operation, independent of the geometric arrangement or sizes of image regions, and can therefore better tolerate rotations. It is based on the Savage definition of rank order [Savage, 1956] and also simple to implement.


Notations

Throughout this paper small Latin letters a, b, \dots represent integers. Small bold letters \mathbf{a}, \mathbf{b} are put for vectors and capital letters A, B for matrices or tensor depending of the context. The dot product between two vectors is denoted $\langle \mathbf{a}, \mathbf{b} \rangle$. We denote by $\|\mathbf{a}\| = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle}$, the ℓ_2 norm of a vector. X_1, \dots, X_n are non ordered variates, x_1, \dots, x_n non ordered observations. "Ordered statistics" means either $p_{(1)} \leq$

^a  <https://orcid.org/0000-0003-3102-4315>

^b  <https://orcid.org/0000-0001-5917-6041>

^c  <https://orcid.org/0000-0002-8641-0312>

^d  <https://orcid.org/0000-0002-1192-7333>

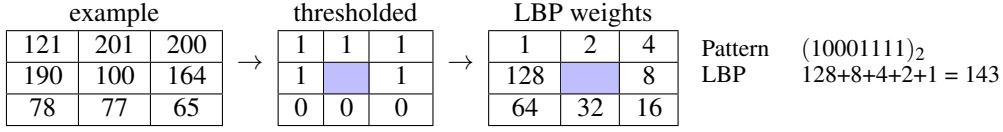


Figure 1: Example of 3×3 image neighborhood ($P = 8$ and $R = 1$).

$\dots \leq p_{(n)}$ (ordered variates) and $p_{(1)} \leq \dots \leq p_{(n)}$ (ordered observations). The extreme order statistics are $p_{(1)} = \min\{x_1, x_2, \dots, x_n\}$, $p_{(n)} = \max\{x_1, x_2, \dots, x_n\}$. The sample range is $p_{(n)} - p_{(1)}$. The $p_{(i)}$ are necessarily *dependent* because of the inequality relations among them.

Definition 1 (Savage [Savage, 1956]). *The rank order corresponding to the n distinct numbers x_1, \dots, x_n is the vector $\mathbf{r} = (r_1, \dots, r_n)^T$ where r_i is the number of x_j 's $\leq x_i$ and $i \neq j$.*

The rank order \mathbf{r} is always unambiguously defined as a *permutation* of the first n integers.

2 Texture coding

Most of image descriptors that encode local patterns *e.g.* local binary patterns (LBP) (and its variants) [Pietikinen et al., 2011, Ojala et al., 1996] depend on (i) the size of the neighborhood (ii) the reading order of the neighbors (iii) and the mathematical function used to calculate the characteristic distance between neighboring pixels. The new pixel value $L_{P,R}$ in the image is an integer in the range of 0 to 255 (for a 8-bit encoding) given by:

$$L_R(P) = \sum_{p=0}^{P-1} 2^p \cdot t(g_p - g_c), \text{ with } t(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where P counts the number of pixels in the neighborhood (not including the central pixel), considering the distance R between central pixel g_c and the neighboring pixels $\{g_p | p = 0, \dots, P-1\}$. In Eq. 1, LBP computes a pixel value from a 8-bit string from the 3×3 neighborhood by computing the Heaviside function $t(\cdot)$ of the difference between the neighboring pixels and the central pixel, $(g_i - g_c)$ (see Fig. 1).

LBP-like texture descriptors have evolved into almost all areas of computer vision, because of their robustness to monotonic gray-scale changes, illumination invariance and computational simplicity. Invariance w.r.t. any monotone transformation of the gray scale is obtained by considering in (1) the signs of the differences $t(g_i - g_c)$, $i = 0, \dots, P-1$. But the *independence* of g_c and $\{|g_0 - g_c|, \dots, |g_{P-1} - g_c|\}$ is not guaranteed in practice. Moreover, under certain circumstances, LBP misses the local structure

as it does not consider the central pixel. The binary data produced by these descriptors are sensitive to noise mainly in uniform regions although [Tan and Triggs, 2007] have proposed a ternary encoding $g \in \{-1, 0, 1\}$ to reduce this noise sensitivity.

In the next section, an algorithm is proposed to generate rank-order importance (ROI) image which could be used in contour detection, segmentation or image quantization.

3 Rank Order statistical model

Let $A = \{a_1, a_2, \dots, a_n\}$ be a set of alternatives, candidates, etc. with cardinality $|A| = n$ and let V be a set of voters, with $|V| = m$.

Each voter/judge k is assumed to have a weak order or ranking $\mathbf{r}^{(k)}$ of the alternatives a_1, a_2, \dots, a_n represented by a vector of integers $\mathbf{r}^{(k)} = (r_1^{(k)}, r_2^{(k)}, \dots, r_n^{(k)})^T$, where $r_1^{(k)}, r_2^{(k)}, \dots$ represent the rank of the alternatives.

The data are collected in a $(n \times m)$ table $R = \{r_i^{(k)}\}$ (see Figure 2.a). R represents either the ranking of the n candidates assigned by the m voters as a *total order*, *i.e.* $r_i^{(k)} \neq r_{i'}^{(k)}, \forall i' \neq i$ [Brüggemann and Patil, 2011], either the ranking of the n candidates in the form such that a voter can give *ex-aequo* positions.

For ease of writing, in the following, $r_{ik} = r_i^{(k)}$.

The rank-aggregation problem consists in finding a *order ranking* \mathbf{r}^* given by a virtual judge minimizing the disagreement of the m judges' opinions, *i.e.*

$$\mathbf{r}^* = \arg \min_{\mathbf{r}} \sum_{k=1}^m d(\mathbf{r}, \mathbf{r}^{(k)}), \quad \text{s.t. } \mathbf{r} \in \mathcal{S}_n, \quad (2)$$

where \mathcal{S}_n is the symmetric group of the $n!$ permutations [Benson, 2016] and the metric $d: \mathcal{S}_n \times \mathcal{S}_n \rightarrow \mathbb{R}^+$ is a distance function chosen *a priori*.

Eq. (2) defines a nonlinear optimization program whose solution \mathbf{r}^* is the distribution of ranks that could have been attributed to these n candidates by a *virtual* voter V resuming the points of view of the m voters [Yadav and Kumar, 2015]. One could also stand the dual problem of the previous one, *i.e.* : is there a distribution of ratings/values that could have been attributed by the m voters to a virtual alternatives a ?

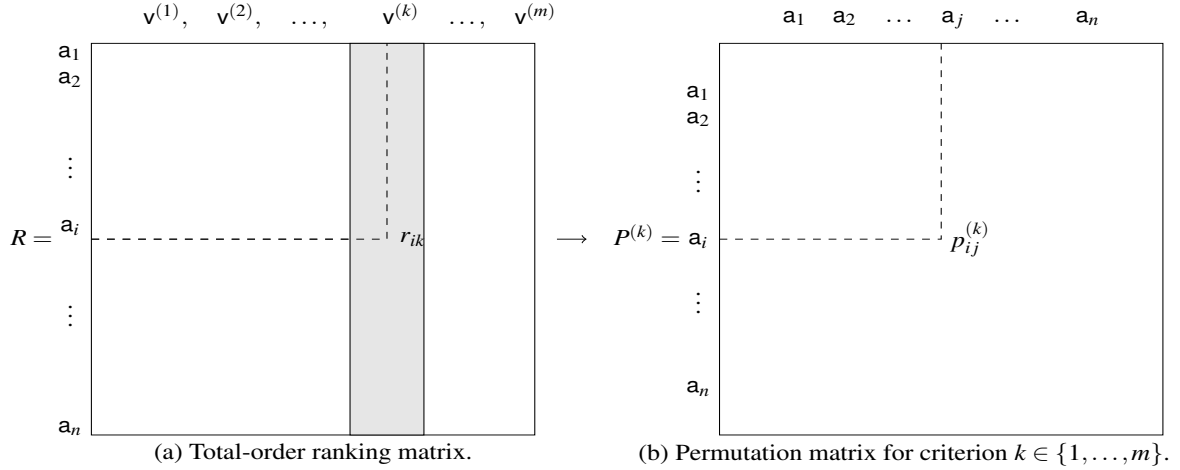


Figure 2: Matrices used with disagreement distance resolution.

The first problem is linked to the idea of aggregating of points of view, the second to the idea of summarizing behaviors.

3.1 Explicit resolution

The optimization problem consists to find a *consensus* distribution of ranks \mathbf{r} given by an unknown virtual voter \mathbf{v} minimizing the discordance of opinions between the m voters, minimizing

$$\sum_{k=1}^m d(\mathbf{r}, \mathbf{r}^{(k)}). \quad (3)$$

The result will be a permutation from S_n . The distance $d(\mathbf{r}^{(k)}, \mathbf{r}^{(k')})$ between the ranking of voter k and the ranking of voter k' can be chosen for instance as the *disagreement* distance $\sum_{i=1}^n \text{sgn} |r_{ik} - r_{ik'}|$, the distance of rank absolute deviation $\sum_{i=1}^n |r_{ik} - r_{ik'}|$, the Euclidean distance between the ranks $\sum_{i=1}^n (r_{ik} - r_{ik'})^2$ or the *Condorcet* distance $\sum_i \sum_j |p_{ij}^{(k)} - p_{ij}^{(k')}|$ (Fig. 2.b), with $p_{ij}^{(k)} = \mathbb{1}_{i < j}$ where $\mathbb{1}$ denotes the indicator matrix for which $p_{ij}^{(k)} = 1$ if the rank of the alternative a_i is *less* than the alternative a_j and 0 otherwise [Gehrlein and Lepelley, 2011]. The choice of these metrics are motivated by a range of properties: (a) they have an intuitive and plausible interpretation as a number of pairwise choices. They provide the best possible description of the process of ranking items as performed by a human (b) their high relevance due to their widespread use (c) they count rather than measure (d) provide very good concordance indicator. [Vigeron and Tomazeli Duarte, 2018] proposed an explicit resolution method with linear programming.

3.2 Euclidean distance (Spearman distance)

When looking for the optimal consensus \mathbf{r}^* of m voters who attributed the votes $\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots, \mathbf{r}^{(m)}$ to the n candidates $\{a_1, a_2, \dots, a_n\}$, we minimize the Euclidean distance defined by $\sum_{k=1}^m \sum_{i=1}^n (r_i^* - r_{ik})^2$, where \mathbf{r}_i denotes the rank of the i th candidate. Note that $\mathbf{r} \in S_n$, with S_n the symmetric group of the $n!$ permutations [Diaconis, 1988]. Hence the constraint $\mathbf{r}^* \in S_n$. The permutation \mathbf{r} can be represented for a voter k by a permutation matrix $P^{(k)} = \{p_{ij}^{(k)}\}_{i,j=1}^n, p_{ij}^{(k)} \in \{0, 1\}$, with $p_{ij}^{(k)} = 1$ if the candidate i is positioned in place j for the k -th voter as pictured in Figure 2.b. Hence we can rewrite the constraint $\mathbf{r}^* \in S_n$ as

$$\sum_{j=1}^n p_{ij} = \sum_{i=1}^n p_{ij} = 1, \forall i, j \quad (4)$$

Example 1 (Condorcet rank-order coding). For instance, the 4th column of matrix R below becomes matrix $P^{(4)}$:

$$R = \begin{bmatrix} 5 & & & & & \\ 3 & & & & & \\ 2 & & & & & \\ 4 & & & & & \\ 1 & & & & & \end{bmatrix} \rightarrow P^{(4)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We will show that replacing \mathbf{r} with the matrix P leads to an optimization function linear in p_{ij} which can be set as follows:

$$\min_{\mathbf{r}} \left(\sum_{ij} \sum_k c_{ij}^{(k)} p_{ij} \right), \text{ s.t. } \sum_{j=1}^n p_{ij} = \sum_{i=1}^n p_{ij} = 1, \forall i, j, \quad (5)$$

where $\sum_k c_{ij}^{(k)}$ corresponds to the affectation cost of the candidate i in place j for all the voters; a cost which

depends of the metric and that is calculated in the following.

As $r_i^* = \sum_{j=1}^n j p_{ij}$, the Euclidean function becomes

$$\sum_{k=1}^m \sum_{i=1}^n \left(\sum_{j=1}^n j p_{ij} - r_{ik} \right)^2 = \sum_{k=1}^m \sum_{i=1}^n \left(\sum_{j=1}^n (j - r_{ik})^2 p_{ij} \right). \quad (6)$$

Hence the minimization problem is equivalent to:

$$\min_P \left(\sum_{i=1}^n \sum_{j=1}^n \phi_{ij} p_{ij} \right) \quad \text{s.t.} \quad \phi_{ij} = \sum_{k=1}^m (j - r_{ik})^2, \quad (7)$$

$$\sum_{i=1}^n p_{ij} = \sum_{j=1}^n p_{ij} = 1, \text{ and } p_{ij} \in \{0, 1\}, \quad (8)$$

where ϕ_{ij} is the cost of attribution alternative \mathbf{a}_i in position j .

If the \mathbf{r}_k are total orders in Eq. (7), Eq. 6 can be simplified in:

$$\sum_{k=1}^m \sum_{i=1}^n (r_i^2 - 2r_i r_{ik} + r_{ik}^2) = \underbrace{\sum_{k=1}^m \sum_{i=1}^n (r_i^2 + r_{ik}^2)}_{2p \times \text{sum of } n \text{ first integers}} - \sum_{i=1}^n (r_i \sum_{k=1}^m r_{ik}). \quad (9)$$

As $\sum_{j=1}^n r_{ik} = p r_i$, where r_i is the mean rank of the alternative i on the m rankings and as $r_i = \sum_{j=1}^n j p_{ij}$, Eq. (9) becomes

$$2p \frac{n(n+1)(2n+1)}{6} - 2p \sum_{i=1}^n \sum_{j=1}^n j r_i p_{ij}. \quad (10)$$

Minimize rank Spearman distance remains to maximize the term $\sum_{i=1}^n \sum_{j=1}^n j r_i p_{ij}$ under the usual constraints of $P^{(k)}$ being a permutation matrix. Finding the optimal permutation p_{ij}^* consists to affect to the alternative \mathbf{a}_i to the rank j the nearest as possible to r_i in the Euclidean sense.

3.3 Rank absolute deviation distance

The problem of finding the virtual voter in the case of the rank absolute deviation distance $\sum_{i=1}^n |r_{ik} - r_{ik'}|$ can be posed as the optimization problem

$$\min_{\mathbf{r}^* \in \mathcal{S}_n} \left(\sum_{k=1}^m \sum_{i=1}^n |r_i^* - r_{ik}| \right), \quad (11)$$

with the notations $r_i^* = \sum_{j=1}^n j p_{ij}$, under the constraints $\sum_{j=1}^n p_{ij} = 1$, $p_{ij} \in \{0, 1\}$. The absolute value in Eq. 11 can be rewritten as

$$\left| \sum_{j=1}^n j p_{ij} - r_{ik} \right| = \left| \sum_{j=1}^n (j - r_{ik}) p_{ij} \right| = \sum_{j=1}^n |j - r_{ik}| p_{ij}. \quad (12)$$

If we note $\phi_{ij} = \sum_{k=1}^m |j - r_{ik}|$ the attribution cost of alternative \mathbf{a}_i in position j , the minimization of the consensus function for $\mathbf{r}^* \in \mathcal{S}_n$ consists to solve the following linear program

$$\min_P \left(\sum_{i=1}^n \sum_{j=1}^n \phi_{ij} p_{ij} \right) \quad \text{s.t.} \quad \phi_{ij} = \sum_{k=1}^m |j - r_{ik}|, \quad (13)$$

$$\sum_{i=1}^n p_{ij} = \sum_{j=1}^n p_{ij} = 1, \text{ and } p_{ij} \in \{0, 1\}. \quad (14)$$

4 Rank-order importance components

4.1 Importance ranking

Several strategies have been proposed in the literature to extract important variables or develop parsimonious models and deal with the dimensionality. The dimension of observed data being generally higher than their intrinsic dimension, it is theoretically possible to reduce the dimension without losing information.

Among the unsupervised tools, principal component analysis (PCA) or factor analysis (FA) are certainly the most used techniques to optimize the understanding insight into of a data set. They aim to project the data onto a lower dimensional subspace in which axes are constructed either by maximizing the variance of the projected data or by explaining the overall covariance structure.

PCA and FA are both linear tools. This means that nonlinear dependencies are not taken into account.

The question is simply: can we extract a set of the most decorrelated rank-order variables to each other capable of capturing distinct information? The overall framework for this objective suggests a rank-order decomposition. The principle remains remarkably simple: it consists into a re-distributive effect of the rank variables – similar to PCA – on a Hilbert space by linear programming.

Lemma 1 (Vigneron and Duarte [Vigneron and Duarte, 2017]). *Consider a collection of rank-orders $R = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m\}$ (data). It is always possible to extract a total rank-order component \mathbf{g}_ℓ with a minimal distance to the data $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m\}$ and simultaneously with the maximal distance to the collection of previously calculated ranks $\{\mathbf{g}_1, \dots, \mathbf{g}_{\ell-1}\}$.*

At stage ℓ , the search of the ℓ th total order \mathbf{g}_ℓ is represented by the $n \times \ell$ matrix $Z^{(\ell)} = \{\mathbf{z}_1, \dots, \mathbf{z}_\ell\}$. In the case of the rank absolute deviation distance (developed in section 3.3) the rank-order decomposition

reduced to solve the following linear program

$$\begin{aligned} & \max_{Z^{(\ell)}} \left(\sum_{i=1}^n \sum_{j=1}^n \phi_{ij} z_{ij}^{(\ell)} - \sum_{i=1}^n \sum_{j=1}^n \beta_{ij} z_{ij}^{(\ell)} \right) \\ \text{s.t. } & z_{ii} = 0, z_{ij}^{(\ell)} + z_{ji}^{(\ell)} - z_{ik}^{(\ell)} \leq 1, i \neq j \neq k, \\ & z_{ij}^{(\ell)} \in \{0, 1\}. \end{aligned} \quad (15)$$

where

$$\phi_{ij} = \sum_{j=1}^n |j - r_{ik}|, \quad (16)$$

and

$$\beta_{ij} = \sum_{k=1}^{\ell-1} z_{ij}^{(k)}, z_{ij}^{(\ell)} + z_{ji}^{(\ell)} = 1, i < j,$$

Algorithm 1 stops when $\ell = m$ and provide $\{\mathbf{g}_1, \dots, \mathbf{g}_m\}$ rank-order vectors such that \mathbf{g}_ℓ is the most decorrelated to the previous ranks $\{\mathbf{g}_1, \dots, \mathbf{g}_{\ell-1}\}$. Until now, on the contrary to PCA, there is no index capable of indicating the quantity of information captured by each vector \mathbf{g}_ℓ .

In the algorithm 1, **LP** is put for the optimal resolution of the linear program Eq. 15.

Algorithm 1 Rank-order decomposition Algorithm.

Require: $P^{(1)}, \dots, P^{(m)} \leftarrow \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m\}$
 $\{\text{Permutation matrices } P^{(k)} = \{p_{ij}^{(k)}\}\} \vee \text{stack}$
 $A = \emptyset \{\text{contain the re-ranked components}\}$
Ensure: $\{\mathbf{g}_1, \dots, \mathbf{g}_m\} \{\text{Postcondition}\}$
1: **for** $\ell = 1$ **to** m **do**
2: Compute $\phi_{ij} = \sum_{j=1}^n |j - r_{ik}|$,
3: Compute $\beta_{ij} = \sum_{k=1}^{\ell-1} z_{ij}^{(k)}$
4: $\Psi = \{\phi_{ij}\}, B = \{\beta_{ij}\}$
5: **LP**($\Psi, B, Z^{(\ell)}$) under constraints (15) {solve linear program}
6: $\mathbf{g}_\ell \leftarrow \mathbf{z}_\ell$
7: **end for**
8: **return** $\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m\}$

4.2 Experimental setup

Example 2 (Top 10 french companies CAC 40). *The composition of the CAC 40 index is based on the ranking on the top 100 companies. By way of illustration, Table 1 shows the data matrix $R = \{r_i^{(k)}\}$ formed with the ranking often company names are listed and ranked according to 4 criteria: turnover induced by innovation (TII), size (SIZ), level of capitalization on the knowledge transferred (CAP), impact of R& D collaborations (IMP) [Vigner and Petit, 2008].*

Company	BOURSORAMA ranks			
	turnover	size	capital	impact
TOTAL	1	4	3	10
LOREAL	2	2	1	2
SANOFI	3	1	2	1
LVMH	4	6	4	3
BNP PARIBAS	5	5	7	5
DANONE	6	3	6	4
AXA	7	8	5	6
VINCI	8	7	8	9
AIRBUS	9	10	10	8
ORANGE	10	9	9	7

Table 1: Ten CAC 40 company rankings according to 4 criteria: turnover induced by innovation (TII), size (SIZ), level of capitalization on the knowledge transferred (CAP), impact of R& D collaborations (IMP).

Table 3 shows two full order ranking consensus, proposed by a virtual judge, minimising the disagreements between the voters (criteria).

Company	aggregated ranks		
	Spearman (Eq. 7)	rank abs. dev. (Eq. 13)	
TOTAL	3		3
LOREAL	2		2
SANOFI	1		1
LVMH	4		4
BNP PARIBAS	5		6
DANONE	6		5
AXA	7		7
VINCI	8		8
AIRBUS	10		10
ORANGE	9		9

Table 2: Left: Ten CAC 40 company rankings according to 4 criteria: innovation induced turnover, size, level of capitalization on transferred knowledge, impact of R& D collaborations. Right: The two proposed ranking using disagreement and Condorcet distances.

For instance, in the case of d_D , the companies are ranked this way: ① SANOFI ② LOREAL ③ TOTAL ④ LVMH ⑤ BNP PARIBAS ⑥ DANONE ⑦ AXA ⑧ VINCI ⑨ ORANGE ⑩ AIRBUS.

The 2 rankings are concordant except for items 5 and 6. Applying Algorithm 1 to table 1 gives the rank decomposition proposed in table 3 in which the first column is more important or explanatory than the second one, itself more explanatory than the third, etc.

Company	Rank-order decomposition			
	comp 1	comp 2	comp 3	comp 4
TOTAL	4	5	6	5
LOREAL	1	3	1	4
SANOFI	2	2	2	2
LVMH	3	6	3	7
BNP PARIBAS	6	1	10	1
DANONE	5	4	7	3
AXA	7	7	5	9
VINCI	8	8	8	8
AIRBUS	10	9	9	6
ORANGE	9	10	4	10

Table 3: Rank order of the 4 first principal components of ten CAC-40 companies obtained using algorithm 1.

Proof of orthogonality can be checked with pairwise Kendall's rank correlation coefficients between the columns of table 3:

$$K^{(2)} = \begin{bmatrix} 1.0 & 0.5556 & 0.5556 & 0.3333 \\ & 1.0 & 0.1111 & 0.4478 \\ & & 1.0 & -0.1111 \\ & & & 1.0 \end{bmatrix}$$

all lower than the Kendall's coefficients¹ of the rank matrix in table 1:

$$K^{(1)} = \begin{bmatrix} 1.0 & 0.6000 & 0.7333 & 0.3778 \\ & 1.0 & 0.6889 & 0.5111 \\ & & 1.0 & 0.4667 \\ & & & 1.0 \end{bmatrix}$$

The Kendall correlations τ_{ij} between two columns i and j will be high when observations have a similar (or identical for a correlation of 1) rank (i.e. relative position label of the observations within the variable: 1st, 2nd, 3rd, etc.) between the two variables, and low when observations have a dissimilar (or fully different for a correlation of -1) rank between the two variables.

So the CAC data were indeed reordered to be the more uncorrelated as possible, as expected.

Example 3 (Application of ROI to textured image). Consider now the neighborhood of a pixel 'p' in a image I , i.e. the set of pixels that touch it. The neighborhood of a pixel can have a maximum of 8 pixels as shown in Fig. 3a. The colored pixels in Fig. 3b are 8-connected to p .

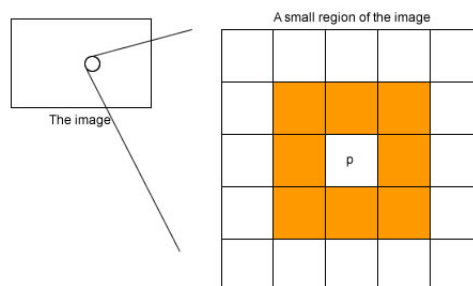
For instance, the 4×4 image I in Fig. 4a. can be decomposed using the 8-connectivity into the 16×8 matrix $R = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_8\}$ (Fig. 4b where \mathbf{r}_1 denotes the column of pixel 1 (clock-wise ordering), \mathbf{r}_2 denotes the column of pixel 2, and so on. The same principle as for LBP is mimic, but with the transformation given in Eq. (15). The 8 neighbors around the central pixel can be seen as "voters" from whom we expect a total rank-order.

¹The Kendall's coefficient is a statistic used to measure the ordinal association between two measured quantities. As the correlation coefficient value goes towards 0, the relationship between the two variables will be weaker.

LBP weights

1	2	3
8		4
7	6	5

(a) Neighbor pixels ordering.



(b) The orange pixels form the neighborhood of the pixel 'p'.

Figure 3

	1	2	3	4
1	4	8	20	18
2	17	12	17	17
3	6	1	17	17
4	4	6	5	14

(a) 4×4 image I .

center pixel	\mathbf{r}_0	\mathbf{r}_1	\mathbf{r}_2	\mathbf{r}_3	\mathbf{r}_4	\mathbf{r}_5	\mathbf{r}_6	\mathbf{r}_7
12	4	8	20	17	17	1	6	17
17	8	20	18	17	17	17	1	6
1	17	12	17	17	5	6	4	6
17	12	17	17	17	14	5	6	1

(b) 8-connectivity matrix R .

center pixel	\mathbf{r}_0	\mathbf{r}_1	\mathbf{r}_2	\mathbf{r}_3	\mathbf{r}_4	\mathbf{r}_5	\mathbf{r}_6	\mathbf{r}_7
12	1	1	4	1	4	1	3	4
17	1	4	3	2	3	4	1	3
1	4	2	2	3	1	3	2	2
17	3	3	1	4	2	2	4	1

(c) Rank matrix R .

Figure 4

The 8-connectivity matrix R is transformed into ranks by simple ordering on which we can apply algorithm 1. Concerning on peripheral pixels, the spatial area is enlarged by adding borders of zeros. In general, if our image is of size $n \times n$, and we examine a neighborhood of $f \times f$, then the size of the resulting output is $(n - f + 1) \times (n - f + 1)$. With $n = 4$, $f = 3$,

indeed, this gives us a 2×2 output channel. For a $n \times n$ image, the encoding values are between 1 and n^2 . The generalization to larger images is straightforward.

a question that might arise is how to choose the ranks for the tied values? it was shown in [Vigneron and Tomazeli Duarte, 2018] that the encoding is insensitive to ties.

As an illustration LBP is applied to Lena's original picture 5a and provides the texture representation given in Figure 5b. From the figure 5a, we obtained a new decomposition (Figures 5c-5f). Visually, the 1st plot is more informative than the 2nd one, which itself is more informative than the 3rd, and so on.

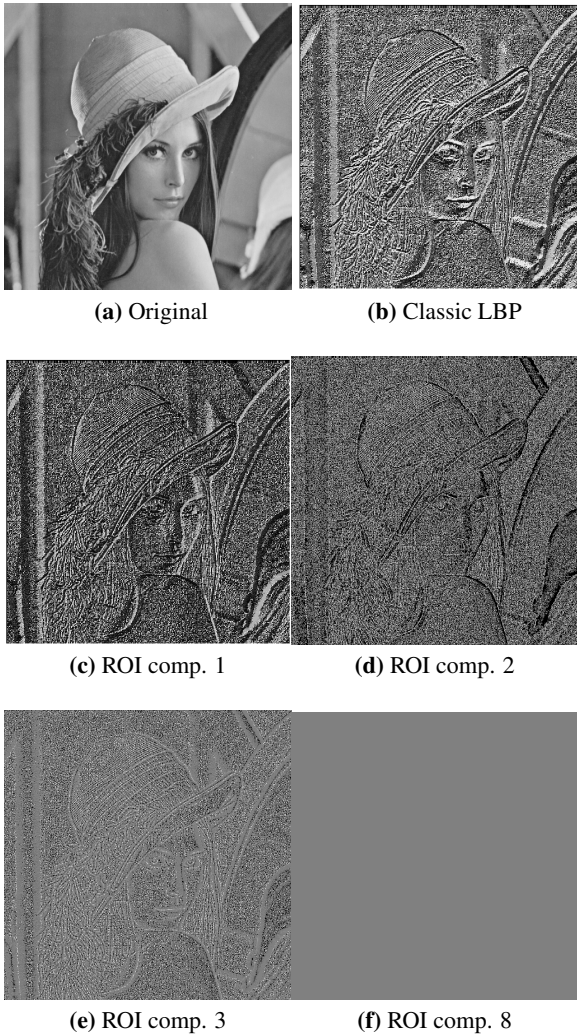


Figure 5: Lena's original (a) is compared to classic LBP representation (b) and with the 1st, 2nd, 3rd and eighth rank-order components obtained from Algorithm 1. The eighth component is apparently the less informative component.

5 ROI-pooling operator

In CNNs maximum pooling operator is defined mathematically for a volume $\mathcal{V} : A \rightarrow \mathbb{R}$, $A = \{(i_1, i_2, \dots, i_N) | i_k \in [1, \dots, n_k], k \in [1, \dots, N]\}$ and a set $B = \{(i_1, i_2, \dots, i_{N-1}, 0) | i_b \in [-K_b, \dots, K_b], b \in [1, \dots, N-1]\}$ called window where either $K_b = K^b$ or $K_b = K^b - 1$ with $K^b \in \mathbb{N}$ as follows:

$$\max \text{pool} = \max_{\mathbf{y} \in B} f(\mathbf{x} - \mathbf{y}). \quad (17)$$

The operation in Eq. 17 looks for the maximum in a neighborhood given by B along the image axis. Unlike the convolution operator, the pixel values in this neighborhood are not combined. Often the maximum pooling operation is used for downsampling the volume by restricting x (striding) with stride $s \in \mathbb{N}$ and A is restricted to $A' = \{(i_1, i_2, \dots, i_N) | i_k \in [1, 1 + s, 1 + 2s, \dots, 1 + n_s s], n_s = \lceil n_k / s \rceil - 1, k \in [1 \dots N]\}$ with the ceiling function $\lceil \cdot \rceil$. The strided maximum pooling operation is then:

$$\mathbf{x}' = 1 + s\mathbf{x}, \quad \max \text{pool} = \max_{\mathbf{y} \in B} f(\mathbf{x}' - \mathbf{y}) \quad (18)$$

where B is simply a binary mask. The strided maximum pooling reduces the size of the input image by only considering every s -th entry along all image axes and discarding all others. ROI-pooling can easily replace maximum pooling in CNN:

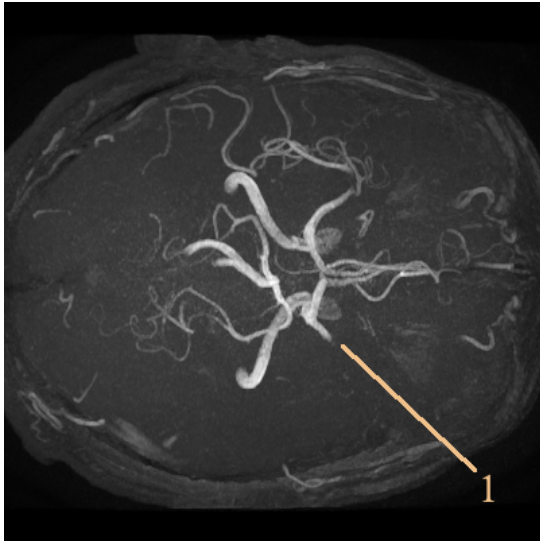
$$\text{ROI pooling}(f, B) = \text{ROI}f(\mathbf{x}_y). \quad (19)$$

In Eq. 19, \mathbf{y} is put for the neighborhood in which \mathbf{x} is selected. As maximum pooling, ROI pooling is parameter free.

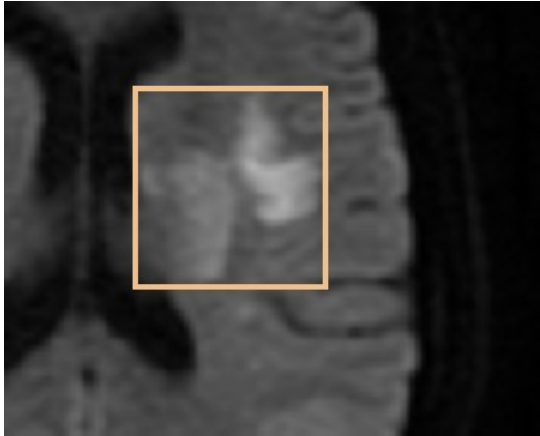
To answer to the question if could ROI pooling perform as well as max-pooling in 19, different grouping operations were performed in a categorization context.

Example 4 (ROI pooling for automatic lesion segmentation of stroke patient). *Lack of expertise or time for interpreting a brain magnetic resonance imaging (MRI) in the case of stroke increases the risk of death and disability. This study aims to enable rapid and accurate assessment of the damage caused by hyperacute ischemic stroke ($< 4.5h$) by quantifying the volume of ischemia. By reducing the variability of interpretation, it enables a more standardized stroke diagnosis, and facilitates rapid and consistent treatment decisions by health professionals regardless of their experience or expertise.*

MRI provides information about damage to the brain. Combined with medical knowledge on blood flow in the human brain, MRI makes it possible to identify the thrombus causing the problems and to decide on the treatment of endovascular recanalization (single or double thrombolysis, thrombectomy).



(a) Maximum intensity projection of a patient with middle carotid artery (MCA) occlusion.



(b) Intensity of injury comparable to that of normal tissue.

Figure 6: Development of a lesion visualized on diffusion imaging (DWI). (a) The artery stops abruptly at the point of occlusion (1) (b) Most of the publications deal only with well developed lesions which take advantage of the high intensity boundaries. In the hyperacute phase, these borders are weak or zero and weak intensities complicate the task of segmentation.

The stroke dataset was provided with the support of the neurology group of the Center Hospitalier Sud Francilien (see [Kobold et al., 2019] for a full description of the dataset). It contains the cranial MRI of 65 stroke patients. The MRI modalities available for this data set are DWI B0 ADC FLAIR ToF and the corresponding phase modalities. 61 of the patients show a lesion on DWI, but some exams lack the phase image.

Thus there are only 58 patients with a visible le-

sion on DWI where all terms are available. The 65 patients all have a visible lesion. MRIs were acquired from a 1.5 T and 3 T General Electrics MRI machine. DWI B0 and ADC share the same resolution, as they are from the same acquisition.

Manual lesion segmentations are available. All lesions were segmented by at least two neurologists. Inter-observer agreement, measured in terms of Dice's coefficient, was 0.69 ± 0.15 . The median lesion size in the data set is approximately 2,920 voxels: the MRI is taken in the hyperacute phase of the stroke, which means that the lesion is growing, may be small and does not have a well-defined border. This makes it a more difficult task than the cases that have already been studied in the literature.

Data augmentation is used because the training set is too small to generate a model that generalizes well. Random crops linked to plot sampling works as follows: the plots are sampled at a size larger than the intended size for training. For example, the model is trained on 64×64 then the patches are sampled at 74×74 . Then, during training, a 64×64 image is cut from the 74×74 patch in a random location, but in such a way that the 64×64 image is entirely contained in the 74×74 image. The number of possible random crops for a patch is determined by the patch size difference and the size of the training image. Once data is co-registered and normalized, automatic lesion segmentation is performed.

The most common evaluation metric for evaluating biomedical image segmentations is Dice's coefficient which measures the overlap of two segmentations but also takes into account their cardinality. This dependence on cardinality makes it difficult to tell which value of Dice's coefficient indicates a good segmentation result. A perfect segmentation is identical to the ground truth and gives a Dice's coefficient of 1. If there is no overlap, then the dice is 0.

The first method tried is 2d U-Net, trained on 64×64 patches using enhanced lesion image with DWI (see Fig. 7). 3d U-net, CNN and improved U-net with ROI were tested.

The segmentation results of the four networks are given in the table 4. The 3D U-Net again suffers from the small training set whereas the U-Net generally learns to detect the lesion but fails in some cases.

The fewer false positive number is found with U-Net with Spearman ROI. Upon manual inspection, it turns out that these are very close to the main lesion and may even be part of the lesion, depending on the definition. It is therefore an almost perfect result. Since the false positives were an insignificant amount of voxels anyway, the Dice coefficient only changes at the fourth digit after the decimal indicate. This result

Model	Dice	FP	FP Size	FN	FN Size	Dr
U-Net	0.65	3.3	29.6	0.39	239.5	0.93
3D U-Net	0.56	229.7	215	1.48	643.5	1.0
CNN	0.53	208.3	83.8	0.25	263.9	0.89
U-Net+Spearman ROI	0.8	133.6	94.0	0.14	353.8	0.96
U-Net+ absolute rank ROI	0.76	29.8	76.3	0.25	121.1	1.0

Table 4: Segmentation results for the thrombus. The columns are the Dice coefficient, the mean number of false positive objects FP and the average "FP Size", the mean number of false negative objects FN and their average size "FN Size" and the detection rate Dr.

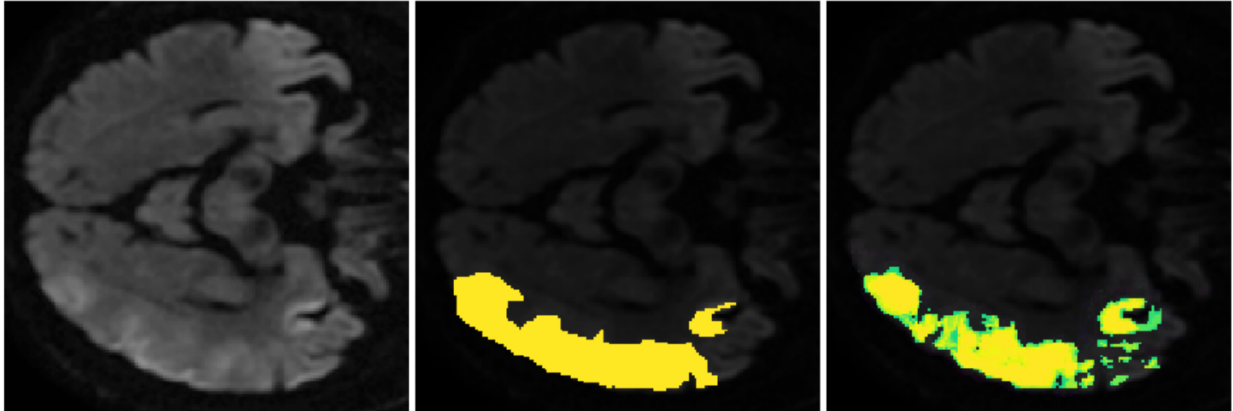


Figure 7: An example of lesion development in the hyperacute phase of stroke (left). The lesion is currently growing and has no clearly defined borders but instead shows gradients towards normal tissue. The image in the center shows the ground truth and the one on the right the probability of injury given by the model. Please note that our algorithm correctly identified the contralateral artifact.

is therefore a segmentation of the lesion at the human level. It is in particular the first lesion segmentation that achieves this performance and a better result cannot be achieved on this type of database. Figure 7 shows the gradual contours of the developing lesion and contralateral artifacts that only trained experts can identify as such.

6 Conclusion

The rank-order pooling layers are non parametric, independent of the geometric arrangement or sizes of the image regions, and can therefore better tolerate rotations. An other asset of the rank-order pooling lies in the number of rank-order components g_e that the algorithm 1 can generate, which are uncorrelated from each other and which guarantees optimal (independent) feature extraction performance.

When should we pool and when should we not? The answer depends upon the following considerations, in descending order of importance: (i) if there is an inadequate number of observations in each of two (or more) subgroups, which would usually necessitate pooling (ii) common sense, necessity, etc.

In statistics, "pooling" means gathering together small sets of data that are assumed to have the same value of a characteristic. ROI pooling is transforming convolution features into a new representation that preserves important information while ignoring irrelevant details.

REFERENCES

- Benson, D. (2016). *Representations of Elementary Abelian p -Groups and Vector Bundles*. Cambridge tracts in mathematics. Cambridge University Press, 1 edition.
- Brüggemann, R. and Patil, G. (2011). *Ranking and Prioritization for Multi-indicator Systems: Introduction to Partial Order Applications*. Environmental and Ecological Statistics. Springer-Verlag New York.
- Diaconis, P. (1988). *Group Representation in Probability and Statistics*, volume 11 of *IMS Lecture Series*. Institute of Mathematical Statistics, Harvard, USA.
- Gehrlein, W. and Lepelley, D. (2011). *Voting Paradoxes and Group Coherence: The Condorcet Efficiency of Voting Rules*. Studies in Choice and Welfare. Springer-Verlag Berlin Heidelberg, 1 edition.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*, volume 1. MIT press Cambridge.
- Haralick, R. (1979). Statistical and structural approaches to

- texture. In *Proceedings of the IEEE*, volume 67, pages 786–804.
- Kobold, J., V., V., Maaref, H., Fourer, D., Aghasaryan, M., Alecu, C., Chausson, N., L'hermitte, Y., Smadja, D., and Läng, E. (2019). Stroke Thrombus Segmentation on SWAN with Multi-Directional U-Nets. In *9th IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA 2019)*, Proc. of the 9th IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA 2019), Istanbul, Turkey.
- Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169 – 2178.
- Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59.
- Pietikinen, M., Hadid, A., Zhao, G., and Ahonen, T. (2011). *Computer Vision Using Local Binary Patterns*, volume 40 of *Computer imaging and vision*. Springer.
- Savage, R. (1956). Contributions to the theory of rank-order statistics – the trend case. *The Annals of Mathematical Statistics*, 27(3):590–615.
- Tan, X. and Triggs, B. (2007). Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *Analysis and Modeling of Faces and Gestures*, volume 4778 of *Lecture Notes in Computer Science*, pages 235–249, Berlin. Springer.
- Vigñeron, V. and Duarte, L. (2017). Toward rank disaggregation: An approach based on linear programming and latent variable analysis. In Tichavský, P., Babaie-Zadeh, M., Michel, O. J., and Thirion-Moreau, N., editors, *Latent Variable Analysis and Signal Separation*, pages 192–200, Cham. Springer International Publishing.
- Vigñeron, V. and Petit, E. (2008). The evaluation of the impact of the technology transfers from public laboratories to private firms : the case of the french nuclear authority. *Fuzzy Economic Review*, 13(1).
- Vigñeron, V. and Tomazeli Duarte, L. (2018). Rank-order principal components. A separation algorithm for ordinal data exploration. In *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pages 1–6.
- Yadav, N.. author. abd Yadav, A. and Kumar, M. (2015). *An Introduction to Neural Network Methods for Differential Equations*. Springer, Gurgaon, Haryana, India.
- Yu, D., Wang, H., Chen, P., and Wei, Z. (2014). Mixed pooling for convolutional neural networks. In *International Conference on Rough Sets and Knowledge Technology*, pages 364–375.