# MAC5921 – Deep Learning

Aula 21 – 21/11/2023

**Interpretabilidade / Explicabilidade**

Nina S. T. Hirata

# XAI – Explainable artificial intelligence

**Uma possível definição (vaga)**

*Set of processes and methods that allows human users to comprehend and trust the results and output created by machine learning algorithms*

**Sinônimos?**

Interpretability

Explainability

# XAI – Explainable artificial intelligence

## Uma possível definição (vaga)

*Set of processes and methods that allows human users to comprehend and trust the results and output created by machine learning algorithms*

## Sinônimos?

Interpretability – HOW?

Explainability – WHY?

# XAI – Explainable artificial intelligence

**Uma possível definição (vaga)**

*Set of processes and methods that allows human users to comprehend and trust the results and output created by machine learning algorithms*

**Sinônimos?**

Interpretability – HOW?

Explainability – WHY?

Promovem transparência
Geram confiança

## Abordagens usadas em XAI

Visualização de dados e modelos

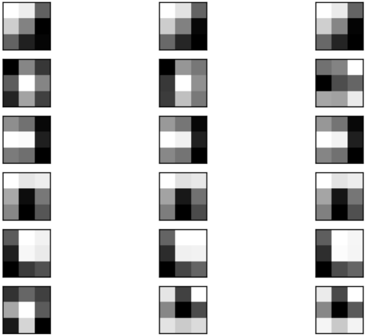
Decomposição de modelos – por exemplo, analisar classificação por classe

Explicação baseada em exemplos – usar exemplo similar ao qual o modelo será aplicado

Métodos Post-hoc – explicar o processo de decisão com modelo já treinado

Apresentação neste slide restrito à imagens/CNNs

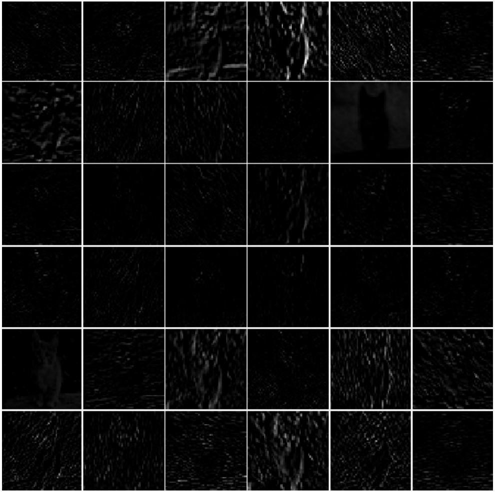
# Filters (kernels)



https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/

Each row corresponds to a 3-channel filter, from the first layer of VGG

# Individual feature maps



## Perguntas que estão tentando responder

Qual parte da imagem está sendo responsável por uma certa ativação?

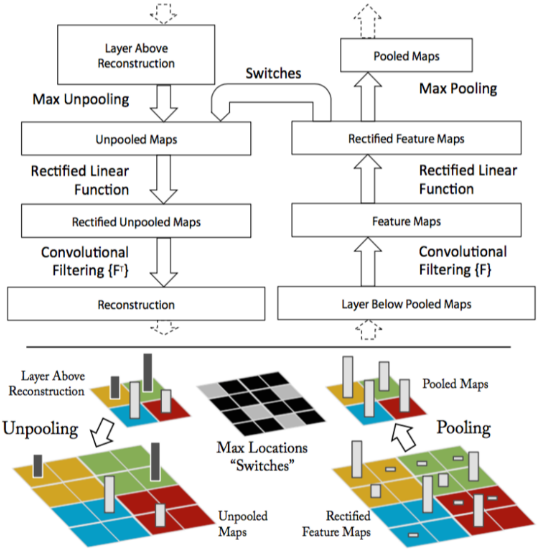
Que tipo de "padrão" uma unidade da rede enxerga?

*Visualizing and Understanding Convolutional Networks*
Matthew D Zeiler, Rob Fergus
https://arxiv.org/abs/1311.2901 (2013)

Propõem o uso de **multi-layered Deconvolutional Network (deconvnet)**: busca inverter o mapeamento; em vez de imagem para *feature map*, busca mapear de *feature map* para o espaço da imagem

Para tanto, *successively (i) unpool, (ii) rectify and (iii) filter to reconstruct the activity in the layer beneath*

# Reconstruction of the input from a single feature map




(Zeiler et al., 2013)

Forward pass:

`Conv ---> ReLU ---> Pooling`

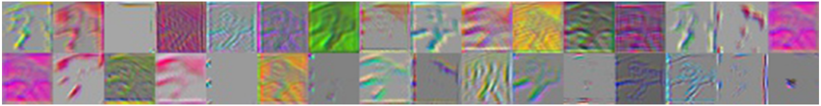
To reconstruct, we just need to perform the opposite sequence

`unPool ---> unReLU ---> deConv`

See Stanford CS230: Deep Learning — Autumn 2018 — Lecture 7 - Interpretability of Neural Network
https://youtu.be/gCJCgQW_LKc

Input image

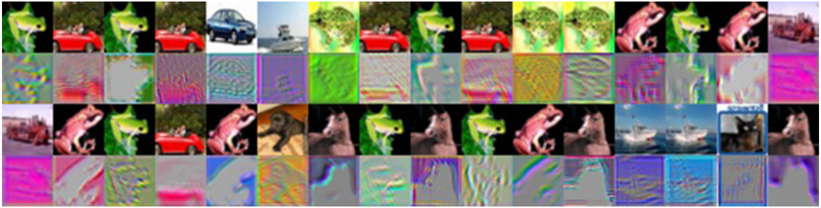
Images reconstructed from feature channels 1 to 32 of layer 1, for the above image



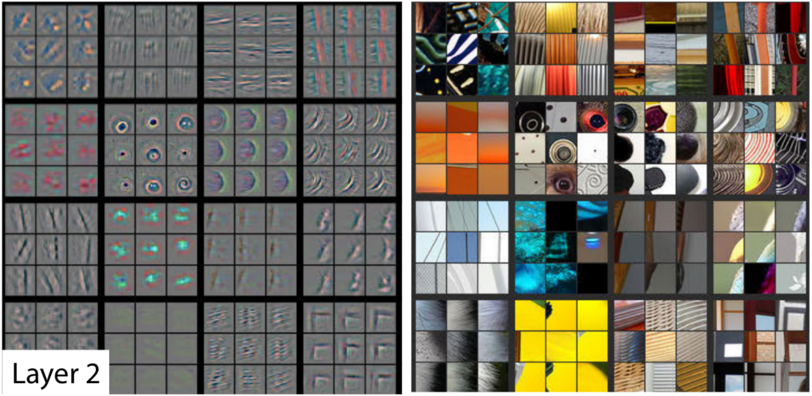
(http://kvfrans.com/visualizing-features-from-a-convolutional-neural-network/)

Images reconstructed from feature channel 7 of layer 1, for multiple input images

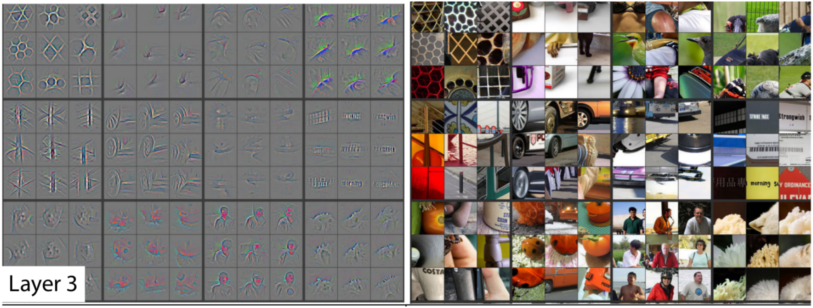
For each of the 32 feature channels, reconstruction of the images that most activated each channel



(Zeiler et al.) 16 feature channels in layer 2; for each channel, reconstruction of the 9 images that generate the strongest activation

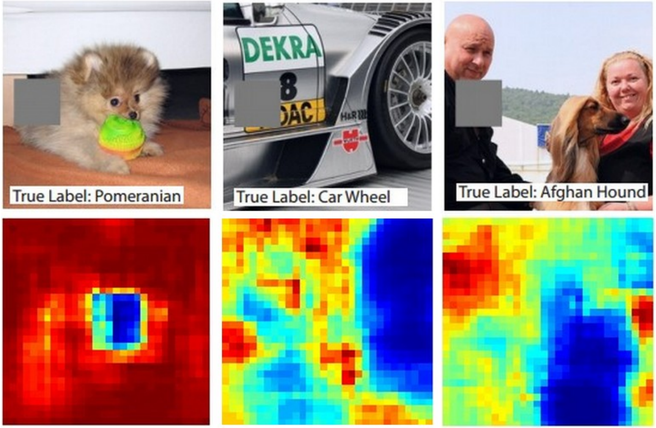


(Zeiler et al.) 12 feature channels in layer 3; for each channel, reconstruction of the 9 images that generate the strongest activation

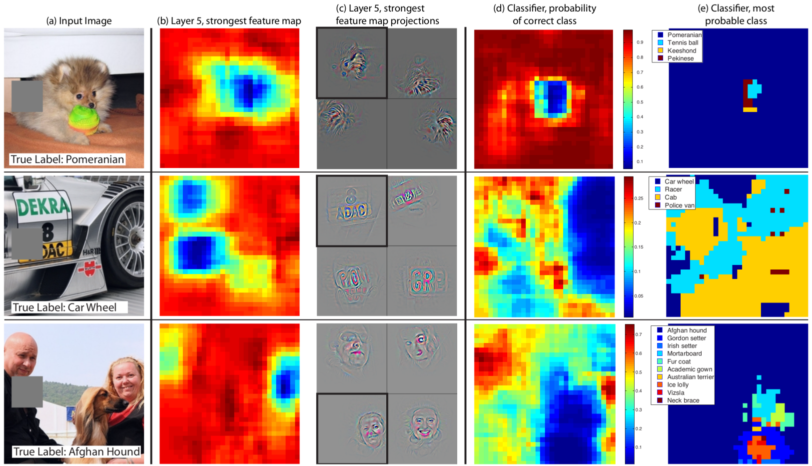


## **Occlusion sensitivity** ( arXiv:1311.2901, Zeiler et al., 2013)

Occlude small regions of input image and check the target class output score




Blue (most sensible; small score for correct the class); red (less sensible; high score for the correct class)




(b) mapa com a ativação do feature channel (originalmente de maior ativação), dependendo da região coberta

(c) reconstrução a partir desse feature channel, destacado em preto

(d) mapa prob. de classificação para a classe correta, em função da posição coberta na imagem de entrada

(e) mapa de classes mais prováveis, em função da posição coberta na imagem de entrada

## Mapa de saliência

(aparentemente chamado de métodos de *Attribution*)
(em contraste a *feature visualization* – mais adiante)

**Attribution:** Métodos que buscam identificar qual parte da imagem é reponsável pela ativação de uma unidade da rede

## **Saliency map** ( arXiv:1312.6034, Simonyan et al., 2014)

$S_c(I)$ score function of class $c$

(antes do softmax)

Saliency map of input image $I_0$:

$$M = \left| \left. \frac{\partial S_c}{\partial I} \right|_{I_0} \right|$$

( $S_c(I)$ is non-linear, but we can approximate it linearly around $I_0$ )

(basicamente, queremos identificar pixels na entrada que, quando perturbados, mais afetam $S_c$)

$c$: class $\quad x$: input image

**Vanilla saliency**

$$M_c(x) = \frac{\partial S_c(x)}{\partial x}$$

**SmoothGrad** (https://arxiv.org/abs/1706.03825, 2017)

$$\hat{M}_c(x) = \frac{1}{n}\sum^{n} M_c(x + \mathcal{N}(0, \sigma^2))$$

Sensitivity map: gradiente local oscila muito; ideia seria suavizar $S_c$ mas como isso não é trivial, a ideia consiste em adicionar perturbações na imagem e depois calcular a saliência média

# SmoothGrad (2017)

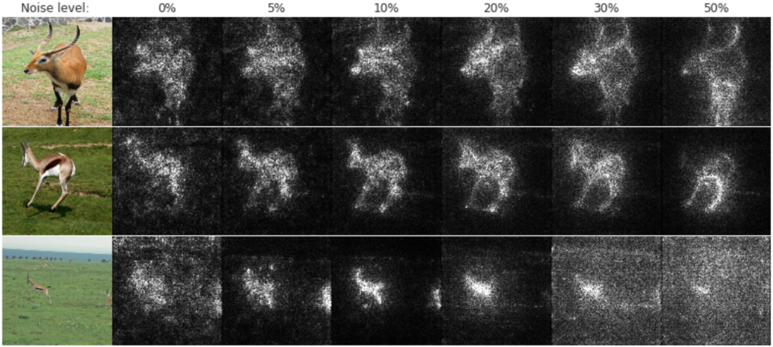


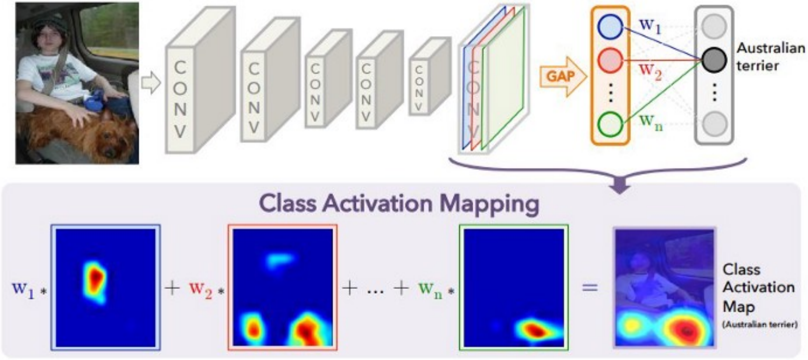

*Figure 3.* Effect of noise level (columns) on our method for 5 images of the gazelle class in ImageNet (rows). Each sensitivity map is obtained by applying Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the input pixels for 50 samples, and averaging them. The noise level corresponds to $\sigma/(x_{max} - x_{min})$.

## CAM – Class Activation Map

Não depende de gradientes

Identificar os mapas de feature que mais contribuem para a saída e ponderar os mesmos de acordo. Regiões mais ativas ficarão destacadas. Em seguida, redimensionar esse mapa ponderado para o tamanho da imagem de entrada e sobrepor a ela.

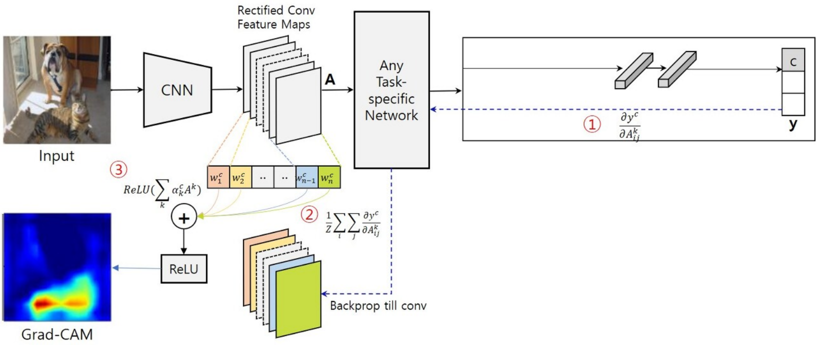
## **Class activation map** (CAM) – https://arxiv.org/abs/1512.04150

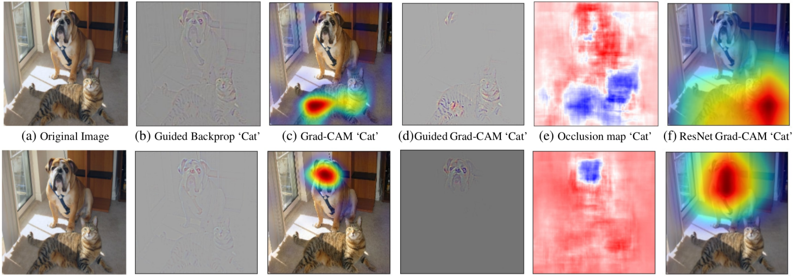



CAM requires a global pooling layer followed by the output layer

# GradCAM – https://arxiv.org/abs/1610.02391

For each feature map $A^k$ in the last convolutional layer, compute $\alpha_k^c = \frac{1}{Z}\sum_{i,j}\frac{\partial S_c(x)}{\partial A_{i,j}^k}$ and use this as weight $w_k$





(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat' (d)Guided Grad-CAM 'Cat' (e) Occlusion map 'Cat' (f) ResNet Grad-CAM 'Cat'

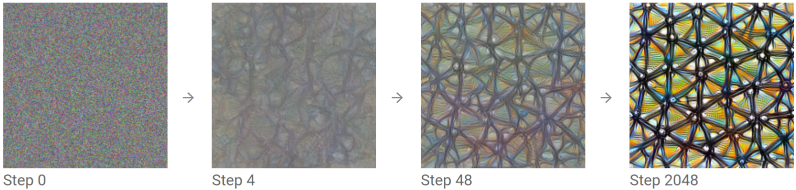
(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog' (j)Guided Grad-CAM 'Dog' (k) Occlusion map 'Dog' (l)ResNet Grad-CAM 'Dog'

Fig. 1: (a) Original image with a cat and a dog. (b-f) Support for the cat category according to various visualizations for VGG-16 and ResNet. (b) Guided Backpropagation [53]: highlights all contributing features. (c, f) Grad-CAM (Ours): localizes class-discriminative regions, (d) Combining (b) and (c) gives Guided Grad-CAM, which gives high-resolution class-discriminative visualizations. Interestingly, the localizations achieved by our Grad-CAM technique, (c) are very similar to results from occlusion sensitivity (e), while being orders of magnitude cheaper to compute. (f, l) are Grad-CAM visualizations for ResNet-18 layer. Note that in (c, f, i, l), red regions corresponds to high score for class, while in (e, k), blue corresponds to evidence for the class. Figure best viewed in color.

# Feature visualization

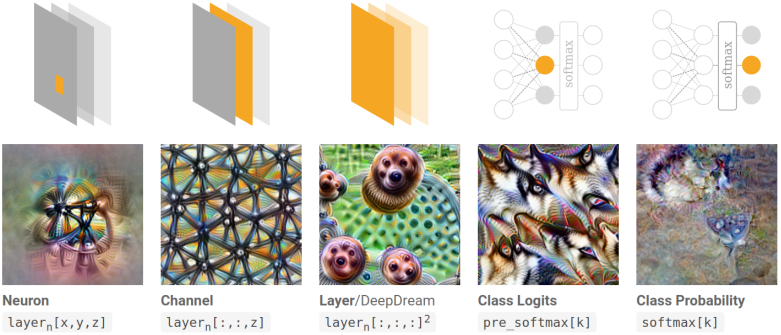
*Feature Visualization: How neural networks build up their understanding of images*
Chris Olah, Alexander Mordvintsev,Ludwig Schubert

https://distill.pub/2017/feature-visualization/ (2017)




Starting from random noise, we optimize an image to activate a particular neuron

# Feature visualization




Podemos otimizar a imagem para ativar qualquer outra "unidade"

Formalmente, seja um neurônio $h$, uma imagem de entrada $img$, as coordenadas $x$ e $y$ do neurônio, a camada $n$ e o canal $z$ do neurônio

Imagem que maximiza a ativação de $h$

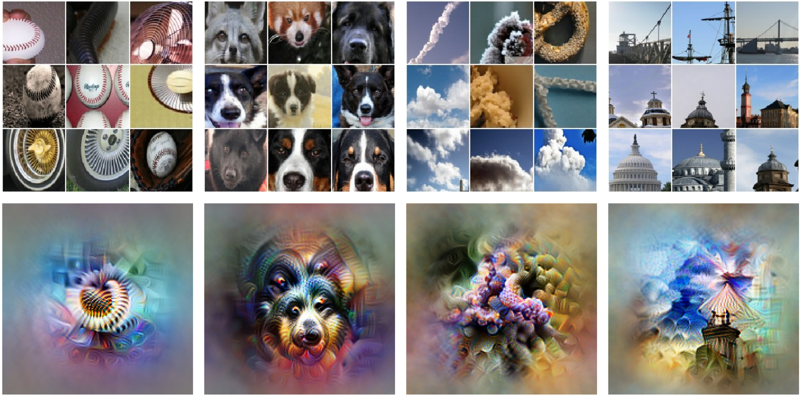
$$img^* = \arg\max_{img} h_{n,x,y,z}(img)$$

Imagem que maximiza a ativação média do canal $z$ na camada $n$:

$$img^* = \arg\max_{img} \sum_{x,y} h_{n,x,y,z}(img)$$

Top: dentre as disponíveis, imagens que maximizam a ativação

Bottom: imagem gerada de tal forma a otimizar a ativação



Baseball—or stripes?
*mixed4a, Unit 6*

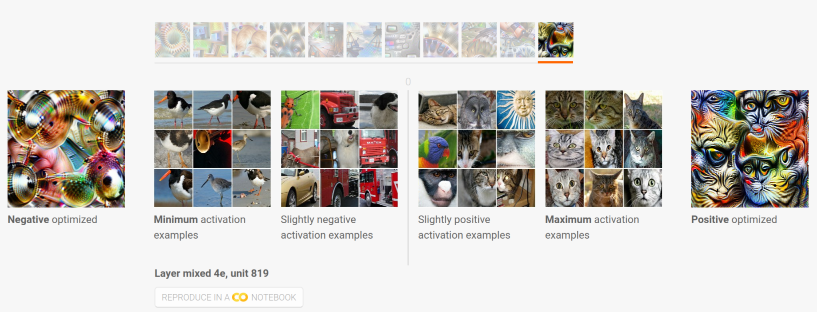
Animal faces—or snouts?
*mixed4a, Unit 240*

Clouds—or fluffiness?
*mixed4a, Unit 453*

Buildings—or sky?
*mixed4a, Unit 492*

Via otimização: apenas mínima e máxima ativação

Via dados: agrupar considerando espectro entre mínima e máxima ativação



## Via otimização: pode-se alcançar diversidade usando-se um termo de diversidade na função a ser otimizada

Simple Optimization

Optimization with diversity reveals four different, curvy facets. *Layer mixed4a, Unit 97*

Dataset examples

Simple Optimization

Optimization with diversity reveals multiple types of balls. *Layer mixed5a, Unit 9*

Dataset examples

## Gradient ascent

Given a trained network, keep its weights fixed, and iteratively update the input (noise) image through backpropagation so as to maximize

$$S_c(I) - \lambda ||I||_2^2$$

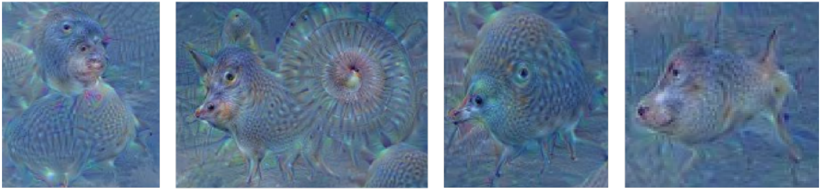
The regularization term is there to keep some smoothness.

goose

Deep dream (Inceptionism: Going Deeper into Neural Networks)

https://blog.research.google/2015/06/inceptionism-going-deeper-into-neural.html

Passar uma imagem pela rede treinada e depois alterar a imagem de forma a otimizar alguma ativação específica



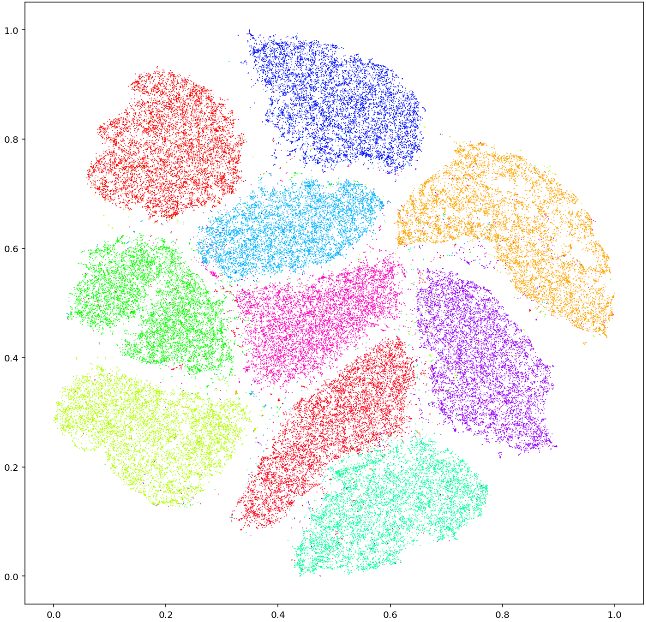
"Admiral Dog!" "The Pig-Snail" "The Camel-Bird" "The Dog-Fish"

## Visualização de dados/embeddings

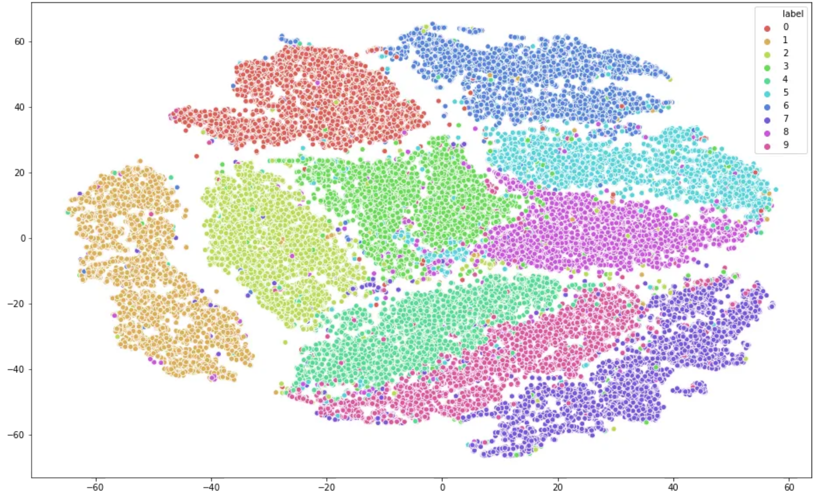
*Stochastic Neighbor Embedding* (t-SNE), 2002
Geoffrey Hinton and Sam Roweis

https://papers.nips.cc/paper_files/paper/2002/hash/6150ccc6069bea6b5716254057a194ef-Abstract.html

# t-SNE embedding do MNIST (Wikipedia)



# 2D Scatter plot of MNIST data after applying PCA (n_components = 50) and then t-SNE




https://towardsdatascience.com/dimensionality-reduction-using-t-distributed-stochastic-neighbor-embedding-t-sne-on-t

## Projeções das *features*

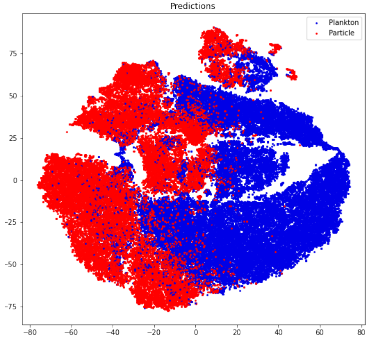

*Features* convolucionais

*Features* morfométricos

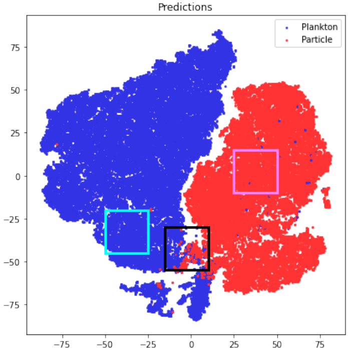
## Exemplo

Separação plankton $\times$ detrito

# Qualitative analysis (D2)



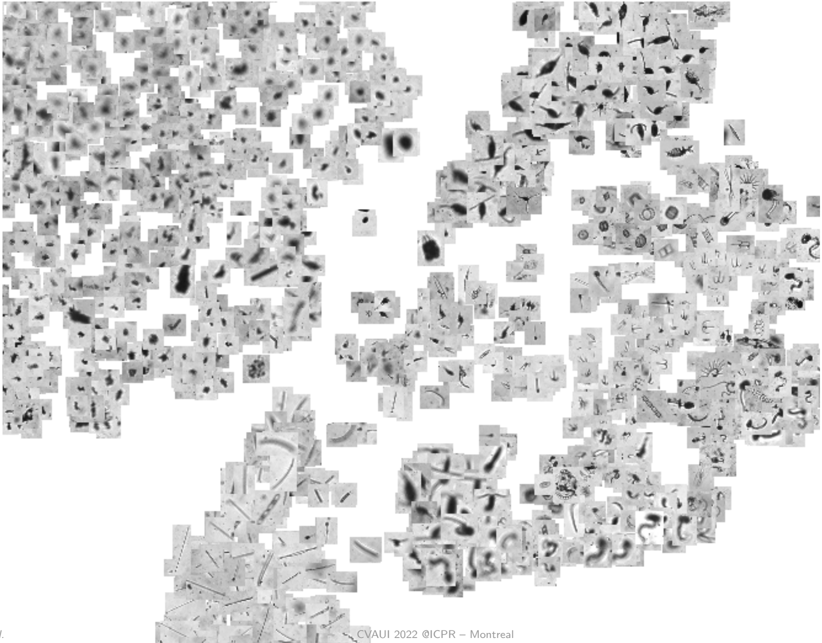

MobileNetV2

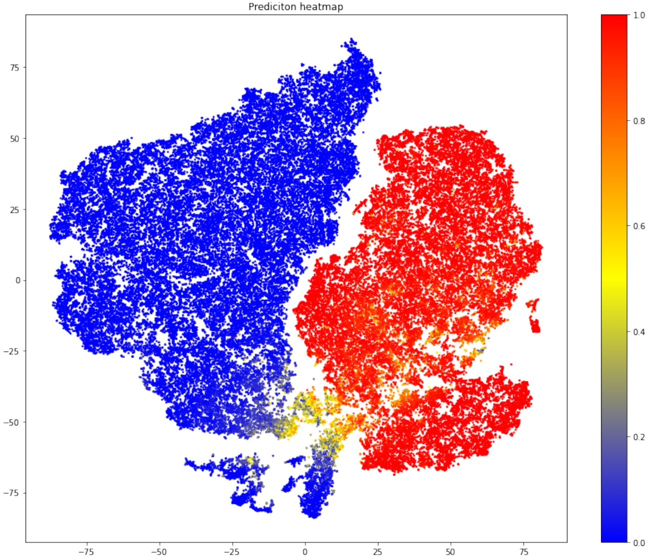



InceptionResNetV2

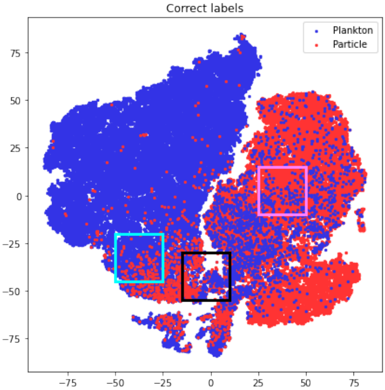
| Network | Training set | Training size | Test size | Batch size | Test Accuracy |
|---|---|---|---|---|---|
| MobileNetV2 | Full | 597,908 | 105,480 | 32 | 95.04% |
| InceptionResNetV2 | 10% | 59,819 | 105,480 | 64 | 95.93% |

# Why t-SNE projections?

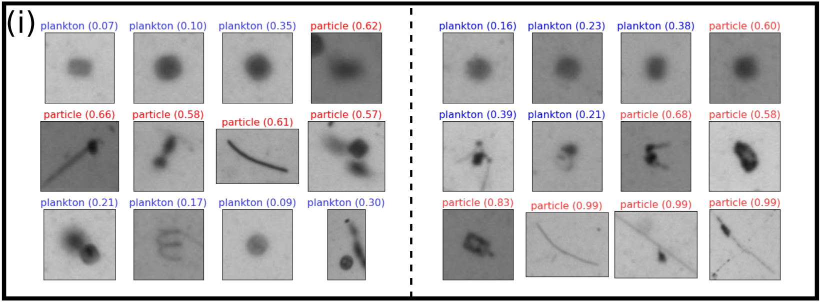



Prediciton heatmap
75
50
25
0
−25
−50
−75
−75
−50
−25
0
25
50
75
1.0
0.8
0.6
0.4
0.2
0.0

# Further analysis of D2

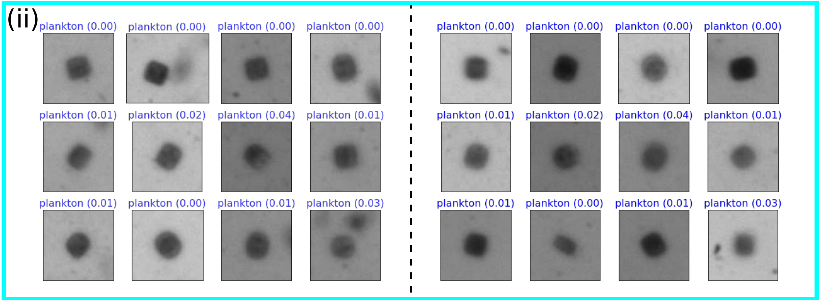


# (black) Region in the classification frontier



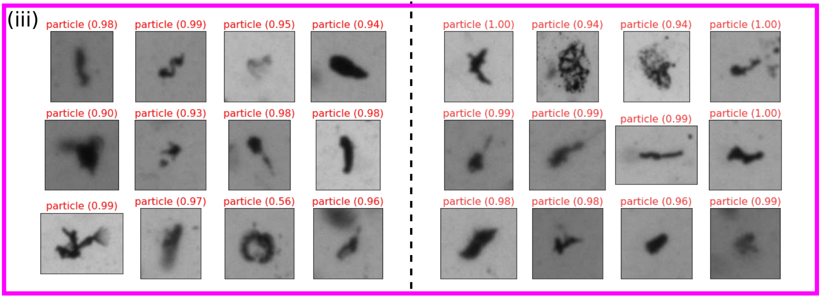

According to user label, left is Plankton and right is particle

# (cyan) Plankton region, according to machine




According to user label, left is Plankton and right is particle

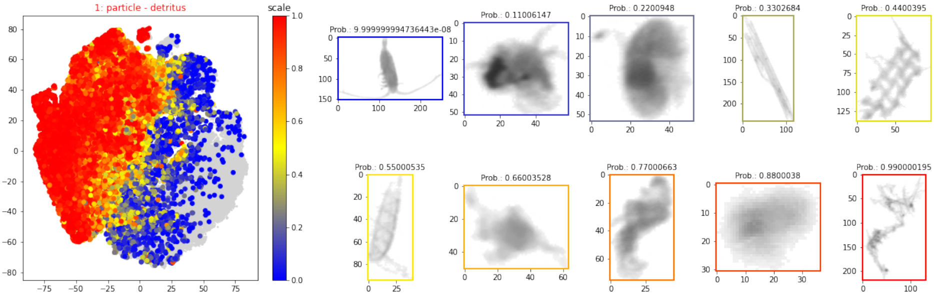
# (magenta) Particle region, according to machine




According to user label, left is Plankton and right is particle

# Another visual analysis (ZooScan)

Random selection of objects in the particle class (according to human label), with scores varying from 0 (plankton) to 1 (particle)




Blue: machine is attributing higher score to the plankton class

Red: machine is attributing higher score to the particle class

## XAI

- Feature importance
- SHAP
- LIME
- Layer-wise relevance propagation (LRP)