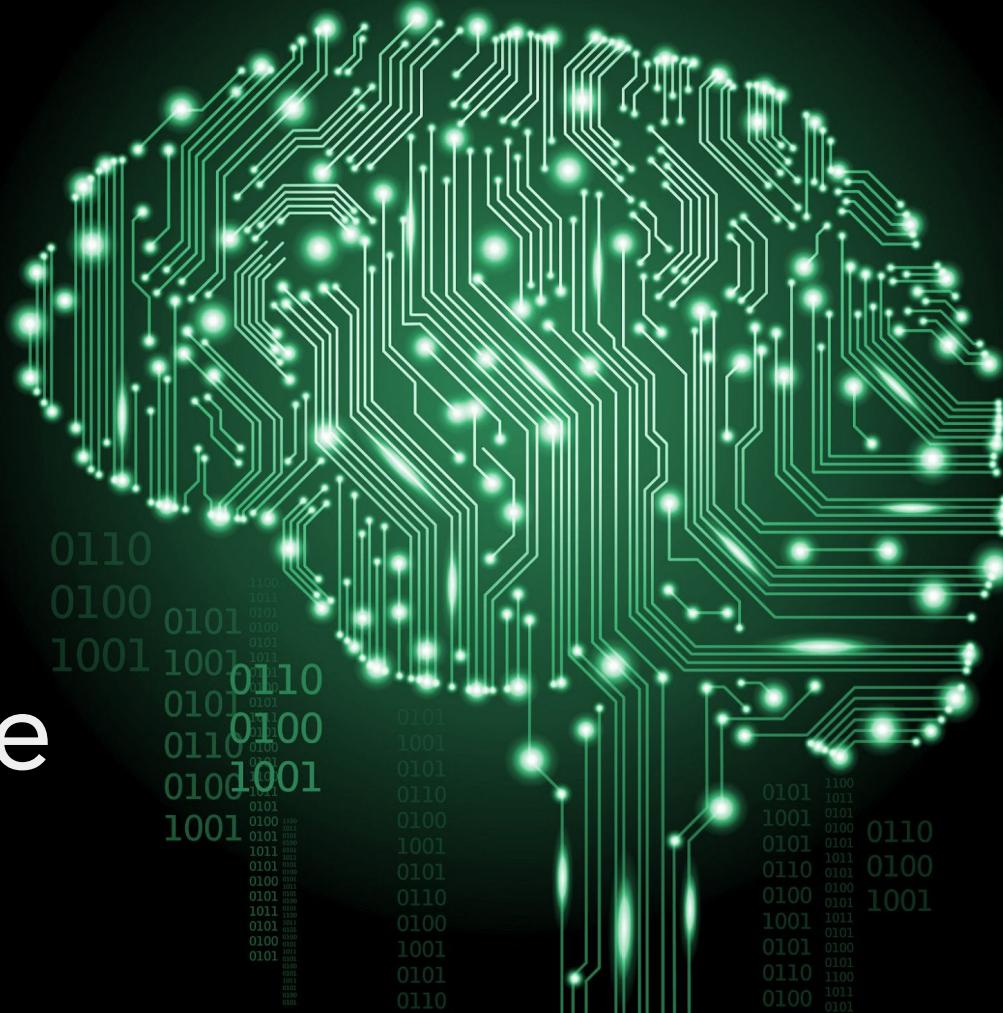


Model Visualization

Understanding models

Why visualize models?



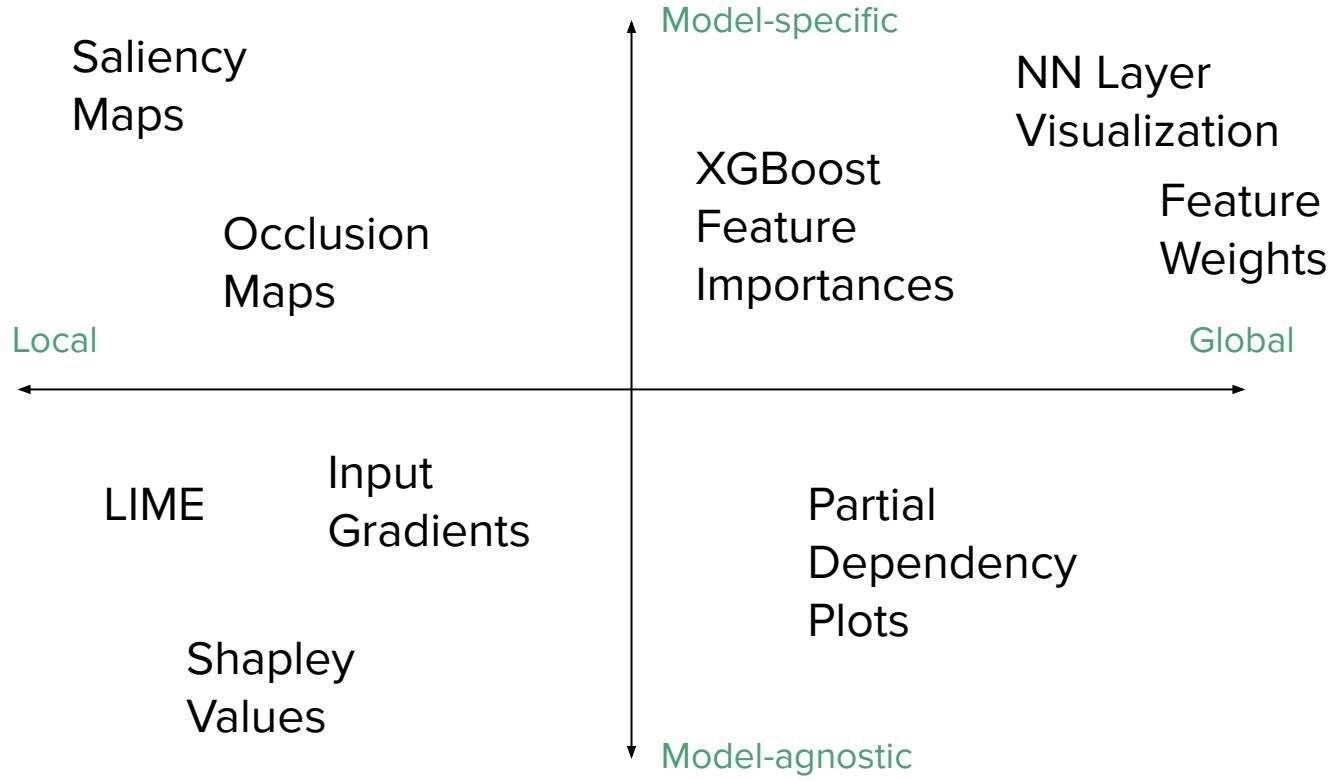


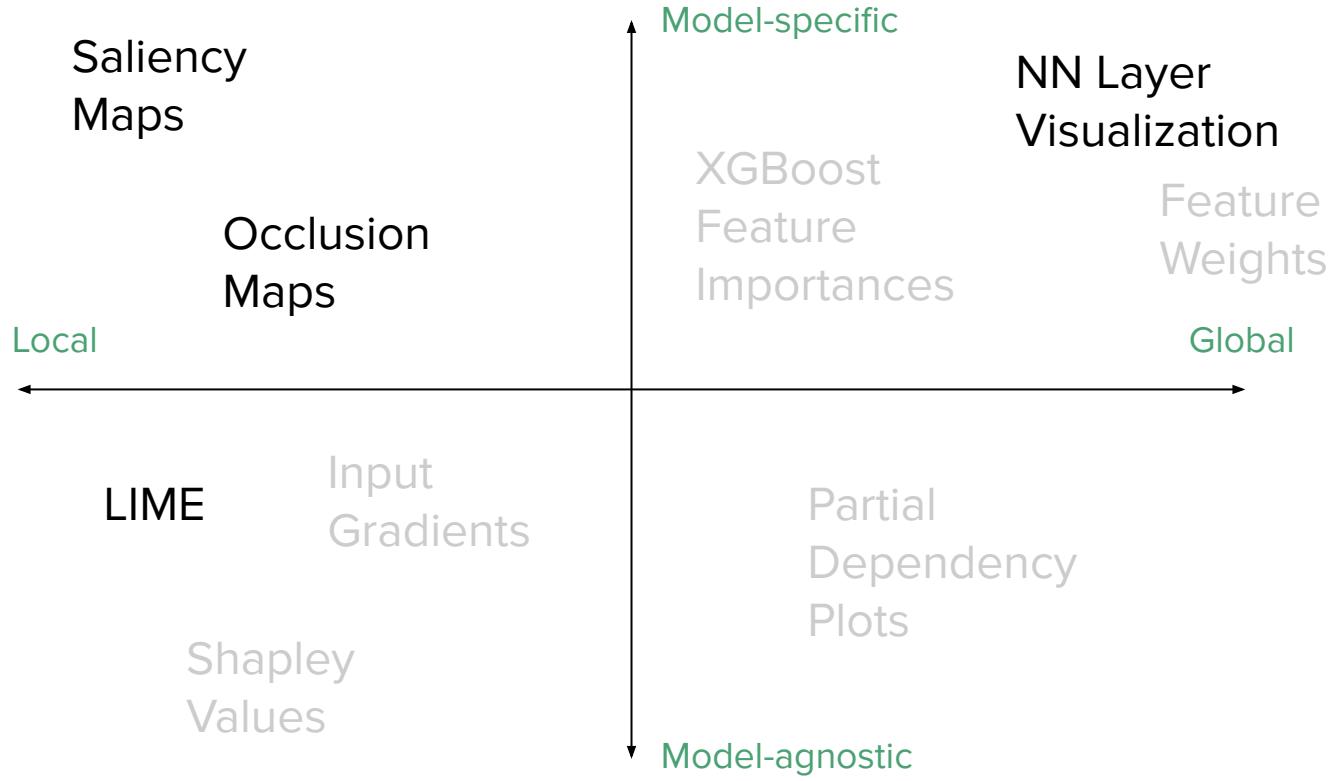
It makes us better data scientists!



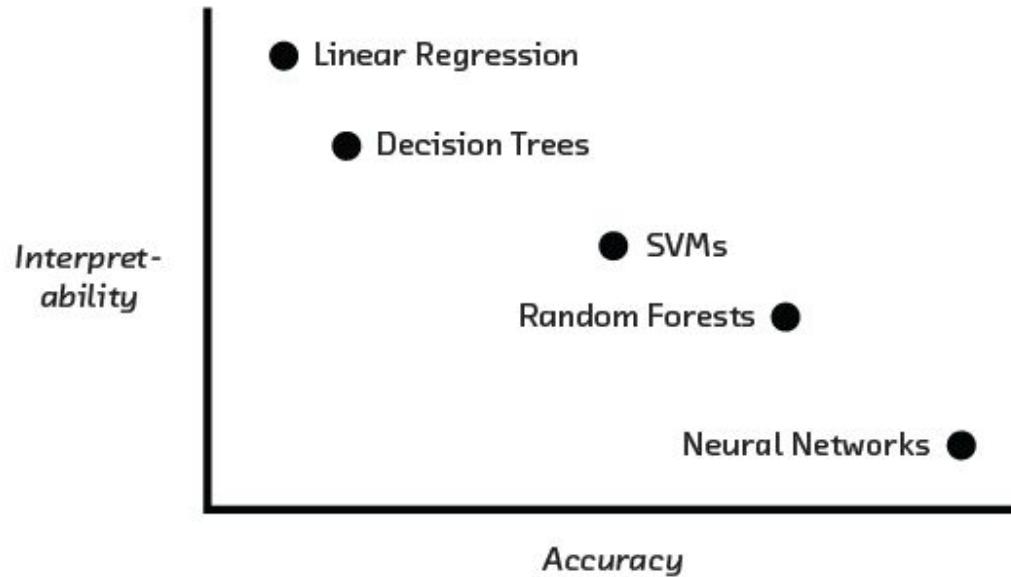
...And model explanations are often more actionable than the predictions







Our problem for this lecture:



Neural Networks are powerful models, but harder to interpret.

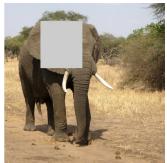
Today:



Layer Visualization



Saliency Maps



Occlusion Maps



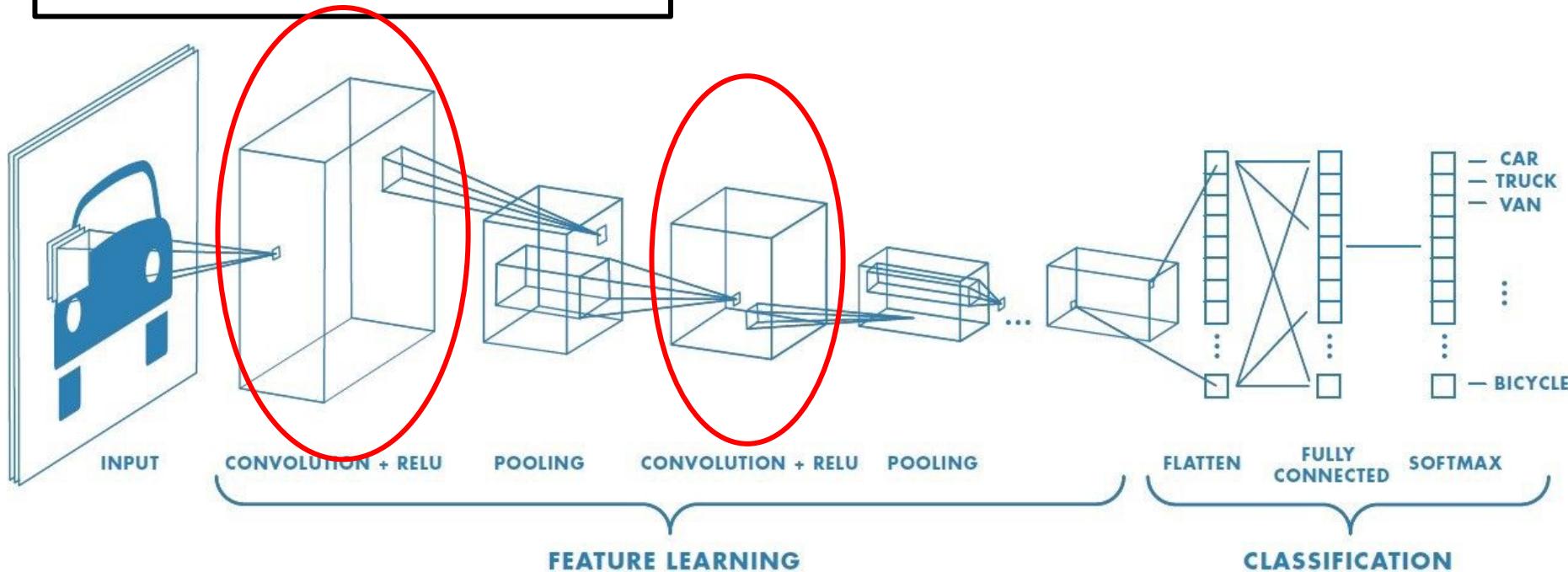
LIME

Layer Visualization

- Techniques shown in this section generalize to any NN architecture
- We will focus on CNNs because their visualizations are more interesting
- Visualization is just the end product of broader interpretation analysis

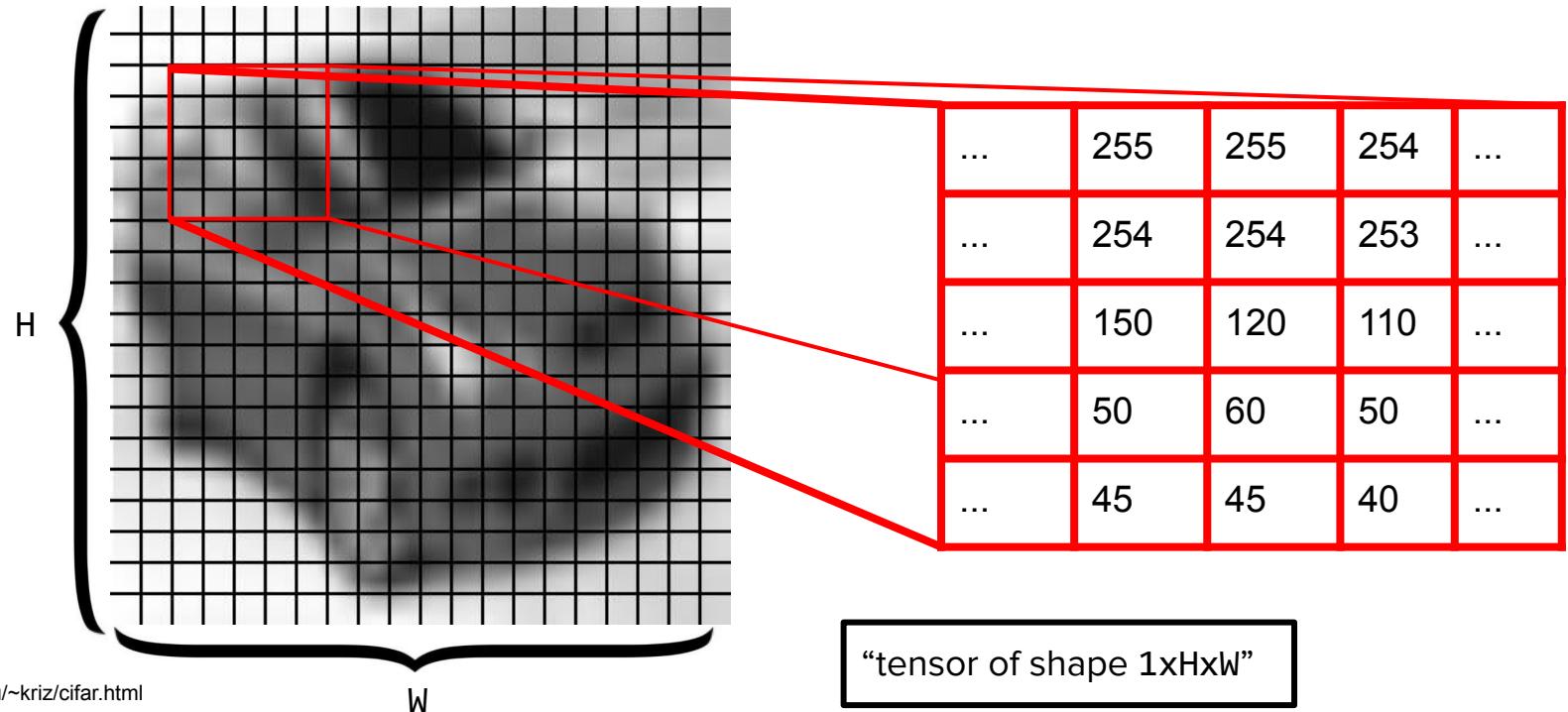
A Brief Review of CNNs

What does it mean to “visualize” a layer?



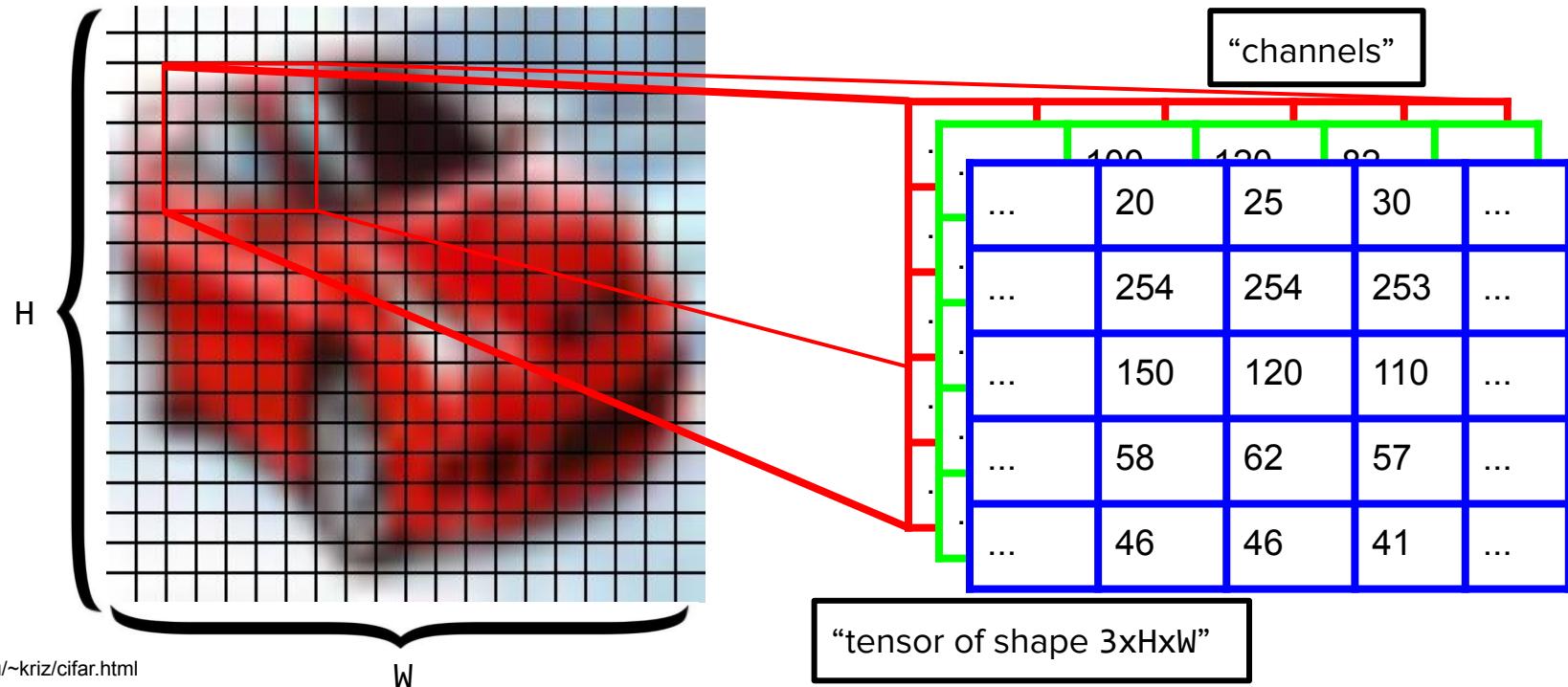
First: What's in an Image?

An image is a multidimensional array (a.k.a. “tensor”):



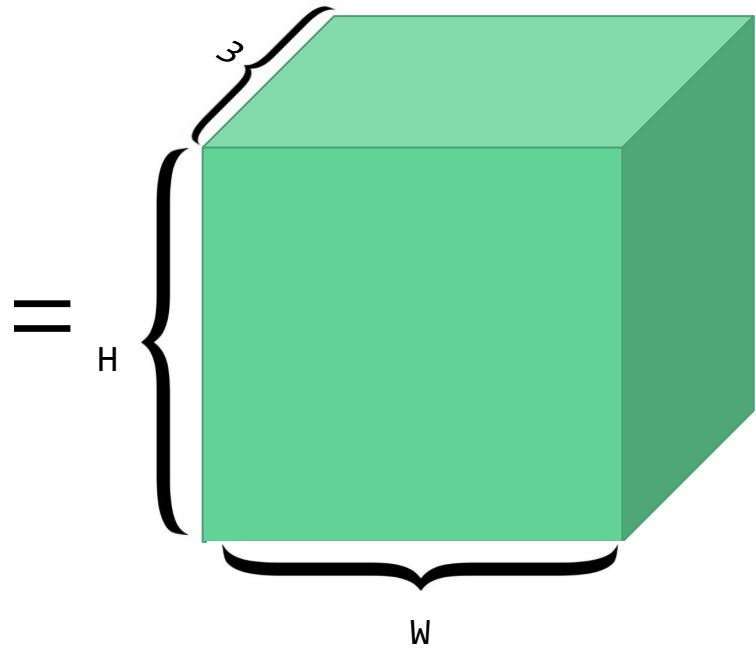
First: What's in an image?

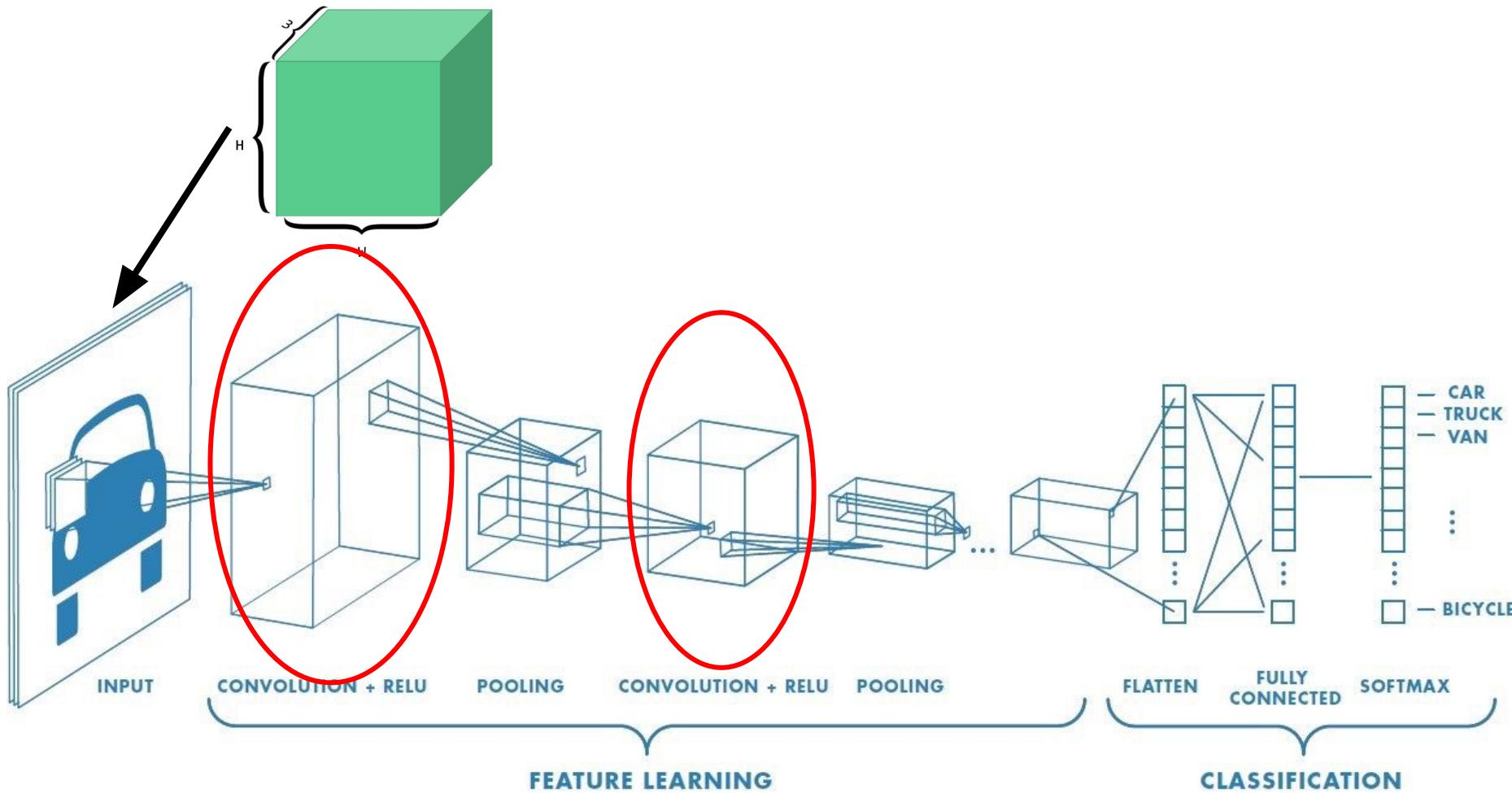
What about the other colors?



First: What's in an Image?

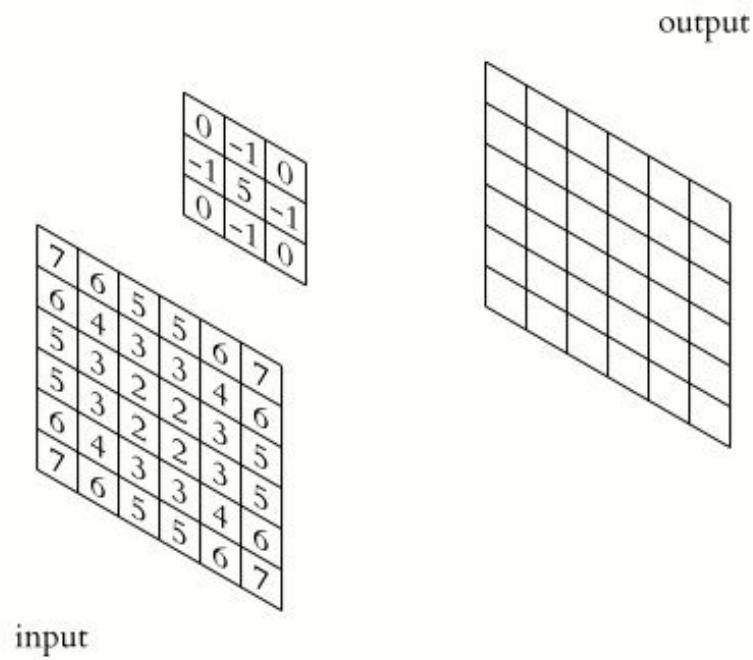
Takeaway:

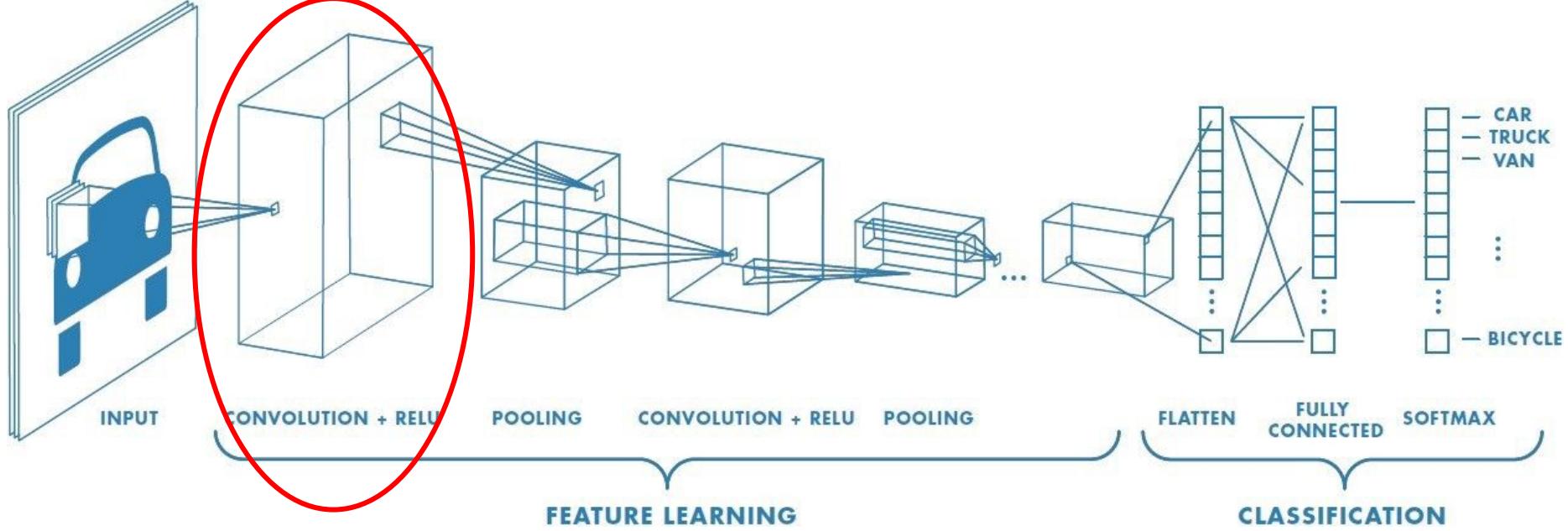




Breaking Down a Convolutional Layer

- Convolutional layer applies a set of **independent** transformations to its input
- Each transformation is characterized by a “kernel”
- Mathematically: convolution is an easy arithmetic operation - **differentiable with respect to the kernels**



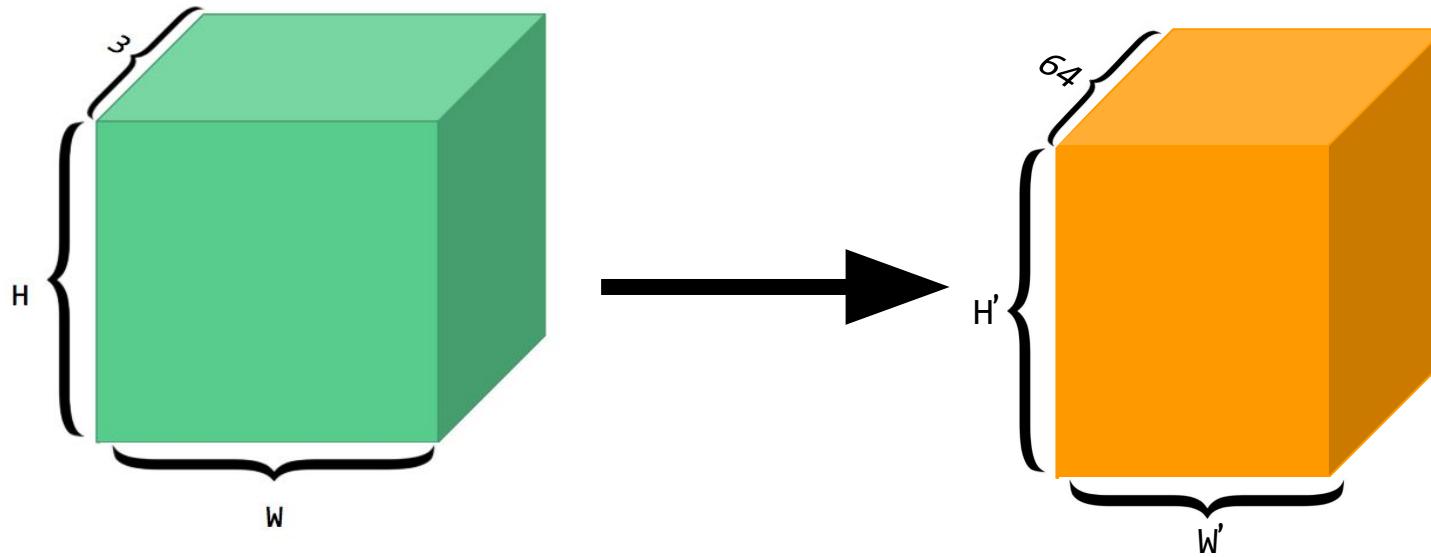


```
Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

3 input
channels

64 output channels
=

64 kernels



```
Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

3 input
channels

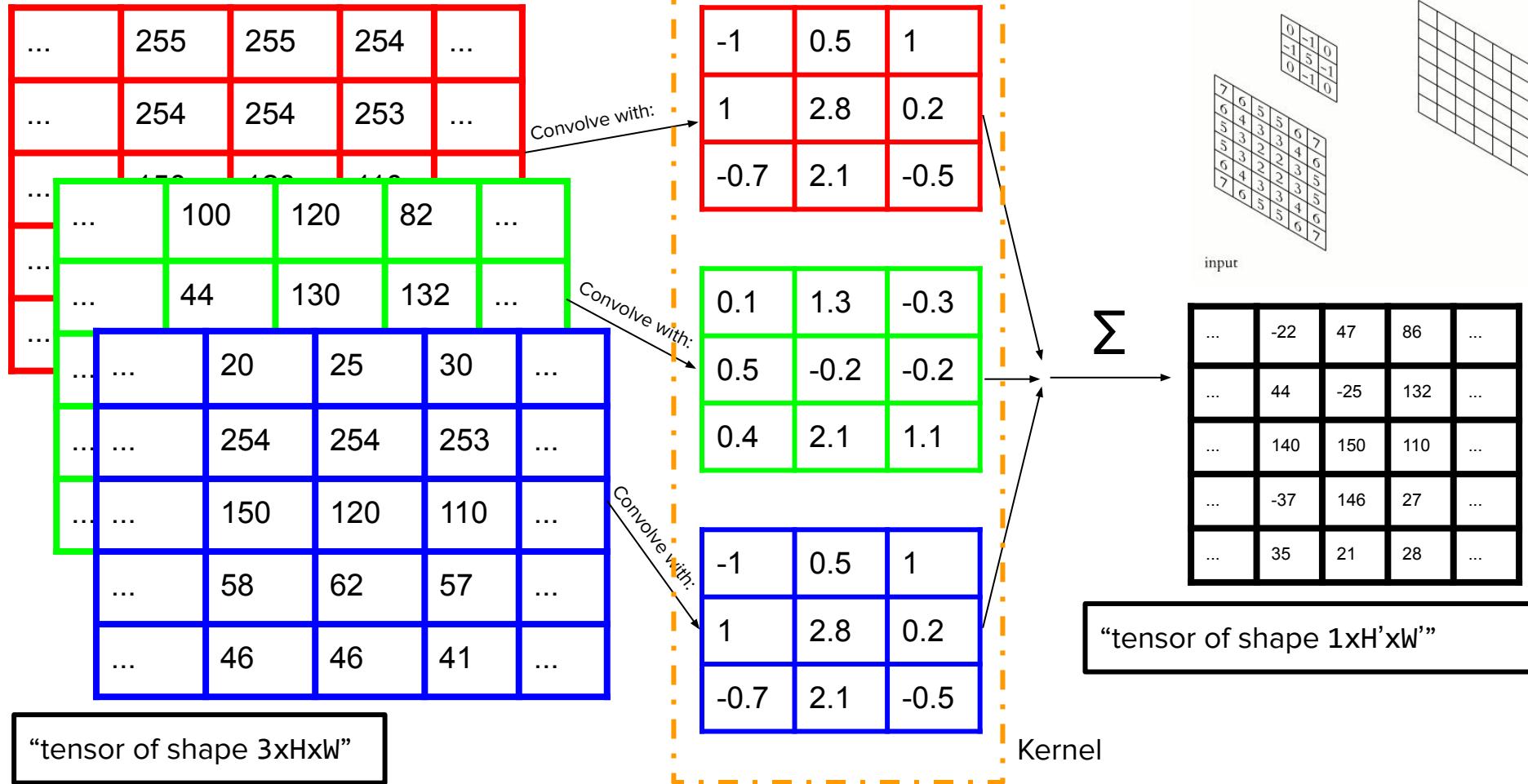
64 output channels
=
64 kernels

Each kernel amplifies a different characteristic of its input.

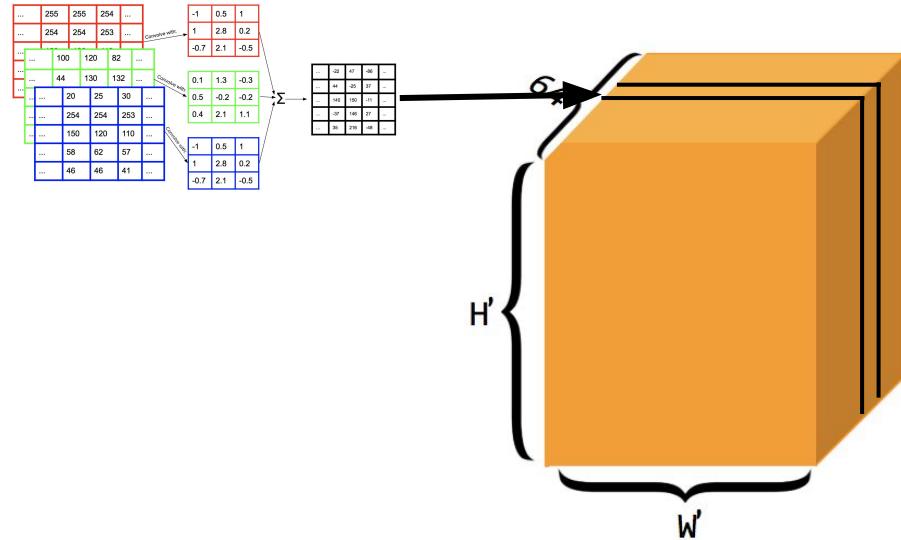
This is an example of “automatic feature engineering.”

The precise values of these kernels are determined during training via backpropagation -

They are the essence of a trained CNN!



```
Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```



Result of layer
given input

What does it mean to “visualize” a layer?

One answer: see what each kernel in that layer “reacts” strongly to; look for broader themes

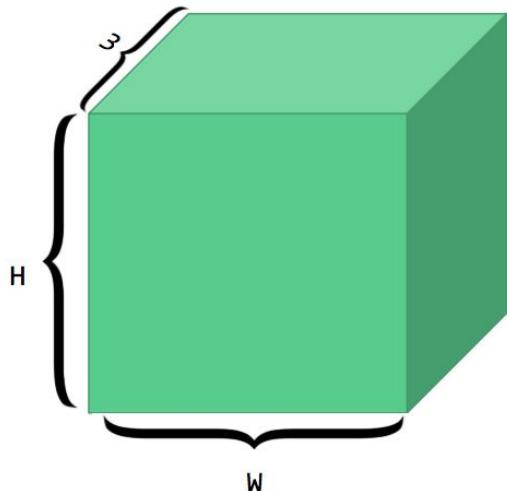
Tying Things Together: Channel Visualizations

- Premise: create an artificial image that maximizes the activation of a channel in a convolutional layer
- Process is exactly the same as training a neural network!
 - i.e. gradient descent

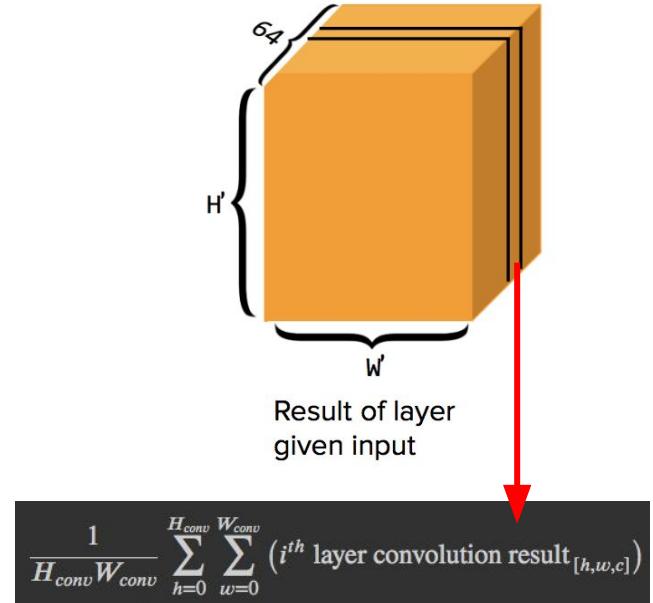
Optimizing over Input Images

Given a pretrained CNN:

Optimize this
input tensor:



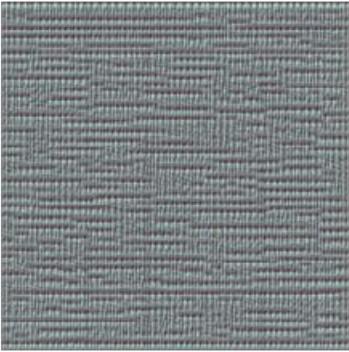
... to maximize an
objective - e.g. :



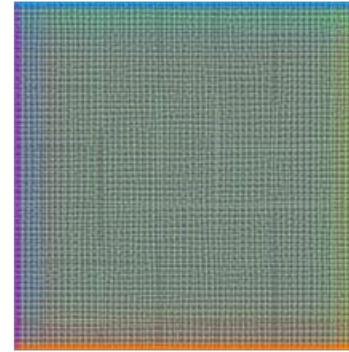
for some layer **i** and channel **c** (in that
layer) of our choosing.

Starting Shallow

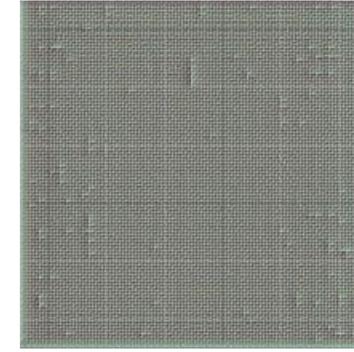
VGG Layer block2_conv1, Channel 1



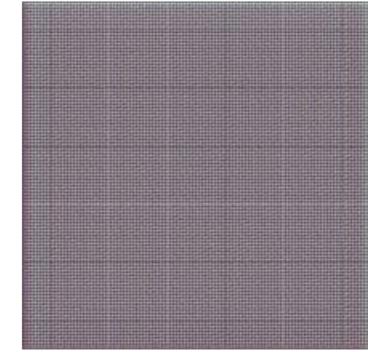
VGG Layer block2_conv1, Channel 3



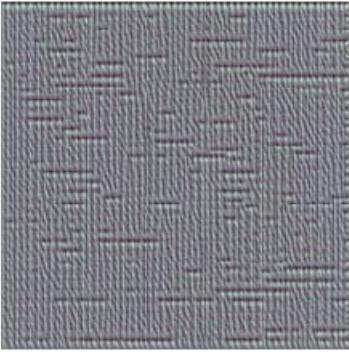
VGG Layer block2_conv1, Channel 5



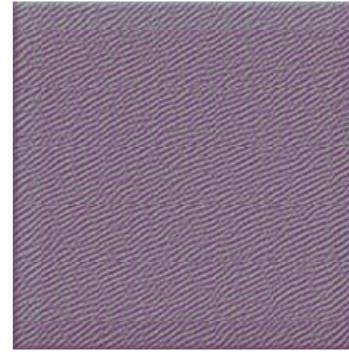
VGG Layer block2_conv1, Channel 7



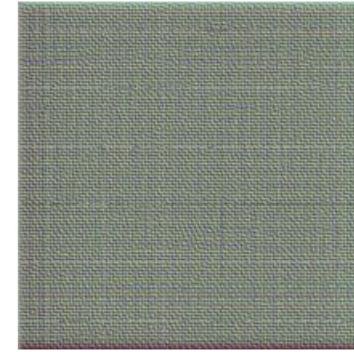
VGG Layer block2_conv1, Channel 8



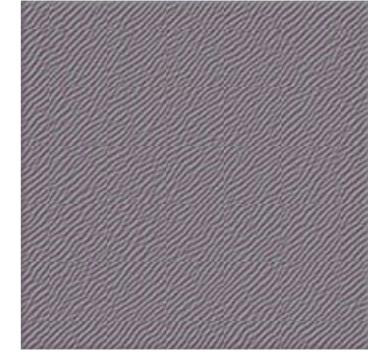
VGG Layer block2_conv1, Channel 11



VGG Layer block2_conv1, Channel 16

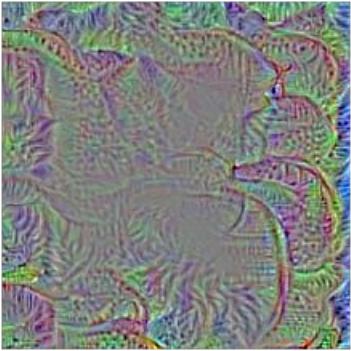


VGG Layer block2_conv1, Channel 18

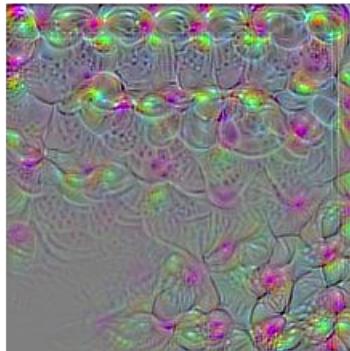


Jumping in the Deep End

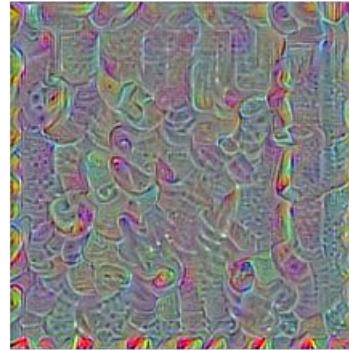
VGG Layer block5_conv1, Channel 0



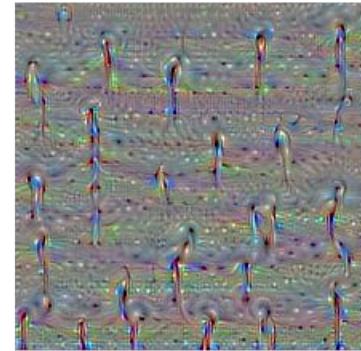
VGG Layer block5_conv1, Channel 1



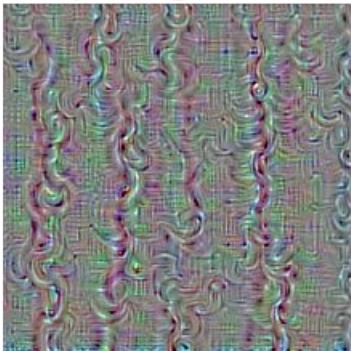
VGG Layer block5_conv1, Channel 3



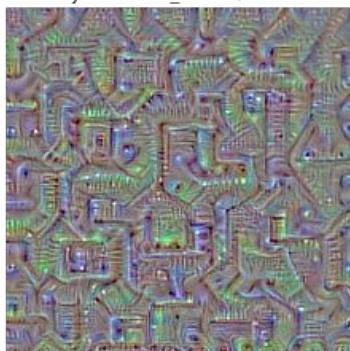
VGG Layer block5_conv1, Channel 7



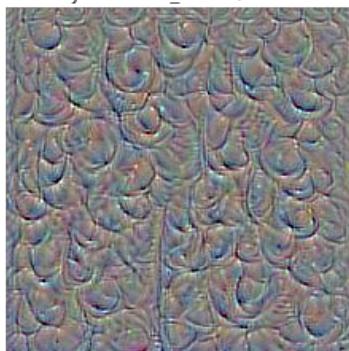
VGG Layer block5_conv1, Channel 8



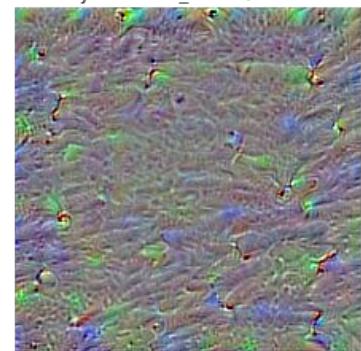
VGG Layer block5_conv1, Channel 11



VGG Layer block5_conv1, Channel 18



VGG Layer block5_conv1, Channel 19



Further Exploration

- Image augmentations
 - Gaussian + other types of blurring
 - Stochastic jitter
 - Affine transformations
- Different objective functions
 - Entire layer, Deep Dream
 - Class-wise visualizations - optimize input image to maximize probability of a given class

To the notebooks!



https://github.com/Harvard-IACS/2020-ComputeFest/tree/master/visualization_workshop

Layer visualization colab notebook

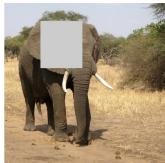
Today:



Layer Visualization



Saliency Maps



Occlusion Maps



LIME

Saliency Maps



Can you identify
this object?

Saliency Maps



How about now?

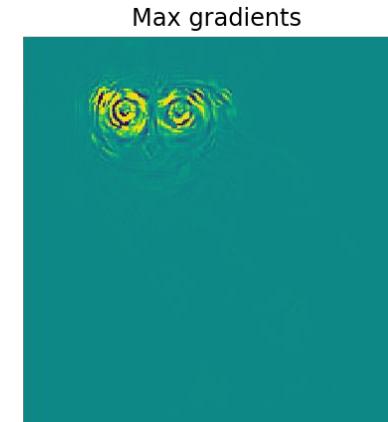
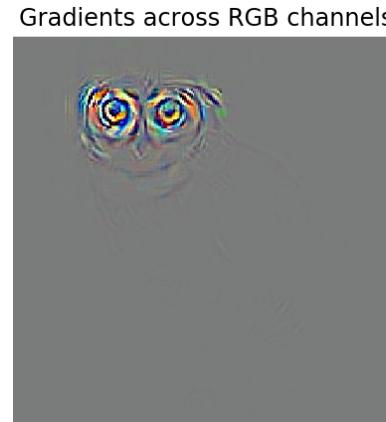
Same source, same zoom.

Saliency Maps



- Saliency maps were originally a model for human attention
- Refers to the likelihood that the observer will focus on certain parts of an image

Saliency Maps

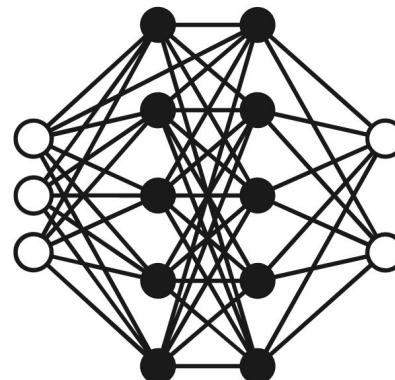


For neural nets - the gradient in the predicted class wrt pixel value, using backpropagation:

$$\frac{\Delta \text{Predicted Class}}{\Delta \text{Pixel Value}}$$

Saliency Maps

Get predicted class



Predicted Class:
Owl



Backpropagate to get the gradient
w.r.t. each pixel of the image

Overlay



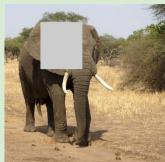
Today:



Layer Visualization



Saliency Maps



Occlusion Maps



LIME

Occlusion Maps

What is brand
is this car?



Occlusion Maps

What is brand
is this car?



Occlusion Maps



Source: <https://unsplash.com/photos/UTw183oZPf0>

Occlusion Maps



Source: <https://unsplash.com/photos/UTw183oZPf0>

Occlusion Maps



Source: <https://unsplash.com/photos/UTw183oZPf0>

Occlusion Maps



Source: <https://unsplash.com/photos/UTw183oZPf0>

Occlusion Maps



Source: <https://unsplash.com/photos/UTw183oZPf0>

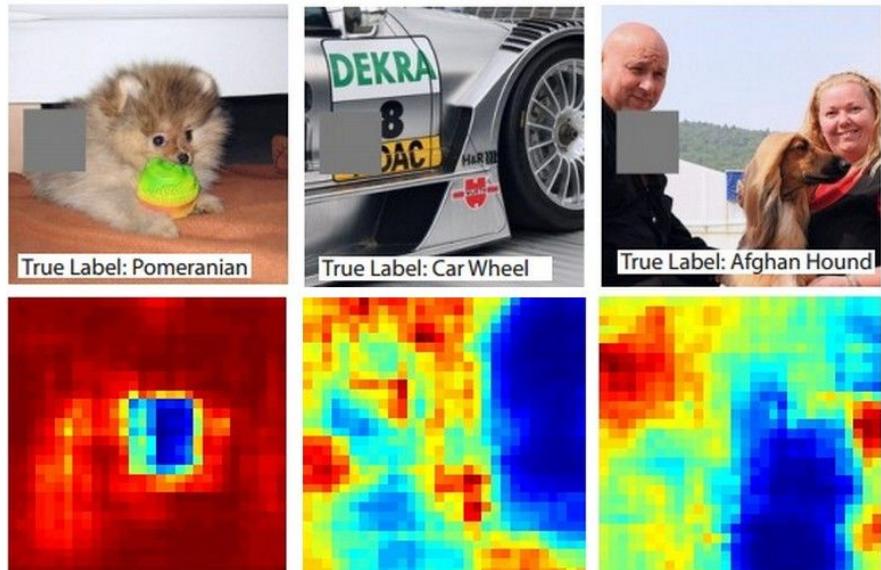
Occlusion Maps



Source: <https://unsplash.com/photos/UTw183oZPf0>

Occlusion Maps

- Zero out a square patch of image, predict and measure accuracy
- Need correct class *a priori*
- nb Saliency is a mathematical technique, Occlusion is an engineering technique



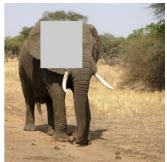
Today:



Layer Visualization



Saliency Maps



Occlusion Maps



LIME

LIME



(a) Husky classified as wolf



(b) Explanation

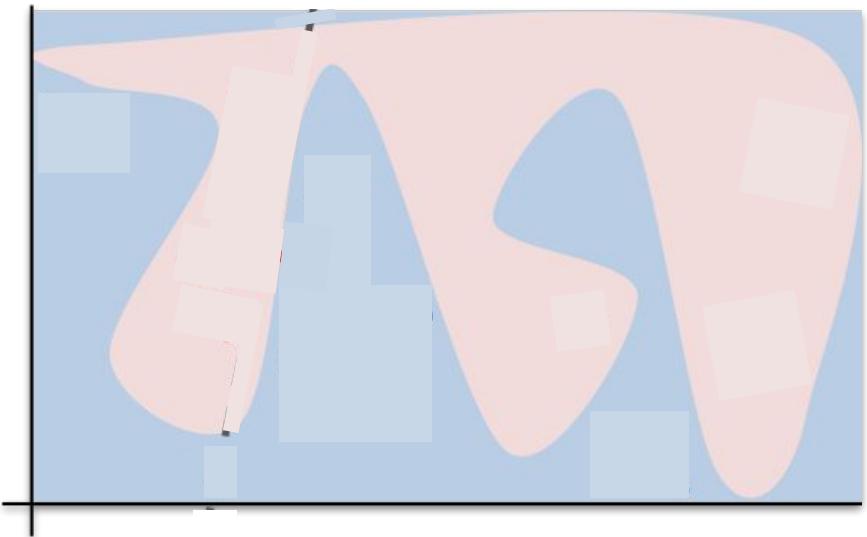
LIME reveals this (bad) CNN
thinks snow=wolf

- Local Interpretation for Model-Agnostic Explanations
- Like Occlusion Mapping, perturb input and watch model accuracy
- Unlike Occlusion Mapping - it's not just for CNNs! Works with general model types

LIME - How it works

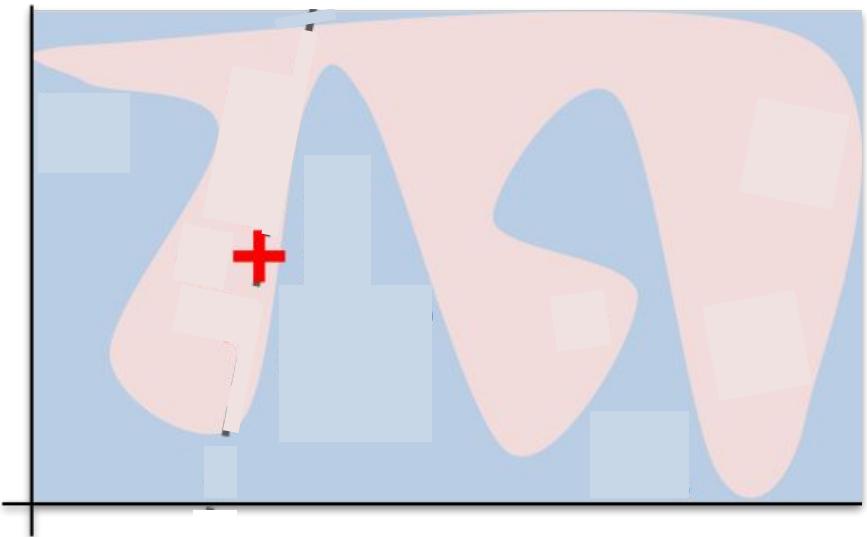
Local interpretability:

1: Take a black-box model with arbitrary decision boundary



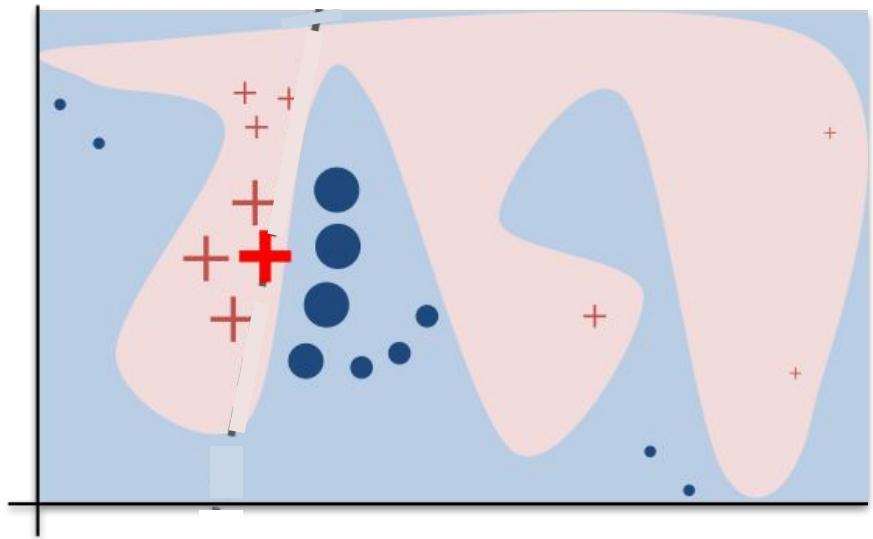
LIME - How it works

2: Select observation to explain



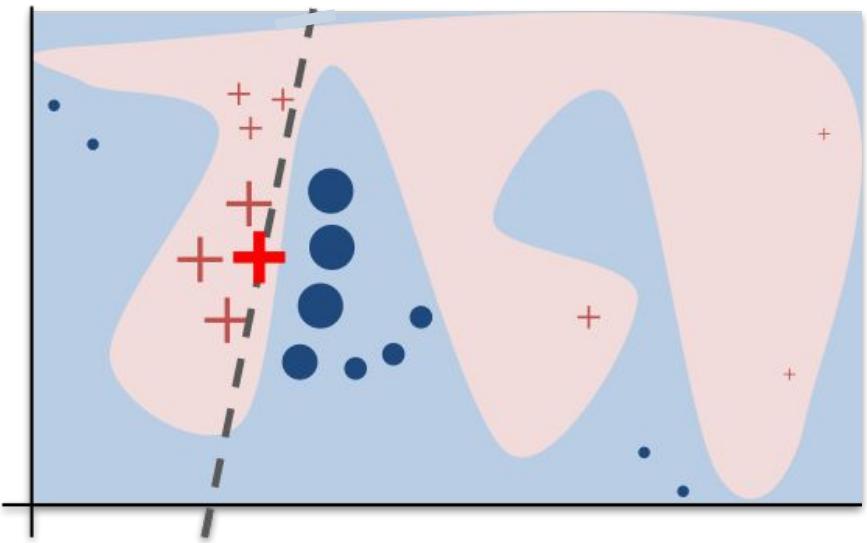
LIME - How it works

3: Sample around the chosen selected point



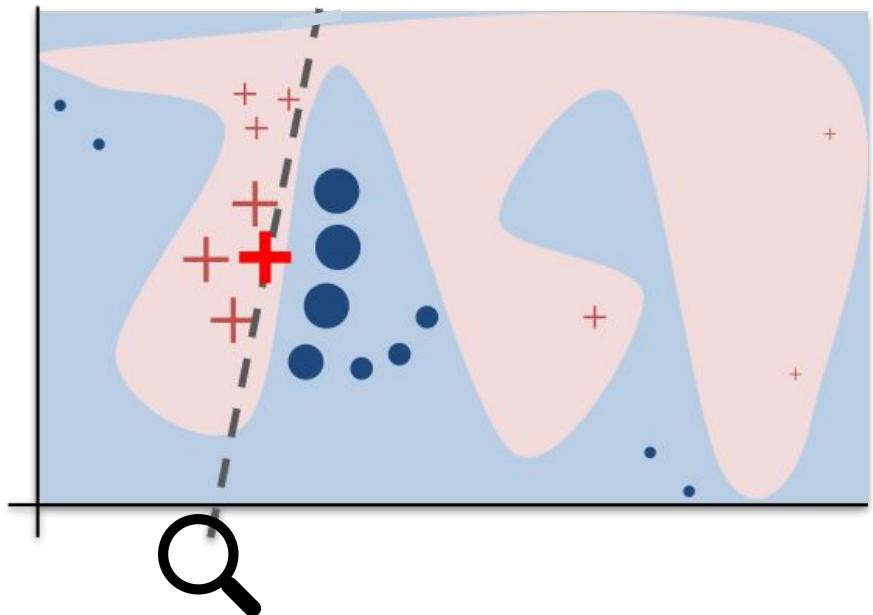
LIME - How it works

4: Build a local linear regression with the samples



LIME - How it works

5: Feature weights correspond to local explanation!





(a) Husky classified as wolf



(b) Explanation

To the notebooks!



https://github.com/Harvard-IACS/2020-ComputeFest/tree/master/visualization_workshop

Saliency / Occlusion / LIME colab notebook

In closing...

All of these are inherently estimations of a high-dimensional space

Don't take these as ground truth. Interpretations require interpretations

Interpret a thorough number of samples, and think critically about each

When used properly, these techniques are essential for us to validate
and understand our models

In closing...



A herd of sheep grazing on a lush green hillside
Tags: grazing, sheep, mountain, cattle, horse

...interpret every single model you build!
Happy modeling, everyone!!