Writing and evaluating reproducible research

Fundamentals of reproducible research and free software

Miguel Colom
http://mcolom.info

CENTRE BORELLI
école normale supérieure paris—saclay
université PARIS-SACLAY

Structure of a scientific research paper

1. Abstract
2. Introduction
3. Method
4. Results and discussion
5. Conclusions
6. References

Let's use YOLO as a use case: https://arxiv.org/pdf/1506.02640.pdf

Structure of a scientific research paper: abstract

- **150-200 word** text **representative** of the entire **article**
- Should be **fair**, not biased in order to attract the readers

- Note that many **bibliographic researches** are based on the **abstract** and **keywords**!

- A proposal to **structure** the abstract: 1. the **purpose** of the study (the central question); 2. a brief statement of **what was done** (Methods); 3. a brief statement of **what was found** (Results); 4. a brief statement of **what was concluded**.

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.

Structure of a scientific research paper: introduction

- **Explaining** the **field**
- **Review** of the **literature**
  - Previous works
  - State of the art
- **Answer** the **knowledge gap**. What's new? Why this should be published?
- **Explain** your **hypothesis**
- **Enumerating** the following **sections** of the article

Structure of a scientific research paper: introduction

Explaining the field

1. Introduction

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little con-

Structure of a scientific research paper: introduction

Review of the literature

Current detection systems repurpose classifiers to perform detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image [10].

More recent approaches like R-CNN use region proposal

Structure of a scientific research paper: introduction

- **Answer the knowledge gap. What's new? Why this should be published?**

nates and class probabilities. Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are.

YOLO is refreshingly simple: see Figure 1. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection.

First, YOLO is extremely fast. Since we frame detection as a regression problem we don't need a complex pipeline. We simply run our neural network on a new image at test

Structure of a scientific research paper: introduction

- **Explain your hypothesis**

Third, YOLO learns generalizable representations of objects. When trained on natural images and tested on artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin. Since YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected inputs.

Structure of a scientific research paper: introduction

- **What is new in this work with respect to the previous**

making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. Fast R-CNN, a top detection method [14], mistakes background patches in an image for objects because it can't see the larger context. YOLO makes less than half the number of background errors compared to Fast R-CNN.

Structure of a scientific research paper: introduction

- **Enumerating the following sections of the article**

cially small ones. We examine these tradeoffs further in our experiments.

All of our training and testing code is open source. A variety of pretrained models are also available to download.

Structure of a scientific research paper: methods (experiments)

- The **experiments performed**, which **support the claims objectively** with data
- **Detailed procedures** → **reproducible research**
- Explain **why the current setup is adequate** to provide **evidence**
- **Positive** and **negative examples**

Structure of a scientific research paper: method

- **The experiments performed, which support the claims. Explain why the current setup is adequate to provide evidence**

4. Experiments

First we compare YOLO with other real-time detection systems on PASCAL VOC 2007. To understand the differences between YOLO and R-CNN variants we explore the errors on VOC 2007 made by YOLO and Fast R-CNN, one of the highest performing versions of R-CNN [14]. Based

4.2. VOC 2007 Error Analysis

To further examine the differences between YOLO and state-of-the-art detectors, we look at a detailed breakdown of results on VOC 2007. We compare YOLO to Fast R-CNN since Fast R-CNN is one of the highest performing detectors on PASCAL and it's detections are publicly available.

We use the methodology and tools of Hoiem et al. [19] For each category at test time we look at the top N predictions for that category. Each prediction is either correct or it is classified based on the type of error:

Structure of a scientific research paper: method

- **Detailed procedures → reproducible research**

dence predictions for boxes that don't contain objects. We use two parameters, λ_{coord} and λ_{noobj} to accomplish this. We set $\lambda_{\text{coord}} = 5$ and $\lambda_{\text{noobj}} = .5$.

Structure of a scientific research paper: method

- **Detailed procedures → reproducible research**

loss function:

$$
\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \qquad (3)
\end{aligned}
$$

Structure of a scientific research paper: method

- **Detailed procedures → reproducible research**

Our learning rate schedule is as follows: For the first epochs we slowly raise the learning rate from 10^{-3} to 10^{-2}. If we start at a high learning rate our model often diverges due to unstable gradients. We continue training with 10^{-2} for 75 epochs, then 10^{-3} for 30 epochs, and finally 10^{-4} for 30 epochs.

Structure of a scientific research paper: method

- **Positive and negative examples**

Figure 6: Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane.

Structure of a scientific research paper: results and discussion

- **Elaborating** on how the **experimental results** presented in the **Methods** part are **significant** to **answer** the **questions** made in the **introduction**
- **Not a repetition** of the evidence already presented, but a **organized presentation**
- The discussion goes **beyond** the **experiments**
  - are the results generalizable? can the experimental setup be applied in different fields or purposes?
- **Substantial explanations** about the new method in the context of **new** or **existing theories** or **frameworks**
- Explanations about the **limitations** of the method
- **This can be two different sections**

Structure of a scientific research paper: results and discussion

- **Discussion on how the experimental results presented in the Methods part are significant to answer the questions made in the introduction**

Third, YOLO learns generalizable representations of objects. When trained on natural images and tested on artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin. Since YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected inputs.

compare mAP to current state-of-the-art methods. Finally, we show that YOLO generalizes to new domains better than other detectors on two artwork datasets.

Structure of a scientific research paper: results and discussion

- **The discussion goes beyond the experiments**

The boost from YOLO is not simply a byproduct of model ensembling since there is little benefit from combining different versions of Fast R-CNN. Rather, it is precisely because YOLO makes different kinds of mistakes at test time that it is so effective at boosting Fast R-CNN's performance.

Structure of a scientific research paper: results and discussion

- **New questions: are the results generalizable? can the experimental setup be applied in different fields or purposes?**

4.5. Generalizability: Person Detection in Artwork

5. Real-Time Detection In The Wild

Structure of a scientific research paper: results and discussion

- **Substantial explanations about the new method in the context of new or existing theories or frameworks**

Not found.

Structure of a scientific research paper: results and discussion

- **Explanations about the limitations of the method**

2.4. Limitations of YOLO

YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class. This spatial constraint limits the number of nearby objects that our model can predict. Our model struggles with small objects that appear in groups, such as flocks of birds.

Since our model learns to predict bounding boxes from data, it struggles to generalize to objects in new or unusual

Structure of a scientific research paper: conclusions

- **Remarking the key ideas from the Discussion section**
  - "YOLO, a **unified model** for object detection"
  - “Our model is **simple** to construct”
  - “Fast YOLO is the **fastest** general-purpose object detector in the literature”
  - “YOLO also **generalizes** well to **new domains**”

Structure of a scientific research paper: conclusions

- **Future work**

(Missing)

Conclusion of the YOLO paper

- Excellent paper
- Well structured
- Uses publicly available datasets
- Gives all details for reproducibility
- Discussion could be improved. It's focused on the performance

- Have a look at the licenses, quite funny!
  https://github.com/pjreddie/darknet

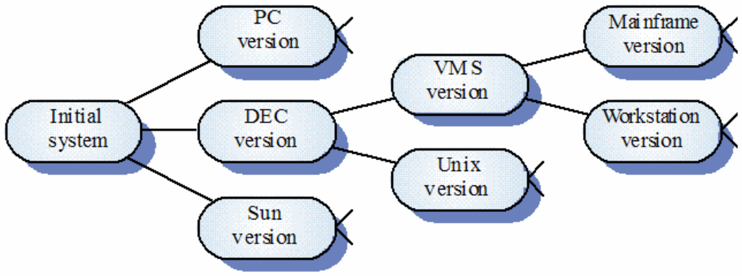
Reviewing reproducible articles

Reminder of the ACM definitions

- **Repeatability**. Can the **same team** with the **same experimental setup** perform the experiment as many times as needed? Existence of a **detailed procedure**, **same measurement system**, **same conditions**
- <u>**Reproducibility**</u>: **different team**, **same experimental setup.**
  The *same* numerical results are obtained.
- <u>**Replicability**</u>: **different team**, **different experimental setup**
  Equivalent results are obtained after following a procedure.

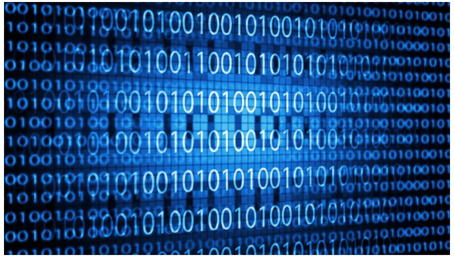
Pre-requisites (1/2)

- **Existence** of a **detailed procedure** for both the **compilation** and **execution**.
- The **exact environment** must be **declared** also (for example, to reconstruct a Docker container)
- The **exact version** of the **code** (commit, SWHID, …) must be declared. The review is only valid for an specific status of the code



Pre-requisites (2/2)

- The **data** must also be **referenced**. The **datasets** must be **public** and **reusable** and allow for comparison.
- **FAIR** principles:
  - **Findability;**
  - **Accessibility;**
  - **Interoperability;**
  - **Reusability**.



- The **reviewer** must be able to **obtain the same** or **comparable results** as in the **paper**. For example: the values in any figures or tables.

Pre-requisites

- The **reviewer** must be able to **obtain** the **same** or **comparable results** as in the paper. For example: the values in any figures or tables.

→ **Let's try!** With **IPOL's DCT denoising**

Checking the pseudocode and code

- As part as the **detailed procedure**, a **pseudocode** must be **available**
- The pseudocode **must describe exactly what the code does. The reviewer must check this.**
    - No hidden hyperparameters
    - No unexplained *magic* numbers
- The **names** of variables and functions **should match** the ones in the **article**
- **Comments** must be added to understand ***why*** the code does some operations (**not how!**)

Checking the pseudocode and code

- **As part as the detailed procedure, a pseudocode must be available. Input and outputs.**

Algorithm 2: DCT Denoising - Hard thresholding

Function $\text{DCTDenoisingHard}(Y, \sigma, s)$

  input : noisy image Y, noise level σ, and patch size s

  output: denoised image

  $X, W \leftarrow 0$

  $Y \leftarrow \text{DecorrelateColors}(Y)$

  for each *patch domain* $\Omega_{patch} \subset \Omega$ *of size* $s \times s$ **do** // Ω is the image support

    $b_{tmp} \leftarrow 0$ // color patch temp variable

    $N_P \leftarrow 0$

    for each *color channel* c **do**

      $\widehat{b} \leftarrow \text{DCT}(\text{ExtractPatch}(Y, \Omega_{patch}, c))$ // uses DCT/IDCT defined in (10)-(11)

      for $\omega \in (\{0, \cdots, s-1\} \times \{0, \cdots, s-1\})$ **do** // scan patch frequency domain

        if $\omega \neq \vec{0}$ **then** // don't filter the zero frequency

          if $|\widehat{b}(\omega)| < 3\sigma$ **then** $\widehat{b}(\omega) \leftarrow 0$

          else $N_P \leftarrow N_P + 1$ // # of nonzero coefficients of $\widehat{b}$

      $b_{tmp}[c] \leftarrow \text{IDCT}(\widehat{b})$ // store channel c of color patch

    $X(\Omega_{patch}) \leftarrow X(\Omega_{patch}) + b_{tmp} \cdot (1 + N_P)^{-1}$

    $W(\Omega_{patch}) \leftarrow W(\Omega_{patch}) + (1 + N_P)^{-1}$ // Adaptive weights, see Section A

  $X \leftarrow X/W$

  return $\text{UndoDecorrelateColors}(X)$

Checking the pseudocode and code

- **Comparison code / pseudocode**

```
inline void ExtractPatch(const Image &src, int pr, int pc, DCTPatch *dst) {
  // src is padded, so (pr, pc) becomes the upper left pixel
  for (int chan = 0; chan < dst->channels(); ++chan) {
    for (int row = 0; row < dst->rows(); ++row) {
      // // the following line copies a line interval to the patch and
      // // is equivalent toi (but faster):
      //   for (int col = 0; col < dst->columns(); ++col) {
      //     dst->space(col, row, chan) = src.val(pc + col, pr + row, chan);
      //   }
      copy(&(src.val(pc, pr + row, chan)),
           &src.val(pc + dst->columns(), pr + row, chan),
           &dst->space(0, row, chan));
```

$$\widehat{b} \leftarrow \mathrm{DCT}(\textsc{ExtractPatch}(Y, \Omega_{patch}, c))$$

- Names OK. Signature of the function different.

Checking the pseudocode and code

- **Comments must be added to understand *why* the code does some operations (not how!)**

Algorithm 2: DCT Denoising - Hard thresholding

Function DCTDenoisingHard(Y, σ, s)
  input : noisy image Y, noise level σ, and patch size s
  output: denoised image
  $X, W \leftarrow 0$
  $Y \leftarrow$ DecorrelateColors(Y)
  for each *patch domain* $\Omega_{patch} \subset \Omega$ *of size* $s \times s$ **do** // Ω is the image support
    $b_{tmp} \leftarrow 0$ // color patch temp variable
    $N_P \leftarrow 0$
    for each *color channel* c **do**
      $\widehat{b} \leftarrow$ DCT(ExtractPatch(Y, Ω_{patch}, c)) // uses DCT/IDCT defined in (10)-(11)
      for $\omega \in (\{0, \cdots, s-1\} \times \{0, \cdots, s-1\})$ **do** // scan patch frequency domain
        if $\omega \neq \vec{0}$ **then** // don't filter the zero frequency
          if $|\widehat{b}(\omega)| < 3\sigma$ **then** $\widehat{b}(\omega) \leftarrow 0$
          else $N_P \leftarrow N_P + 1$ // # of nonzero coefficients of $\widehat{b}$
      $b_{tmp}[c] \leftarrow$ IDCT($\widehat{b}$) // store channel c of color patch
    $X(\Omega_{patch}) \leftarrow X(\Omega_{patch}) + b_{tmp} \cdot (1 + N_P)^{-1}$
    $W(\Omega_{patch}) \leftarrow W(\Omega_{patch}) + (1 + N_P)^{-1}$ // Adaptive weights, see Section A
  $X \leftarrow X/W$
  return UndoDecorrelateColors(X)

Checking the pseudocode and code

- Any **pre/post processing must** be **explained** in the **paper**
- The **structure** (functions) of the code should be **reflected** in the **pseudocode**
- The **pseudocode** should have a **proper granularity**
    - No need to describe how to compute a cosine with a Taylor series!
    - However, any significant computation should be described. Example: in DCT denoising it's important that the DCT matrix is orthogonal to keep the isometric property.

Checking the pseudocode and code

- **Granularity**

$$\widehat{b} \leftarrow \mathrm{DCT}(\textsc{ExtractPatch}(Y, \Omega_{patch}, c)) \qquad \texttt{// uses DCT/IDCT defined in } (10)\text{-}(11)$$
$$\widehat{g} \leftarrow \mathrm{DCT}(\textsc{ExtractPatch}(G, \Omega_{patch}, c))$$

Checking the pseudocode and code

Isometric DCT transform. The type-II DCT transform implemented in the FFTW library and its inverse (type-III) are not isometric, so in order to implement the frequency domain denoising they must be normalized. The FFTW transforms (identified by the w superindex) compute for $k = 0, \cdots, N-1$

$$\mathrm{DCT}^{\mathrm{w}}(X)_k = 2\sum_{j=0}^{N-1} X_j \cos\left[\pi\left(j+\frac{1}{2}\right)\frac{k}{N}\right], \tag{5}$$

$$\mathrm{IDCT}^{\mathrm{w}}(Y)_k = Y_0 + 2\sum_{k=1}^{N-1} Y_k \cos\left[\pi\left(j+\frac{1}{2}\right)\frac{k}{N}\right], \tag{6}$$

which are unnormalized, hence $\mathrm{IDCT}^{\mathrm{w}}(\mathrm{DCT}^{\mathrm{w}}(X)) = 2N\,X$.
The isometric transforms $Y = \mathrm{DCT}(X)$ and $X = \mathrm{IDCT}(Y)$ that satisfy Parseval's equality $\sum_k |Y_k|^2 = \sum_j |X_j|^2$ are obtained as

$$Y_k = \mathrm{DCT}(X)_k = \alpha_k\ \mathrm{DCT}^{\mathrm{w}}(X)_k = \alpha_k\, 2\sum_{j=0}^{N-1} X_j \cos\left[\pi\left(j+\frac{1}{2}\right)\frac{k}{N}\right], \tag{7}$$

$$X_j = \mathrm{IDCT}(Y)_j = \mathrm{IDCT}^{\mathrm{w}}(\beta\cdot Y)_j = \beta_0\, Y_0 + \sum_{k=1}^{N-1} \beta_k\, 2Y_k \cos\left[\pi\left(j+\frac{1}{2}\right)\frac{k}{N}\right], \tag{8}$$

$$\text{with}\quad \alpha_k = \begin{cases}\sqrt{1/(4N)}, & k=0\\ \sqrt{1/(2N)}, & k=1\dots,N-1\end{cases} \quad\text{and}\quad \beta_k = \begin{cases}\sqrt{1/N}, & k=0\\ \sqrt{1/(2N)}, & k=1\dots,N-1.\end{cases} \tag{9}$$

The normalization factors corresponding to the 2D-DCT of a $N \times M$ image are given by

$$Y_{k,m} = \alpha_k\, \alpha'_m\ \mathrm{DCT2D}^{\mathrm{w}}(X)_{k,m}, \tag{10}$$

$$X_{j,l} = \mathrm{IDCT2D}^{\mathrm{w}}(\widetilde{Y})_{j,l} \qquad \text{with}\quad \widetilde{Y}_{k,m} = \beta_k\, \beta'_m\, Y_{k,m}, \tag{11}$$

where α' and β' are defined as in Equation (9) but for the range $[0\dots,M]$.

Checking the pseudocode and code

- **Optimizations**: the might cause the **source code** and **pseudocode look** quite **different**
- **Potential problem**: the pseudocode is **describing something different** to what has been **implemented**
- The paper need to **explain carefully** why **they're equivalent**. **Not granted**.

Checking the pseudocode and code

- **The paper need to explain carefully why they're equivalent. Not granted.**

```
for (int chan = 0; chan < dst->channels(); ++chan) {
  for (int row = 0; row < dst->rows(); ++row) {
    // // the following line copies a line interval to the patch and
    // // is equivalent toi (but faster):
    //   for (int col = 0; col < dst->columns(); ++col) {
    //     dst->space(col, row, chan) = src.val(pc + col, pr + row, chan);
    //   }
    copy(&(src.val(pc, pr + row, chan)),
         &src.val(pc + dst->columns(), pr + row, chan),
         &dst->space(0, row, chan));
```

Checking the pseudocode and code

- The **pseudocodes must** be **referenced** in the **paper**
- Each **pseudocode** must contain a **brief explanation** what's about, with its inputs and outputs.

Checking the pseudocode and code

- **The pseudocode s must be referenced in the text**

Using (3) and (4) for this procedure guarantees that white Gaussian noise remains so under the DCT transform, so the noise model remains the same in every layer of the pyramid. A scaling factor s used (Algorithm 4 lines 14 and 25) to guarantee that the values of the image remain on the same range after resizing, which also implies that the standard deviation of the noise gets halved at each

Checking the pseudocode and code

- **Each pseudocode must have a brief explanation what's about, with its inputs and outputs. Discuss about the following pseudocode:**

Algorithm 5: Multiscale DCT Denoising

Function $\textsc{MultiscaleDCT}(Y, \sigma, s, n_{scales}, f_{rec})$

  input : noisy image Y, noise level σ, patch size s, number of scales n_{scales}, and multiscale recomposition factor f_{rec}

  output: denoised image

  for $l \leftarrow n_{scales} - 1, \ldots, 0$ **do**

    $Y_l \leftarrow \textsc{ExtractScale}(Y, l)$

    $X_l \leftarrow \textsc{DCTdenoising2step}(Y_l, \sigma/2^l, s)$

    if $l == n_{scales} - 1$ **then** $combined \leftarrow X_l$

    else $combined \leftarrow \textsc{MergeCoarse}(X_l, combined, f_{rec})$

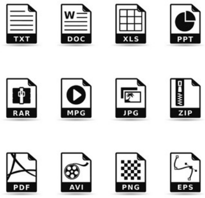
  return $combined$

Recommendations when writing a reproducible article

Recommendations about code availability, referencing, and environment

- **Make your code available**:
    - Github, Gitlab
    - Software Heritage → Version tracking. Easy referencing. Permanent archiving.
    - ...
- **Data repositories**
    - Zenodo
    - ...
- **Control** and **describe** your **environment**
    - Guix, Nix
    - Docker
    - Singularity
    - Virtual machines
    - TerraForm
    - ...

Recommendations about formats

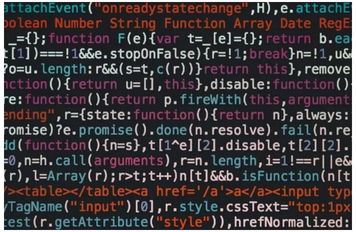


- Use **standards** and **reusable formats**. For both **documentation, code**, and **data**. **Avoid proprietary formats**.
- You can use, for example:
  - CSV
  - HDF
  - LaTex
  - And many others

- Use **public datasets**. Make your own **research artifacts FAIR** (for example, in Zenodo)

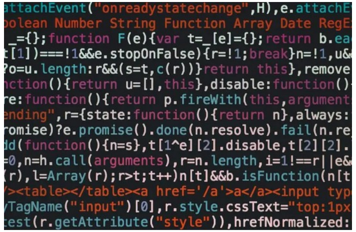
Recommendations about code quality (1/2)

- Use **asserts** to **control errors**. Specially during active development.
- Save an example of **execution** and **compare** with the **output** if you change anything
- **Comment** the **code**: **why it does** something, **not how!** "# sum a and b" vs "# Compute the accumulated cost"



Recommendations about code quality (2/2)

- **Document** your software. To avoid that it gets **unsynced** with the code you can use **automatic documentation** (Doxygen and others)
- Give a **version number** or **commit version** to your **released software**
- **Ask** your **colleagues** to **review** your code and article before submitting it



Recommendations to write the article

- **Cite** the **work** of **others**. Statements must be **cited** or **proven**! **Reproducing** existing work **without citation** may be considered **plagiarism**!
- **Scientific writing** should be **factual**, **concise** and **evidence-based**, but that doesn't mean it can't also be **creative**, **appealing to the readers**. **It must be.**
- **Avoid speculation** in the **discussion** section. You can **add** some in the **conclusions**. For example, about the evolution of the field.
- **Focus** your paper on a **single and clear key message or claim**. The **title** should reflect this, and it should be **clear** in the abstract.

Recommendations to write the article

- Use **institutional emails**. You're **working in a group**!
- The **abstract** typically is 1**50-200 words**, but check the journal/conf. rules
- A proposal to **structure** the **abstract**: 1. the **purpose** of the study (the central question); 2. a brief statement of **what was done** (Methods); 3. a brief statement of **what was found** (Results); 4. a brief statement of **what was concluded**.
- **Avoid *"I"*** and use ***"we"*** (even if you alone! *On the shoulders of giants*)
- **Tense**: **methods** section in **past tense**. **Conclusions** in **present tense**.

Recommendations to write the article

- The **captions** of the **figures** must be **complete**, even if some **text is repeated from a section**. They should explain **what the figure is showing**, along with any **information needed** for the **interpretation**. Help the lazy reader.

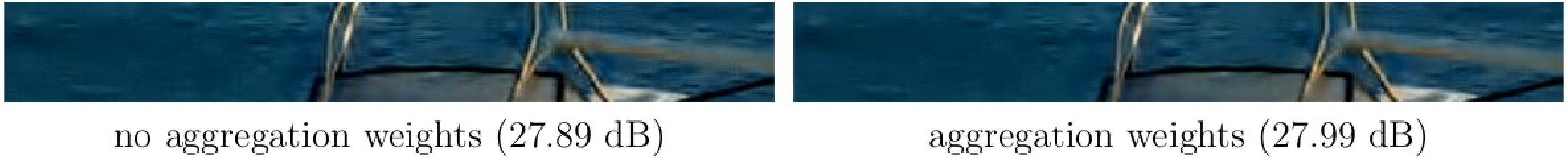


Figure 1: Detail of a result from MS DCT denoising with 8×8 patches computed without and with aggregation weights for a noise level $\sigma = 50$. Note the reduced oscillations in the sky.

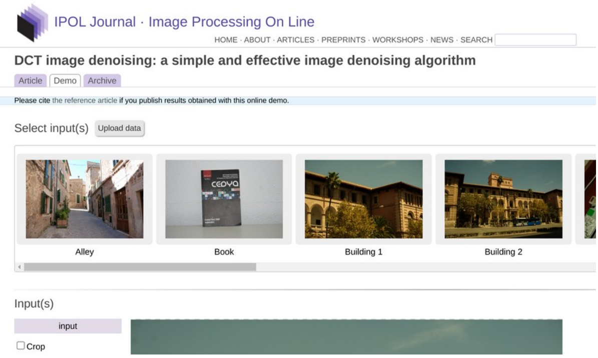
Recommendations to write the article

- Any **important equations** must be **numbered**, and **referenced** in the text.
- **Graphics**: use **vector graphics** whenever possible (PDF, SVG)
- **Review** the **bibliography**. Review the **format** of the **citations**. Check that it's **complete**.

Recommendations for the online demos (1/3)

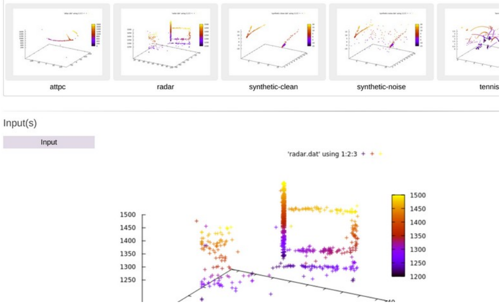
→ **Online demos** are very **useful**. They allow other researchers to quickly **obtain results** and **compare**. They increase the **impact** of your **publication**.

- **Minimize** the number of **parameters**: it's a demo, not a complete app. If needed, add an **"expert mode"** to show the rest of the parameters.



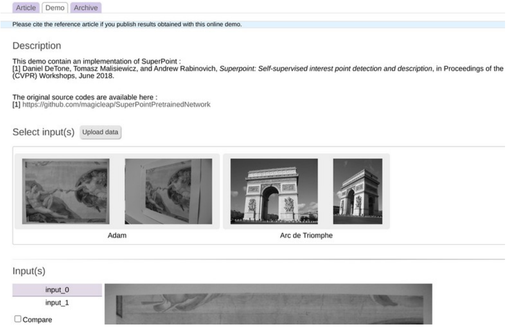
Recommendations for the online demos (2/3)

- Add a **short explanation** of each **parameter**
- Choose **typical default values**
- Choose a **reasonable range** of values (min, max, default) for the **parameters**
- **Limit** the **range** of the **parameters** which cause **too-long executions**. A user typically waits no more than **30 seconds**. (“*who waits forever, anyway?*”)



Recommendations for the online demos (3/3)

- **Show results** in a way that they **illustrate** the **method** and are **easy** to **interpret**
- Add a **small introduction** in the **demo**. Some users might land directly there from a Google's search. The **demo** must be **auto-contained**.
- Check the **online archive** now and then, since you'll find **unexpected results** which will bring you **insights** for **your research**.



References

- Mack, C. (2014). How to write a good scientific paper: structure and organization. J. Micro/Nanolith. MEMS MOEMS, 13(4), 040101.
- Alston, J. M., & Rick, J. A. (2021). A beginner's guide to Conducting reproducible research. Bulletin of the Ecological Society of America, 102(2), 1-14.
- Steingraber, S., Jolls, C., & Goldberg, D. (1985). Guidelines For Writing Scientific Papers. Honors Organismal Biology Laboratory Manual. Web. Retrieved July, 11, 2014.
- Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. arXiv preprint arXiv:1709.05943.

References

- Nicola Pierazzo, Jean-Michel Morel, and Gabriele Facciolo (2017). Multi-Scale DCT Denoising. Image Processing On Line, 7, 288–308.