Consistent K-fold cross validation:

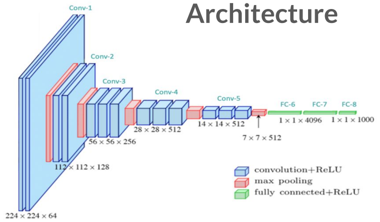
Application to image/video object detection datasets

C. Barrelet, M. Chaumont, G. Subsol, V. Creuze, and M. Gouttefarde

What's a K-fold?

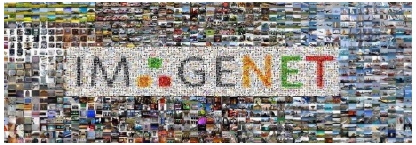
→

Architecture



$\oplus$

Database



Train | Test

Partition

Model

→

- Dog: 60%
- Cat: 40%

Model	Folds:		Train	Test		θ
M_1						$\theta(M_1)$
M_2						$\theta(M_2)$
M_3						$\theta(M_3)$
M_4						$\theta(M_4)$
M_5						$\theta(M_5)$

Architecture performance:

$$\theta(\mathcal{M}) \approx \bar{\theta}_M = \sum_{i=1}^{K} \theta(M_i)/K$$

Performance sensitivity:

$$\sigma^2(\mathcal{M}) \approx \sigma_M^2 = \sum_{i=1}^{K} (\theta(M_i) - \bar{\theta}_M)^2/K$$

Ron Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proceedings of the 14th International,Joint Conference on Artificial Intelligence - Volume 2, San Francisco, CA, USA, 1995, IJCAI'95, p. 1137–1143, Morgan Kaufmann Publishers Inc

Why a K-fold?

- Not necessary for “large” reference datasets (ImageNet, COCO, etc.)
  - Law of large numbers
- Necessary for “smaller” datasets
  - Law of large numbers cannot hold anymore
- Easier architectures comparison

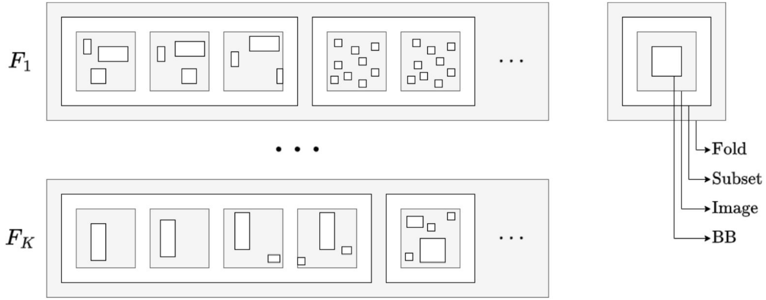
How to K-fold?

Randomly partition all the images into *K* folds

But, what if the dataset can be subdivided into subsets of images sharing similar features?

It's not fair...

What does it mean for object detection datasets?



Internal variability you say?

K matters!

- Moderate *K values* (10-20) $\rightarrow \sigma^2_M \searrow$
- Low *K values* (2-5) $\rightarrow \sigma^2_M \nearrow$

Size too!

- Dataset size N$\nearrow$ $\rightarrow \sigma^2_M \searrow$
- Better use *K=5* or *K=10* than *K=N*

There is a performance-variance tradeoff…

- Estimator must maximize the performance and minimize the variance

Random induces internal variability.

Maybe smooth the results with J-K-folds?

- How to choose J and K?
- Dataset more heterogeneous $\rightarrow$ More repetitions…
- More complexity…

What about saving complexity by ensuring internal consistency ?

Ron Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in Proceedings of the 14th International,Joint Conference on Artificial Intelligence - Volume 2, San Francisco, CA, USA, 1995, IJCAI’95, p. 1137–1143, Morgan Kaufmann Publishers Inc

Juan D. Rodriguez, Aritz Perez, and Jose A. Lozano, “Sensitivity analysis of k-fold cross validation in prediction error estimation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 3, pp.569–575, 2010

Ron Kohavi and David H. Wolpert, “Bias plus variance decomposition for zero-one loss functions,” in International Conference on Machine Learning, 1996

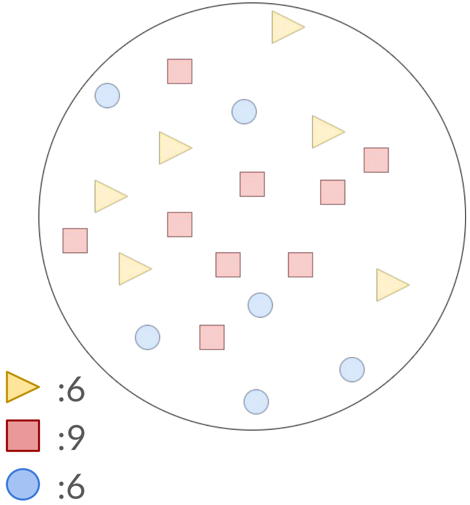
Gaoxia Jiang and Wenjian Wang, “Error estimation based on variance analysis of k-fold cross-validation,” Pattern Recognition, vol. 69, pp. 94–106, 2017

Tzu-Tsung Wong and Po-Yang Yeh, “Reliable accuracy estimates from k-fold cross validation,” IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 8, pp. 1586–1594, 2020.

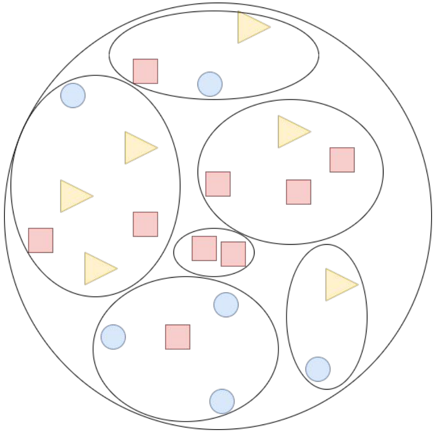
Henry Moss, David Leslie, and Paul Rayson, “Using J-K-fold cross validation to reduce variance when tuning NLP models,” in Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, Aug. 2018, pp. 2978–2989, Association for Computational Linguistics.

Let's ensure consistency!

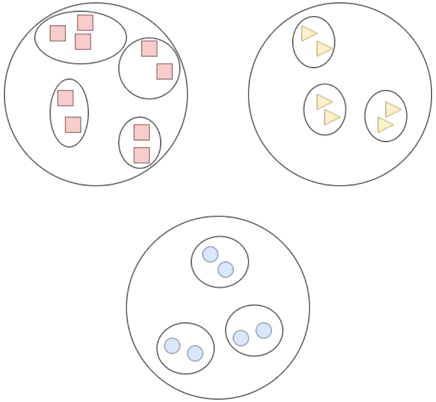
Stratified-**KCV**



Distributed-**KCV**



Distributed-Stratified-**KCV**



Predetermined and hand-crafted features...

Classification task only

L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen, Classification and Regression Trees, Taylor & Francis, 1984

N.A. Diamantidis, D. Karlis, and E.A. Giakoumakis, "Unsupervised stratification of cross-validation for accuracy estimation," Artificial Intelligence, vol. 116, no. 1, pp. 1–16, 2000

Xinchuan Zeng and Tony Martinez, "Distribution-balanced stratified cross-validation for accuracy estimation," Journal of Experimental & Theoretical Artificial Intelligence, vol. 12, 04 2001

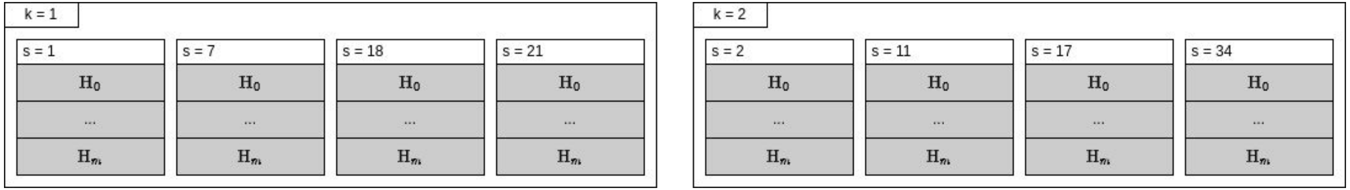
A general method to build consistent KCV

Definition of descriptors at BB and images levels

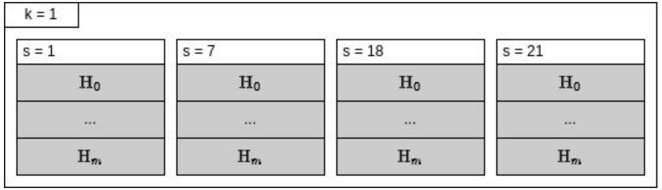
- **BB-level**:
  - a class
  - features describing geometrical and intensity properties of the **thumbnail**
- **Image-level**:
  - features describing geometrical and intensity properties of the **image**

An histogram-based framework

- Consider an image dataset of N subsets identified by the index *s* ∈ {1, ..., N}
- Assume each subset *s* characterized by a set of *m* histograms denoted $\mathbf{H}_h^s$ with *h* ∈ {0, ..., m}



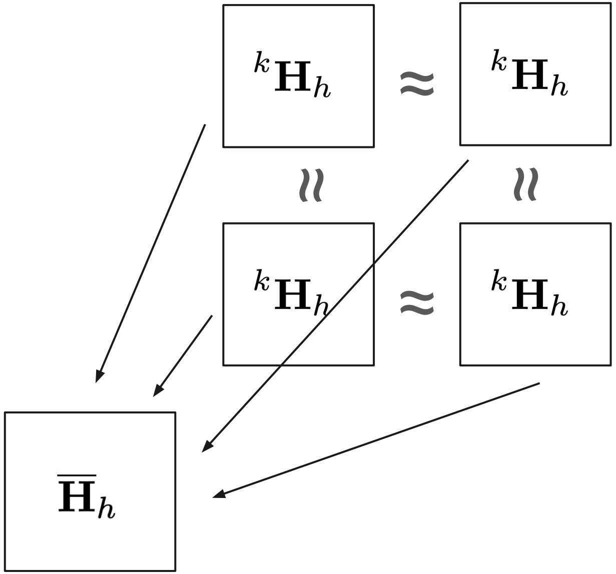
A general method to build consistent KCV



The average of the hth histogram for the fold k $\longrightarrow$ $${}^{k}\mathbf{H}_h = \frac{1}{|{}^{k}\mathcal{S}|} \sum_{s \in {}^{k}\mathcal{S}} \mathbf{H}_h^s$$

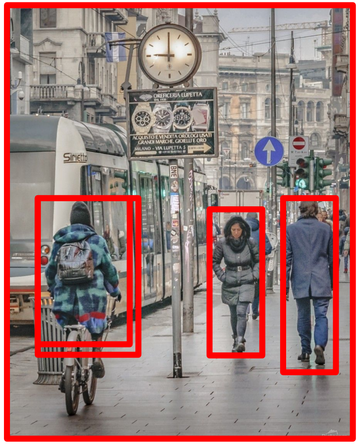
The desired list of histograms $\longrightarrow$ $$\overline{\mathbf{H}}_h = \frac{1}{N} \sum_{s=1}^{N} \mathbf{H}_h^s$$

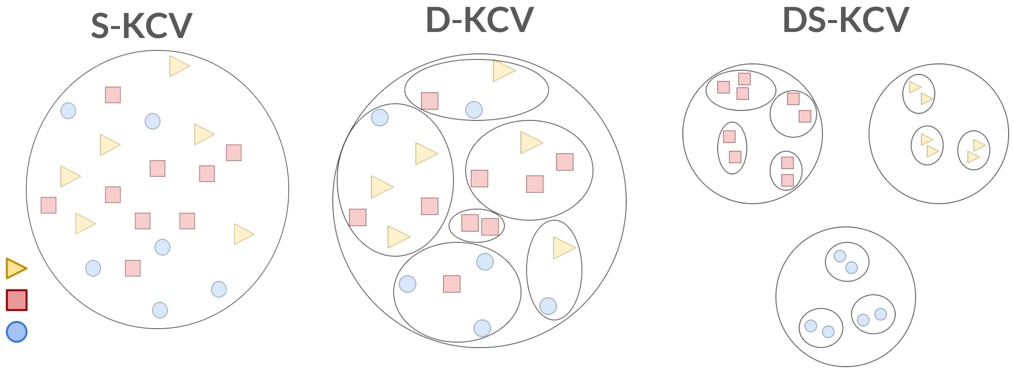
$$\mathbf{a}^* = \underset{\mathbf{a} \in \{1,\ldots,K\}^N}{\operatorname{argmin}} \left(\sum_{k=1}^{K} \sum_{h=0}^{m} D\left({}^{k}\mathbf{H}_h, \overline{\mathbf{H}}_h\right) \right)$$



A general method to build consistent KCV

3 different descriptors:





$$\mathbf{a}^* = \underset{\mathbf{a}\in\{1,\dots,K\}^N}{\operatorname{argmin}} \left(\sum_{k=1}^{K} \alpha_0 \, \mathrm{D}\left({}^k\mathbf{H}_0, \overline{\mathbf{H}}_0\right) + \alpha_1 \, \mathrm{D}\left({}^k\mathbf{H}_1, \overline{\mathbf{H}}_1\right) + \alpha_2 \, \mathrm{D}\left({}^k\mathbf{H}_2, \overline{\mathbf{H}}_2\right) \right)$$

S-KCV D-KCV DS-KCV

A greedy optimization algorithm

- There are K^N potential assignments
- Similar to the *Multi-way number partitioning problem* (NP-hard)

Algorithm 1: A greedy algorithm to build a balanced K-fold

Inputs : K, N, $\{\forall h, \overline{\mathbf{H}}_h\}$

Output: $\mathbf{a}$ ▷ Videos assignation

$\mathbf{a} \leftarrow [0, 0, \dots, 0]$ ▷ Initialization

$\mathcal{V} \leftarrow \{0, \dots, N\}$

`/* While there is a video to dispatch among the folds */`

while $\mathcal{V} \neq \emptyset$ **do**

    foreach k *randomly picked from* $\{1, \dots, K\}$ **do**

        `/* Find the best video to assign to the fold` k `*/`

        $\mathbf{e} \leftarrow [0, 0, \dots, 0]$ ▷ Error vector

        forall j *in* $\mathcal{V}$ **do**

            ${}^{k}\mathcal{V}^{(temp)} \leftarrow {}^{k}\mathcal{V} \cup j$

            $\forall h,\ {}^{k}\mathbf{H}_h^{(temp)} \leftarrow \frac{1}{|{}^{k}\mathcal{V}^{(temp)}|} \sum_{i \in {}^{k}\mathcal{V}^{(temp)}} H_h^{(i)}$

            $\mathbf{e}[j] \leftarrow \sum_{h=0}^{m} \alpha_h \cdot dist\left({}^{k}\overline{\mathbf{H}}_h^{(temp)}, \overline{\mathbf{H}}_h\right)$

        $j^* \leftarrow \underset{j}{argmin}(\mathbf{e})$ ▷ Best video to assign

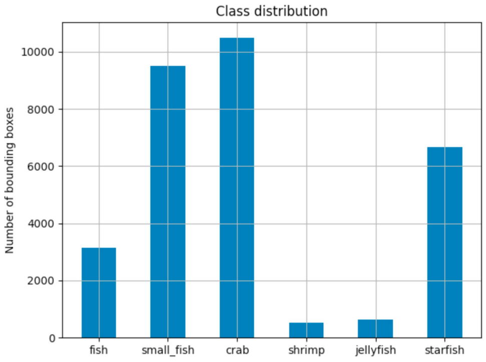
        $a[j^*] \leftarrow k$ ▷ Assign it to the fold

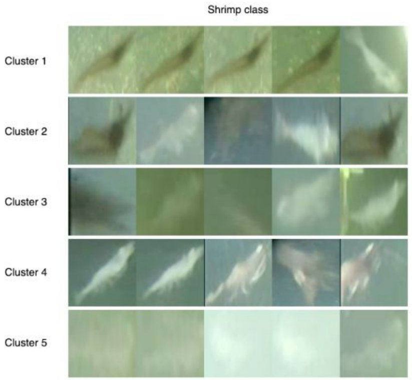
        $\mathcal{V} \leftarrow \mathcal{V} \backslash j^*$ ▷ Remove it from $\mathcal{V}$

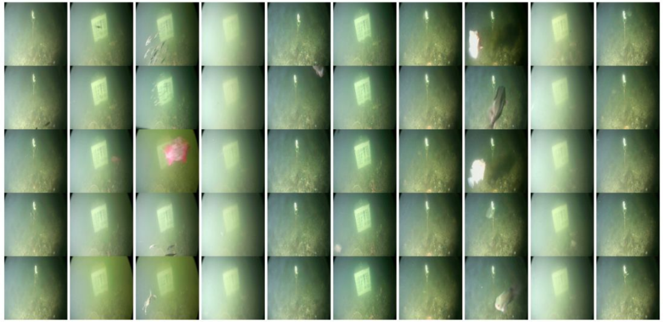
Application to image and video datasets

Scenario I: A video dataset with class imbalance (**Brackish**)

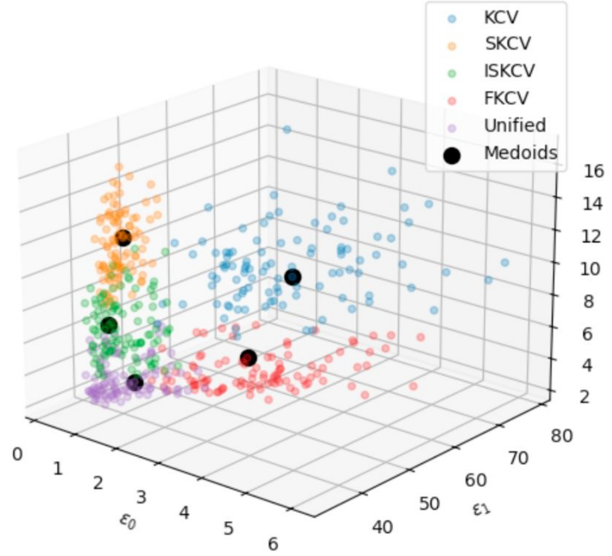
$$\mathbf{a}^* = \underset{\mathbf{a}\in\{1,\ldots,K\}^N}{\operatorname{argmin}} \left(\sum_{k=1}^{K} \alpha_0 \, \mathrm{D}\left({}^{k}\mathbf{H}_0, \overline{\mathbf{H}}_0\right) + \alpha_1 \, \mathrm{D}\left({}^{k}\mathbf{H}_1, \overline{\mathbf{H}}_1\right) + \alpha_2 \, \mathrm{D}\left({}^{k}\mathbf{H}_2, \overline{\mathbf{H}}_2\right) \right)$$

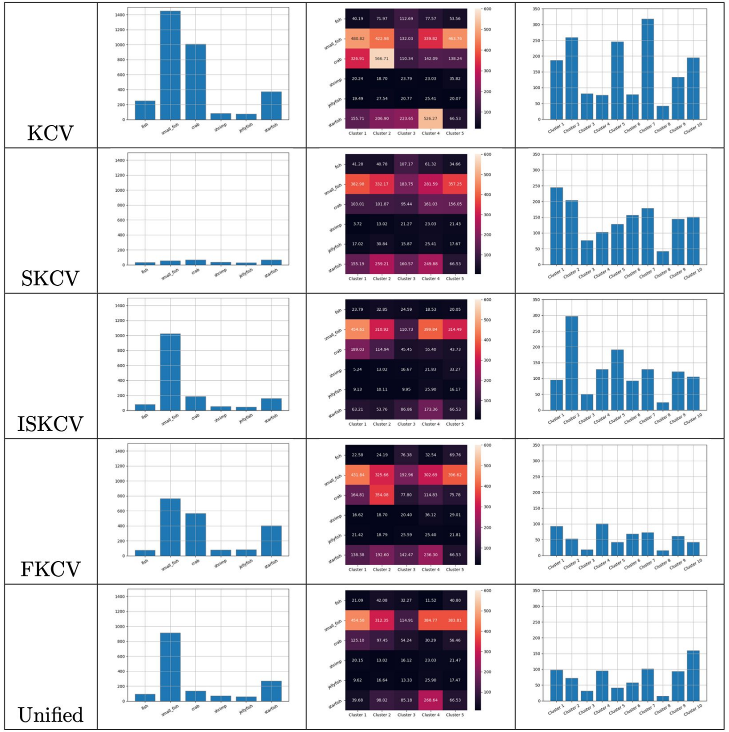






Consistency results

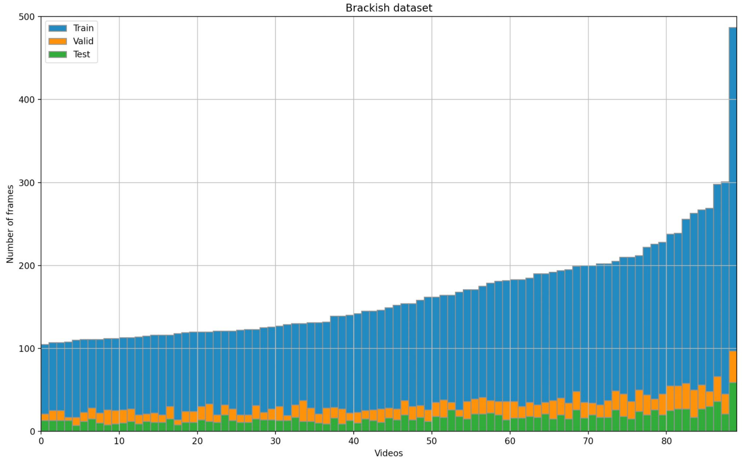




$$\mathbf{a}^* = \underset{\mathbf{a}\in\{1,\ldots,K\}^N}{\operatorname{argmin}} \left(\sum_{k=1}^{K} \alpha_0 \, \mathrm{D}\left({}^k\mathbf{H}_0, \overline{\mathbf{H}}_0\right) + \alpha_1 \, \mathrm{D}\left({}^k\mathbf{H}_1, \overline{\mathbf{H}}_1\right) + \alpha_2 \, \mathrm{D}\left({}^k\mathbf{H}_2, \overline{\mathbf{H}}_2\right) \right)$$

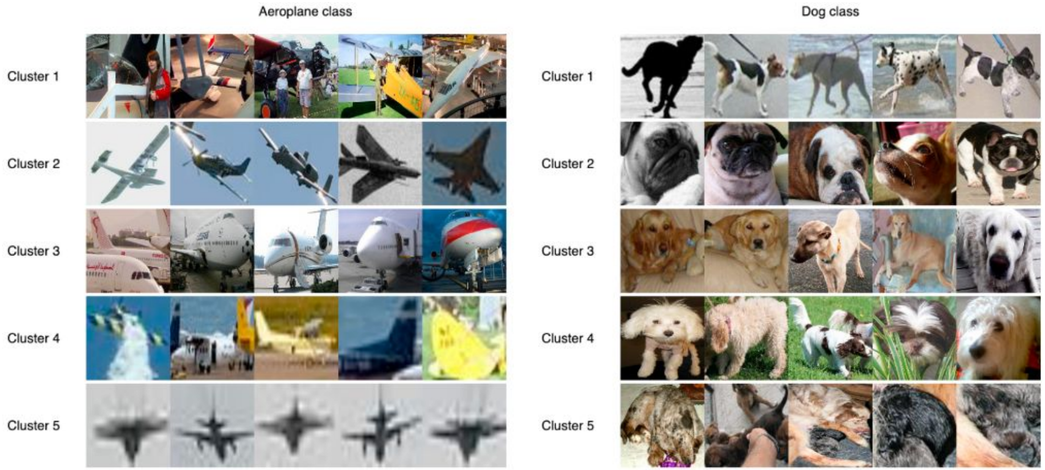
Results

Partitioning	P (%)	R (%)	F1 (%)	mAP@.5 (%)	mAP@.5:.95(%)
*Biased split	98.10	98.10	98.10	99.10	83.60
KCV	63.24 ± 9.61	43.90 ± 6.75	48.64 ± 5.74	47.83 ± 7.16	29.39 ± 6.35
SKCV	62.76 ± 9.58	44.55 ± 6.43	49.47 ± 6.26	48.97 ± 6.46	28.54 ± 3.62
ISKCV	64.06 ± 10.01	46.02 ± 6.32	50.92 ± 7.19	**50.25** ± 6.14	**29.65** ± 3.73
FKCV	62.94 ± **7.17**	44.95 ± 5.33	49.15 ± 5.30	48.89 ± **5.54**	29.43 ± 4.00
Unified	**66.38** ± 8.07	**46.31** ± **5.22**	**51.53** ± **5.29**	50.12 ± 5.98	29.54 ± **3.52**



Application to image and video datasets

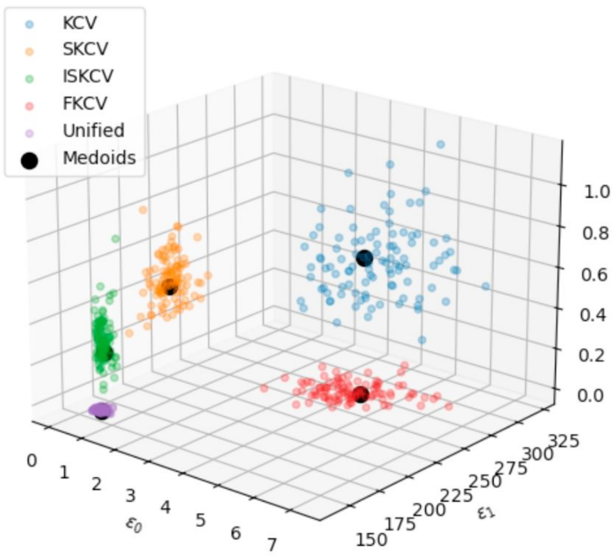
Scenario II: An image dataset with class imbalance (**PVOC10**)



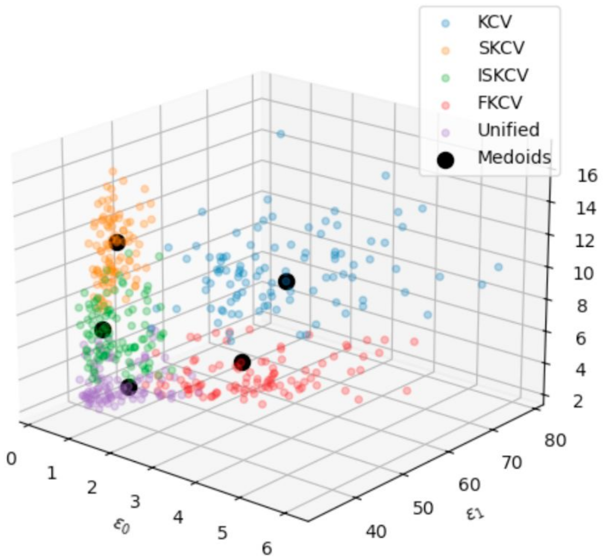
Conclusion

- New paradigm for creating consistent K-fold
- Application to object detection
- Significant results

Consistency results



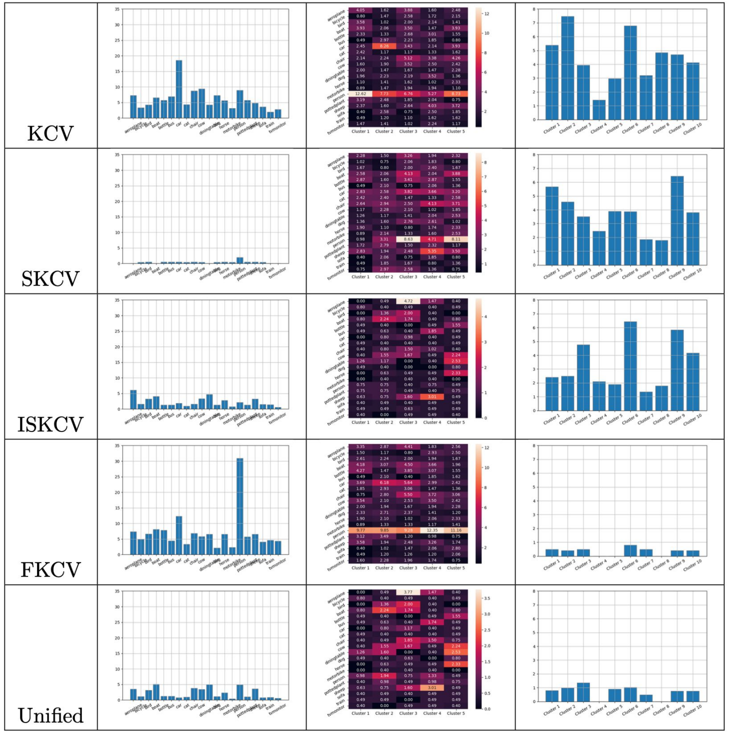
PVOC10



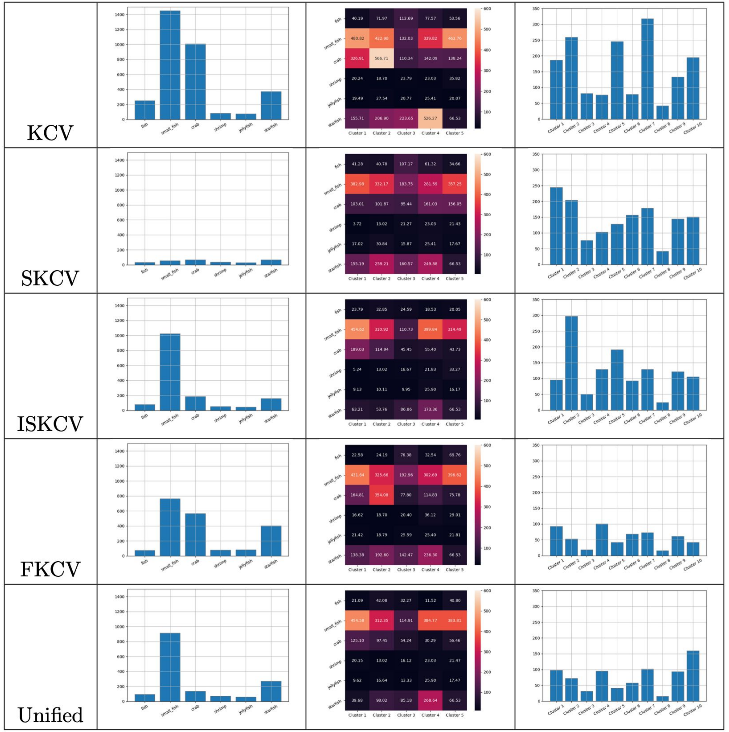
Brackish

Consistency results

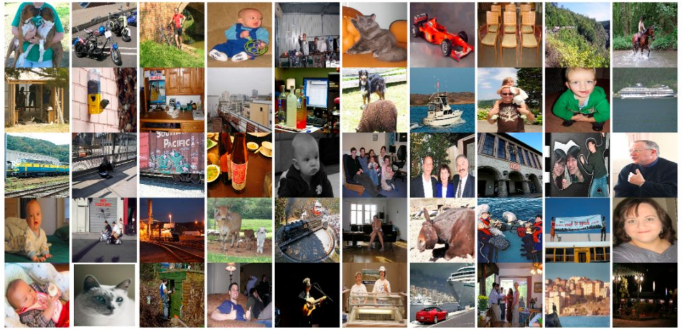
PVOC10

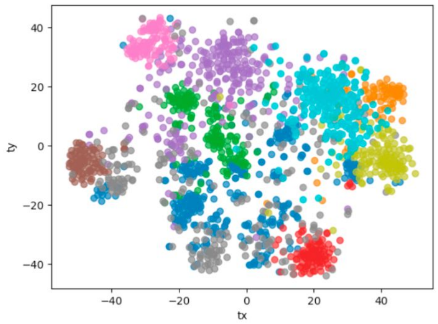


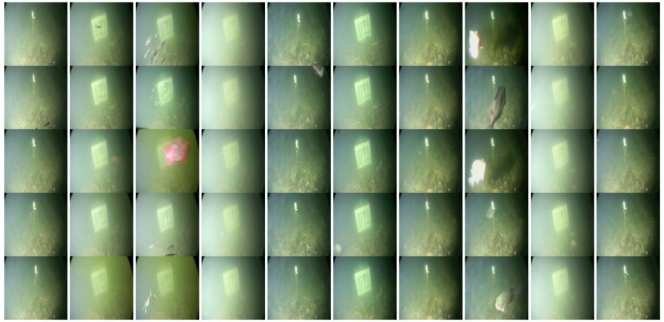
Brackish

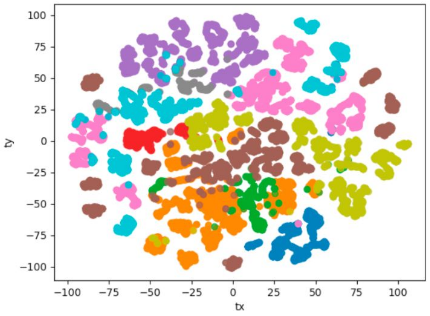


Clustering to get H_2









Experiments and results

Table 6: YOLOv5s mean $\overline{\overline{\theta}}(\mathcal{M})$ and standard deviation $\overline{\sigma}(\mathcal{M})$ on different PVOC10 partitioning.

Partitioning	P (%)	R (%)	F1 (%)	mAP@.5 (%)	mAP@.5:.95 (%)
KCV	61.33 ± 5.06	48.15 ± 3.05	51.97 ± 3.04	50.13 ± 2.66	31.13 ± 2.72
SKCV	59.62 ± 6.22	47.51 ± 3.47	50.79 ± 3.19	49.52 ± 3.53	30.51 ± 3.00
ISKCV	60.34 ± 5.02	**48.20** ± **2.35**	52.02 ± 2.09	50.42 ± 2.65	31.31 ± 2.49
FKCV	61.64 ± **4.68**	47.95 ± 3.14	51.64 ± **1.70**	50.07 ± **2.24**	31.27 ± **1.94**
Unified	**62.63** ± 4.72	47.59 ± 2.77	**52.29** ± 2.31	**50.43** ± 2.76	**31.60** ± 2.91

Table 7: YOLOv5s mean $\overline{\overline{\theta}}(\mathcal{M})$ and standard deviation $\overline{\sigma}(\mathcal{M})$ on different Brackish partitioning. *Note that the biased split does not consider a video as an image subset and randomly splits all the images among the folds as shown in Figure 7.

Partitioning	P (%)	R (%)	F1 (%)	mAP@.5 (%)	mAP@.5:.95(%)
*Biased split	98.10	98.10	98.10	99.10	83.60
KCV	63.24 ± 9.61	43.90 ± 6.75	48.64 ± 5.74	47.83 ± 7.16	29.39 ± 6.35
SKCV	62.76 ± 9.58	44.55 ± 6.43	49.47 ± 6.26	48.97 ± 6.46	28.54 ± 3.62
ISKCV	64.06 ± 10.01	46.02 ± 6.32	50.92 ± 7.19	**50.25** ± 6.14	**29.65** ± 3.73
FKCV	62.94 ± **7.17**	44.95 ± 5.33	49.15 ± 5.30	48.89 ± **5.54**	29.43 ± 4.00
Unified	**66.38** ± 8.07	**46.31** ± **5.22**	**51.53** ± **5.29**	50.12 ± 5.98	29.54 ± **3.52**

Conclusion

- New paradigm for creating consistent K-fold in object detection tasks
- application to object detection
- results are significant
-
- We adapted previous works on classification to object detection (S-KCV, D-SKCV, and DS-KCV) in a unified framework
- We showed interesting results both maximizing the performance estimation and minimizing the variance estimation, facilitating unbiased comparison of different algorithms on the same database