Questions

Pablo Mesejo

pmesejo@go.ugr.es

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

UNIVERSIDAD DE GRANADA

DaSCI

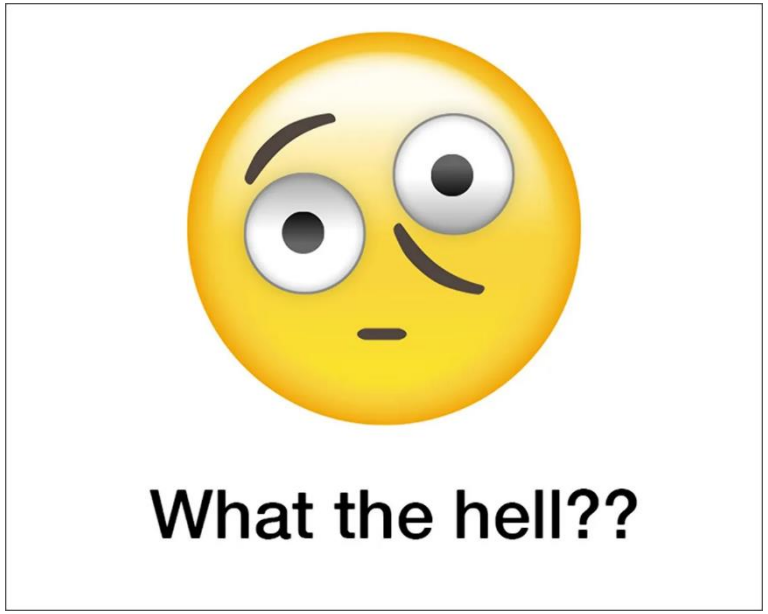
Instituto Andaluz de Investigación en **Data Science** and **Computational Intelligence**

Progress in Deep Learning (and science in general)

This paper says “X”

But this paper proposes “Y” (and even “no X”)

And this says that “Z” is the correct thing to do (and suggests that “X” and “Y” are both wrong)



In science, the coexistence of contradictory statements is common. Specially, in very dynamic and young research fields (like DL).

Science make progress in this way (advancing from conflicting views)

What to do? My advice:

- Follow the conclusions of the most recently published paper.
  - Always that you consider that the evaluation performed is rigorous.
  - These conclusions can change in the future (“nothing is written in stone”) but, at least, is the most up-to-date information we have.
- The devil is in the details.
  - Regarding DL: some recommendations may be for a particular type of network, or may only be valid in combination with other elements (such as the problem or a set of layers). Pay much attention to details.
- Prioritize scientific publications from well-established researchers, institutions and venues (and/or highly cited).

Furthermore, in general, and Deep Learning is not an exception, theory goes behind practice.

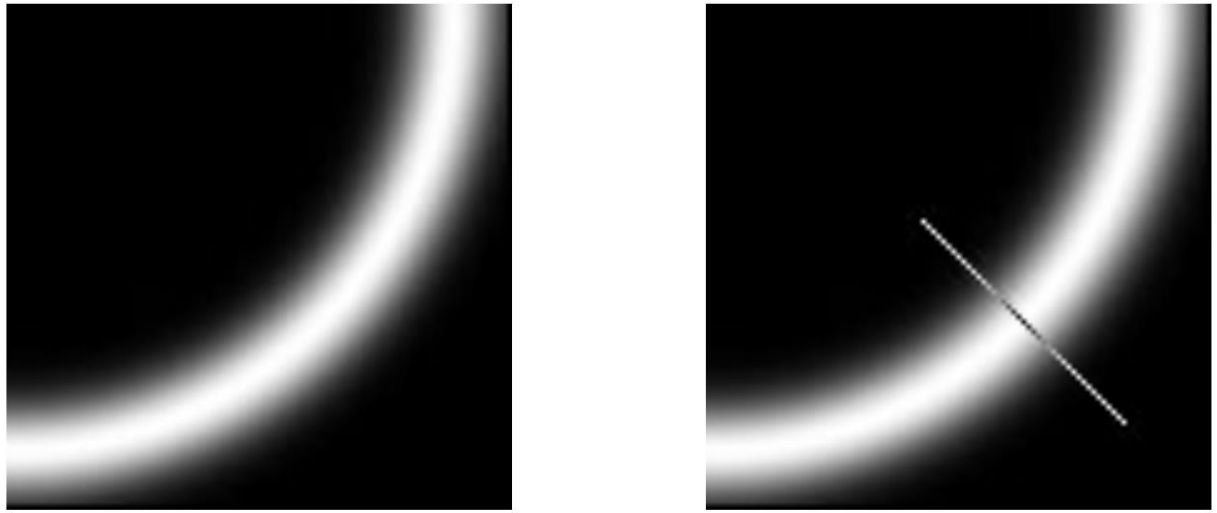
"In global terms, mathematics is very far behind computer science [in making fast progress in deep learning]. That's a very classic situation in science. If you look the history of physics, from the first experimental discoveries to the theory, you typically have several centuries. Here, we are speaking of 10, 20, 30 years. This is not only normal but it's great for maths. Because that means you have new problems, and I think what is the beauty of these problems is that the range of mathematics is very diverse (Fourier transform, convolution, harmonic analysis, algebra, differential equations, high-dimensional statistics, geometry,...). It's going to take some time, but I'm impressed by how fast the field is moving, including the math's side. Mathematics has always been moving slow, but it's moving... there are a lot of ideas coming in."

(Stéphane Mallat, DaSCI seminar, 16/05/2023)

Non-maximal suppression

Many ways to do it depending on your application

- Canny Edge Detector.
  - Thin multi-pixel wide “ridges” down to single pixel width



Forsyth & Ponce (1st ed.) Figure 8.11

Select the image **maximum point** across the width of the edge

Many ways to do it depending on your application

- SIFT.

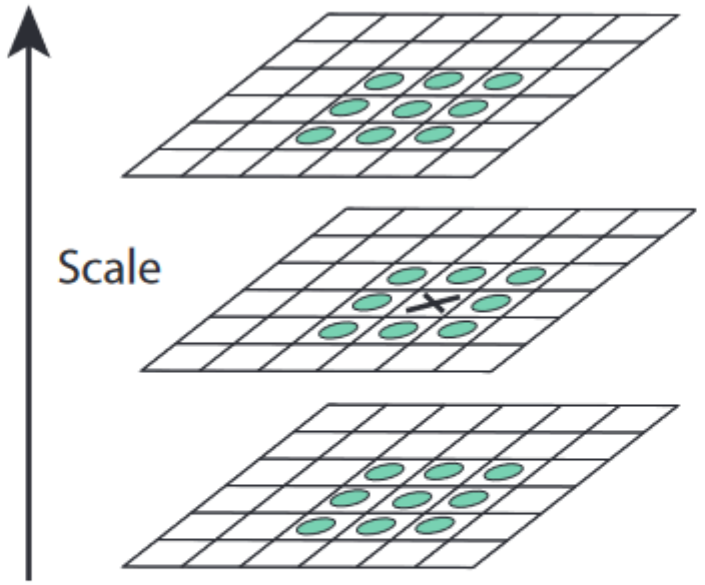
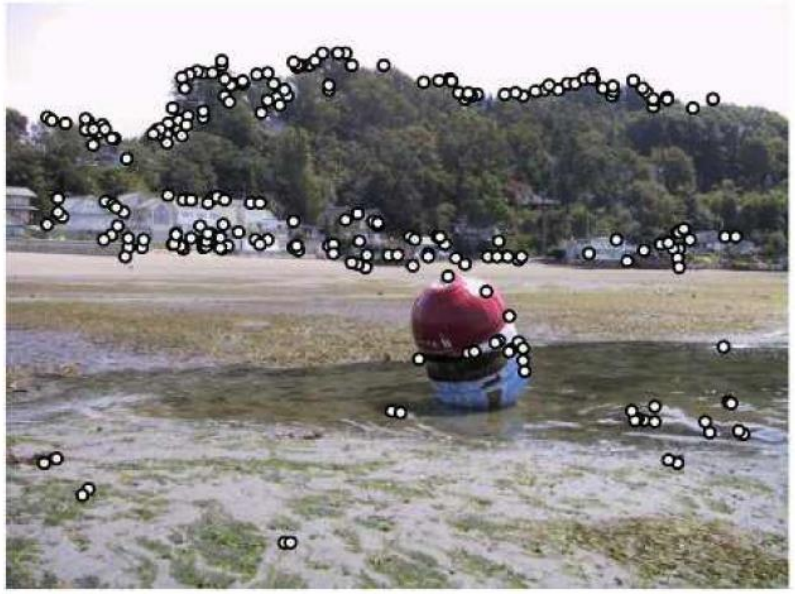


Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

Many ways to do it depending on your application

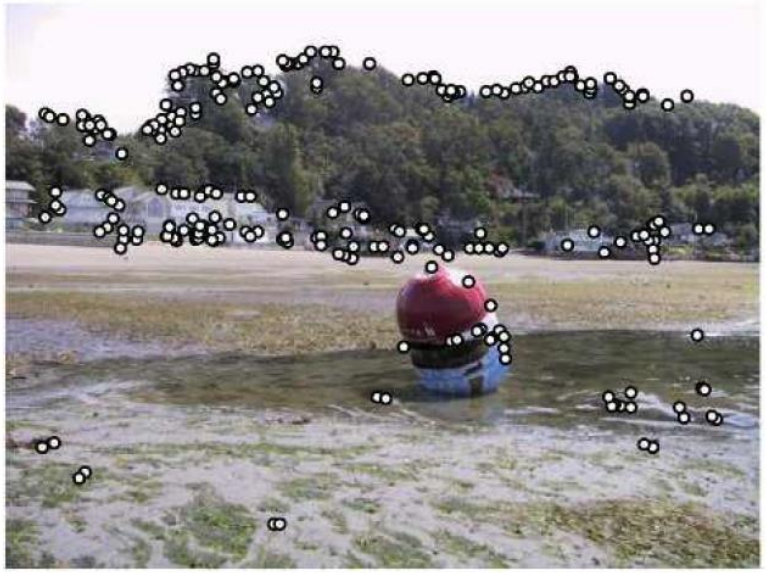
- Adaptive non-maximal suppression (ANMS). Szeliski 7.1.1
  - If you simply look for local maxima → this can lead to an uneven distribution of feature points across the image.



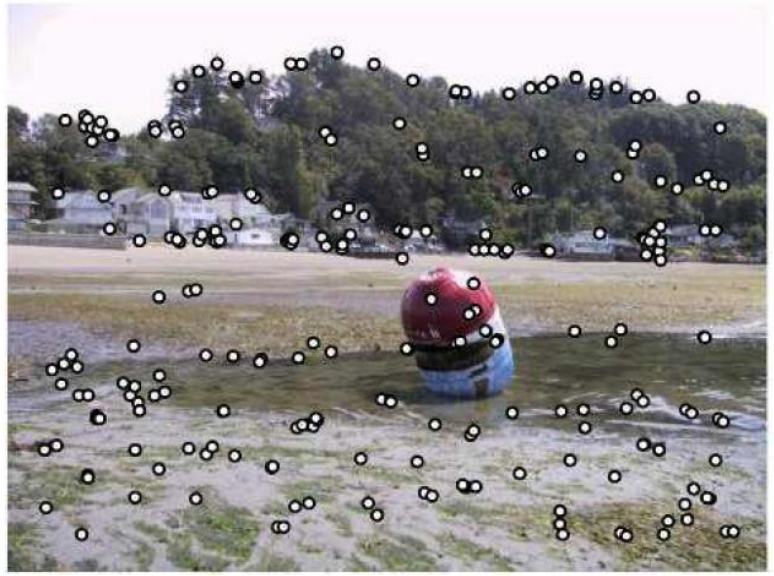
(a) Strongest 250

Many ways to do it depending on your application

- Adaptive non-maximal suppression (ANMS). Szeliski 7.1.1
  - detect features that are both local maxima and whose response value is significantly (10%) greater than that of all of its neighbors within a radius r



(a) Strongest 250



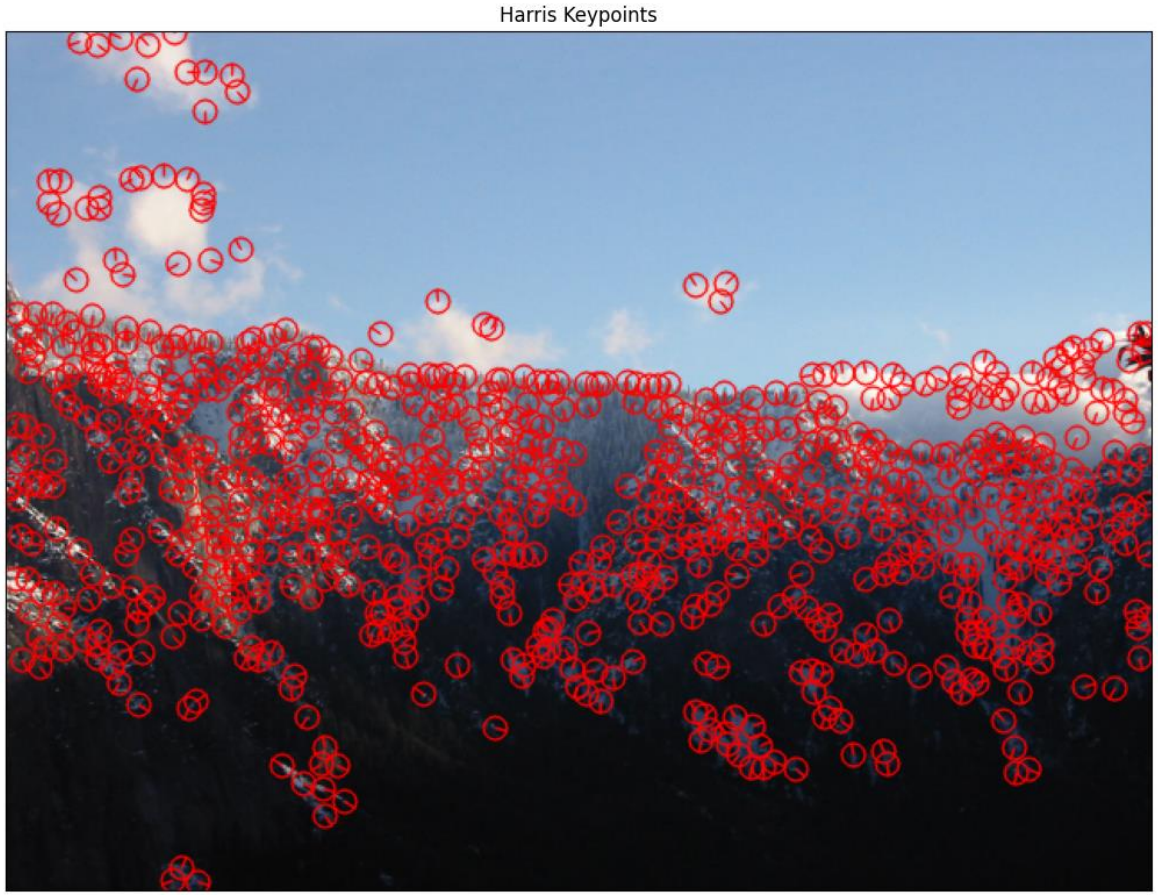
(c) ANMS 250, $r = 24$

Many ways to do it depending on your application

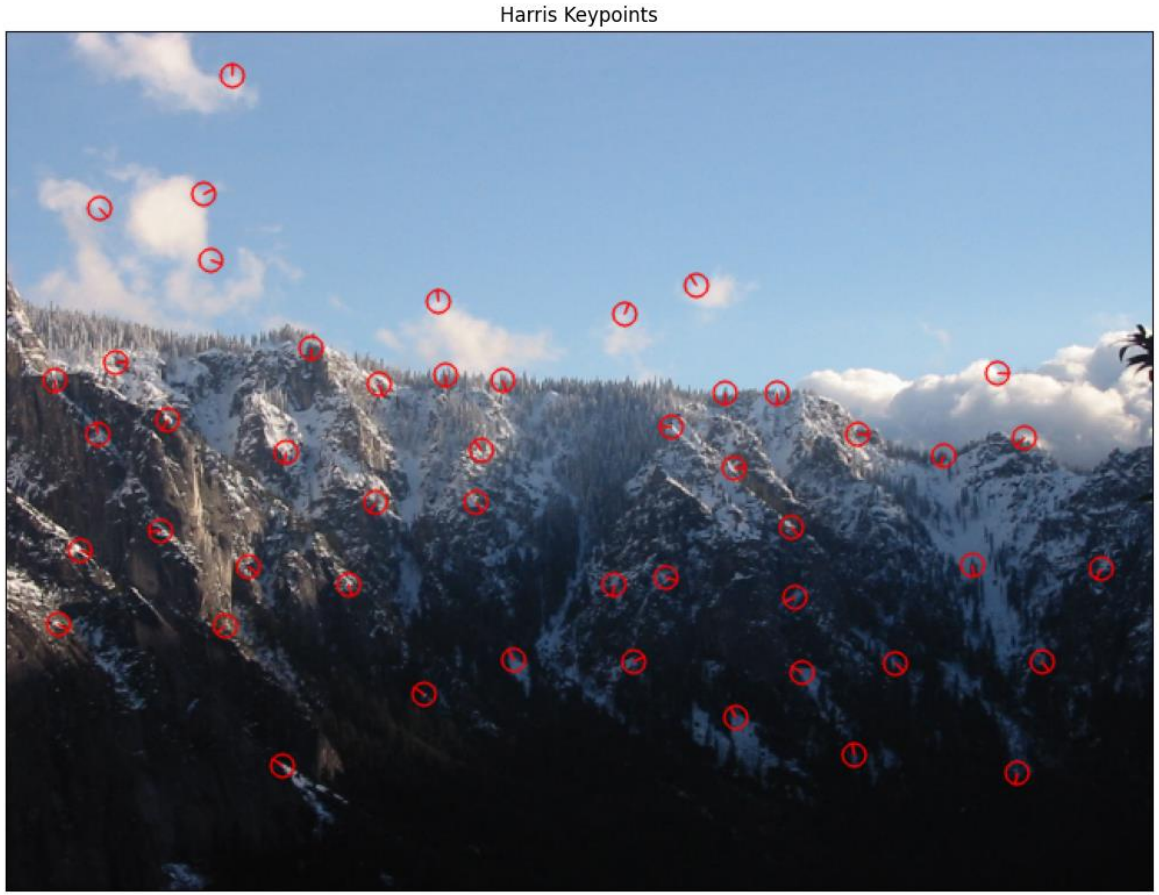
- Adaptive non-maximal suppression (ANMS). Szeliski 7.1.1
  - We'll do something similar when we implement Harris Corner Detector.
  - We threshold the Harris Map + we set a minimal allowed distance separating peaks (**skimage.feature.corner_peaks**)

- Using the same threshold…

min_distance=1



min_distance=25



Regarding batch sizes, learning rates, and the interaction between hyperparameters

- Intuitively, larger batch size should increase “effectiveness”
  - Because gradient update takes into account a significant portion of the dataset.
- Smaller batch size → train with gradients estimated from a smaller portion of the dataset.
  - A smaller “chunk” of the dataset will be less representative of the overall relationship between the features and the labels.

- However, it has been empirically observed that **increasing the batch size of a model typically decreases its ability to generalize** to unseen datasets → **Generalization Gap**

- Wilson and Martinez. "The general inefficiency of batch training for gradient descent learning." *Neural networks* 16.10 (2003): 1429-1451.

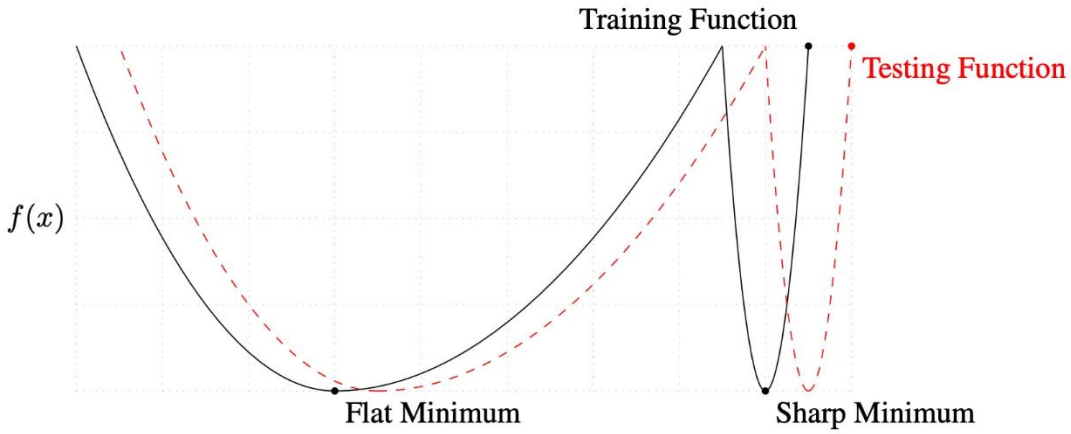
Learning rate	Batch size	Max word accuracy (%)	Training epochs
0.1	1	96.49	21
0.1	10	96.13	41
0.1	100	95.39	43
0.1	1000	84.13 +	4747 +
0.01	1	96.49	27
0.01	10	96.49	27
0.01	100	95.76	46
0.01	1000	95.20	1612
0.01	20,000	23.25 +	4865 +
0.001	1	96.49	402
0.001	100	96.68	468
0.001	1000	96.13	405
0.001	20,000	90.77	1966
0.0001	1	96.68	4589
0.0001	100	96.49	5340
0.0001	1000	96.49	5520
0.0001	20,000	96.31	8343

The smaller the batch size, the better (for generalization)

- Keskar et al. "On large-batch training for deep learning: Generalization gap and sharp minima." *ICLR 2017*.

– **Large batches tend to overfit** compared to the same network trained with smaller batch size.

– Large batches tend to zoom in on the closest relative minima that it finds, whereas networks trained with a **smaller batch size tend to "explore" the loss landscape** before settling on a promising minimum.



Small batch size adds noise to training
→ This noise seems to be beneficial to avoid sharp minima.

- Hoffer et al. "Train longer, generalize better: closing the generalization gap in large batch training of neural networks." *NeurIPS* 2017.
  - The larger the batch size, the smaller the number of updates → with a lower number of weight updates, the chances of approaching a minimum are smaller
  - “The "generalization gap" stems from the relatively small number of updates rather than the batch size”
  - They proposed an **algorithm to decrease the effects of the Generalization Gap whilst being able to keep a relatively large batch size**.

Note: the larger the batch, the faster you train (and more memory you need)

- Smith et al. "Don't decay the learning rate, increase the batch size." *ICLR 2018*.
  - We already know:
    - Small learning rate → overfitting
    - Large batch size → overfitting
    - It's common to decay the learning rate → first you explore, then you exploit
  - In this paper, instead of decaying learning rate, the authors increase batch size.

"It reaches equivalent test accuracies after the **same number of training epochs**, but with **fewer parameter updates**, leading to greater parallelism and **shorter training times**. "

- Smith, Leslie N. "A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay." *US Naval Research Laboratory Technical Report 5510-026* (2018).
  - Small batch sizes have regularization effects.
  - He recommends using a larger batch size when using the 1cycle learning rate schedule.
  - Goal: “obtaining the highest performance while minimizing the needed computational time.”
  - Faster training:
    - larger batch sizes (overfitting) + larger learning rates (exploration)

Yann LeCun ∞

@ylecun

Training with large minibatches is bad for your health.
More importantly, it's bad for your test error.
Friends dont let friends use minibatches larger than 32.

Traducir post

arxiv.org

Revisiting Small Batch Training for Deep Neural Networks

Modern deep neural network training is typically based on mini-batch stochastic gradient optimization. While the us...

11:00 p. m. · 26 abr. 2018

https://arxiv.org/abs/1804.07612

"The best performance has been consistently obtained for **mini-batch sizes between m=2 and m=32**, which contrasts with recent work advocating the use of mini-batch sizes in the thousands."

- Remember to double-check the slides, for instance:
  - Slides 43 and 44 from P2.pdf

Questions

Pablo Mesejo

pmesejo@go.ugr.es

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

UNIVERSIDAD DE GRANADA

DaSCI

Instituto Andaluz de Investigación en **Data Science** and **Computational Intelligence**