# CS396: Selected CS2

## (Deep Learning for visual recognition)

**Spring 2022**


**Dr. Wessam EL-Behaidy**

Associate Professor, Computer Science Department,
Faculty of Computers and Artificial Intelligence,
Helwan University.


**Lectures (Course slides) are based on** Stanford course : Convolutional Neural Networks for Visual Recognition (CS231n): [http://cs231n.stanford.edu/index.html](http://cs231n.stanford.edu/index.html)

# Lecture 9: Visualization

# Visualization

For understanding how our deep CNN model is able to classify the input image, we need to understand how our model sees the input image by using **visualization.**

Three techniques can be used for learning about conv-nets through visualization:

(1) **Visualizing Intermediate Layer Activations**
(2) **Visualizing ConvNet Filters**
(3) **Visualizing Heat maps of class activations**

# (1) Visualizing Intermediate Layer Activations

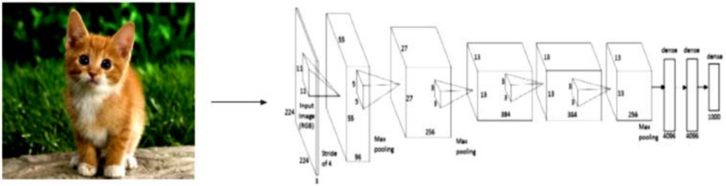
Examine the **activations** (i.e. feature maps) and discover which features the network learns by comparing areas of activation with the original image.

# Visualization with a Deconvnet

Understanding the operation of a convnet requires interpreting the feature activity in intermediate layers.

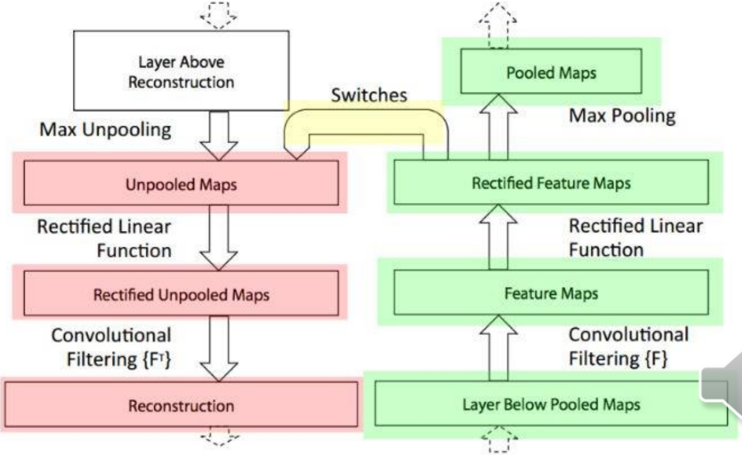
Zeiler et al. showing what input pattern originally caused a given activation in the feature maps with a **Deconvolutional Network (deconvnet)**.

1. Feed image into net



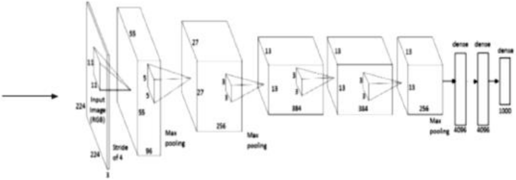
# Visualization with a Deconvnet

2- Pick a layer, set the gradient there to be all zero except for one 1 for some neuron of interest.
3- Pass unpooled map through ReLU.
4- Deconvolution using the transpose of the original filters

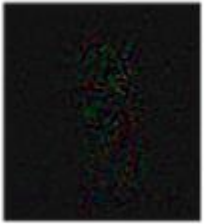


# Backpropagation Vs Guided Backpropagation

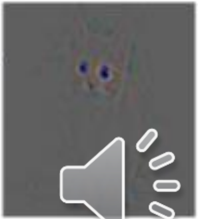
1. Feed image into net




Backprop to image:



**"Guided backpropagation:"** instead



# ReLU in Backward Pass

Images come out nicer if you only backprop positive gradients through each ReLU (guided backprop)

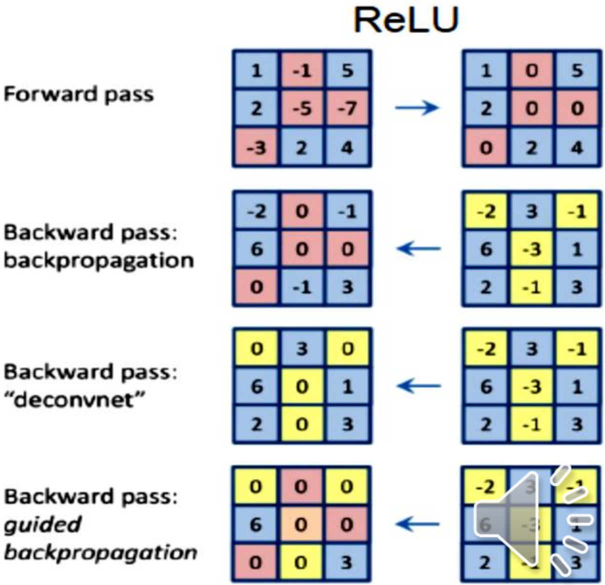
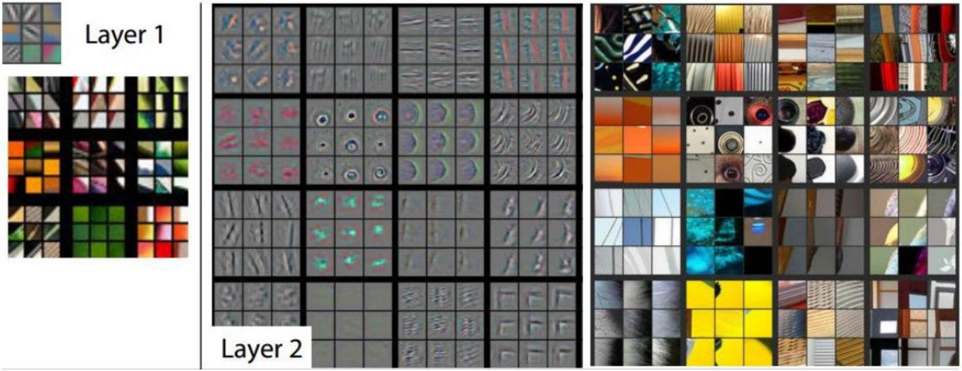



Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015.

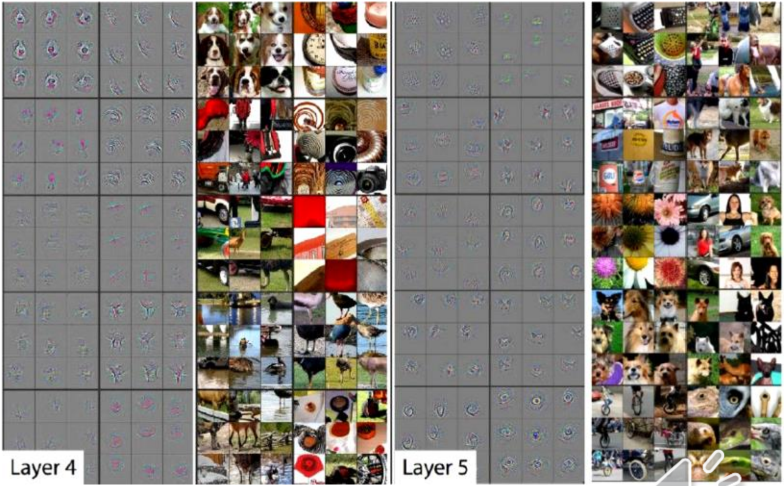
# DeconvNet Visualizing arbitrary neurons




*Visualizing and Understanding Convolutional Networks, Zeiler & Fergus, 2013*

# DeconvNet Visualizing arbitrary neurons

Visualizing arbitrary neurons along the way to the top...



# Visualizing Intermediate Layer Activations

The convolutional layers perform convolutions with learnable parameters. The network learns to identify useful features, often with one feature per channel (i.e filters, or kernels ).

**For example:**

We choose to observe the first convolutional layer which has 64 channels in used model.

# Activations of specific channel in conv1

All activations are scaled so that the minimum activation is 0 and the maximum is 1. **White pixels** represent strong positive activations and **black pixels** represent strong negative activations. You can see that this channel activates on red pixels, because the whiter pixels in the channel correspond to red areas in the original image.

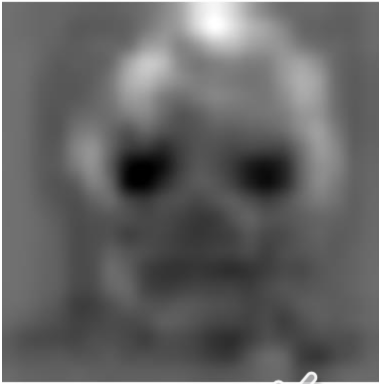
# Find the Strongest Activation Channel

Find the channel with the largest activation.
**Notice that** this channel activates on edges. It activates positively on light left/dark right edges, and negatively on dark left/light right edges

# Investigate a Deeper Layer

Display the **strongest** activation in the conv5 layer.

In this case, the maximum activation channel is not as interesting for detailed features as some others, and shows strong negative (dark) as well as positive (light) activation. This channel is possibly focusing on faces.



# Investigate a Deeper Layer

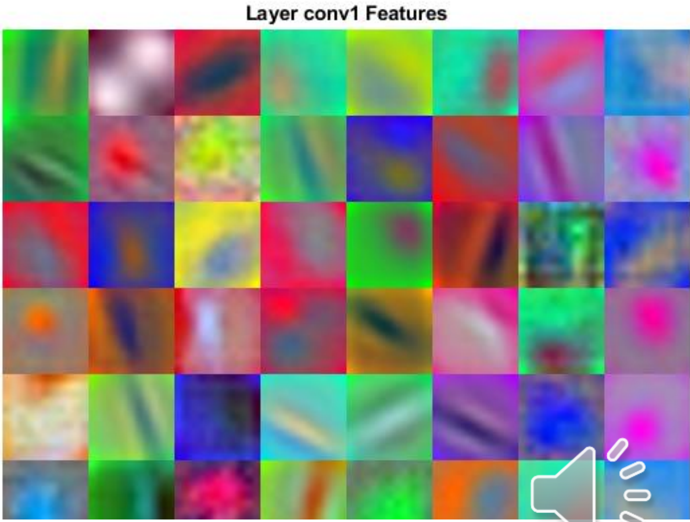
There are channels that might be activating on eyes.
Many of the channels contain areas of activation that are both light and dark. However, only the positive activations are used because of the rectified linear unit (ReLU) that follows the conv5 layer.

# (2) **Visualizing ConvNet Filters**

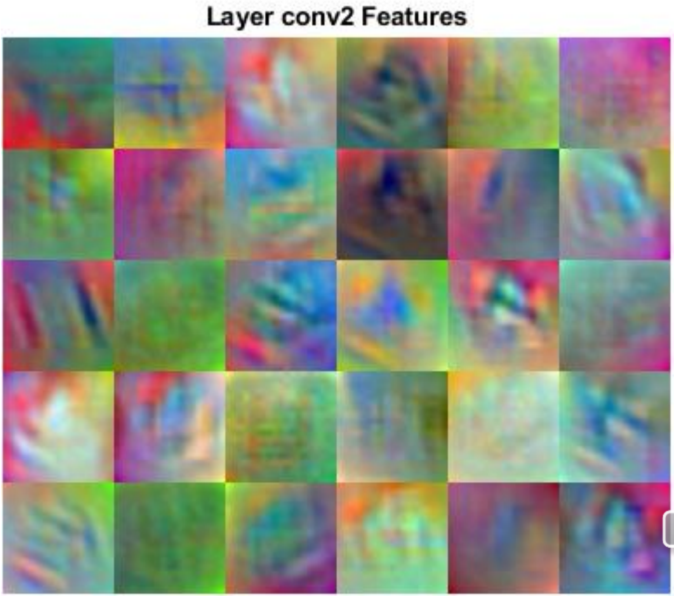
Find out that **channels** (i.e. filters) in earlier layers learn simple features like color and edges, while channels in the deeper layers learn complex features like eyes.

# Features of 1$^{st}$ convolutional layer

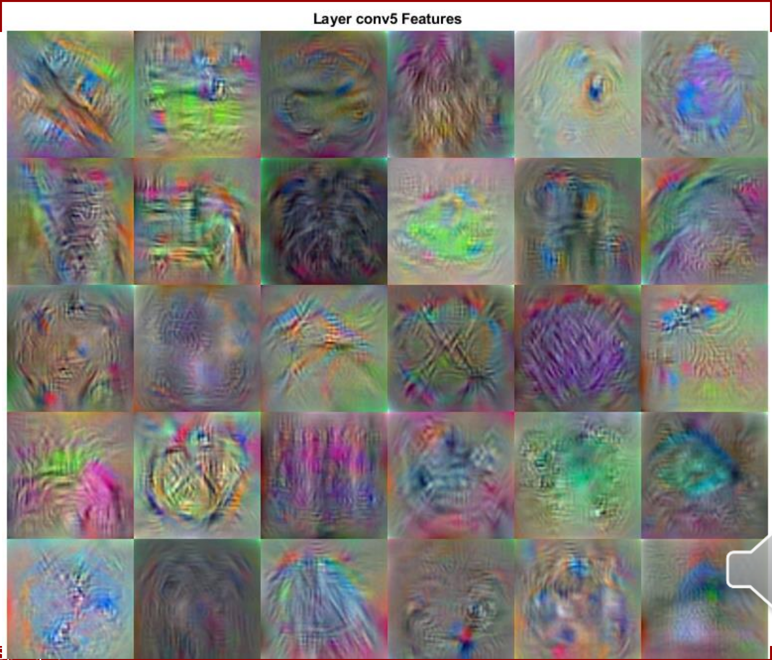
The features of the first convolutional layer mostly contain edges and colors, which indicates that the filters at layer 'conv1' are edge detectors and color filters. The edge detectors are at different angles, which allows the network to construct more complex features in the later layers.



# Features of 2nd convolutional layer



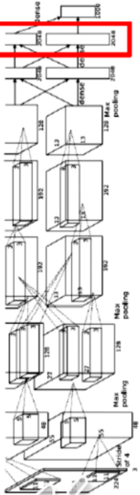
# Features of 5th convolutional layer



# Last Layer

Last Layer

FC7 layer



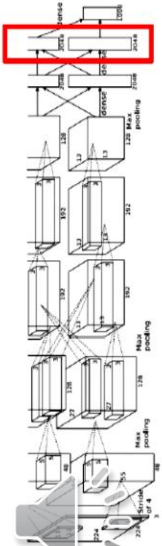
4096-dimensional feature vector for an image
(layer immediately before the classifier)

Run the network on many images, collect the feature vectors

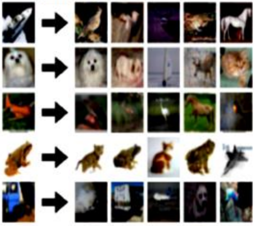
# Last Layer

## Last Layer: Nearest Neighbors

4096-dim vector



**Recall**: Nearest neighbors in pixel space



Test image L2 Nearest neighbors in feature space

Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.

# (3) Visualizing Heat Maps of class activations

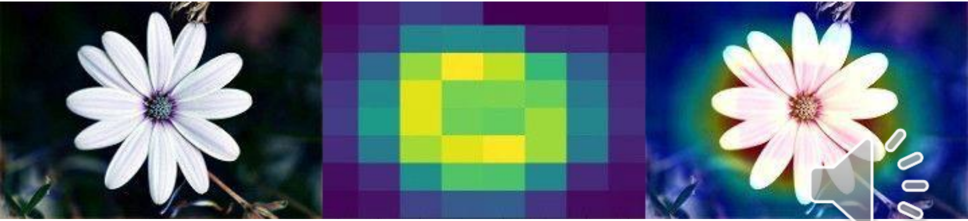
While predicting the class labels for images, sometimes your model will predict wrong label for your class, i.e. the probability of the right label will not be maximum. In cases such as these, it will be helpful if you could visualize which parts of the image is your convNet looking at and deducing the class labels.

**Class Activation Map** (CAM) visualization:
One of the techniques of using CAM is to producing heatmaps of class activations over input images.
A class activation heatmap is a 2D grid of scores associated with a particular output class, computed for every location for an input image, indicating how important is each location is with respect to that output class

# CAM visualization

So basically, the heatmap is trying to tell us the locations in the image which are important for that particular layer to classify it as the target class, which is **Daisy** in this case**.**

It is pretty clear that the network has no problem in classifying the flower, as there are no other objects in the entire image.



# CAM visualization

In this image, the network could not classify the image as **Daisy,** but if you take a look at the heatmap of the activation map, it is clear that the network is looking for the flowers in the correct parts of image.

# This lecture references


- Dr. Ghada's Slides of Pattern recognition course Spring 2018
http://www.fcih.net/ghada/pattern-recognition/

- https://towardsdatascience.com/understanding-your-convolution-network-with-visualizations-a4883441533b
- https://www.mathworks.com/help/deeplearning/examples/visualize-activations-of-a-convolutional-neural-network.html
- Visualizing and Understanding Convolutional Networks, Matthew D. Zeiler and Rob Fergus, Dept. of Computer Science, New York University, USA