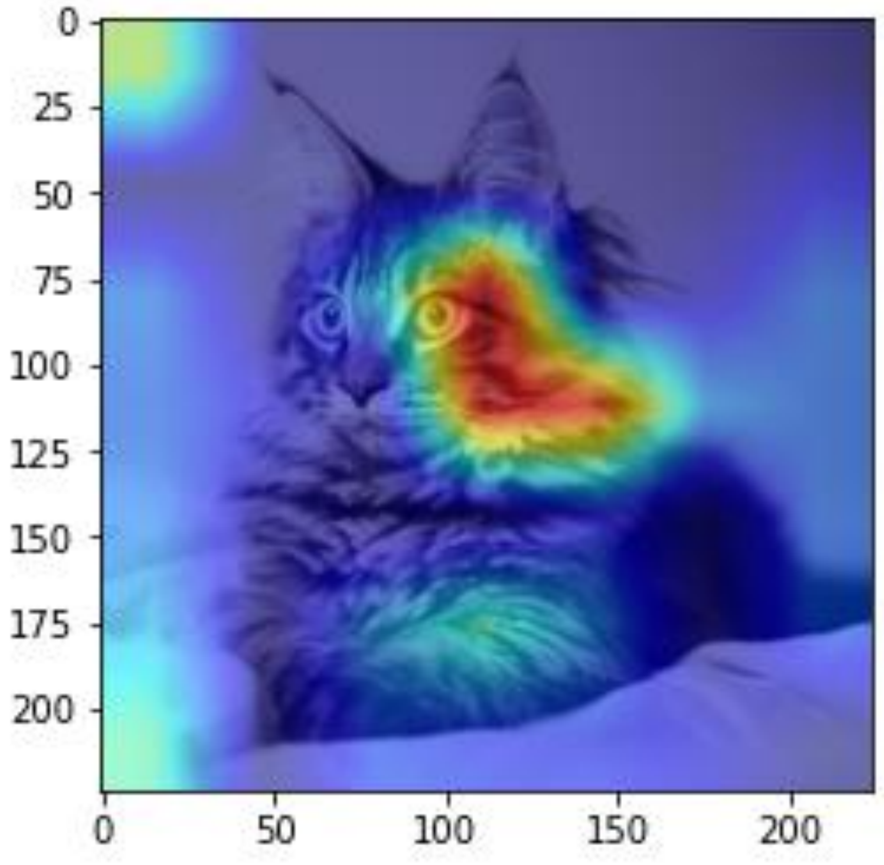
Making sense of CNNs: Visualization techniques

Alberto TONDA, Ph.D. (Senior permanent researcher, DR)

UMR 518 MIA-PS, INRAE, AgroParisTech, Université Paris-Saclay
UAR 3611, Institut des Systèmes Complexes de Paris Île-de-France

Outline

- xAI: eXplainable AI
- Visualization of feature patterns
- Saliency maps
- Grad-CAM



Black-box effect

- Black-box effect is common to all ML algorithms
  - We know that the prediction is, but not why the model did it
  - Too many parameters to analyze, even for RF or big Decision Trees

- For CNNs, it's even *worse*
  - Feature construction/extraction step
  - What are the features used? What do they look like?
  - What parts of the images is the CNN analyzing to give a decision?

xAI: eXplainable AI

- Relatively new research field
  - "Open the black box", answer "Why is it behaving like that?"
  - **Local explanation**: why behavior for **this sample**?
  - **Global explanation**: why behavior in general, or for a class?

- Ongoing discussion
  - What is the definition of **explanation**?
  - Difference between *explanation* and *interpretation*
  - Is it possible to build white-box ML algorithms? Yes, but…*

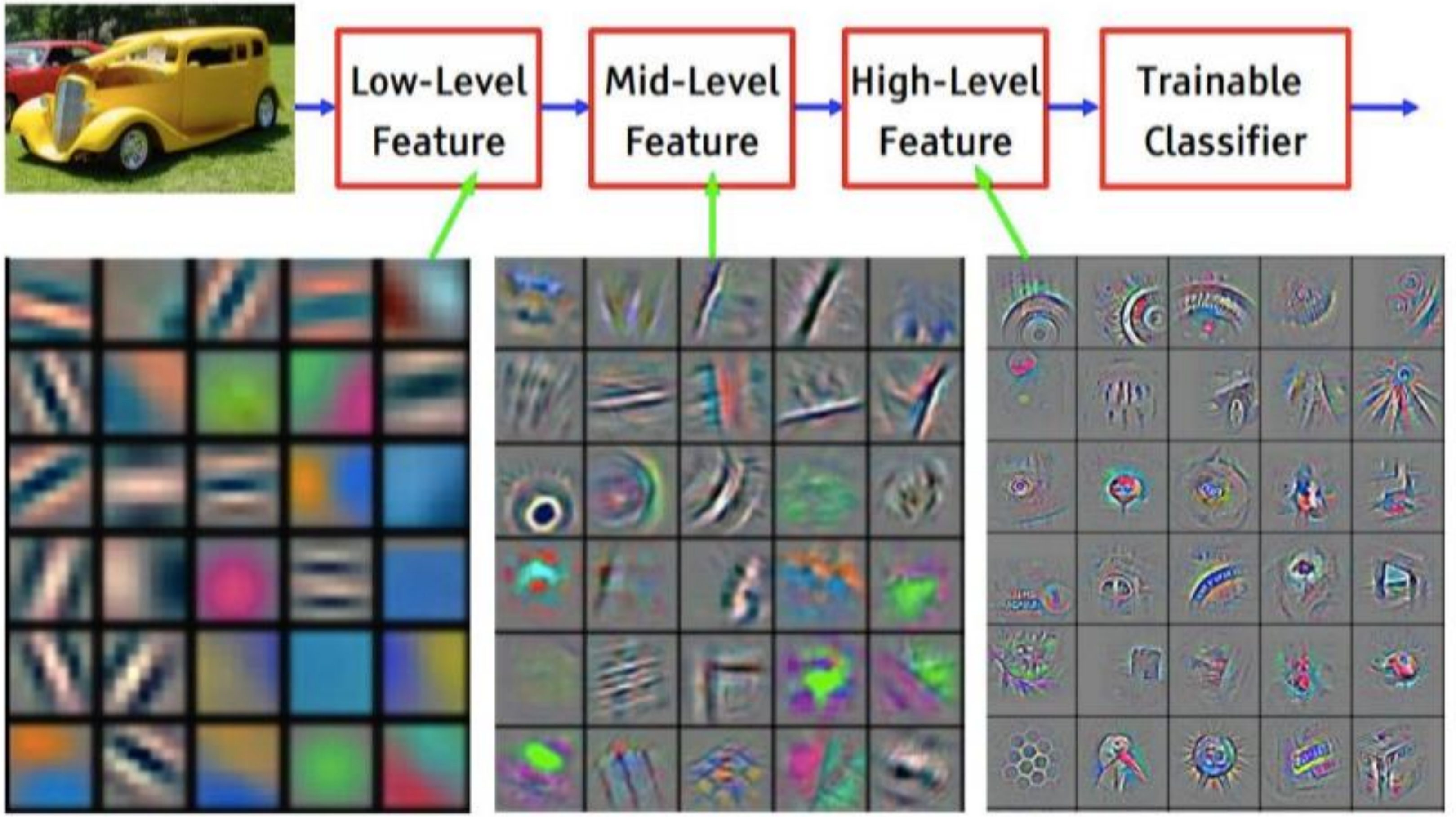
*…this is a long discussion. The short version is that there seems to be a **trade-off** between **interpretability** and **performance**

xAI: eXplainable AI

- Model-agnostic methods
  - Local Interpretable Model-agnostic Explanations (LIME)
  - SHapley Additive exPlanations (SHAP/SHAPLY)
  - Relative feature importance (e.g. permutation importance)

- For CNNs, we have some model-specific methods
  - Since the models are performing **feature construction/extraction**
  - We are not even sure of *what the features are*!

Visualization of feature patterns

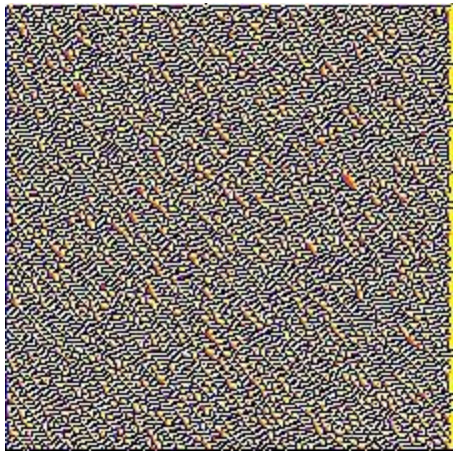
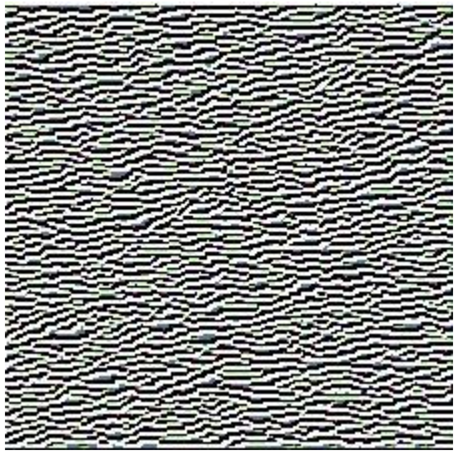
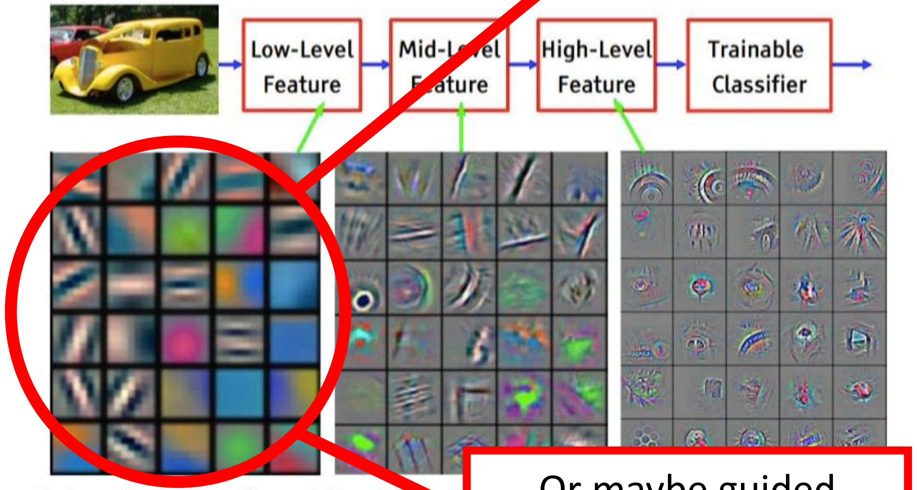


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Visualization of feature patterns

- Optimize an *image*
  - Treat pixel values as variables in the optimization problem
  - Generate image that *maximizes* output of a target filter
  - Backpropagating gradients to the *pixels*

Zoomed in

Feature visualization of convolutional net t...

Or maybe guided backpropagation…?

Hooks?

- The implementation uses **pytorch hooks**
  - Not very well documented
  - Essentially, connect a function to a module or tensor
  - Every time computes outputs (**forward**)…
  - Every time computes gradient (**backward**)…
  - …the function is invoked!
- Useful for debugging or visualization
  - Without having to tweak with model
  - E.g. without writing extra methods

Saliency maps

- What are the **most important pixels** for the decision?
  - Forward pass of an image through the network
  - Take relevant output (classification: tensor value for target class)
  - Compute derivative of that output w.r.t. to *pixels* (backward pass)

- Pixels with **high values of gradient** are more important
  - Changing their value might greatly impact decision
  - Visualize which pixels impact decision the most

Grad-CAM

- Looking at *one single pixel* at the time not very informative
  - Later CNN filters (high-level features) map to *image areas*
  - Can’t we do the same as saliency maps, but image areas?
- Gradient-weighted Class Activation Maps
  - Focus is on a **target class** for a classification problem
  - Outputs of **target modules(s)** more important for final decision?
- Obstacle: last part of CNN is a very complex set of non-linearities, hard to interpret
- Solution: replace it with a linear classifier, retrain it

Questions?

Bibliography

- Lundberg, S. M., & Lee, S. I. (2017). *A unified approach to interpreting model predictions*. Advances in neural information processing systems, 30. [**SHAP**]
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *"Why should I trust you?" Explaining the predictions of any classifier*. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144). [**LIME**]
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). *Grad-cam: Visual explanations from deep networks via gradient-based localization*. In Proceedings of the IEEE international conference on computer vision (pp. 618-626).
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). *Striving for simplicity: The all convolutional net*. arXiv preprint arXiv:1412.6806. [**Visualizing Feature Patterns**]