Explanations of ARA network

Weronika Hryniewska
w.hryniewska@mini.pw.edu.pl
20/02/2020

Politechnika Warszawska

MI DATA LAB

Presentation outline

1) Importance of XAI (Explainable AI)
2) Importance of XAI in ARA network
3) Explanation methods
    a) LIME
    b) Visualizing intermediate activations
    c) Other solutions
4) Summary

Importance of Explainable AI

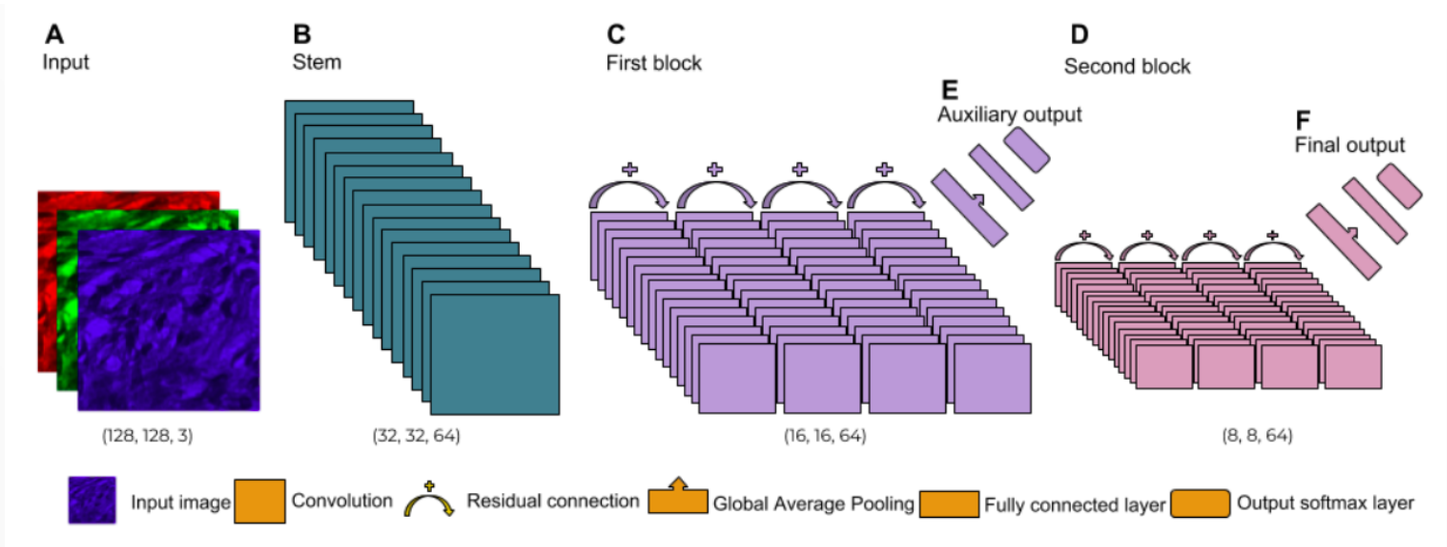
- understand the decision-making mechanism to ascertain that the AI makes accurate and fair decisions (self-driving, credit card approval)
- end up with many applications that produce outcomes for the wrong reasons (human bias, ethics)
- produce decisions transparent for users
- determine the limits of the model

ARA

ARA: accurate, reliable and active histopathological image classification framework with Bayesian deep learning

Łukasz Rączkowski[1], Marcin Możejko[1], Joanna Zambonelli[2], and Ewa Szczurek[1,*]

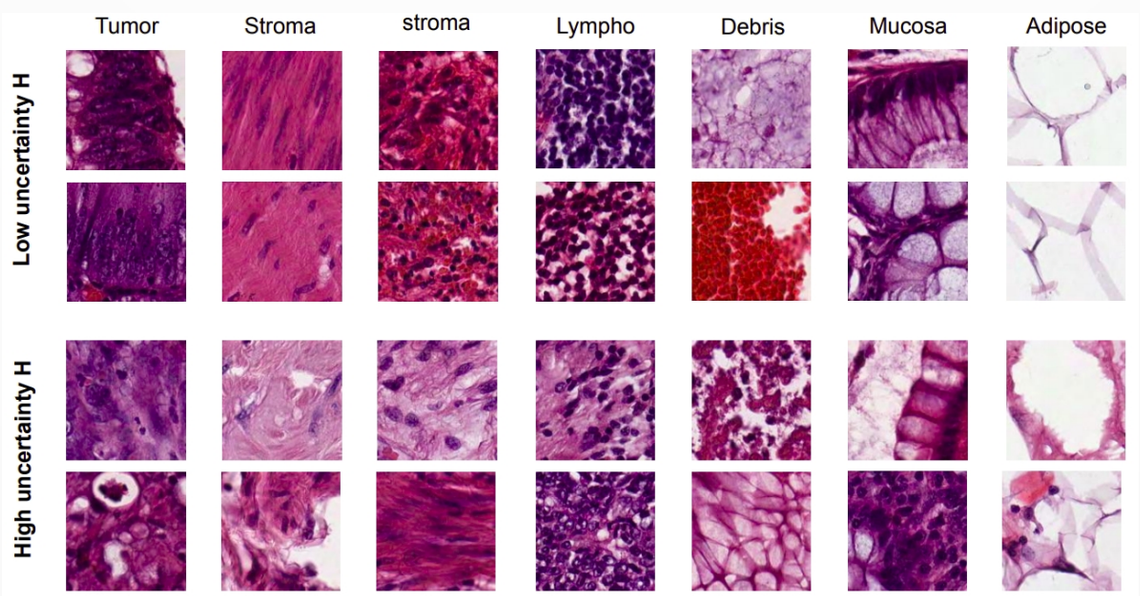
[1]Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland
[2]Department of Pathology, Medical University of Warsaw, Warsaw, Poland
[*]szczurek@mimuw.edu.pl



Ł. Rączkowski, M. Możejko, J. Zambonelli, and E. Szczurek, "ARA: accurate, reliable and active histopathological image classification framework with Bayesian deep learning," bioRxiv, p. 658138, Jun. 2019, doi: 10.1101/658138.

Importance of explaining ARA model

- understand decision process – which image features, patterns take part in decision process
- understand reasons of model uncertainty and wrong classifications
- eliminate possibility of bias



Ł. Rączkowski, M. Możejko, J. Zambonelli, and E. Szczurek, “ARA: accurate, reliable and active histopathological image classification framework with Bayesian deep learning,” bioRxiv, p. 658138, Jun. 2019, doi: 10.1101/658138.

My ARA model

```
from keras import layers
from keras import models

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
                        input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(8, activation='softmax'))
```

class name	ACC
01_TUMOR	64.23%
02_STROMA	35.05%
03_COMPLEX	71.61%
04_LYMPHO	98.49%
05_DEBRIS	88.94%
06_MUCOSA	74.50%
07_ADIPOSE	84.05%
08_EMPTY	100.00%
	76.45%

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               3211776
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 4104
=================================================================
Total params: 3,456,712
Trainable params: 3,456,712
Non-trainable params: 0
```

LIME

Times Cited: 483
(from Web of Science Core Collection)

★ Star 7.1k

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "'Why Should I Trust You?': Explaining the Predictions of Any Classifier." ArXiv:1602.04938 [Cs, Stat], August 9, 2016. http://arxiv.org/abs/1602.04938.

Divide image into superpixels

- Use representation that is understandable to humans, regardless of the actual features
- A contiguous patch of similar pixels (a super-pixel)

Original Image

Interpretable Components

M. T. R. Guestrin Sameer Singh, Carlos, “Local Interpretable Model-Agnostic Explanations (LIME): An Introduction,” O’Reilly Media, 12-Aug-2016. [Online]. Available: https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/. [Accessed: 21-Feb-2020].

Fidelity-interpretability trade-off

- The explanation produced by LIME is obtained by the following:

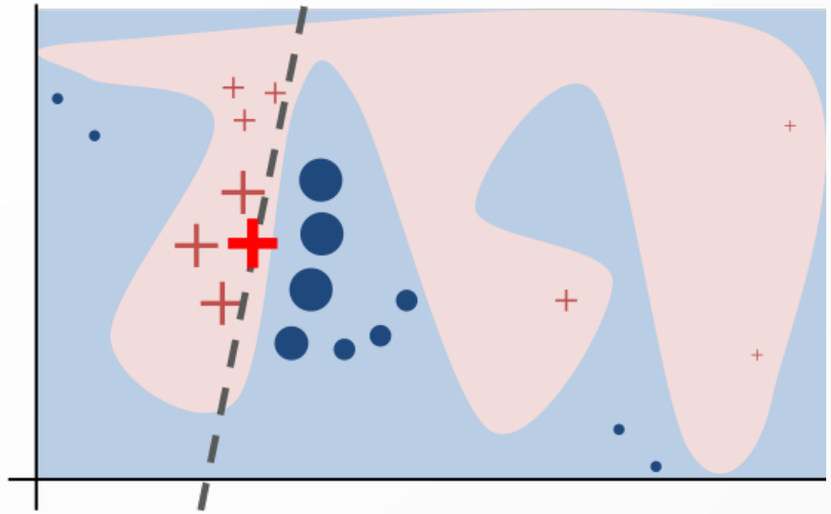
$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \; \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

G is a class of potentially interpretable models, i.e. a model g ∈ G can be readily presented to the user with visual or textual artifacts

a measure of how unfaithful *g* is in approximating *f* in the locality defined by π_x

a measure of complexity

The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background.
The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weights them by the proximity to the instance being explained (represented here by size). The dashed line is the learned.



M. T. R. Guestrin Sameer Singh, Carlos, "Local Interpretable Model-Agnostic Explanations (LIME): An Introduction," O'Reilly Media, 12-Aug-2016. [Online]. Available: https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/. [Accessed: 21-Feb-2020].

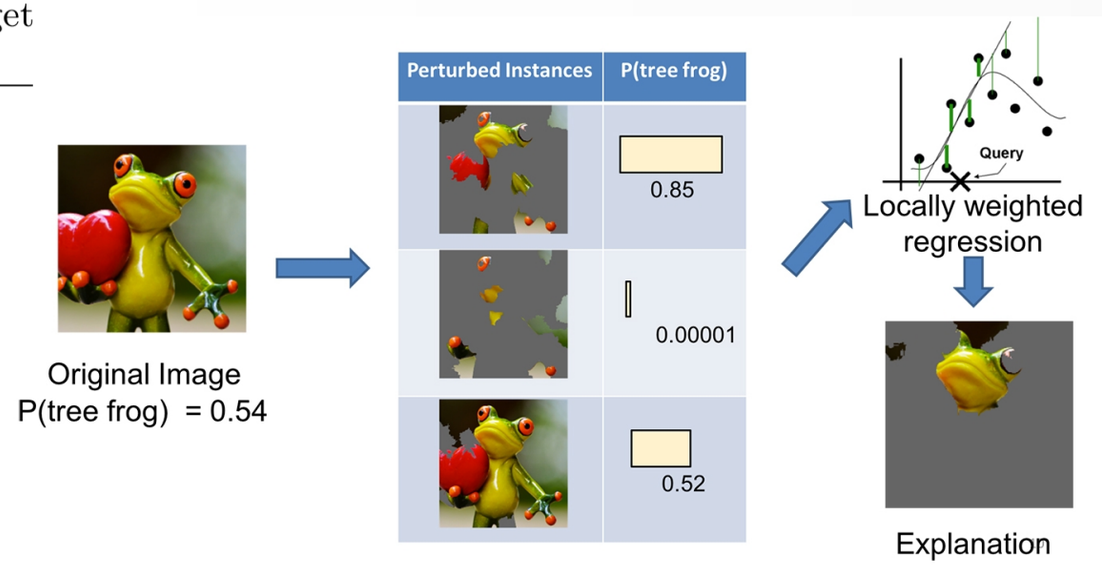
Sparse linear explanations

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

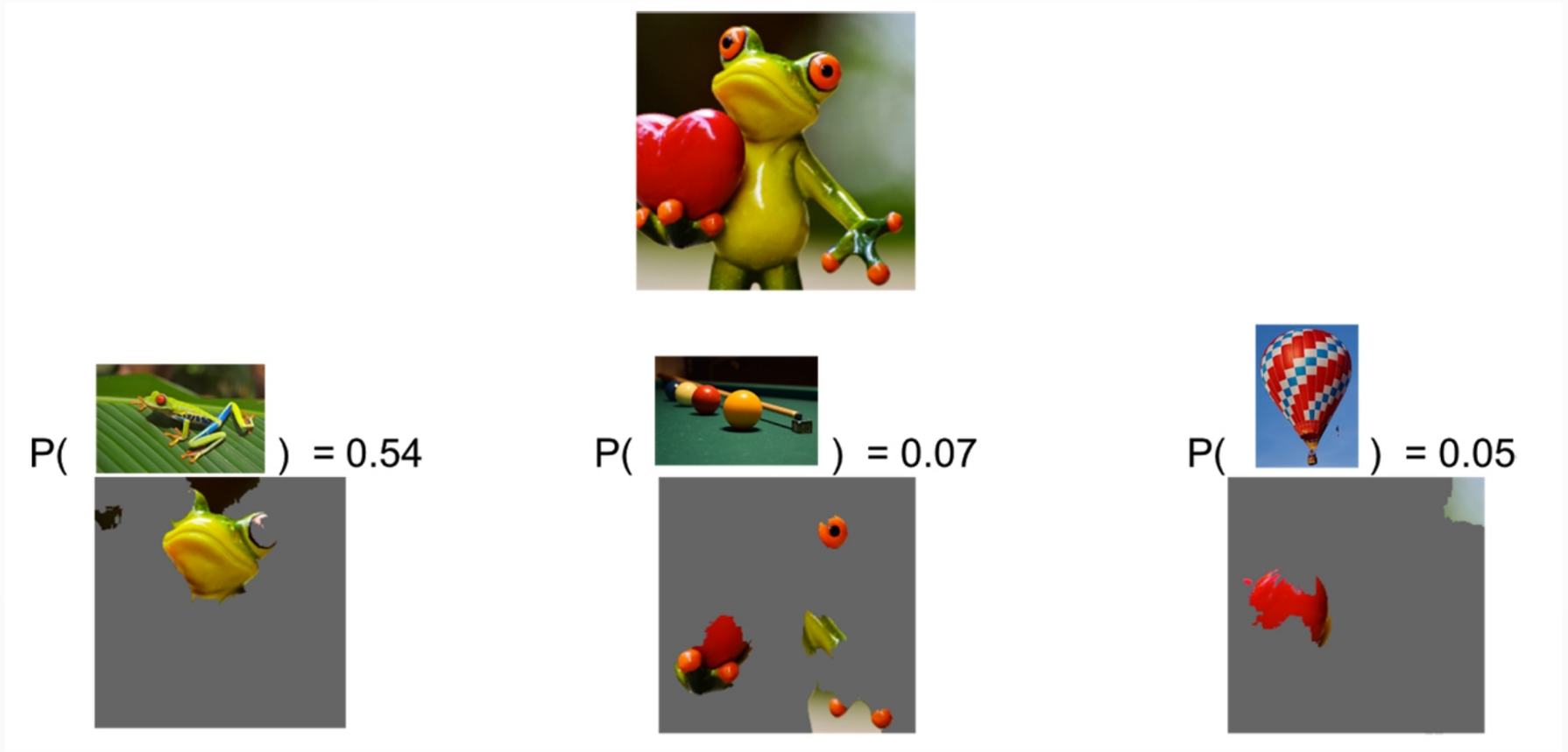
$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w

Directly solving previous equation is intractable, so Algorithm 1 is an approximation, which first selects K features with Lasso and then learns the weights via least squares.



M. T. R. Guestrin Sameer Singh, Carlos, "Local Interpretable Model-Agnostic Explanations (LIME): An Introduction," O'Reilly Media, 12-Aug-2016. [Online]. Available: https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/. [Accessed: 21-Feb-2020].

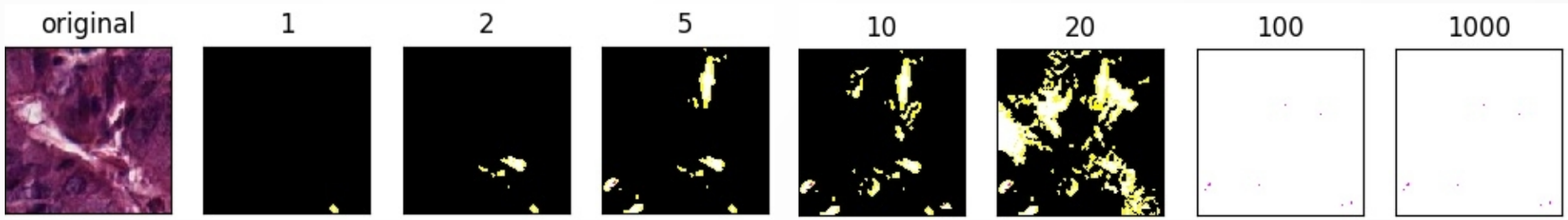
The top predicted classes are tree frog, pool table and balloon.



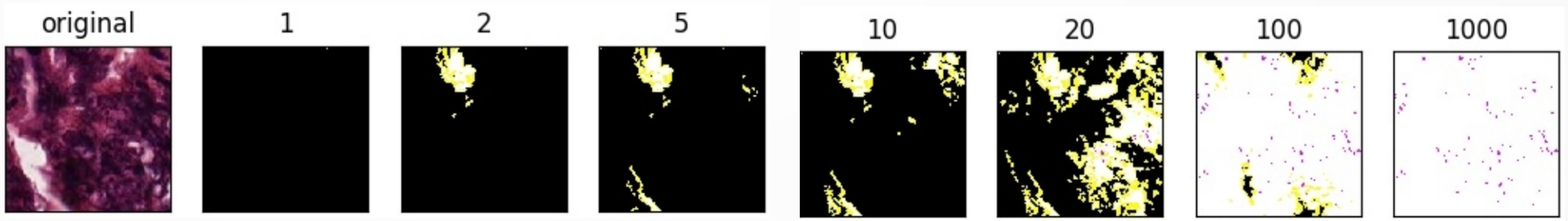
M. T. R. Guestrin Sameer Singh, Carlos, “Local Interpretable Model-Agnostic Explanations (LIME): An Introduction,” O’Reilly Media, 12-Aug-2016. [Online]. Available: https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/. [Accessed: 21-Feb-2020].

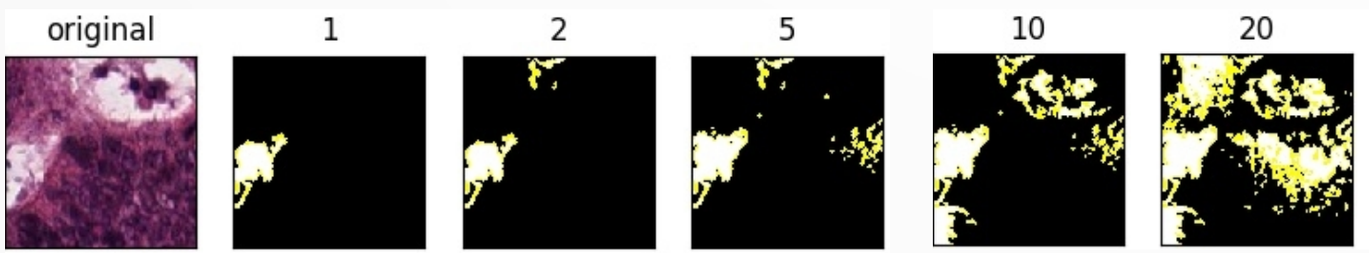
01_Tumor

02_Stroma

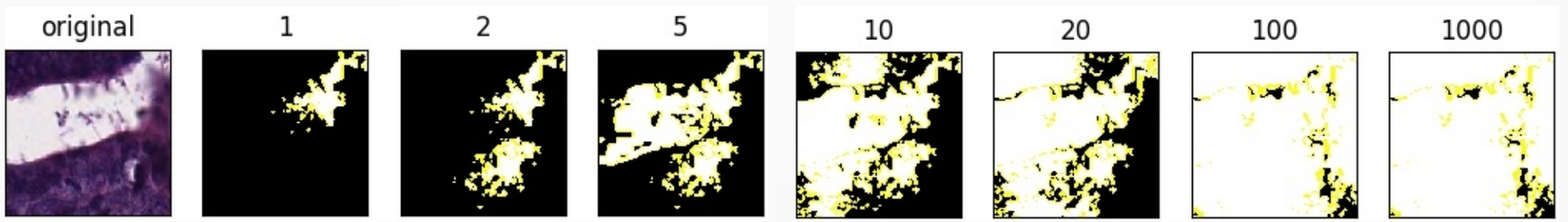


03_Complex



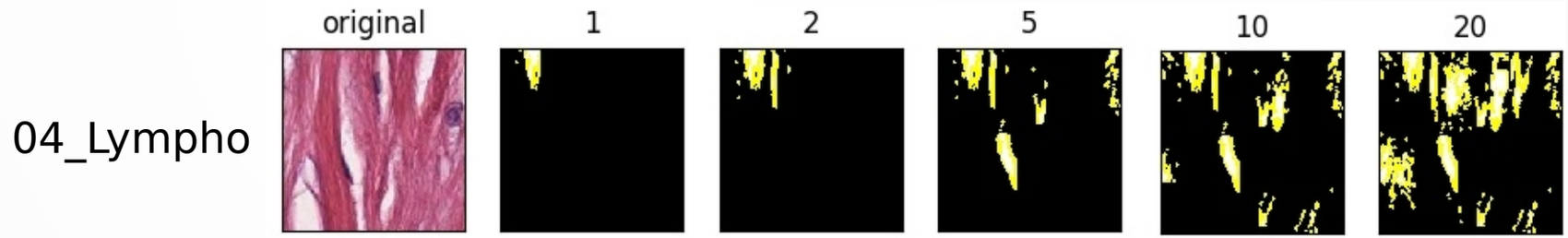


01_Tumor

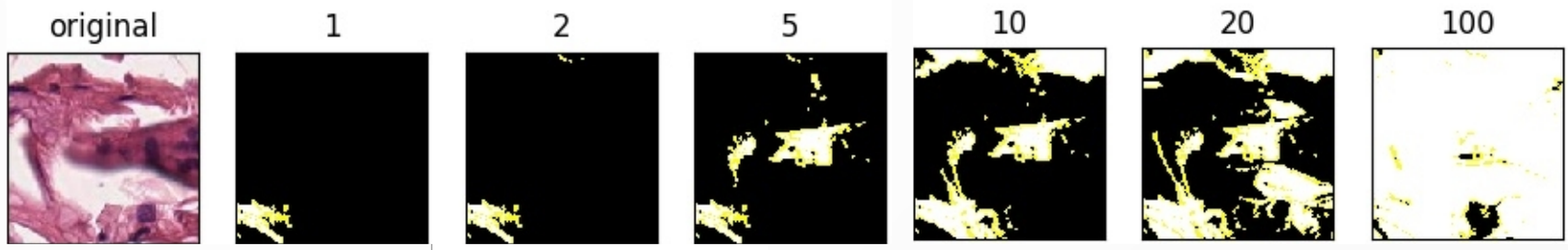


01_Tumor

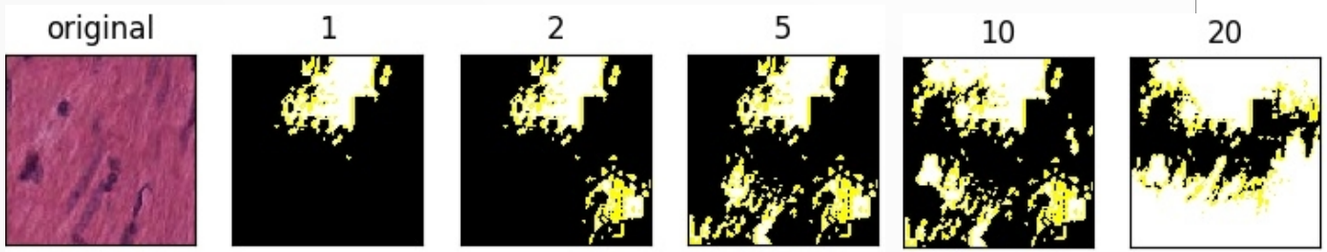
02_Stroma



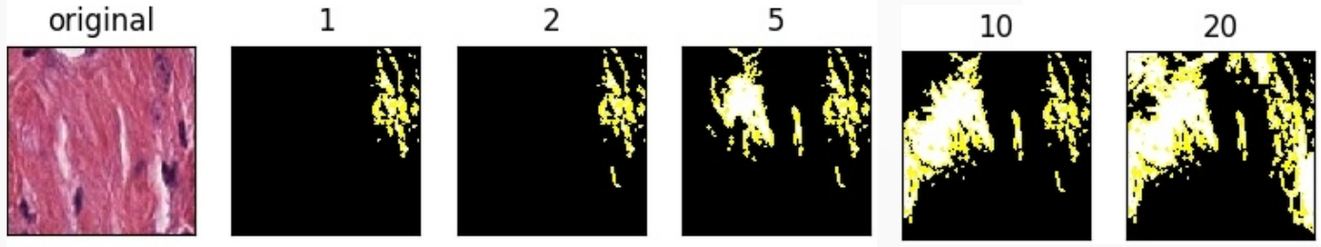
04_Lympho



04_Lympho



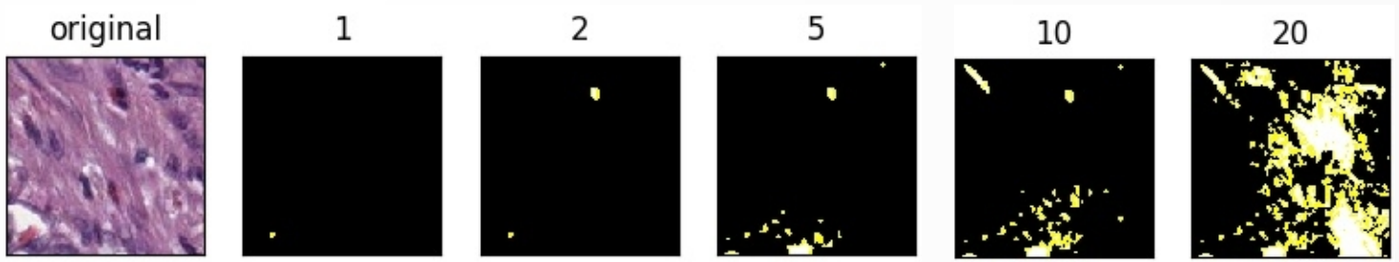
02_Stroma



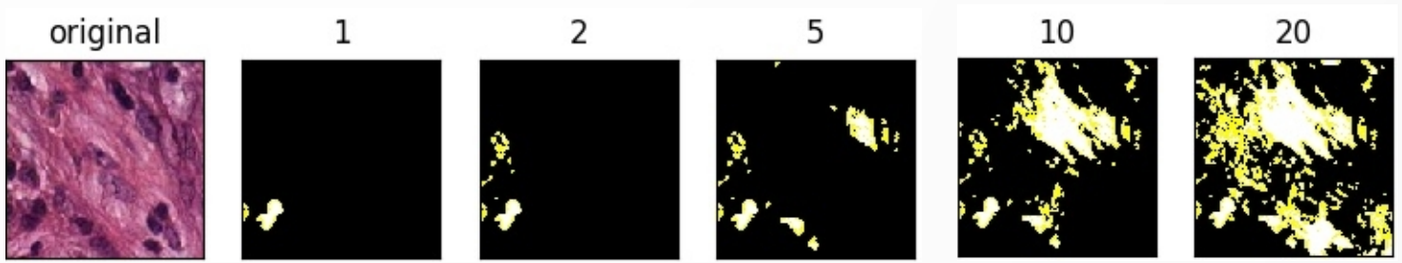
02_Stroma

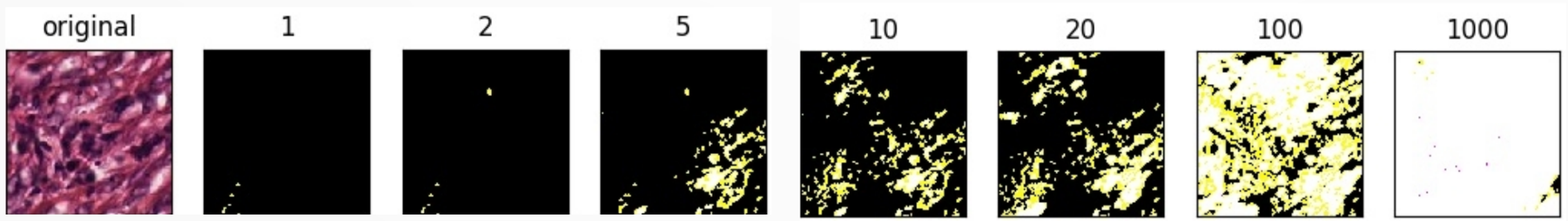
03_Complex

05_Debris

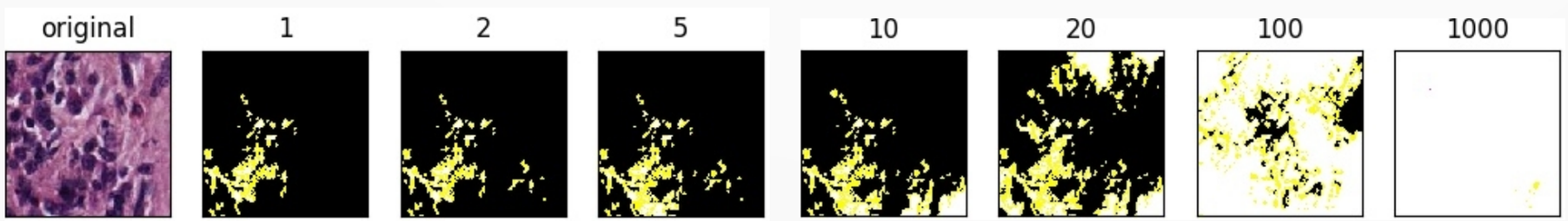


02_Stroma



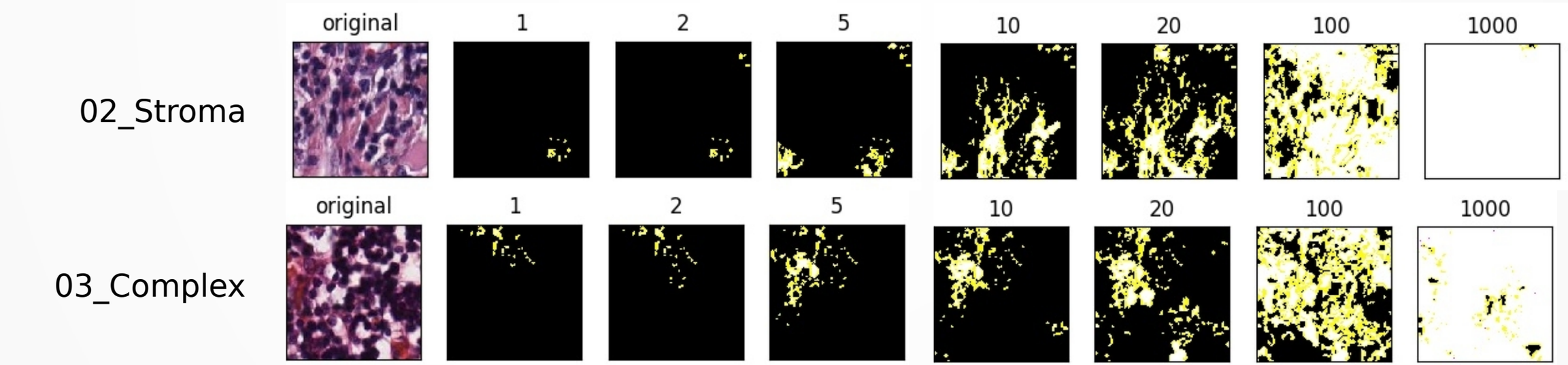


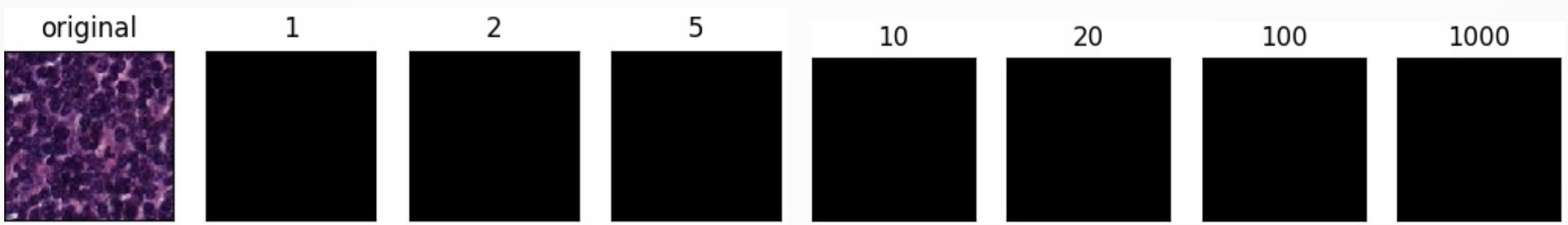
03_Complex

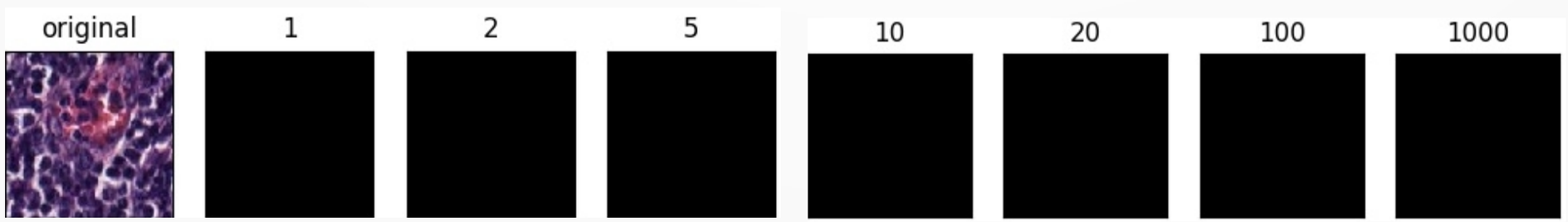


03_Complex

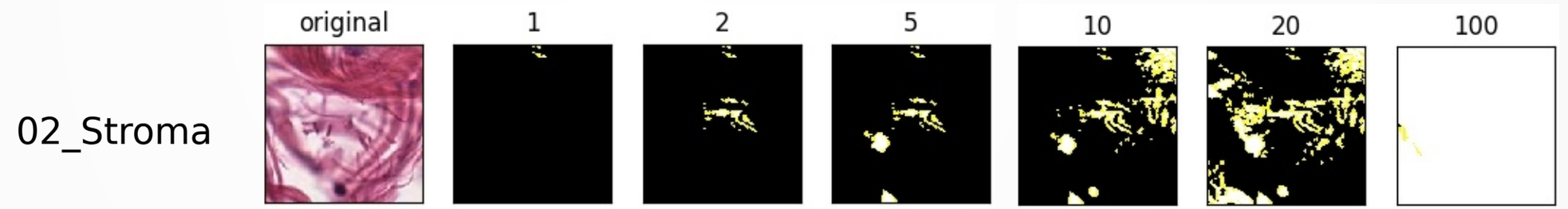
04_Lympho


original
1
2
5
10
20
100
1000
02_Stroma
original
1
2
5
10
20
100
1000
03_Complex

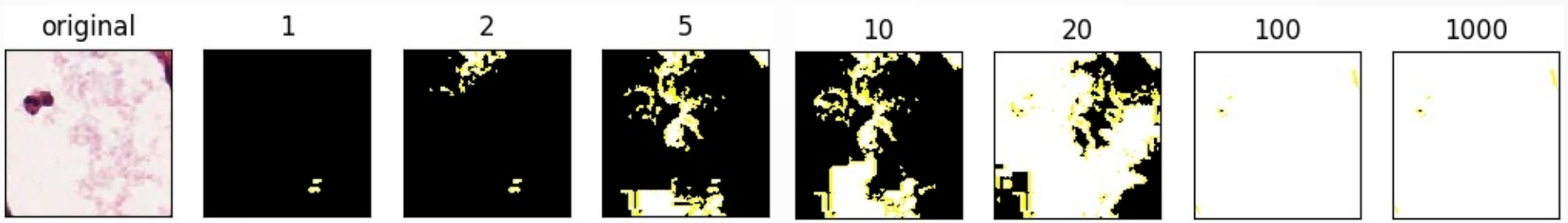

original
1
2
5
10
20
100
1000
04_Lympho


original
1
2
5
10
20
100
1000
04_Lympho

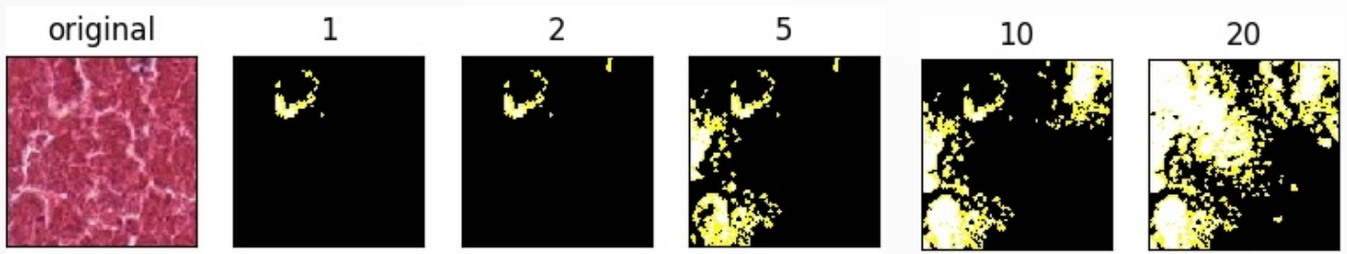
05_Debris



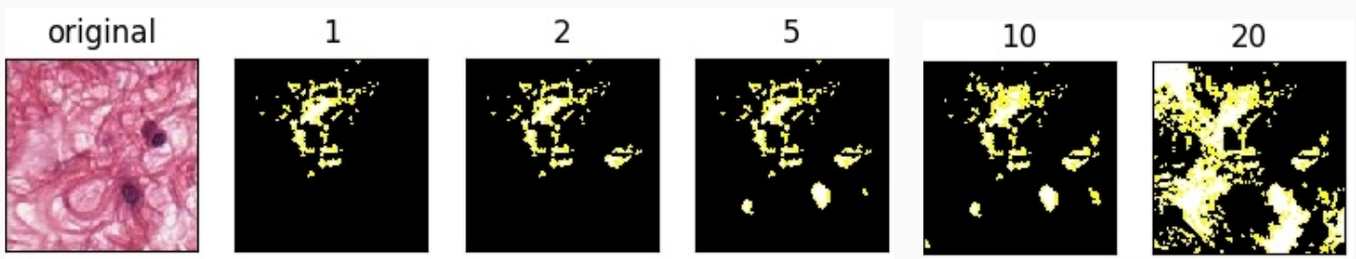
02_Stroma



07_Adipose



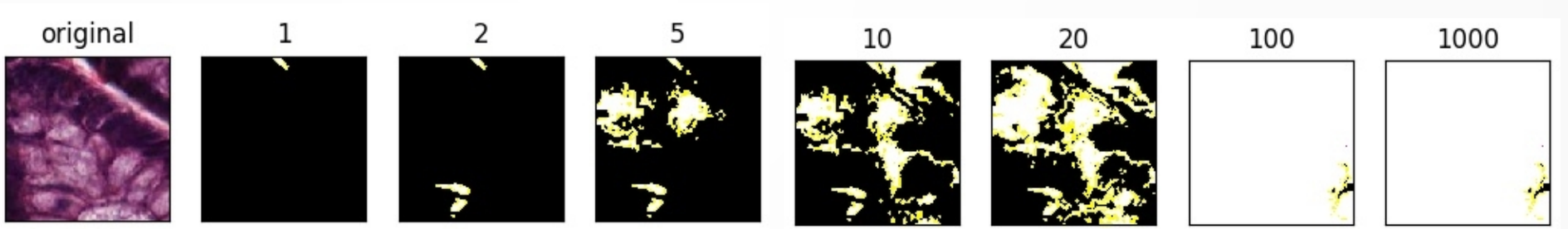
05_Debris



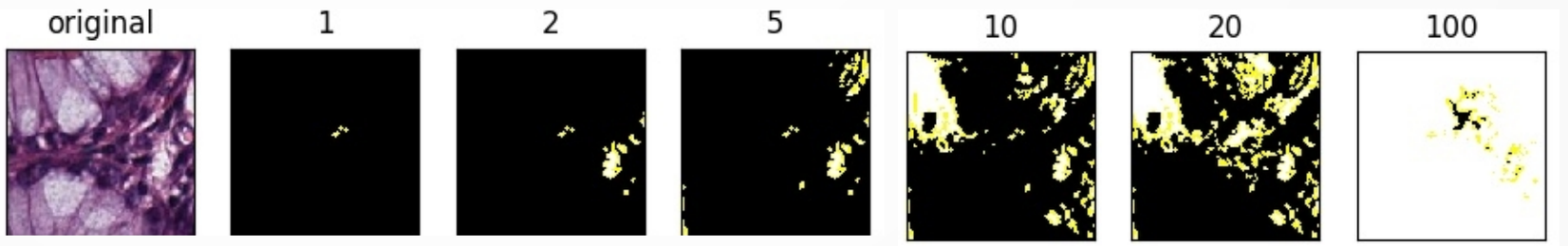
05_Debris

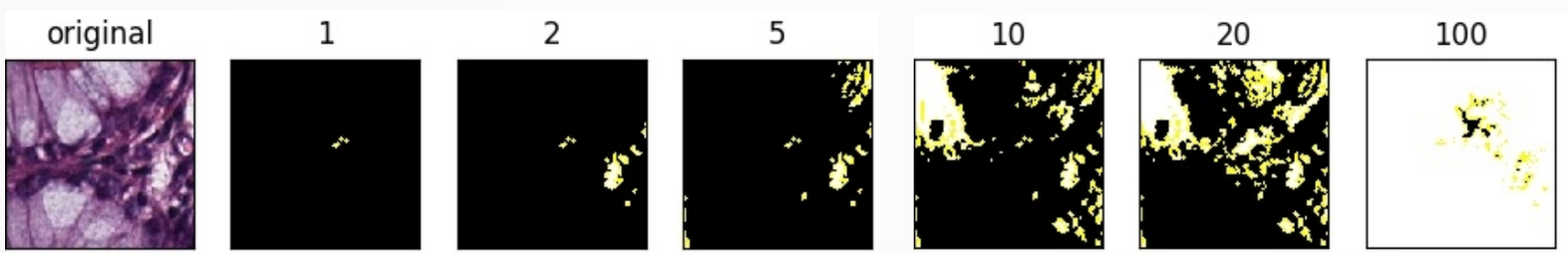
06_Musoca

02_Stroma

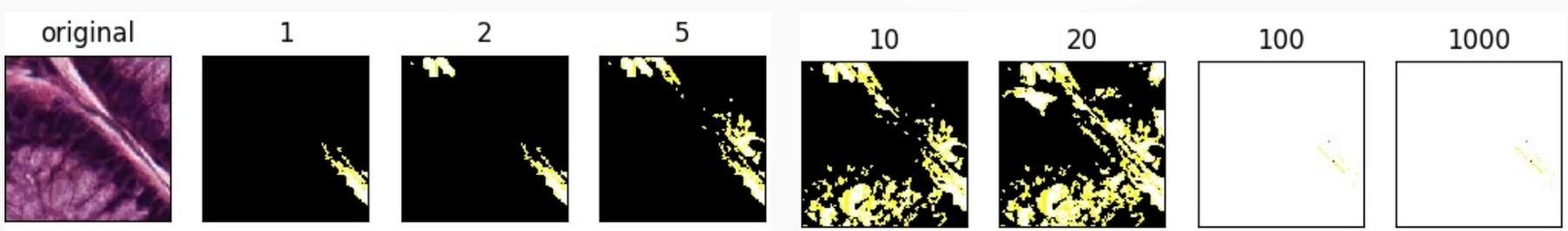


03_Complex





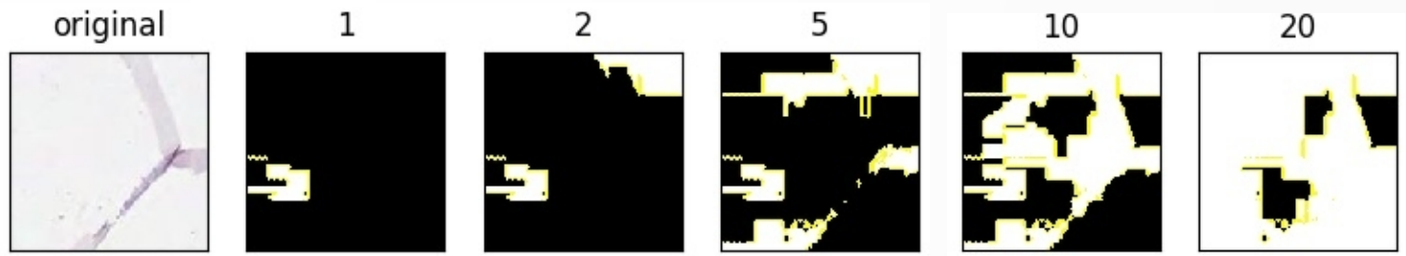
06_Mucosa



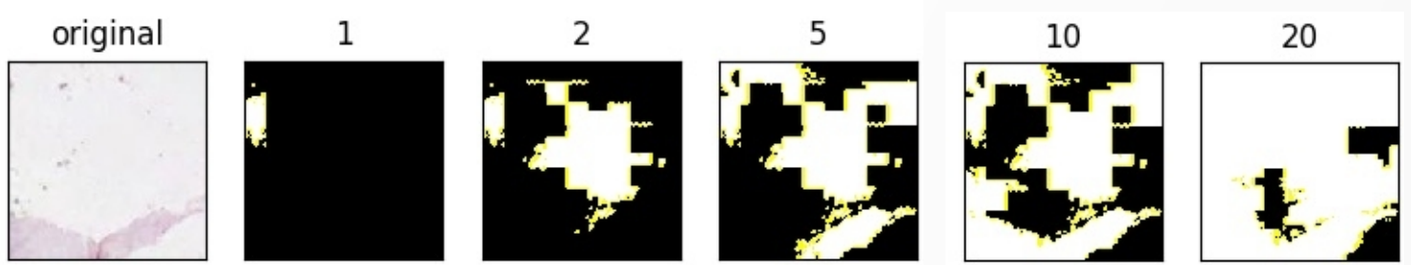
06_Mucosa

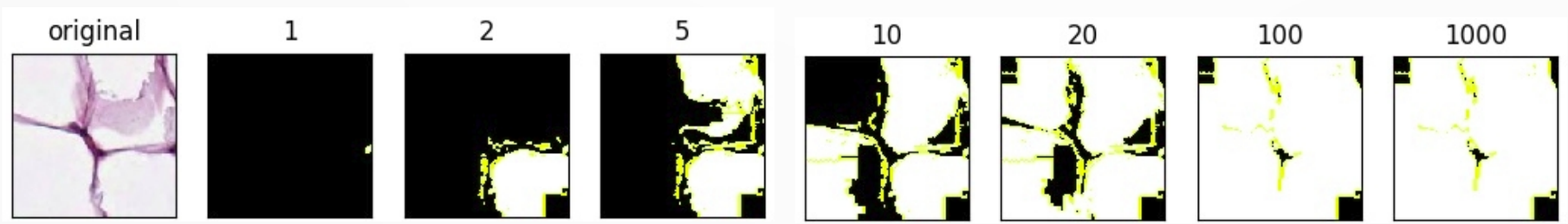
07_Adipose

08_Empty

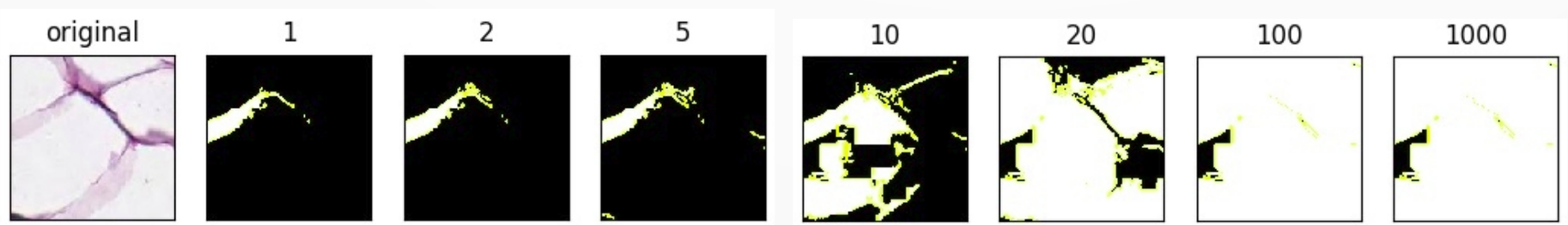


08_Empty



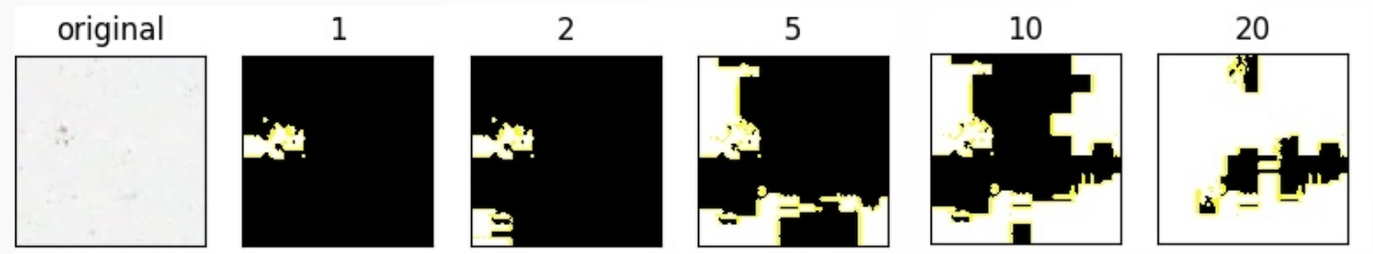


07_Adipose

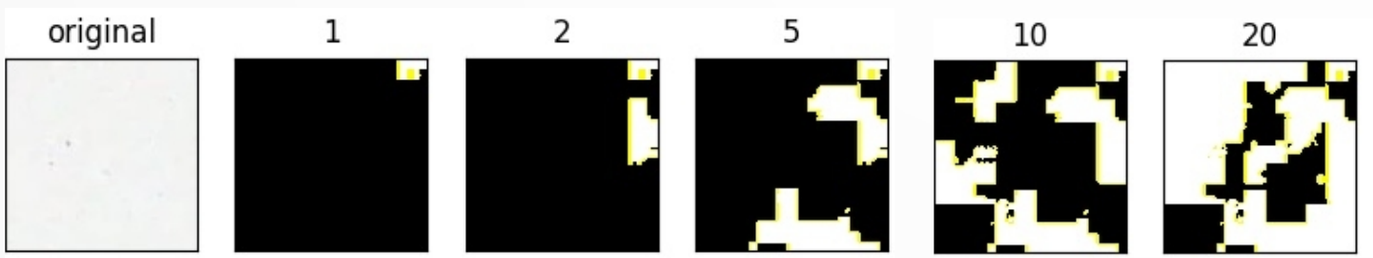


07_Adipose

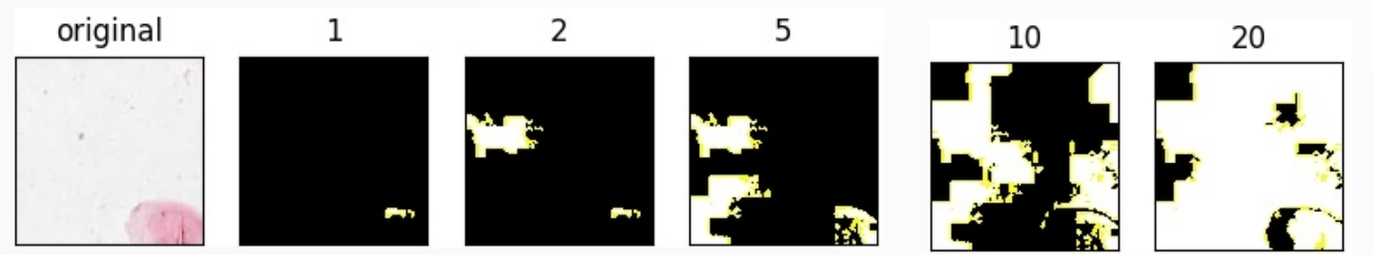
08_Empty



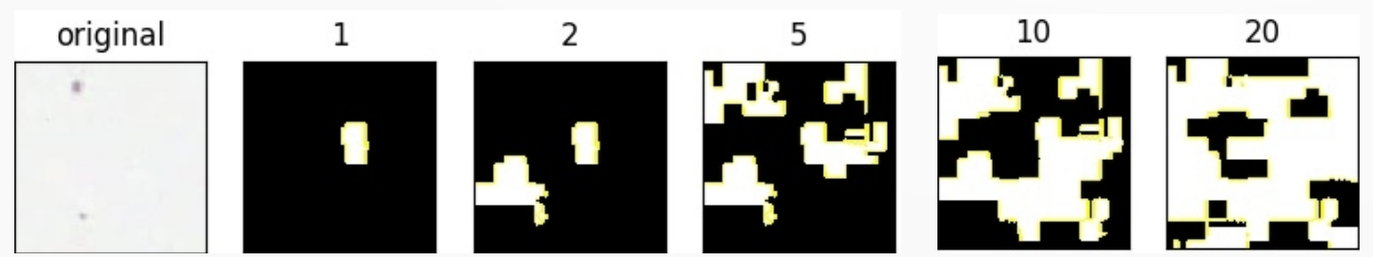
08_Empty



08_Empty



08_Empty

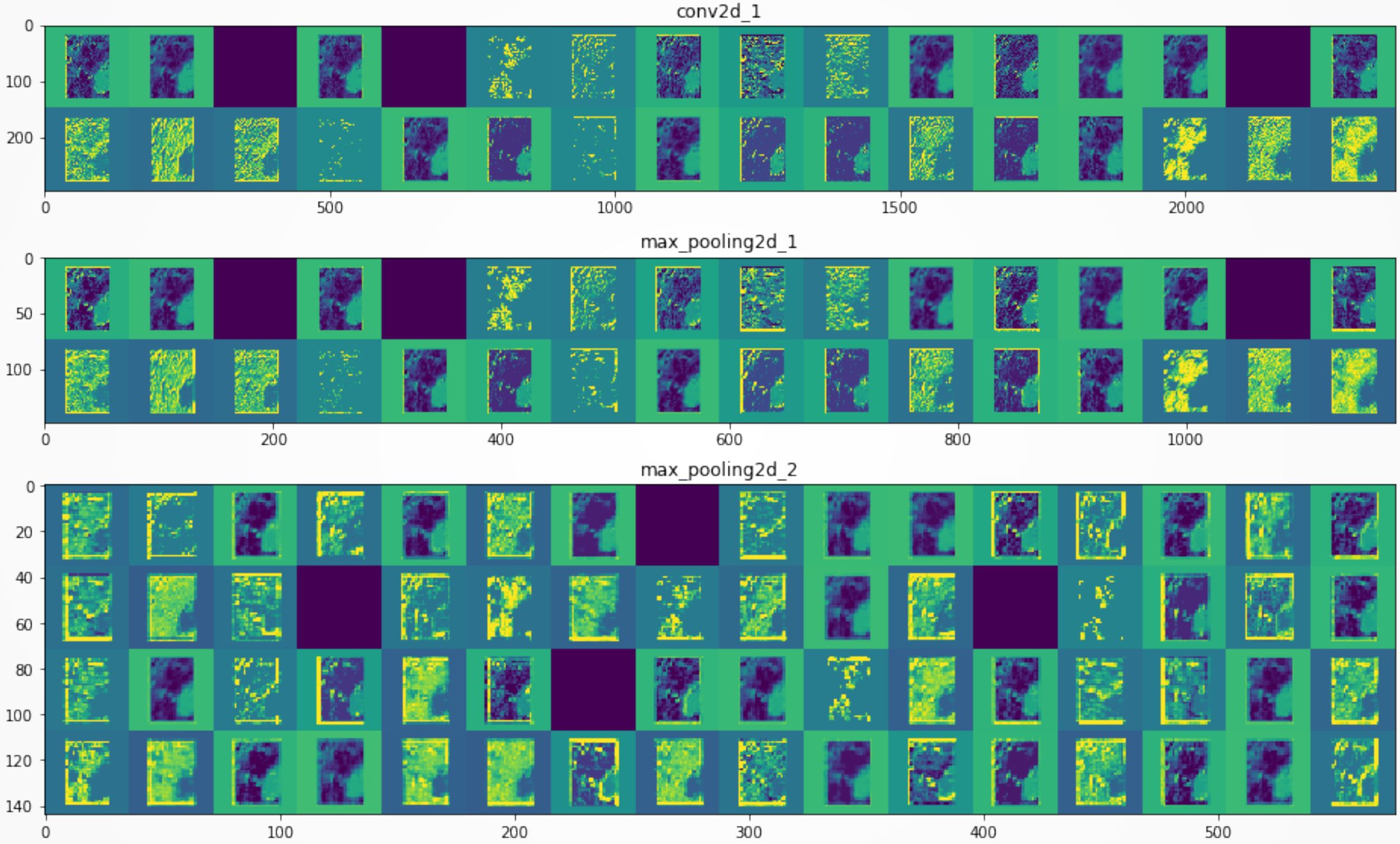


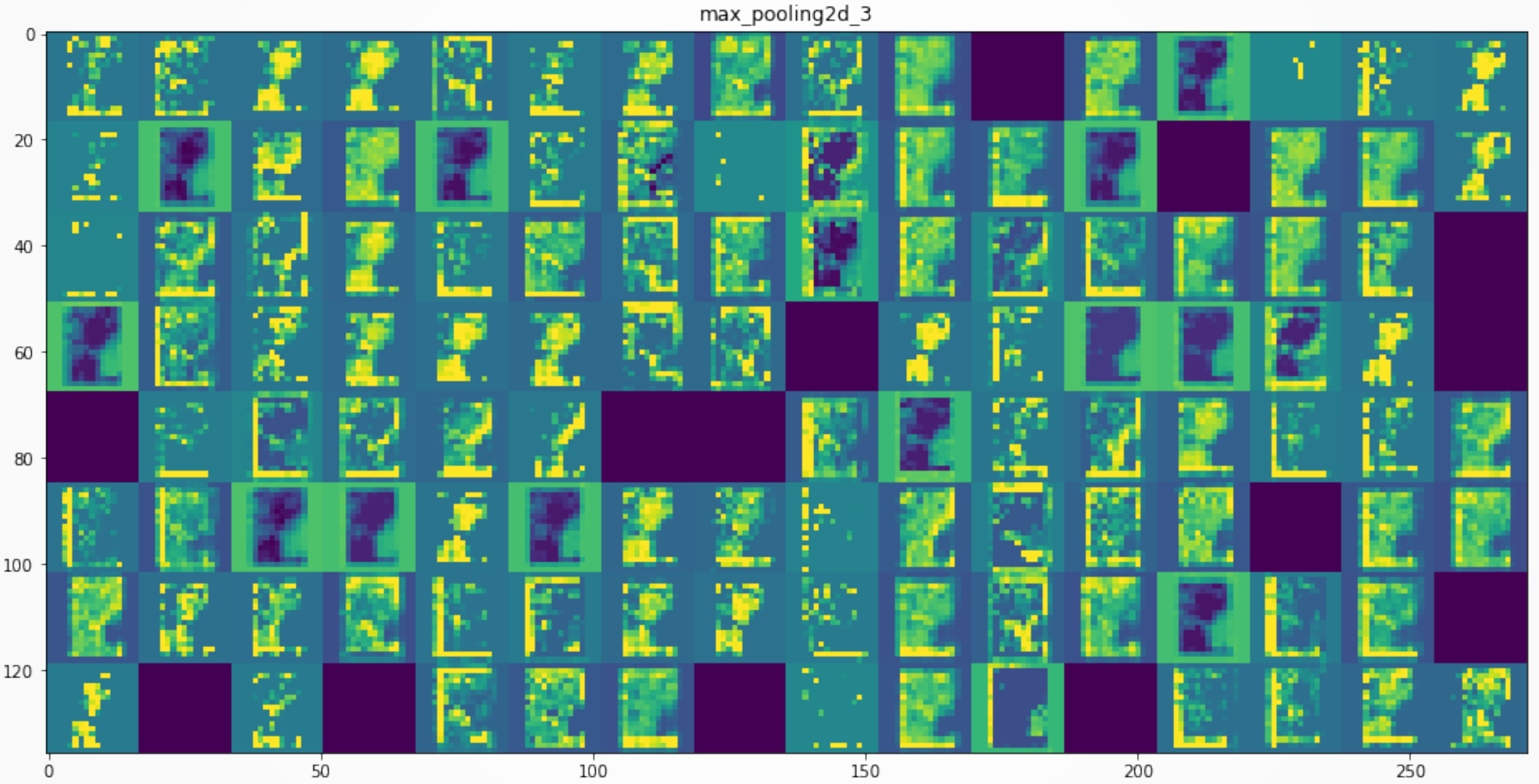
08_Empty

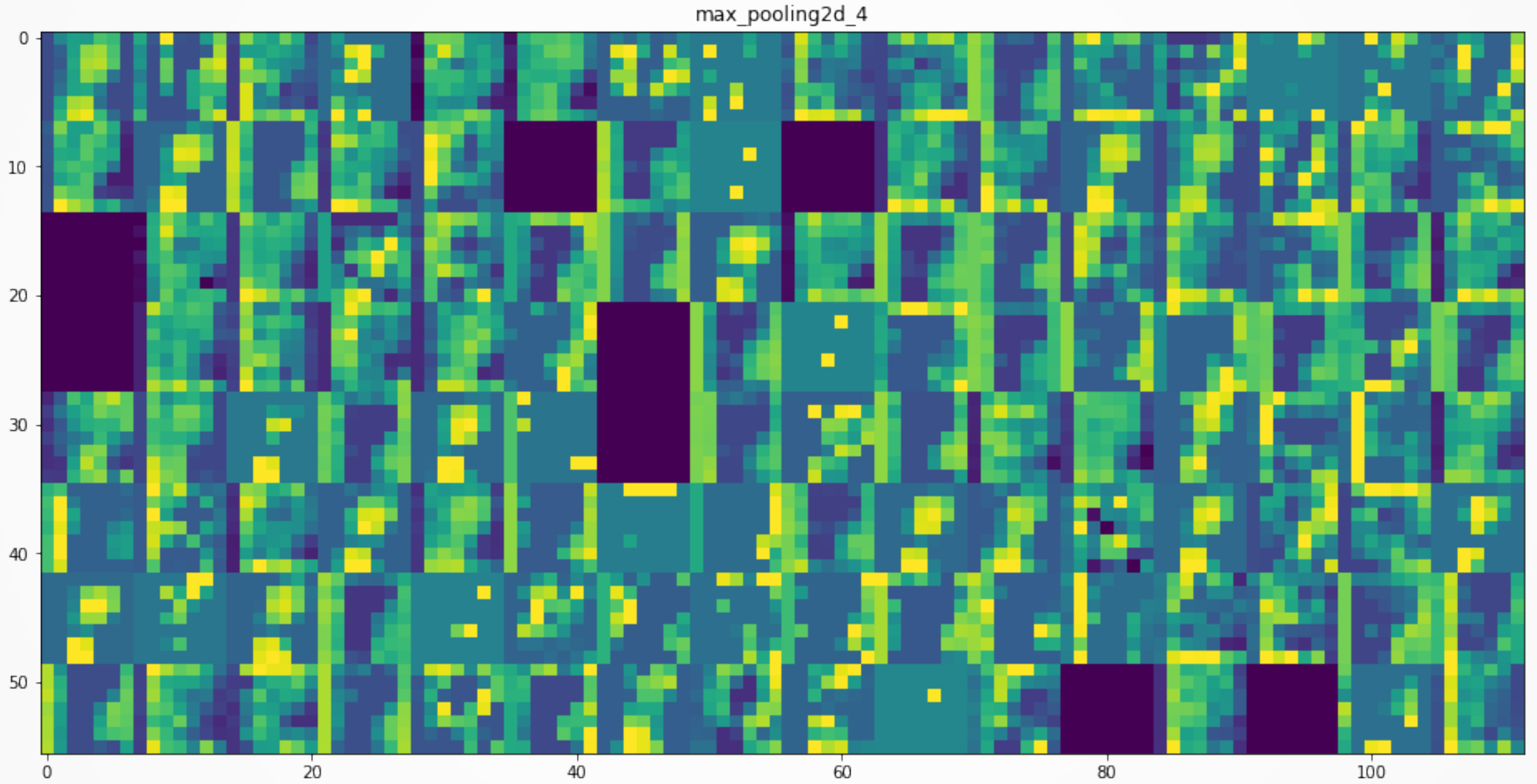
Visualizing intermediate activations

- It consists in displaying the feature maps that are output by various convolution and pooling layers in a network, given a certain input. This gives a view into how an input is decomposed unto the different filters learned by the network.
- These feature maps we want to visualize have 3 dimensions: width, height, and depth (channels). Each channel encodes relatively independent features, so the proper way to visualize these feature maps is by independently plotting the contents of every channel, as a 2D image.

F. Chollet, Deep Learning with Python, 1st ed. USA: Manning Publications Co., 2017.


conv2d_1
0
100
200
0
500
1000
1500
2000
max_pooling2d_1
0
50
100
0
200
400
600
800
1000
max_pooling2d_2
0
20
40
60
80
100
120
140
0
100
200
300
400
500

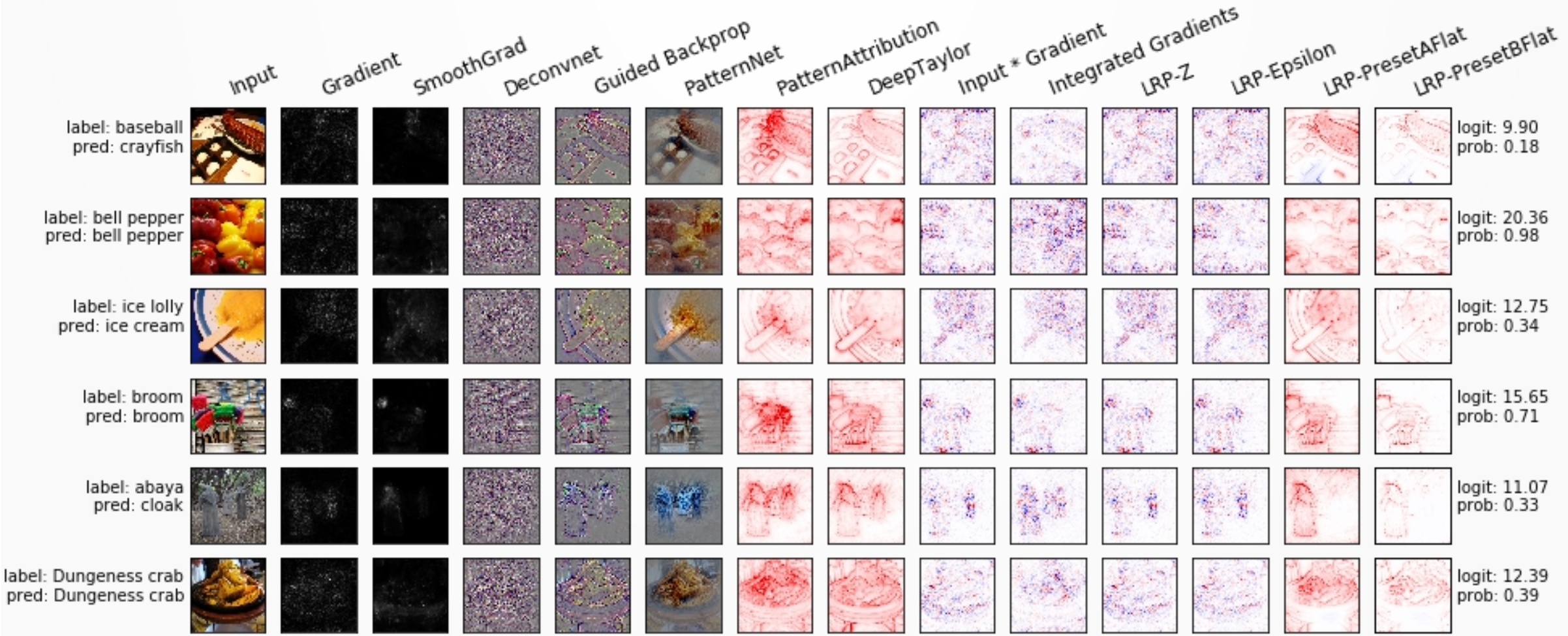

max_pooling2d_3
0
20
40
60
80
100
120
0
50
100
150
200
250


max_pooling2d_4
0
10
20
30
40
50
0
20
40
60
80
100

Other solutions

- iNNvestigate neural networks
- Visualizing heatmaps of class activation
- SHAP (SHapley Additive exPlanations)
- ...

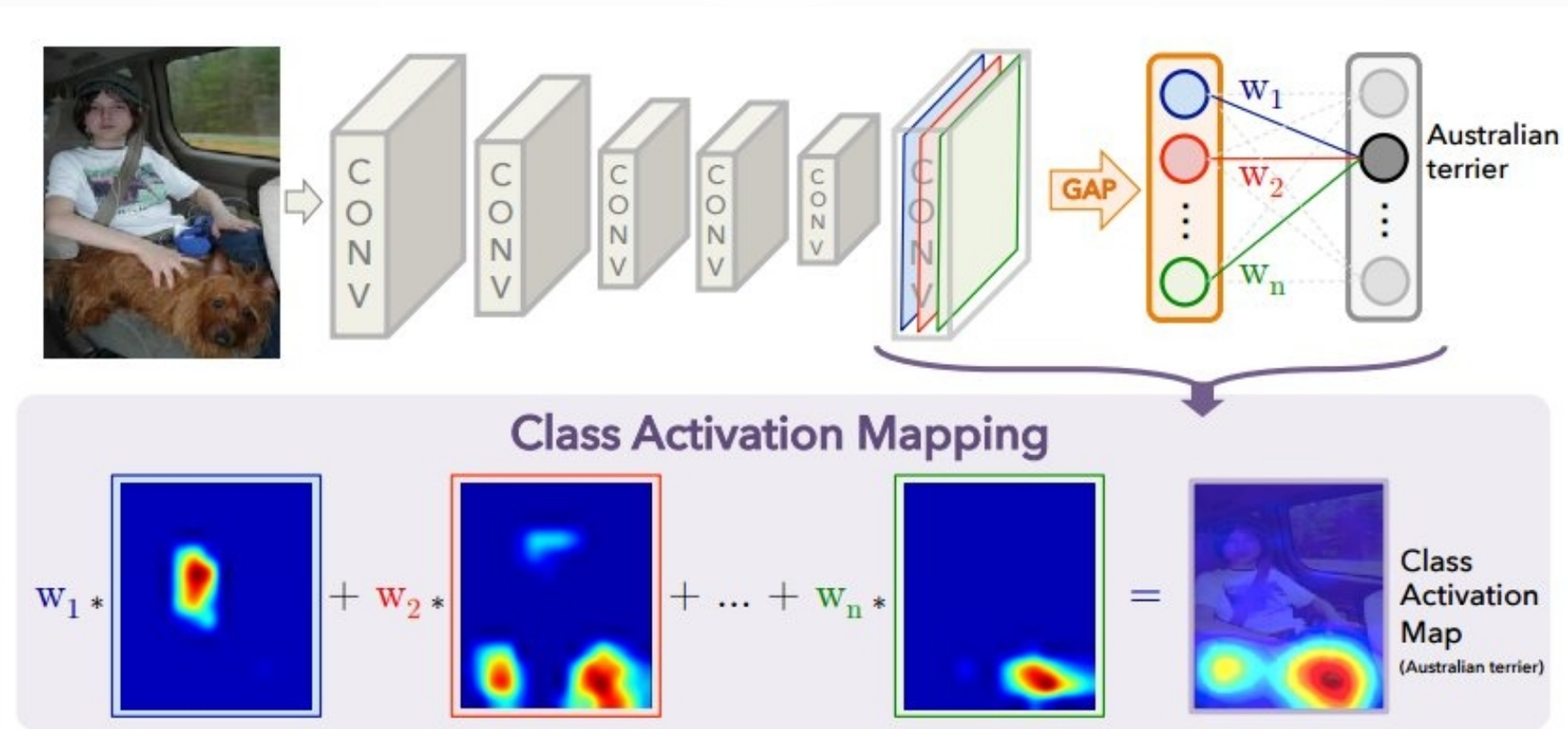
iNNvestigate neural networks



M. Alber et al., “iNNvestigate neural networks!,” arXiv:1808.04260 [cs, stat], Aug. 2018.

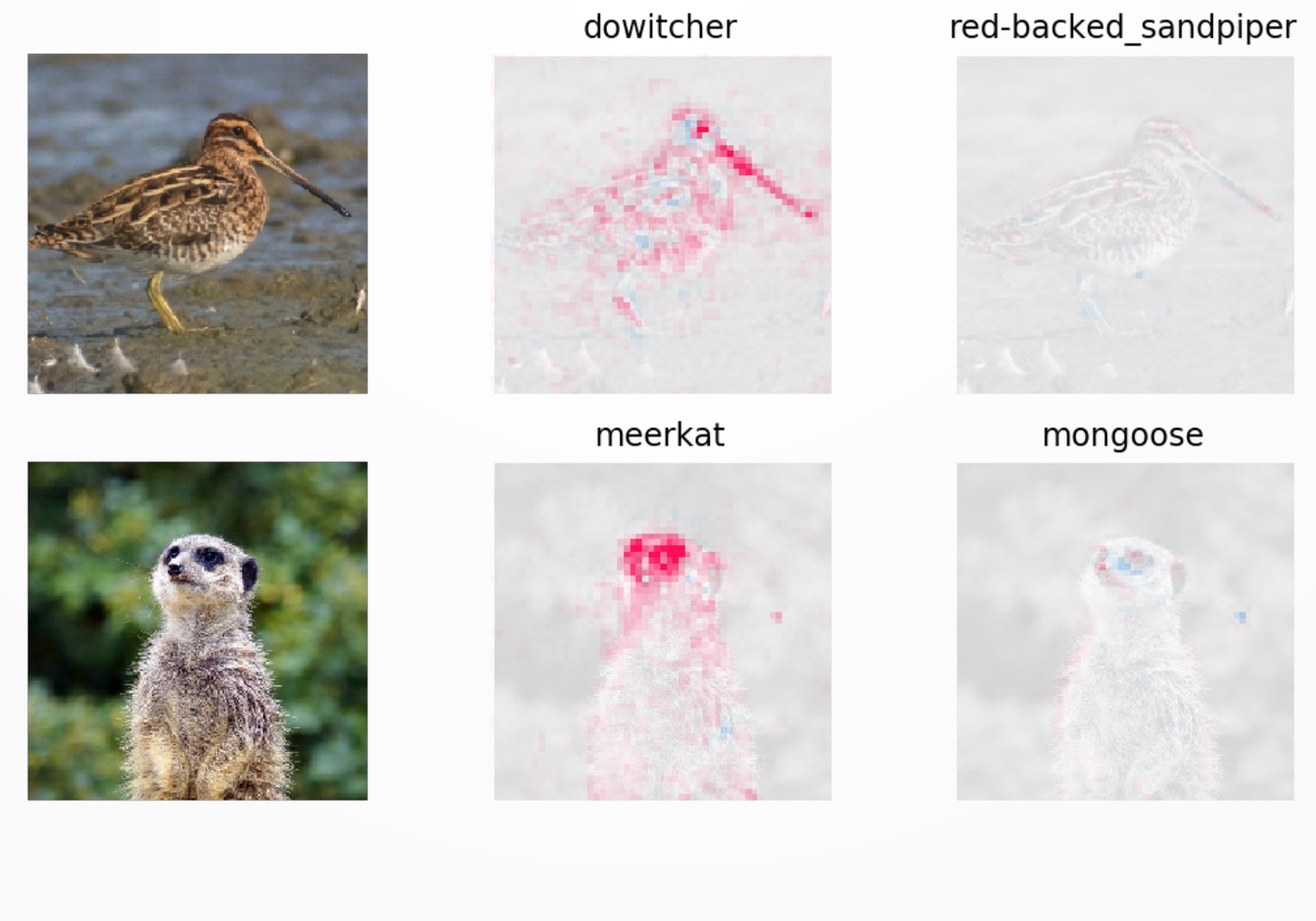
Visualizing heatmaps of class activation

Technique for generating class activation maps using the global average pooling (GAP) in CNNs. A class activation map for a particular category indicates the discriminative image regions used by the CNN to identify that category. The procedure for generating these maps is illustrated as follows:



B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2921–2929, doi: 10.1109/CVPR.2016.319.

SHAP (SHapley Additive exPlanations)



S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” 2017.

Thank you for your attention!