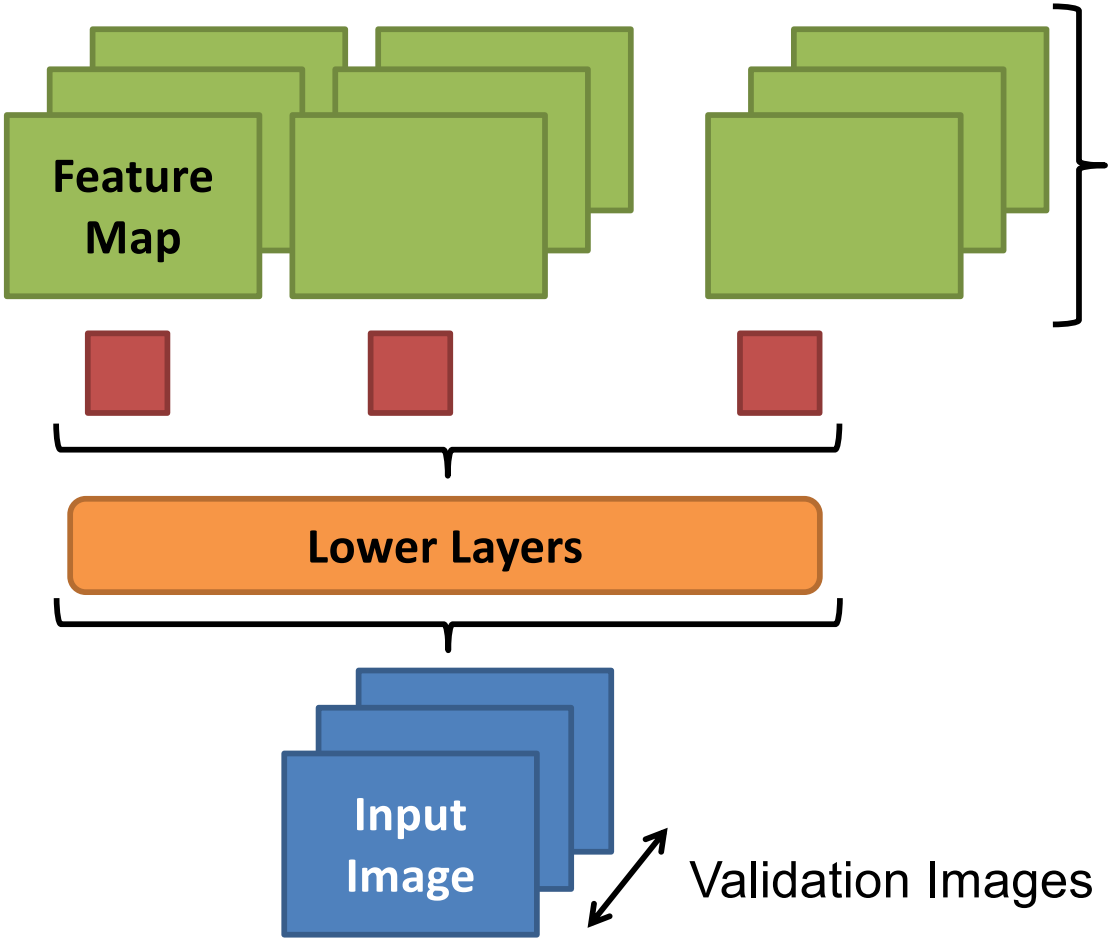
Computer Vision

Convolutional Neural Networks as Feature Extractors

Luis F. Teixeira | M.EIC | FEUP

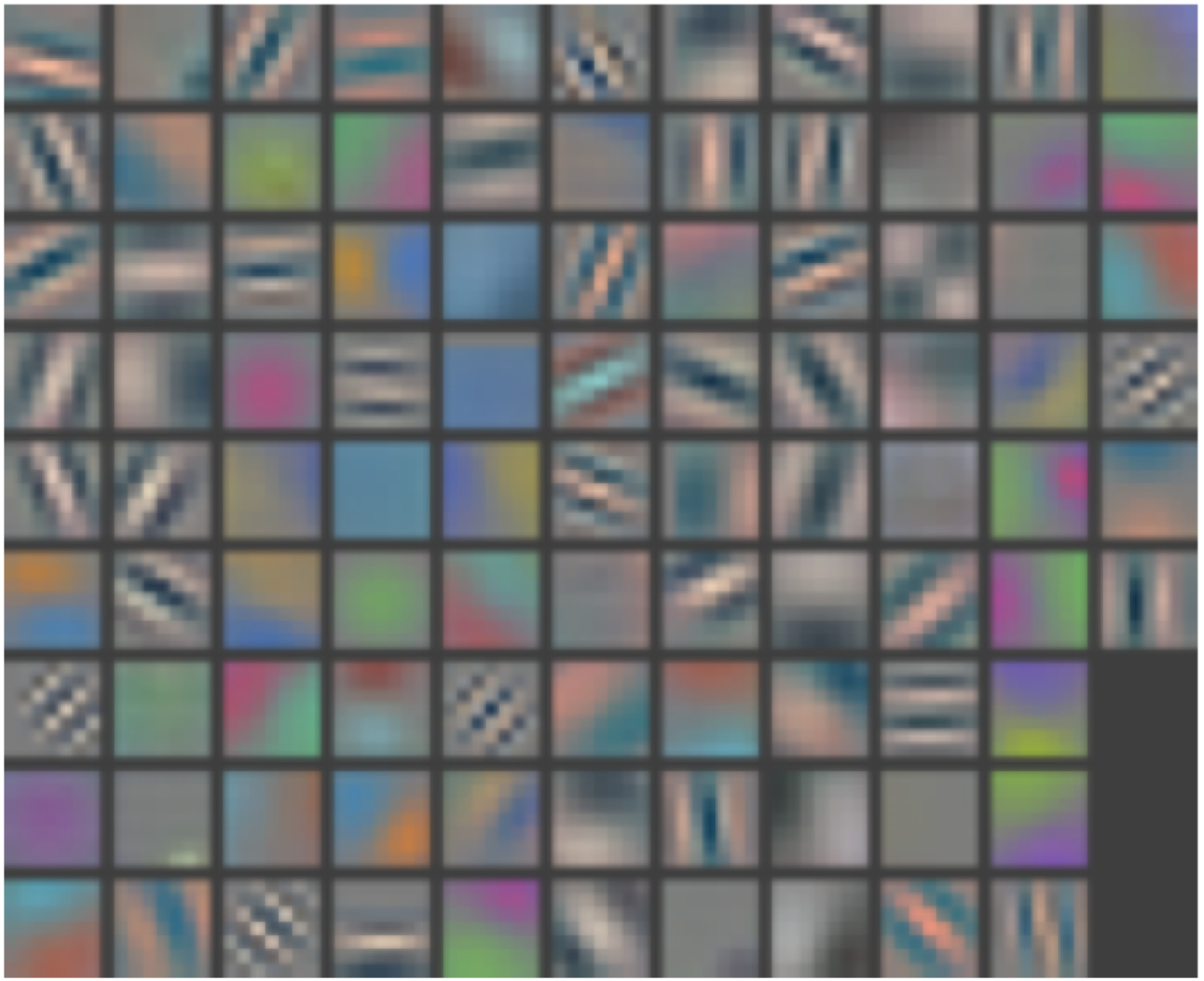
Visualize CNN filters

- Use ImageNet 2012 validation set
- Push each image through network



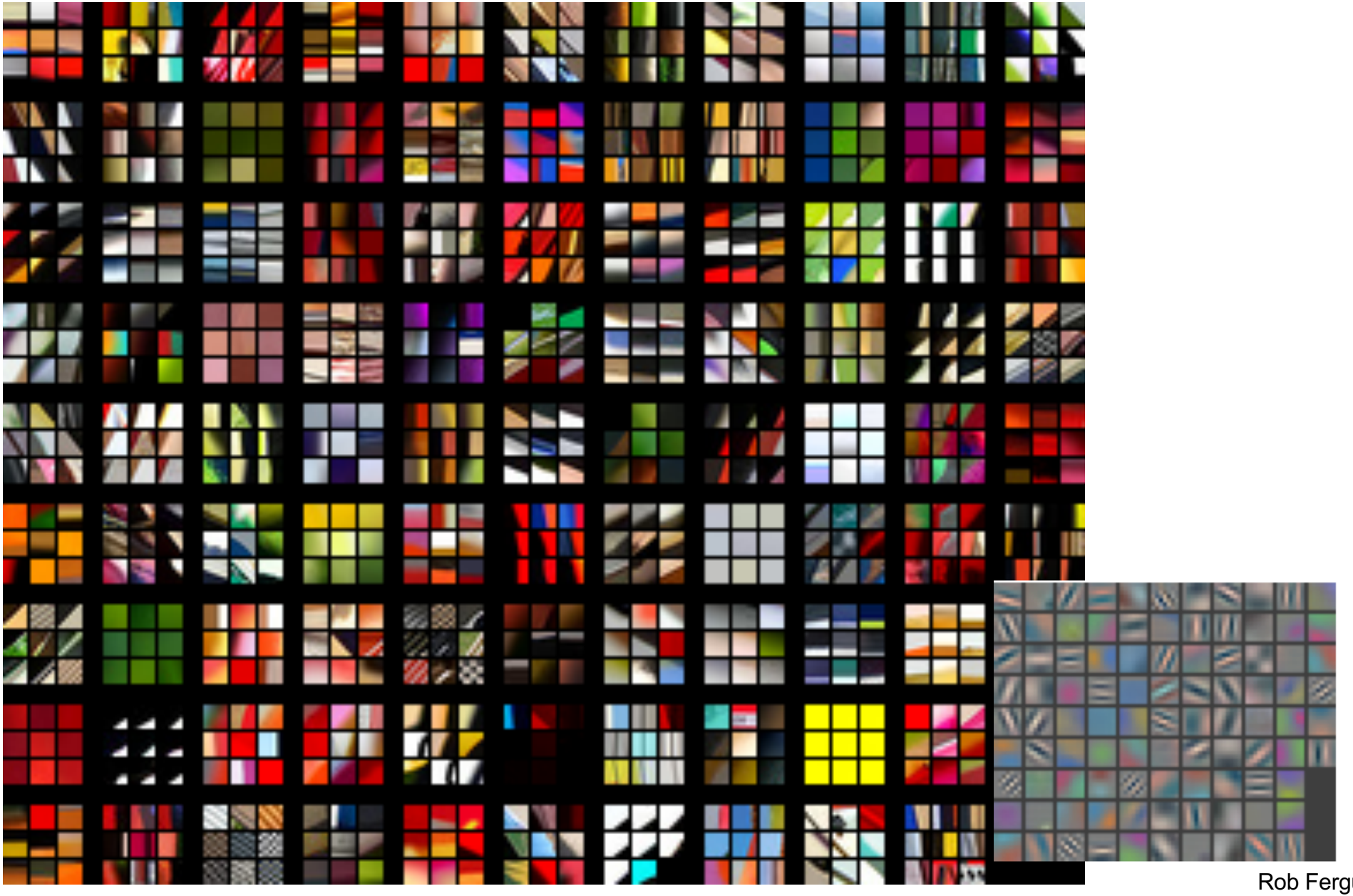
- Take max activation from feature map associated with each filter
- Use Deconvnet to project back to pixel space (we will see later how it works)

Layer 1: Filters



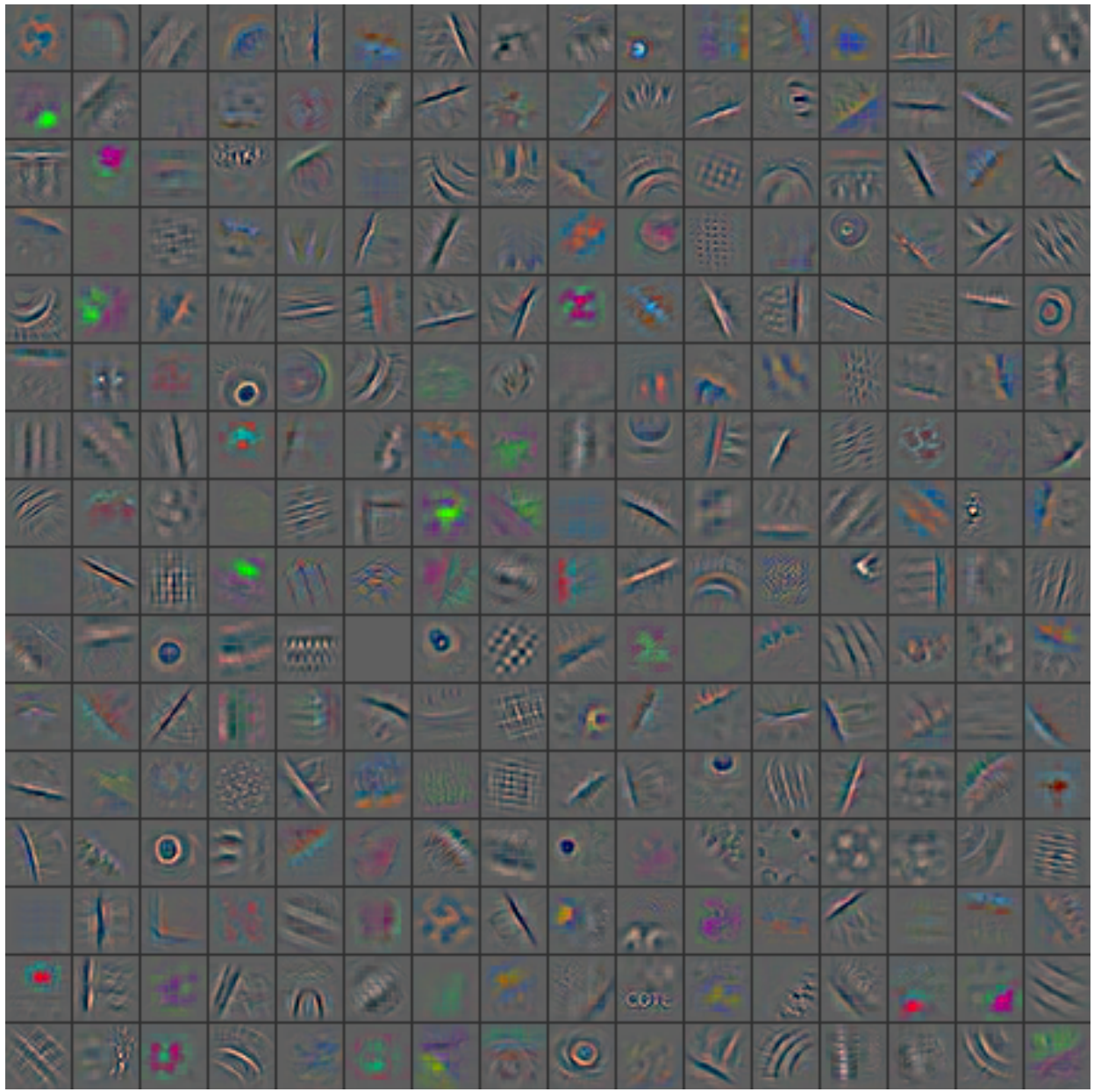
Rob Fergus

Layer 1: Top-9 Patches

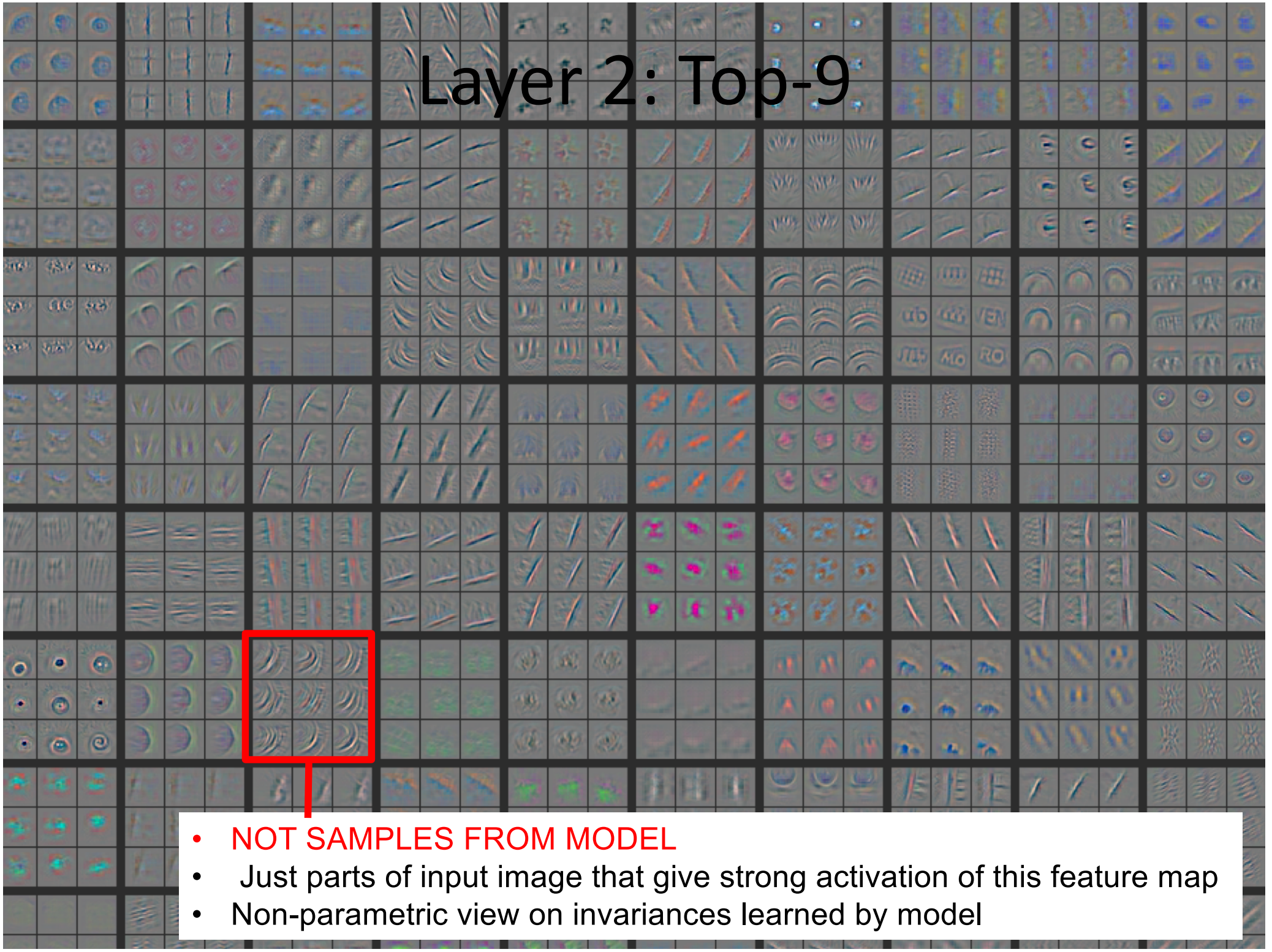


Rob Fergus

Layer 2: Top-1



Rob Fergus

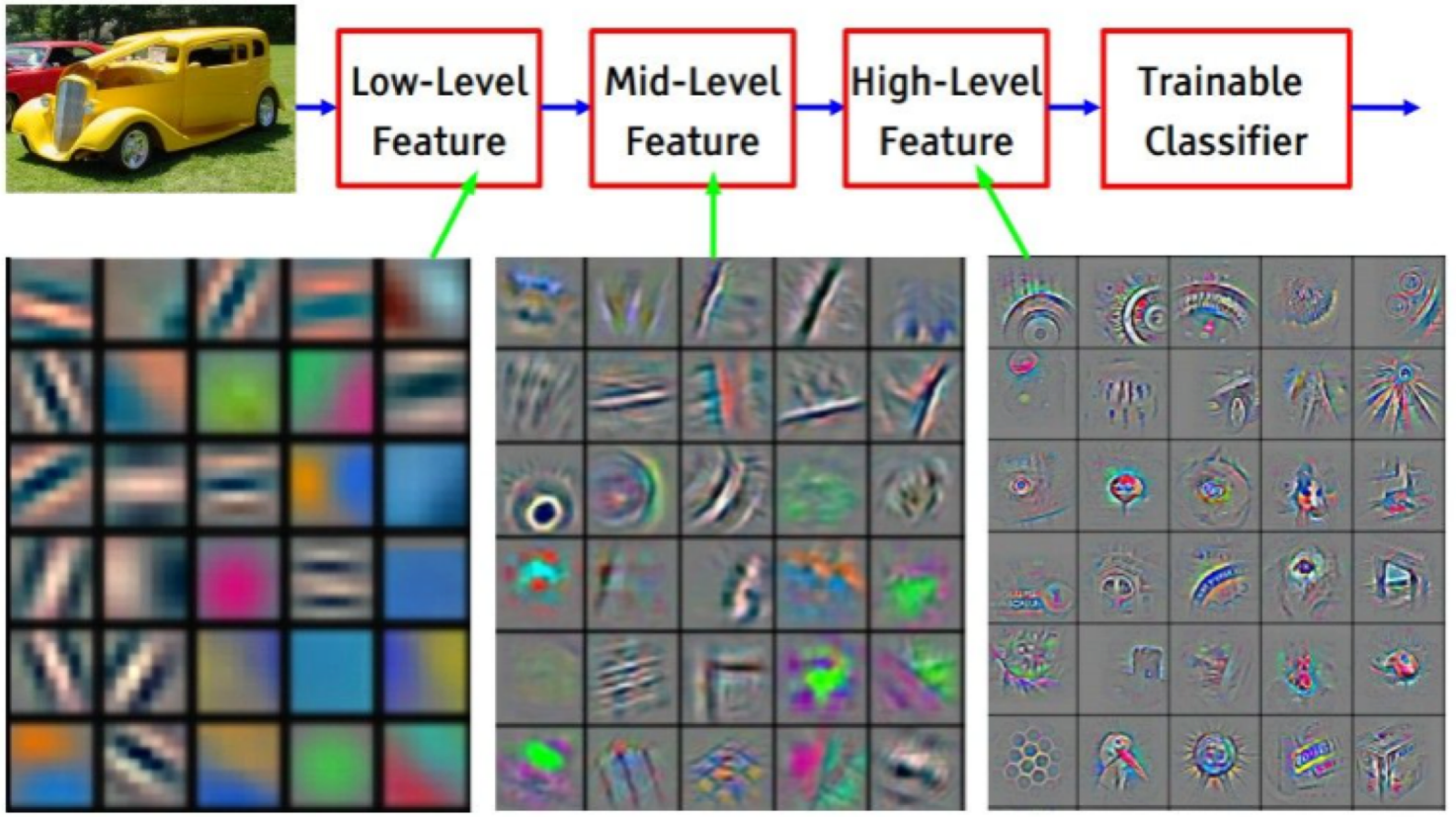

Layer 2: Top-9
• NOT SAMPLES FROM MODEL
• Just parts of input image that give strong activation of this feature map
• Non-parametric view on invariances learned by model


Layer 2: Top-9 Patches
Patches from validation images that give maximal activation of a given feature map


Layer 3: Top-9

Pretrained models as feature extractors

- Convolutional layers act as linear and non-linear filters able to **extract important features** for the demanded task

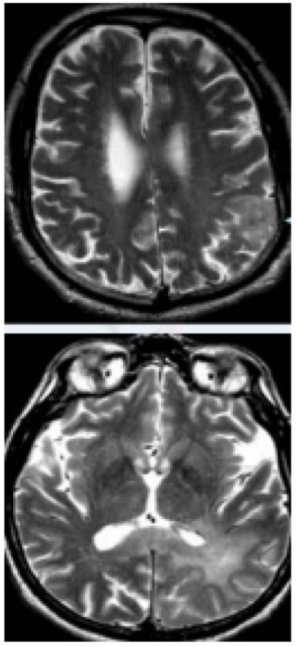


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Exercise

- Suppose you have the task of creating a model to classify what is the **fish species**. What strategy would you adopt to train the model?

- And a model to classify if a **brain MRI image** presents some tissue potentially with cancer



Pretrained models as feature extractors

- Convolutional layers as first block for another trainable computer vision problem → **transfer learning**
- Change the model slightly for a similar problem with same or different classes → **fine tuning**
- CNN trained as a vector representation of the image (**embedded vector**) and use it as input in other vision or non-vision related tasks, e.g.: captioning, translating, etc.
- Nowadays there are many **pretrained models** in all the development platforms