Lecture Notes for

Neural Networks and Machine Learning

CNN Visualization

Circuits

Logistics and Agenda

- Logistics
    - Student Presentations (worksheet)
        - If distance, can submit one page summary, rather than presentation
- Agenda
    - Visualizing Convolutional Architectures
    - Circuits in CNNs (time permitting)

CNN Visualization

Paper Gestalt

Carven von Bearnensquash
Department of Computer Science
University of Phoenix
bearensquash@live.com

Abstract

Peer reviews of conference paper submissions is an integral part of the research cycle, though it has unknown origins. For the computer vision community, this process has become significantly more difficult in recent years due to the volume of submissions. For example, the number of submissions to the CVPR conference has tripled in the last ten years. For this reason, the community has been forced to reach out to a less than ideal pool of reviewers, which unfortunately includes uninformed junior graduate students, disgruntled senior graduate students, and tenured faculty. In this work we take the simple intuition that the quality of a paper can be estimated by merely glancing through the general layout, and use this intuition to build a system that employs basic computer vision techniques to predict if the paper should be accepted or rejected. This system can then be used as a first cascade layer during the review process. Our results show that while rejecting 15% of "good papers", we can cut down the number of "bad papers" by more than 50%, saving valuable time of reviewers. Finally, we fed this very paper into our system and are happy to report that it received a posterior probability of 88.4% of being "good".

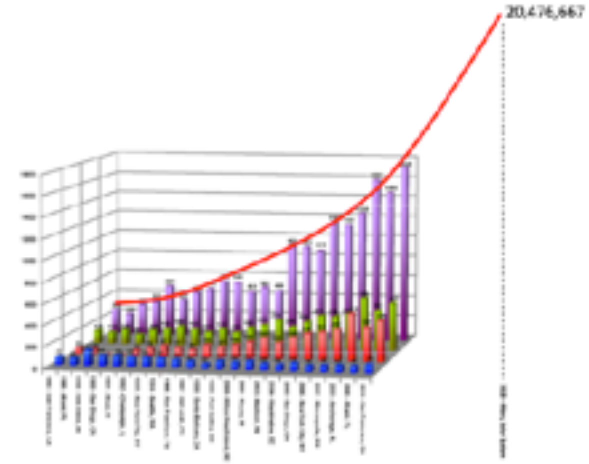


Figure 1. **Paper submission trends.** The number of submitted papers to CVPR, and other top tier computer vision conferences, is growing at an alarming rate. In this paper we propose an automated method of rejected sub-par papers, thereby reducing the burden on reviewers.

and tenured faculty. Although many excellent research pa-

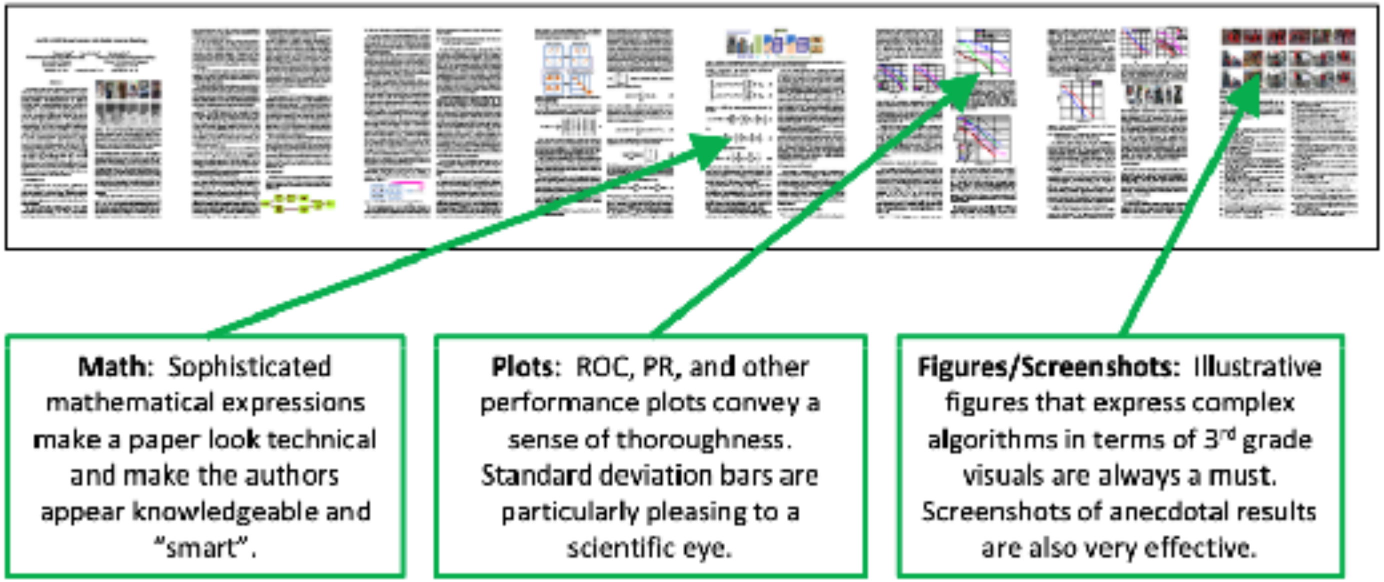


Figure 6. Characteristics of a "Good" paper.

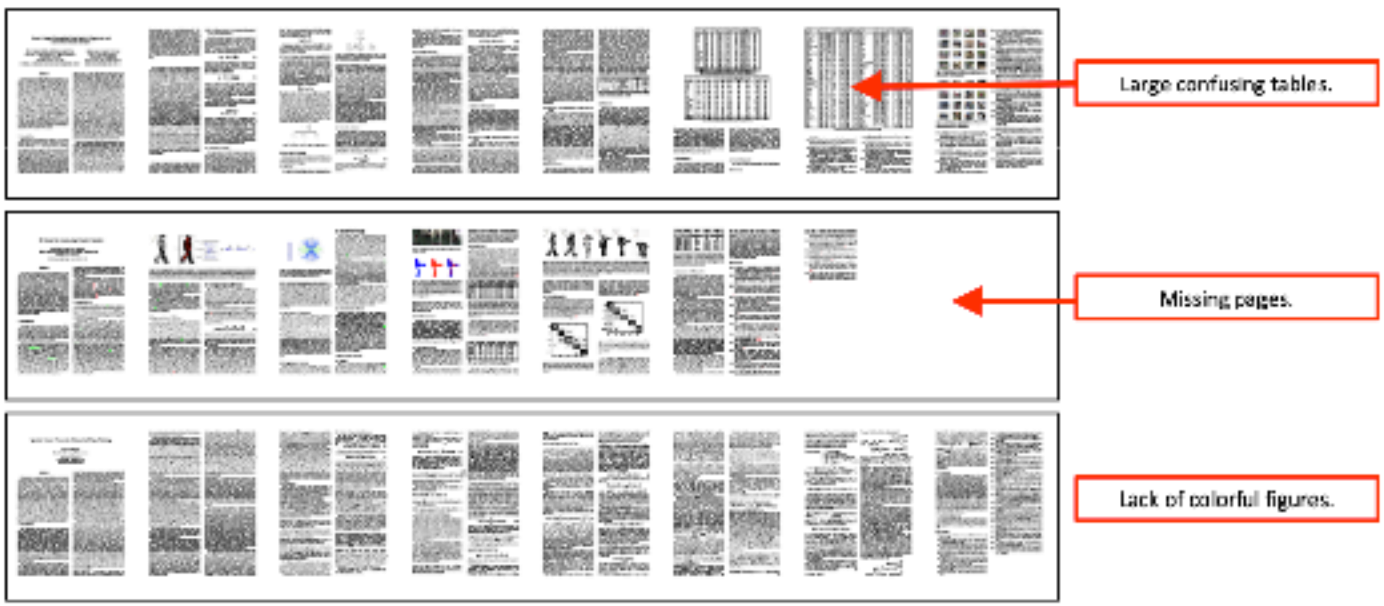
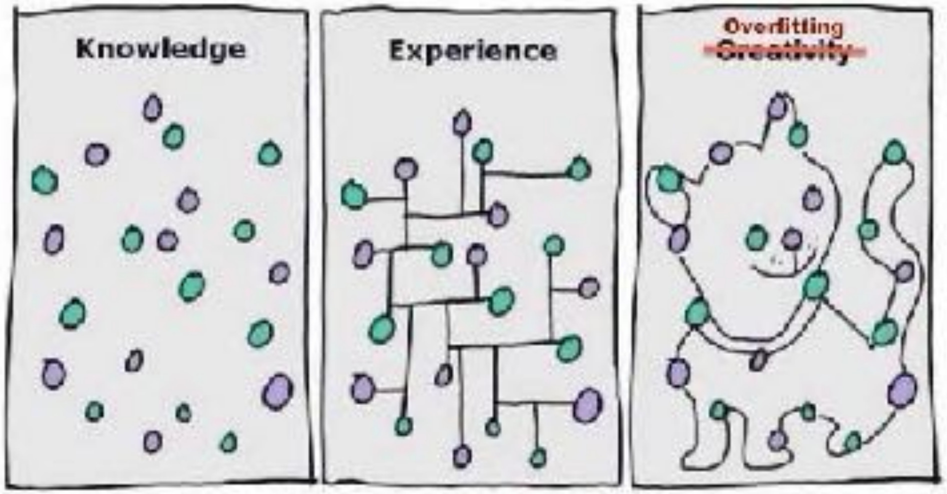


Figure 7. Characteristics of a "Bad" paper.

Basics of Convolutional Neural Network Visualization

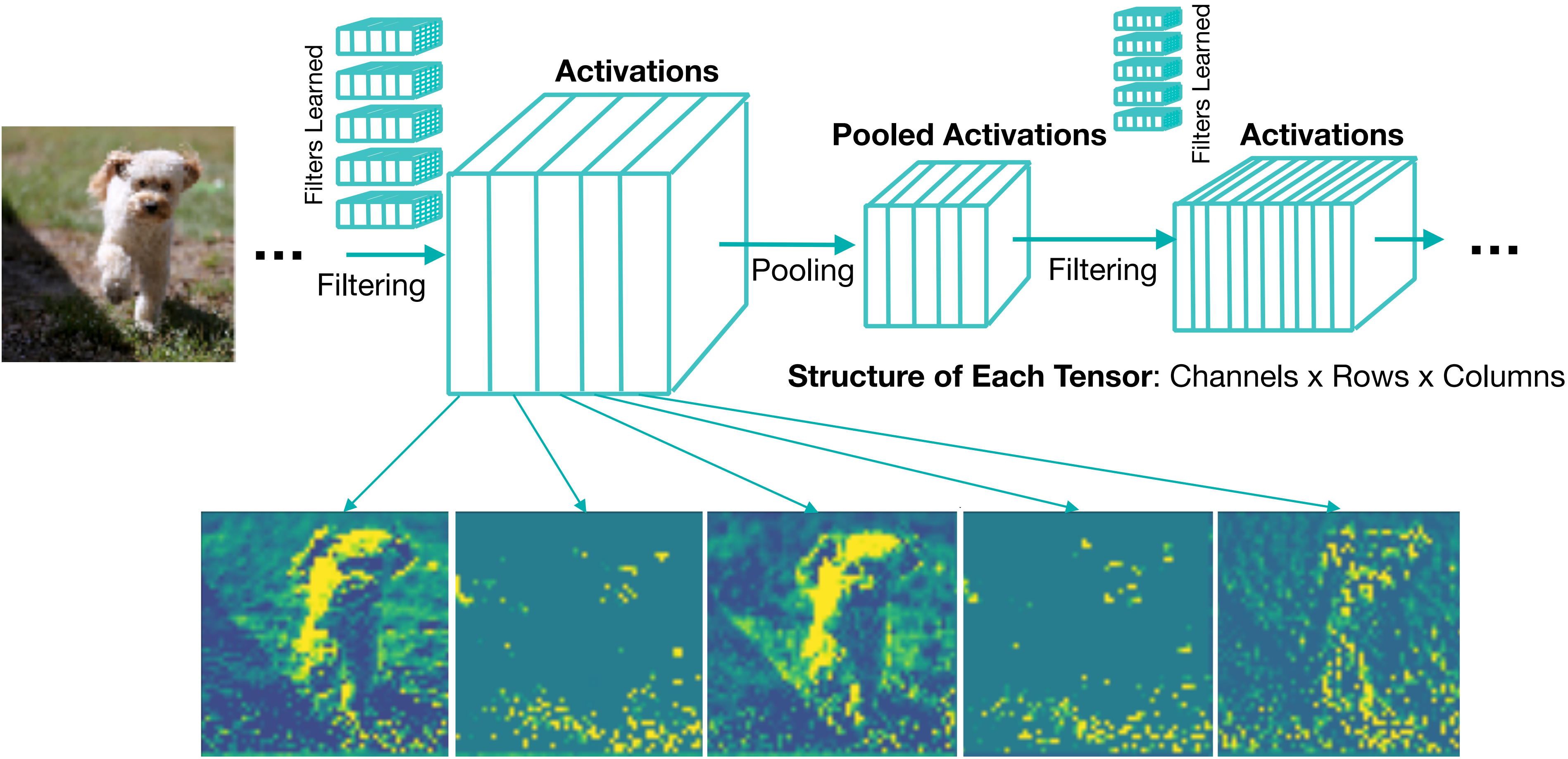


Tools to Visualize Neurons and Filters

- Visualize **Filter Activation Maps**
  - What parts of the inputs activate each filter?
- Visualize **Filters**
  - What does each filter look like? Is it similar to other filters?
  - Can we excite a certain filter by updating the input image?
- **Heatmaps** of Class Activation
  - What part of an input image most influences each final output?

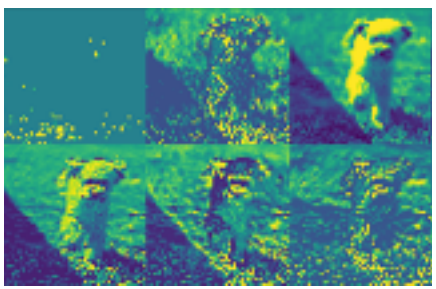
Visualizing Intermediate Activations

- Look layer by layer
- **Assume**: each filter learns something useful



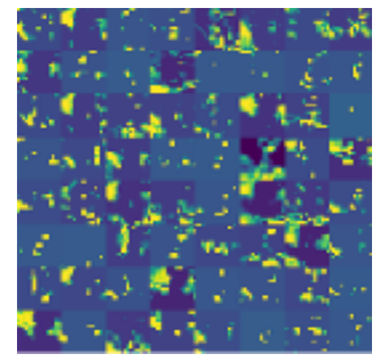
Visualizing Intermediate Activations

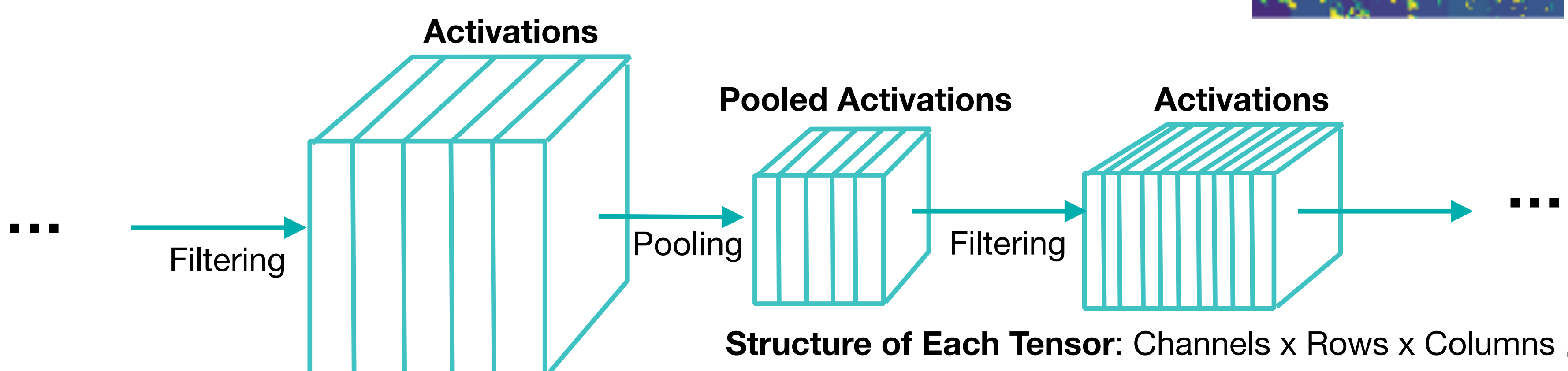
- **Recall**: general structure of most CNNs
  - Small kernels throughout (3x3)
  - Filtering followed by Pooling (spatial downsampling)
  - More filters in later layers



Early Activations are larger but not as numerous

Later Activations are smaller and more numerous

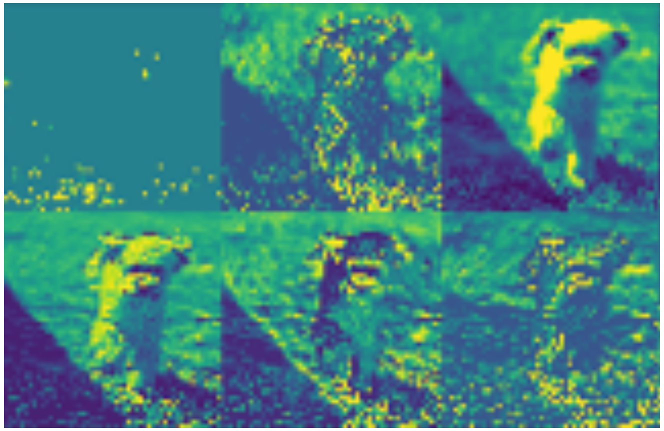




Structure of Each Tensor: Channels x Rows x Columns

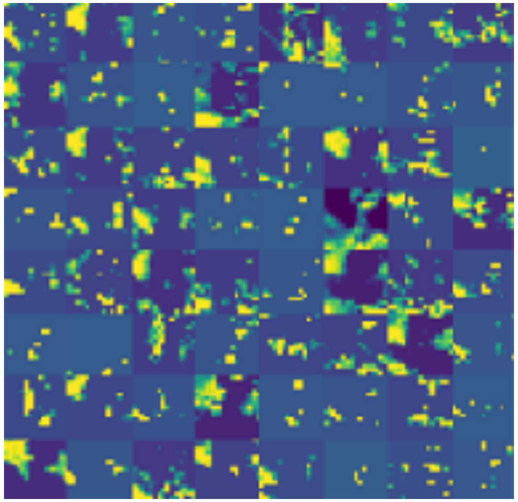
Visualizing Intermediate Activations

- **Result:** Information Distillation Pipeline
  - Deeper layers have more abstract triggers
  - Deeper activations are increasingly sparse
  - Early layers are texture and edge detectors
  - Notion of "High Level Abstraction," has biological motivation



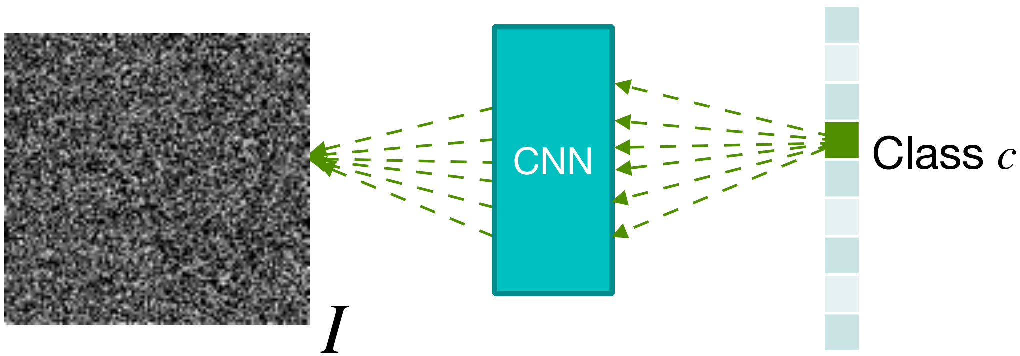
Early Activations are larger but not as numerous

Later Activations are smaller and more numerous

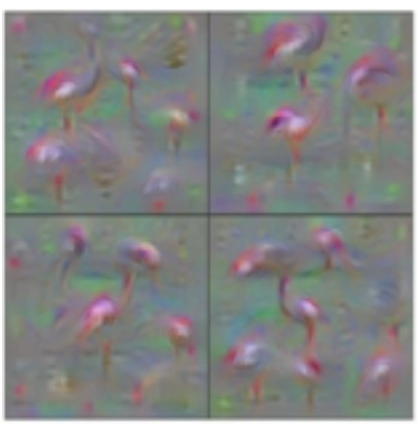


Visualizing Filters: Class Neuron

- **Idea:** What Maximally Activates a Class Output?
  - Gradient Ascent in the Input Space



$$I \leftarrow I + \eta \nabla f_c(I)$$


Flamingo

where c is a specific neuron in output layer

f is the activation before softmax

I is the input image, init to zeros (or random)

gradient update is for I

CNN weights stay unchanged

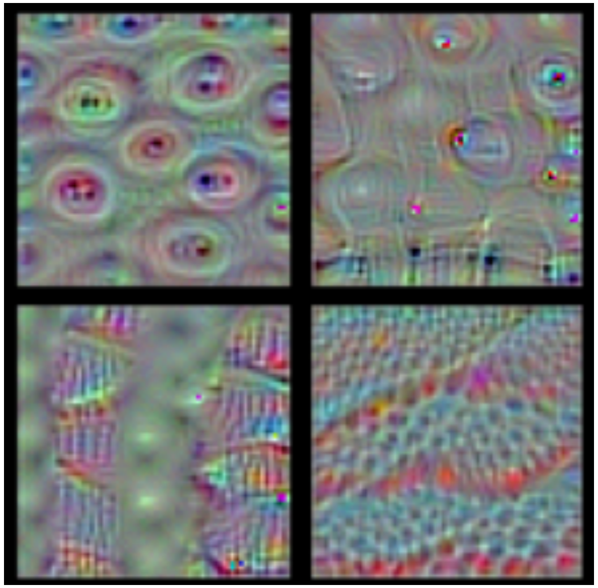
http://cs231n.github.io/understanding-cnn/

Visualizing Filters: Maximal Activations

- **Idea:** What Maximally Activates a **Filter**?
  - **Again**: Gradient Ascent in the Input Space

$$I \leftarrow I + \eta \sum_{i,j} \nabla f_n(I)_{i,j}$$

"trick" must use norm of gradient



where n is a specific **filter** in a layer

f is the activation of n^{th} filter in layer

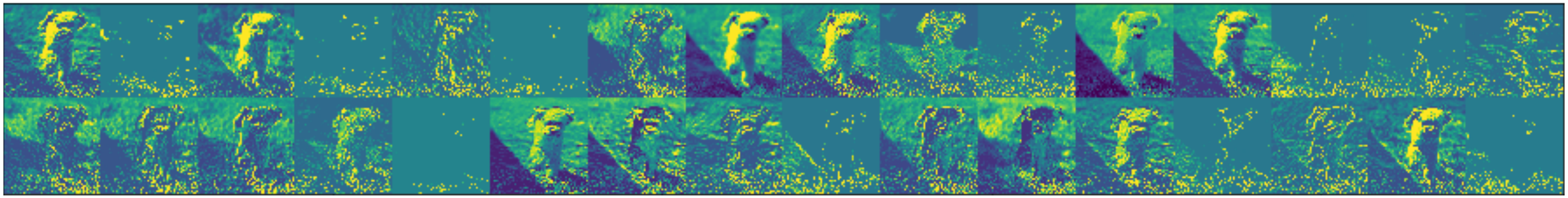
I is some random image, or zeros

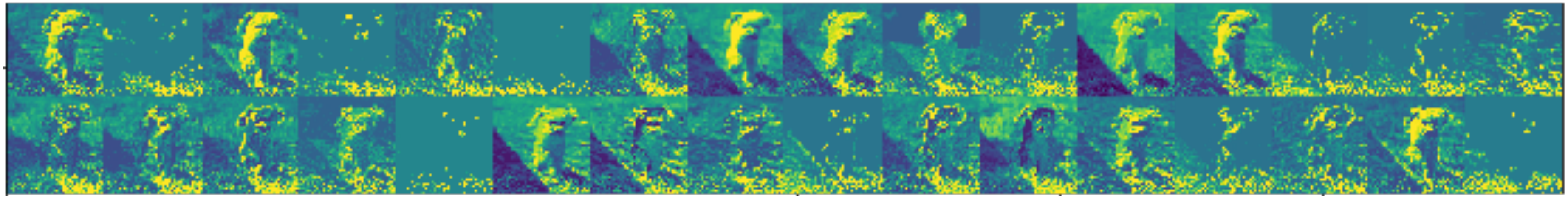
gradient update is for I

Visualizing ConvNets

Part One: Filter Activations

Part Two: Image Gradients





```
Follow Along: LectureVisualizingConvnets.ipynb
activation-demo
```

Class Activation Mapping (CAM)

- **Idea:** What areas of the image contributed most to the classification result?
- Also, for each class, what areas of the image exhibit features of that class?
- Use change in output, w.r.t. final conv layer

final layer output in response to image I
c is class of interest

$$\alpha_k^c = \frac{1}{|A_k^{(L)}|} \sum_{i,j} \frac{\partial f_c(I)}{\partial A_{i,j,k}^{(L)}}$$

final convolutional layer, L, activations
for row, column, channel

gradient weight for channel k and class c in layer L
k in $1 \ldots K$ activations in final layer

Ramprasath et al., **Grad-CAM:** Visual Explanations from Deep Networks via Gradient-based Localization

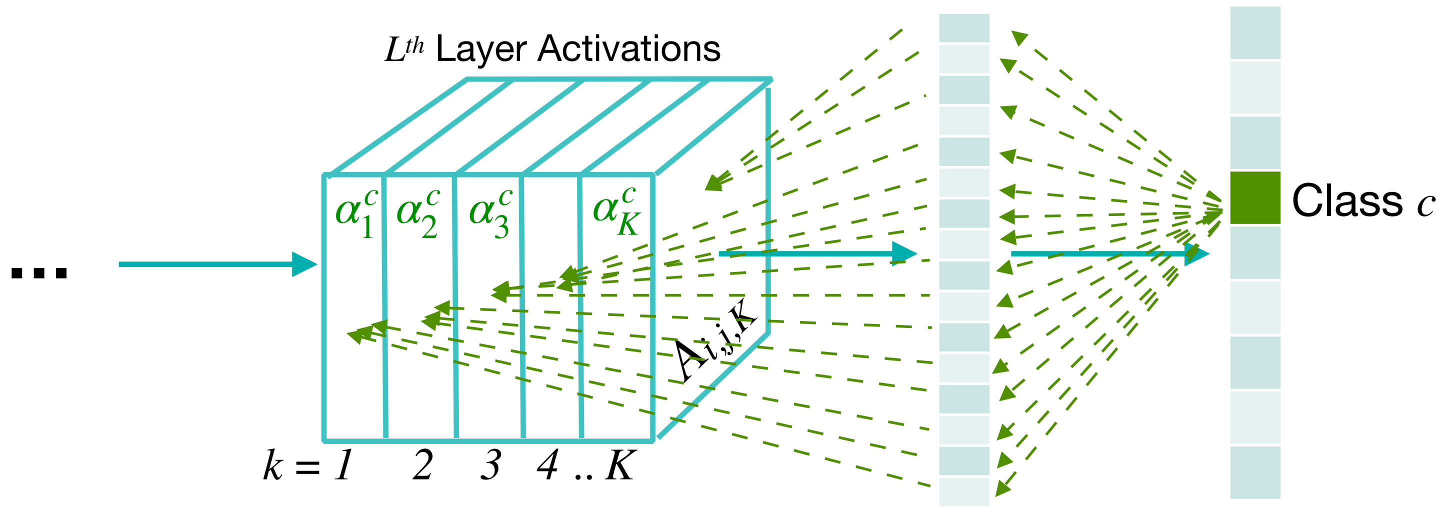
Class Activation Mapping (CAM)

$$\alpha_k^c = \frac{1}{|I \times J|} \sum_{i,j} \frac{\partial f_c(I)}{\partial A_{i,j,k}^{(L)}}$$

final layer output in response to image I
c is class of interest

final convolutional layer, L, activations
for row, column, channel

gradient weight for channel k and class c in layer L
k in $1 \ldots K$ activations in final layer



Sensitivity of Class to Activations

Ramprasath et al., **Grad-CAM:** Visual Explanations from Deep Networks via Gradient-based Localization

Class Activation Mapping (CAM)

$$\alpha_k^c = \frac{1}{|I \times J|} \sum_{i,j} \frac{\partial f_c(I)}{\partial A_{i,j,k}^{(L)}}$$

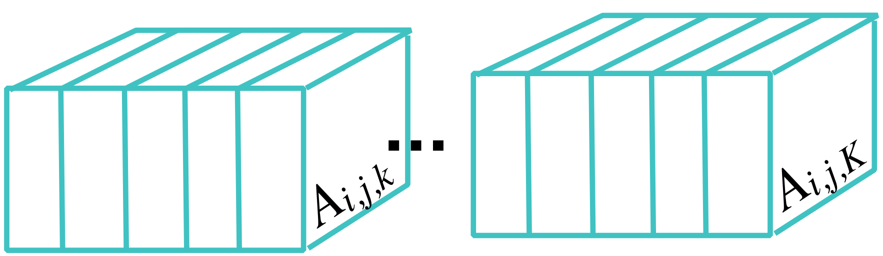
final layer output in response to image I
c is class of interest

final convolutional layer, L, activations
for row, column, channel

gradient weight for channel k and class c in layer L
k in $1 \ldots K$ activations in final layer

Heatmap, S, is the **weighted sum** of final layer activations:

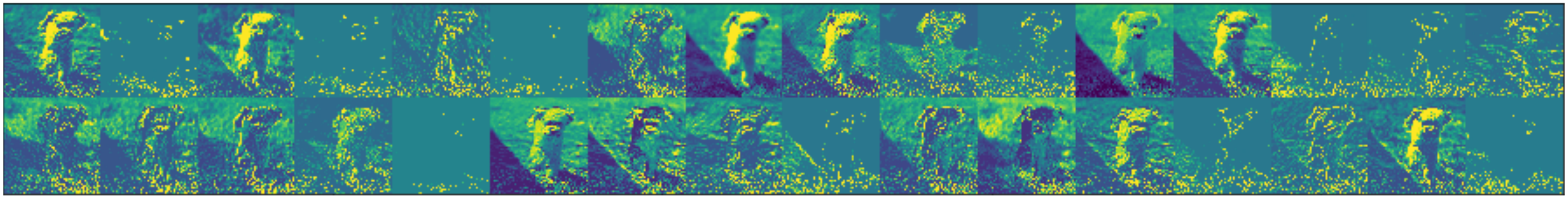
$$S_{i,j} = \frac{1}{S_{max}} \sum_k \alpha_k^c A_{i,j,k}^{(L)}$$

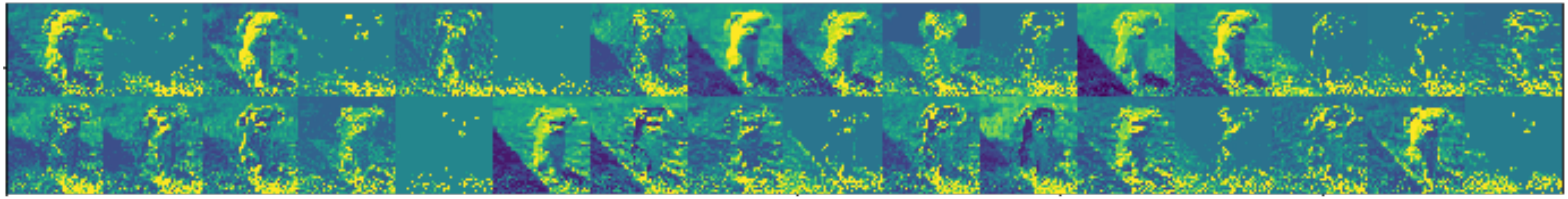


Ramprasath et al., **Grad-CAM:** Visual Explanations from Deep Networks via Gradient-based Localization

Visualizing ConvNets

Part Three: Grad-CAM





Follow Along: LectureVisualizingConvnets.ipynb
activation-demo