भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Visualizing the Deep CNN Models

Konda Reddy Mopuri

Agenda

- Motivation
  - Why do we need to visualize ?
  - What are the implications of it ?
- Some early/interesting attempts

Motivation

Why do we need to visualize ?

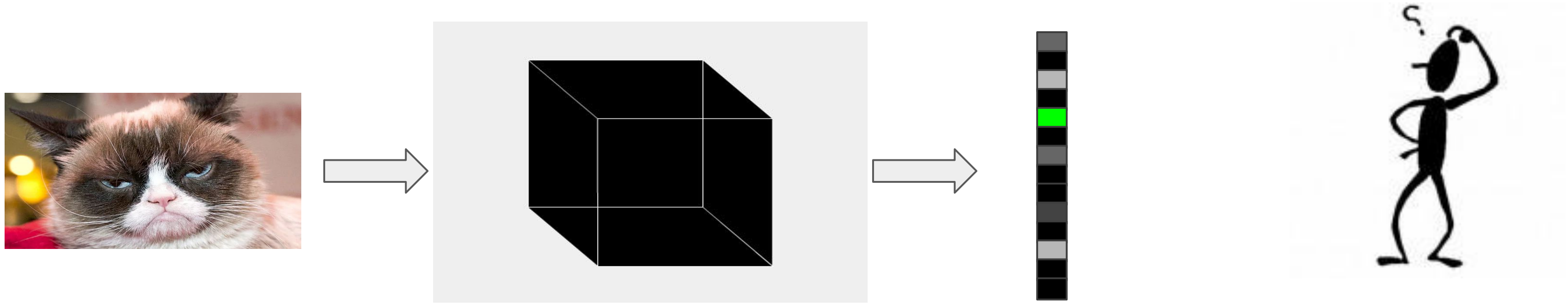
CNNs - complex ML systems

- CNNs is a success story
- However, they are complex models
- 10s of layers, 100s of feature maps, 100000000 of parameters

CNNs - what do they learn ?

CNNs are black boxes ?

- Often don't provide detailed information about the inference



Interpretability matters

- These CNN classifiers suffer
  - Lack of decomposability
  - No transparency
    - when they fail → no warning, no explanation
  - From the trade-off b/w “Accuracy” and “Interpretability”

What gets better ?

Implications

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Information supporting the inference

- Reason an inference
- E.g. Visual explanations

Can enable Human verification

- Incorrect predictions can be costly
    - Ex: Medical diagnosis, defence applications, etc.
- Predictions need to be verified by an expert

Approaches

Some of them

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

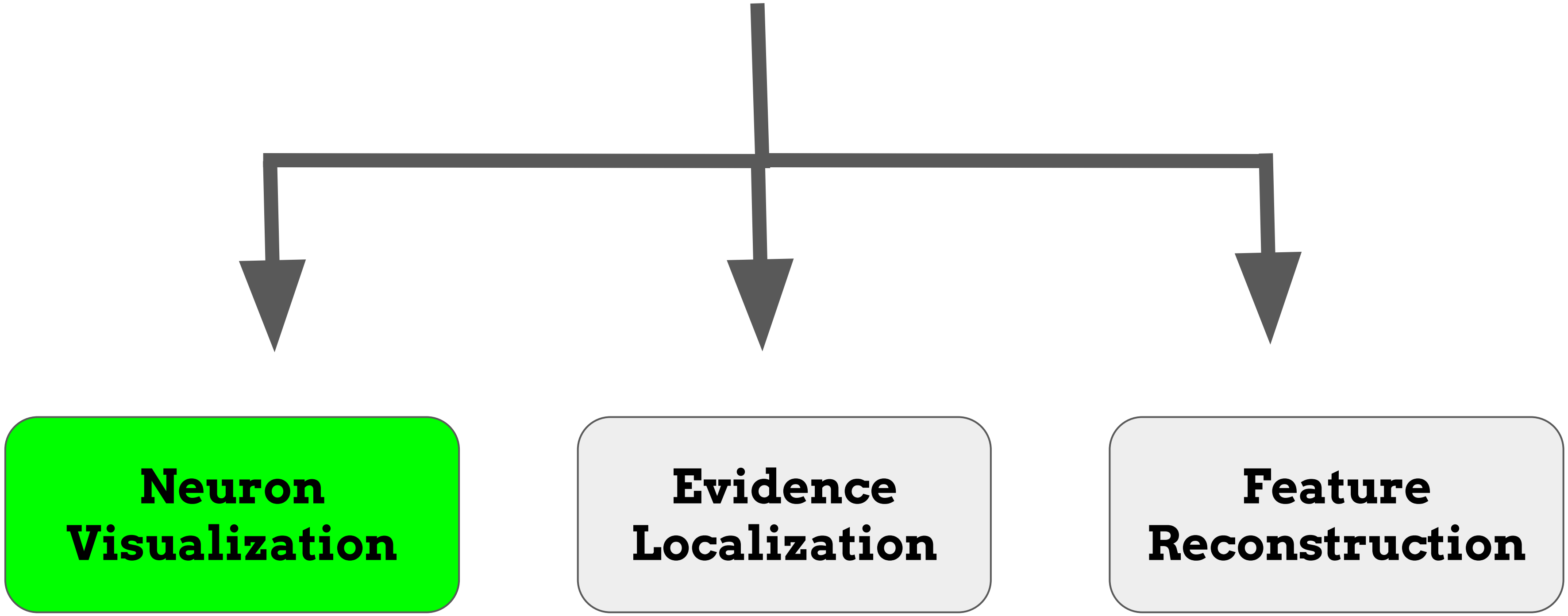
CNN Visualization

Neuron Visualization

Evidence Localization

Feature Reconstruction

CNN Visualization



RCNN

Neurons and stimuli

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

- What do the neurons learn ?

Ross Girshick et al. CVPR 2014

Neurons and stimuli

- Recognize visual attributes/concepts/topics

Ross Girshick et al. CVPR 2014

Visualizing with a Deconvnet

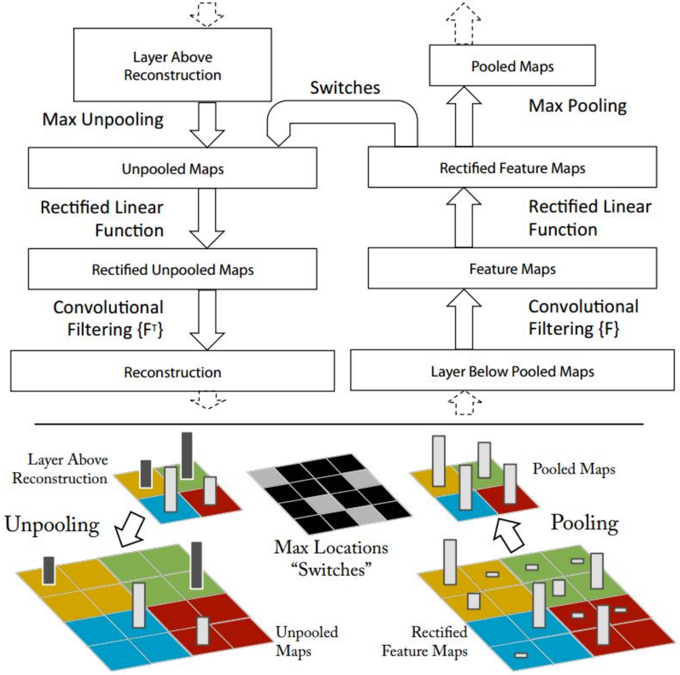
Visualizing with a Deconvnet

- Understanding a convnet requires interpreting the feature activity @differernt layers
- Map neuron activations onto i/p pixel space
- Show what patterns caused it

Zeiler et al. ECCV 2014

Visualizing with a Deconvnet

- Deconv layers
- Switches
- Transposed filters



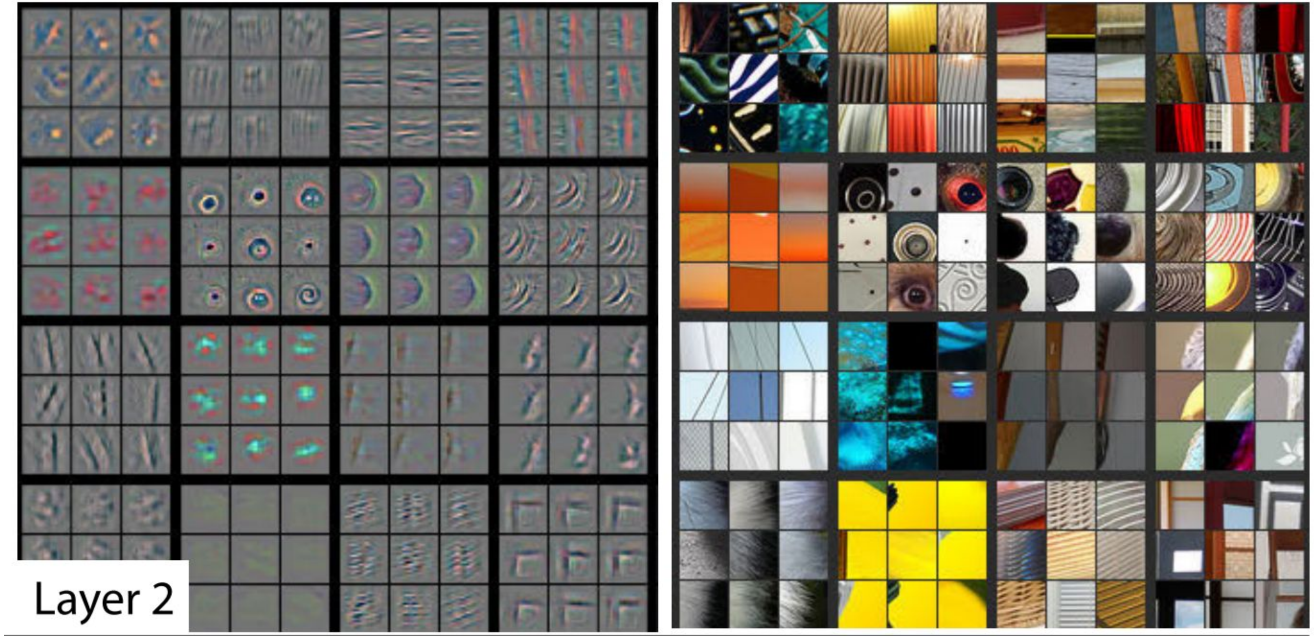
[Zeiler et al. ECCV 2014]

Visualizing with a Deconvnet

- Pick a unit to visualize
- Zero out all the remaining units in the layer
- Project back via the deconv layers

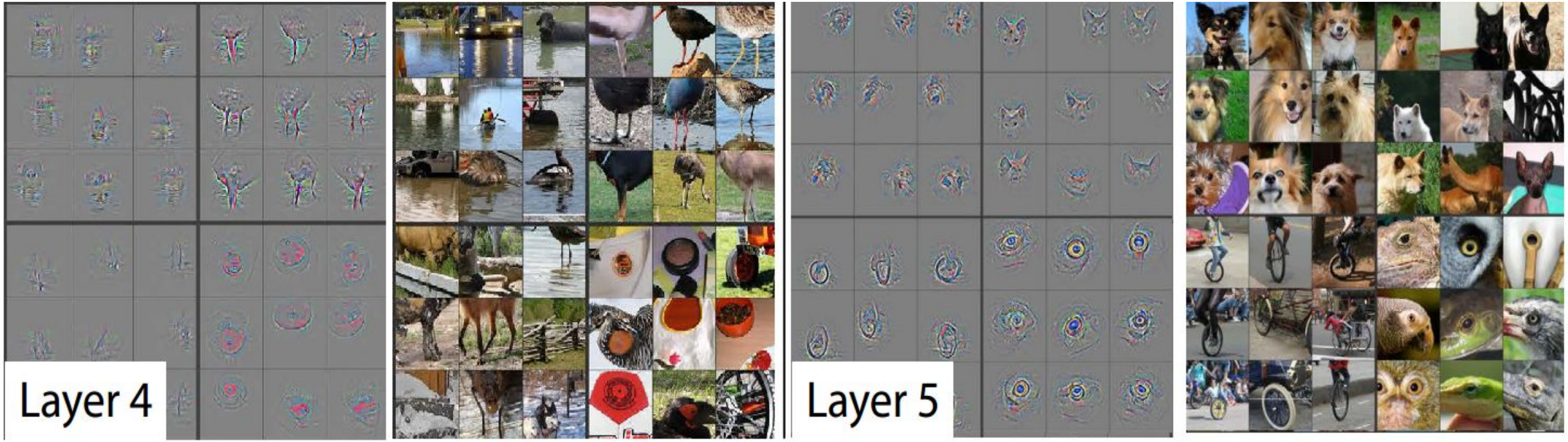
Zeiler et al. ECCV 2014

Visualizing with a Deconvnet



Zeiler et al. ECCV 2014

Visualizing with a Deconvnet



[Zeiler et al. ECCV 2014]

Observations

- Patches have greater variation than visualizations
- Strong grouping w/i feature map
- Greater invariance at higher layers
- Exaggeration of discriminative parts

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Visualization

- **Neuron Visualization**
- **Evidence Localization**
- **Feature Reconstruction**

Evidence localization

- Provide visual explanations
- Grounding the inference
- Ex: Classification network
  - Which pixels are responsible to the predicted label ?

Deep inside a CNN

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Deep inside CNNs

- Class model visualization
- Image-specific class saliency visualization

[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps [S Karenyan et al.] [ICLRW 2014]]

Class model visualization

- Numerically generate an image for chosen class

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2,$$

[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps [S Karenyan et al.] [ICLRW 2014]](#)

Class model visualization

- Numerically generate an image for chosen class

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2,$$

dumbbell cup dalmatian

[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps [S Karenyan et al.] [ICLRW 2014]]

Image specific visualization

- Query the CNN about the spatial support for a class
- Compute the gradients wrt the image

[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps [S Karenyan et al.] [ICLRW 2014]]

Image specific visualization

- Equivalent to performing gradient ascent on score function wrt image

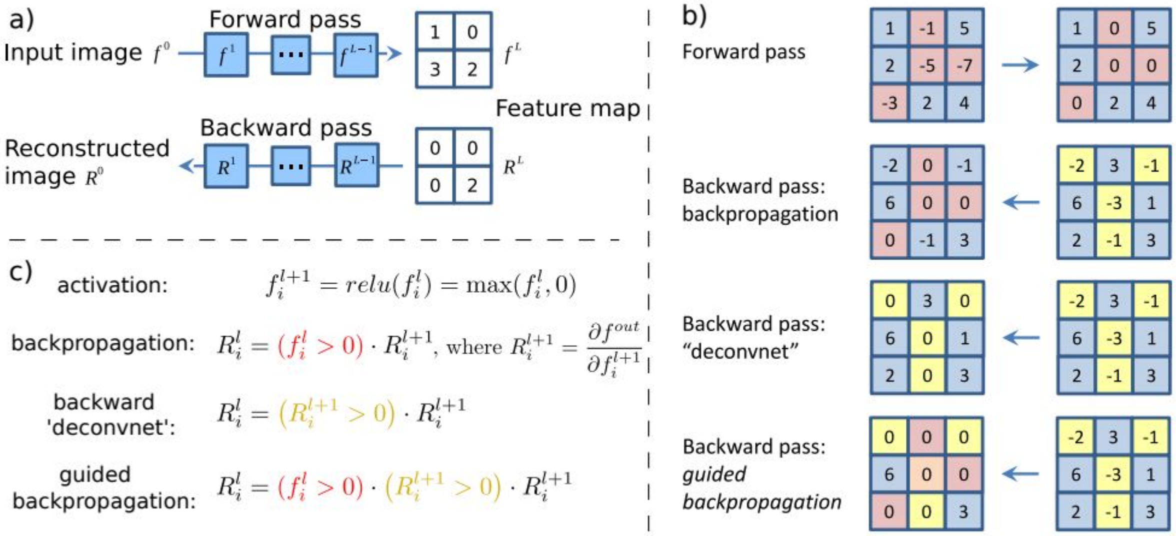
$$S_c(I) \approx w^T I + b \qquad\qquad w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}$$

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps [S Karenyan et al.] [ICLRW 2014]

Image specific visualization

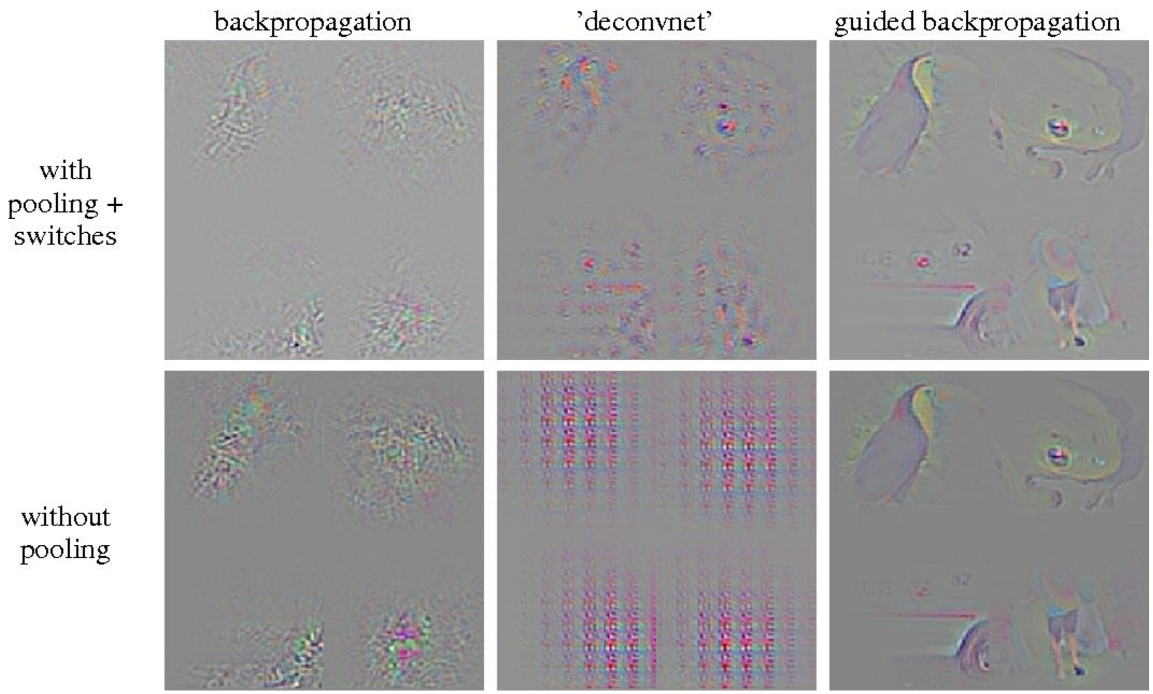
Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps [S Karenyan et al.] [ICLRW 2014]

Guided backprop for better reconstruction



[Guided backprop]() by Springenberg et al. ICLR 2015

Guided backprop for better reconstruction



Guided backprop by Springenberg et al. ICLR 2015

Class activation maps

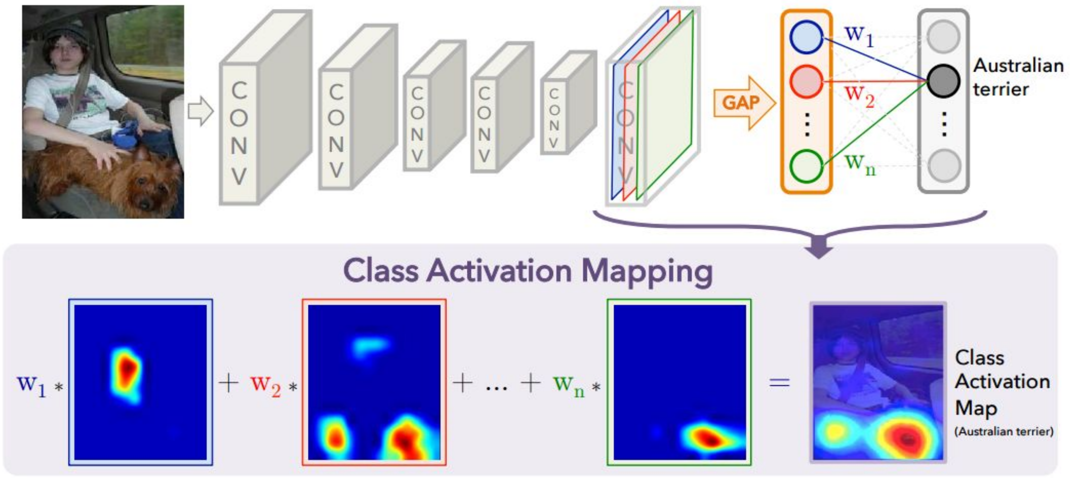
Class Activation Maps (CAM)

- Class discriminative image regions used by CNN to identify the category

B Zhou et al. Learning Deep Features for Discriminative Localization, CVPR 2016

Class Activation Maps (CAM)

- Perform GAP on the final conv feature map



[B Zhou et al. Learning Deep Features for Discriminative Localization, CVPR 2016]

Class Activation Maps (CAM)

- $f_k(x,y)$ - activation of unit 'k' in last conv layer at location (x,y)
- GAP → $F^k = \Sigma_{x,y} \; f_k(x,y)$
- Score predicted for class 'c' → S_c

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c f_k(x,y).$$

$$M_c(x,y) = \sum_k w_k^c f_k(x,y).$$

Class Activation Maps (CAM)

- $f_k(x,y)$ - activation of unit 'k' in last conv layer at location (x,y)
- GAP → $F^k = \Sigma_{x,y} \; f_k(x,y)$
- Score predicted for class 'c' → S_c

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c f_k(x,y).$$

$$\boxed{M_c(x,y) = \sum_k w_k^c f_k(x,y).}$$

Weighted summation of feature maps

Advantages of CAM

- Doesn’t require a backprop operation
- Class discriminative

Drawbacks of CAM

- CNN needs GAP in the architecture
    - If not, it needs retraining (last layer) with a GAP

WSL with CAM

- Remove fc layers
- Add GAP layer retrain
- Threshold the map
- Fit a BB
- Detection on ILSVRC 2012 validation set

Table 1. Classification error on the ILSVRC validation set.

Networks	top-1 val. error	top-5 val. error
VGGnet-GAP	33.4	12.2
GoogLeNet-GAP	35.0	13.2
AlexNet*-GAP	44.9	20.9
AlexNet-GAP	51.1	26.3
GoogLeNet	31.9	11.3
VGGnet	31.2	11.4
AlexNet	42.6	19.5
NIN	41.9	19.6
GoogLeNet-GMP	35.6	13.9

Table 2. Localization error on the ILSVRC validation set. *Back prop* refers to using [23] for localization instead of CAM.

Method	top-1 val.error	top-5 val. error
GoogLeNet-GAP	**56.40**	**43.00**
VGGnet-GAP	57.20	45.14
GoogLeNet	60.09	49.34
AlexNet*-GAP	63.75	49.53
AlexNet-GAP	67.19	52.16
NIN	65.47	54.19
Backprop on GoogLeNet	61.31	50.55
Backprop on VGGnet	61.12	51.46
Backprop on AlexNet	65.17	52.64
GoogLeNet-GMP	57.78	45.26

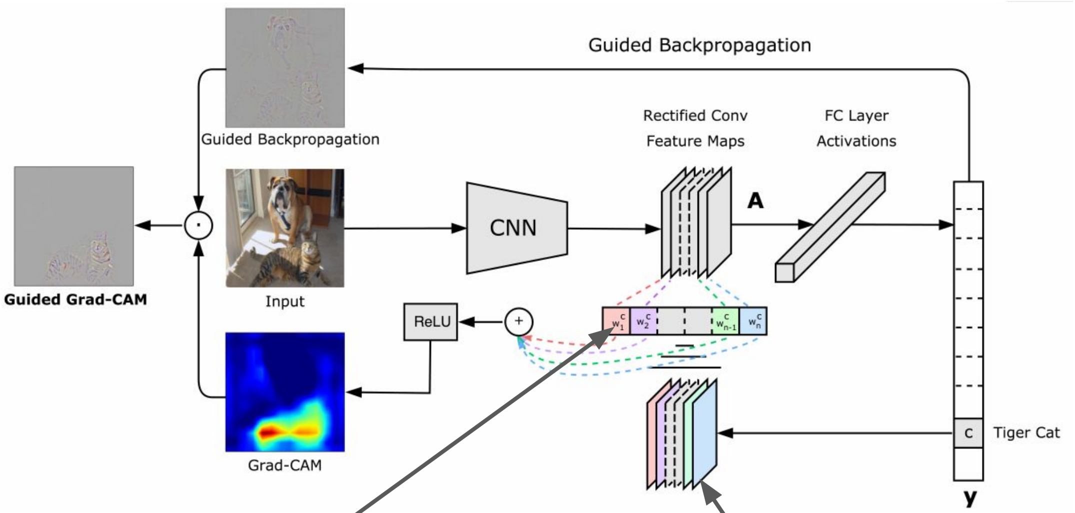
Gradient weighted CAM

Grad-CAM

- Gradient weighted CAM
- Combines class specific gradient info. with pixel visualization
- Generalizes CAM for all architectures

[Grad-CAM, NIPSW 2016, ICCV 2017]

Grad-CAM



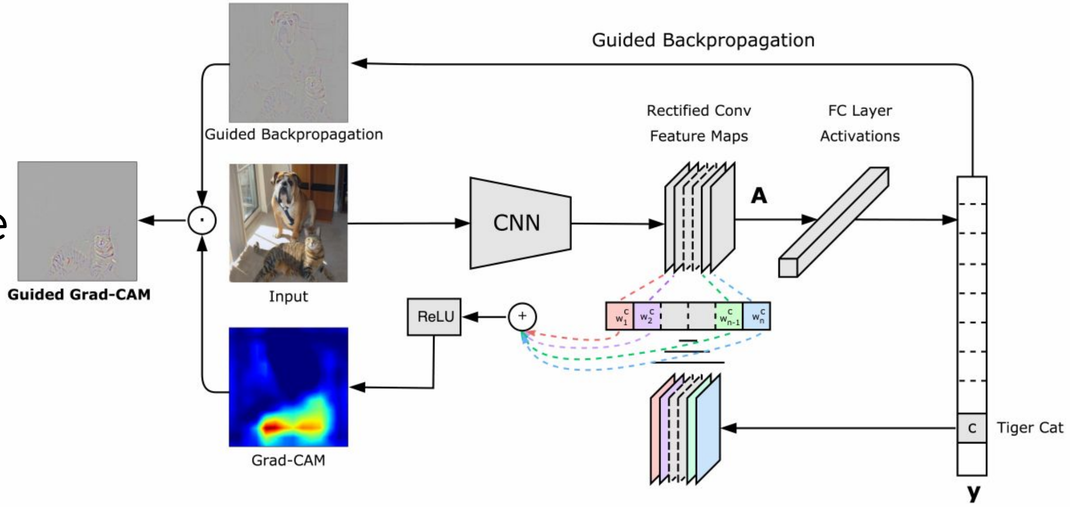
Sum of these gradient becomes the weight for feature map

Gradient of score w.r.t the conv layer o/p

[Grad-CAM, NIPSW 2016, ICCV 2017]

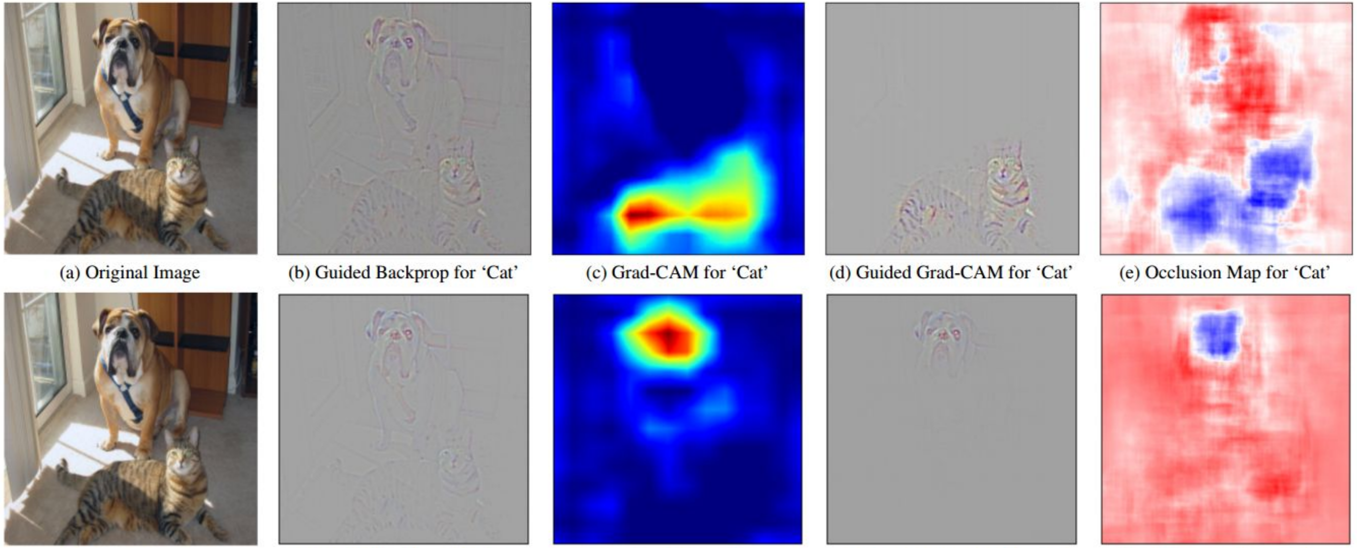
Grad-CAM

- Gives the weights to combine w/o GAP/retraining
- ReLU captures only the +ve correlations



[Grad-CAM, NIPSW 2016, ICCV 2017](#)

Grad-CAM results



(a) Original Image (b) Guided Backprop for 'Cat' (c) Grad-CAM for 'Cat' (d) Guided Grad-CAM for 'Cat' (e) Occlusion Map for 'Cat'

(f) Original Image (g) Guided Backprop for 'Dog' (h) Grad-CAM for 'Dog' (i) Guided Grad-CAM for 'Dog' (j) Occlusion Map for 'Dog'

[Grad-CAM, NIPSW 2016, ICCV 2017]

More results

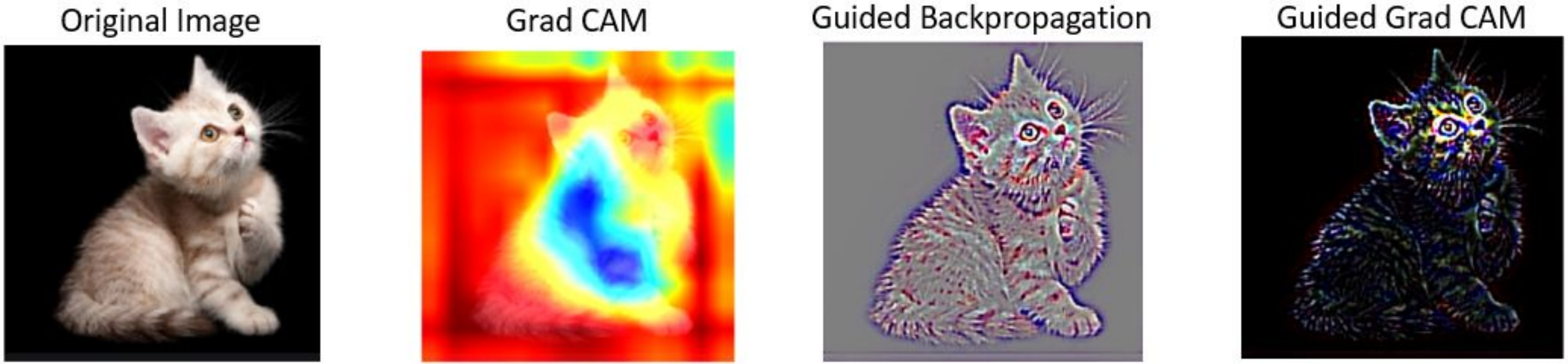
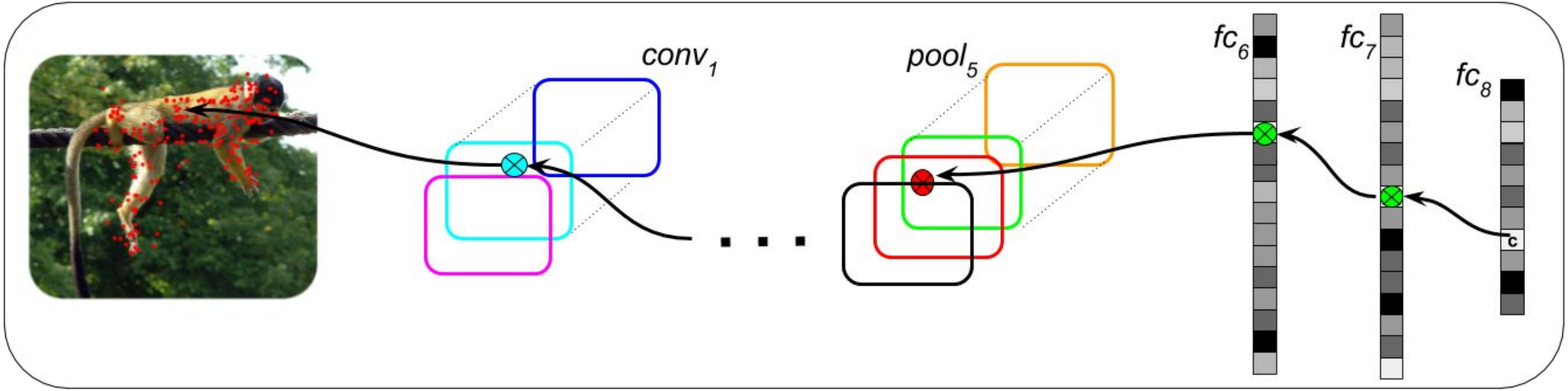


Figure from Renu Khandelwal and towardsdatascience.com

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

CNN-Fixations

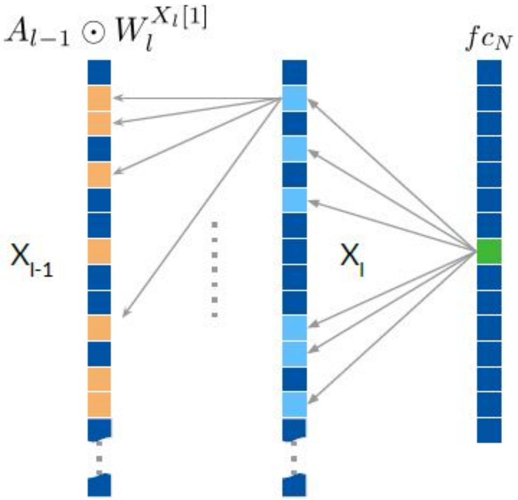
CNN-Fixations

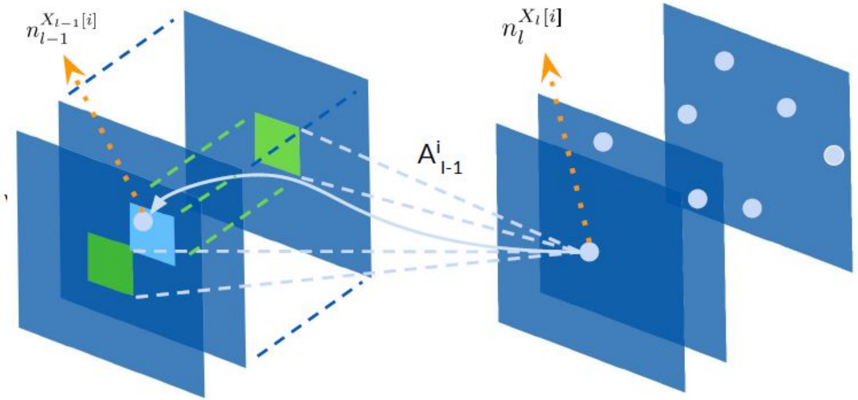


(a)

[CNN fixations: An unraveling approach to identify discriminative image regions, Konda Reddy Mopuri, Utsav Garg, R V Babu, **Trans. On IP 2018**]

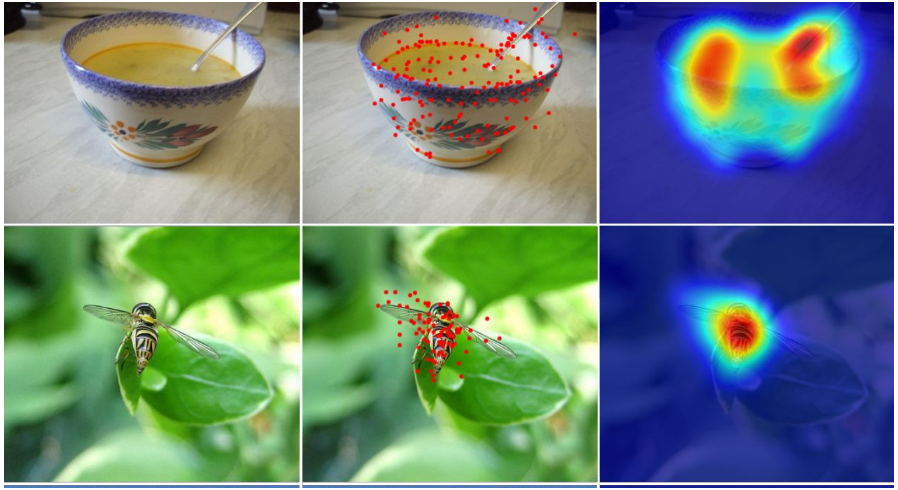
CNN-Fixations

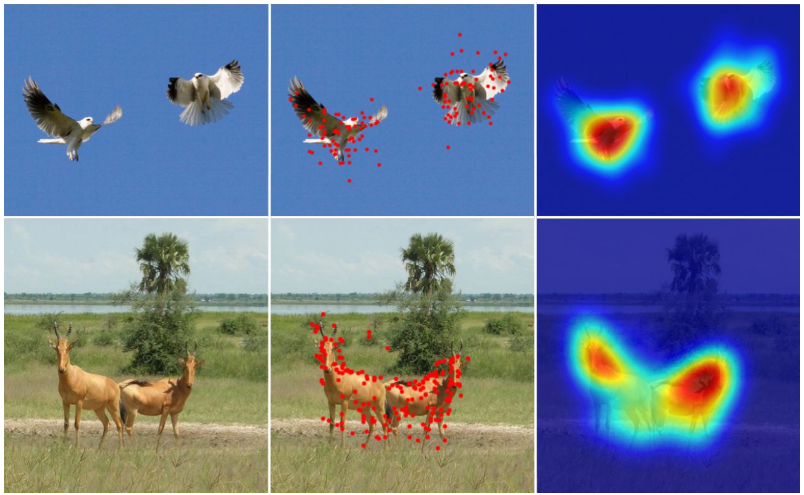




[CNN fixations: An unraveling approach to identify discriminative image regions, Konda Reddy Mopuri, Utsav Garg, R V Babu, **Trans. On IP 2018**]

CNN-Fixations





[CNN fixations: An unraveling approach to identify discriminative image regions, Konda Reddy Mopuri, Utsav Garg, R V Babu, **Trans. On IP 2018**]()

Other similar works

- Layerwise relevance propagation for neural networks, ICMLW 2016, PLOS 2015
- Excitation backpropagation, ECCV 2016
- Visualizing Higher-Layer Features of a Deep Network [Y Bengio et al.] [Tech Report]
- Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks [ICCV 2015]
- Grad-CAM++ from Prof. Vineeth's group, IIT Hyderabad
- ....

Visualization

- **Neuron Visualization**
- **Evidence Localization**
- **Feature Reconstruction**

Inverting deep representations

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Understanding deep image representations by inverting them

[Mahendran et al. CVPR 2015]

Inverting deep features

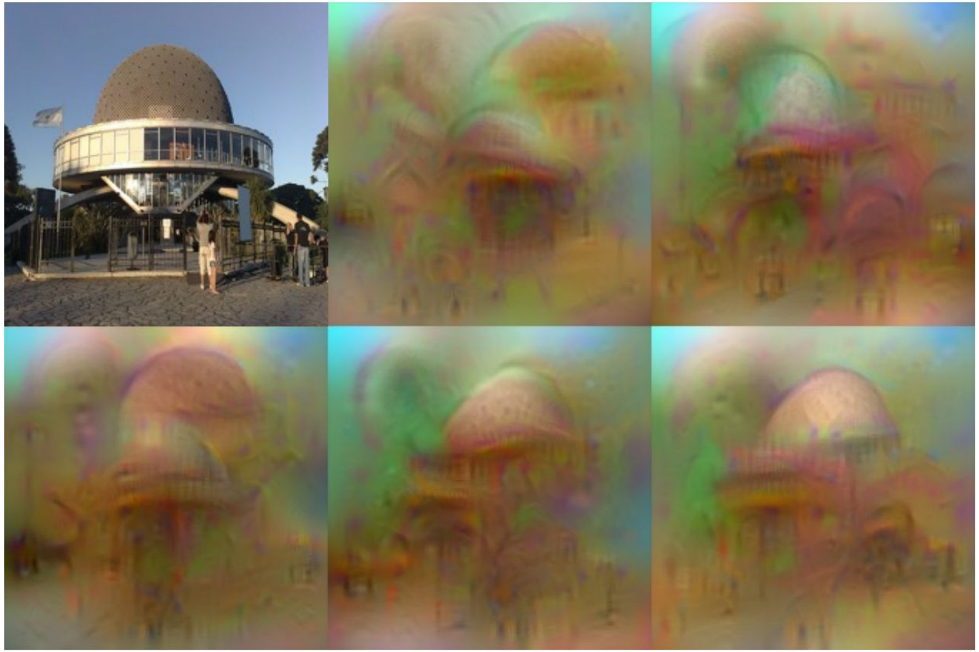


Figure 1. **What is encoded by a CNN?** The figure shows five possible reconstructions of the reference image obtained from the 1,000-dimensional code extracted at the penultimate layer of a ref-

Inverting deep features

- Given an image encoding, to what extent we can reconstruct the image ?
- No unique solution

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^{H\times W\times C}}{\operatorname{argmin}} \ \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda\mathcal{R}(\mathbf{x})$$

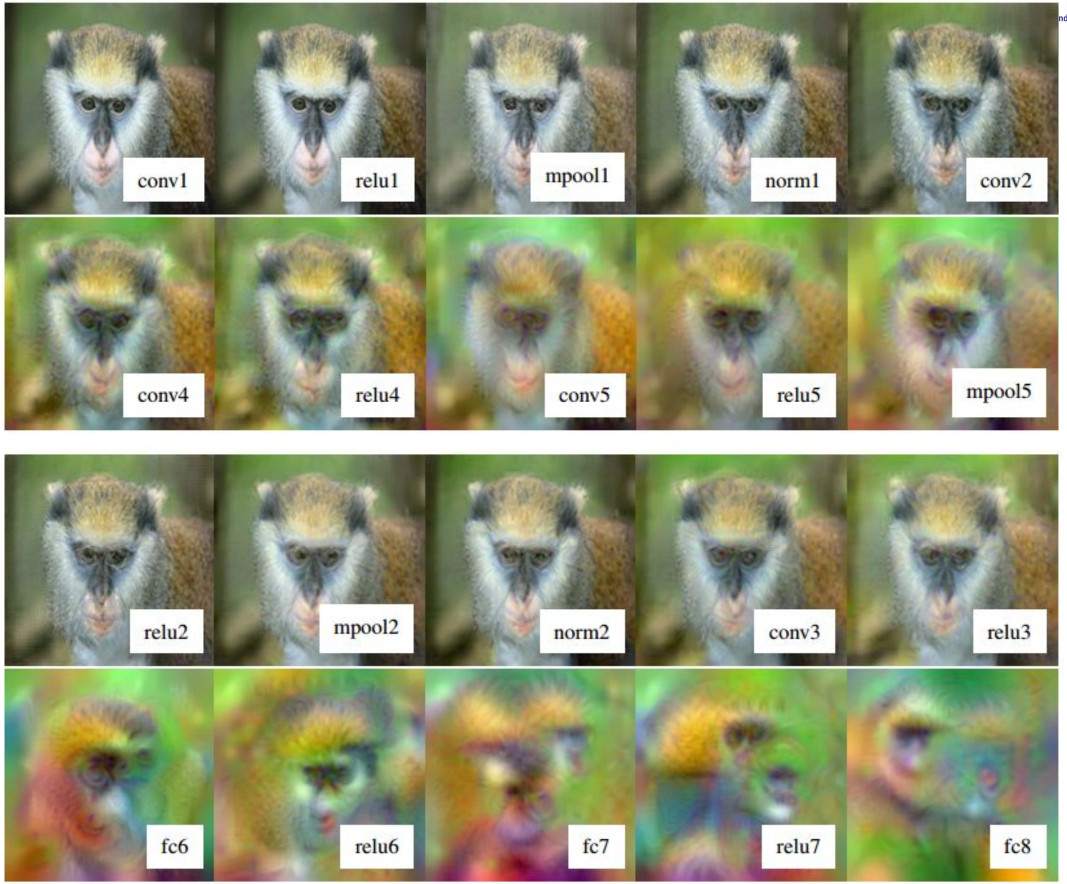
Loss function & Regularizer

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

- R - restrict the reconstruction to natural images
- Challenge : modelling it → TV norm prior

$$\mathcal{R}_{V^\beta}(\mathbf{x}) = \sum_{i,j} \left((x_{i,j+1} - x_{ij})^2 + (x_{i+1,j} - x_{ij})^2 \right)^{\frac{\beta}{2}}$$

Results


conv1
relu1
mpool1
norm1
conv2
conv4
relu4
conv5
relu5
mpool5
relu2
mpool2
norm2
conv3
relu3
fc6
relu6
fc7
relu7
fc8

Inverting CNN representation with another CNN

Inverting Visual Representations with Convolutional Networks

Alexey Dosovitskiy et al.
CVPR 2016

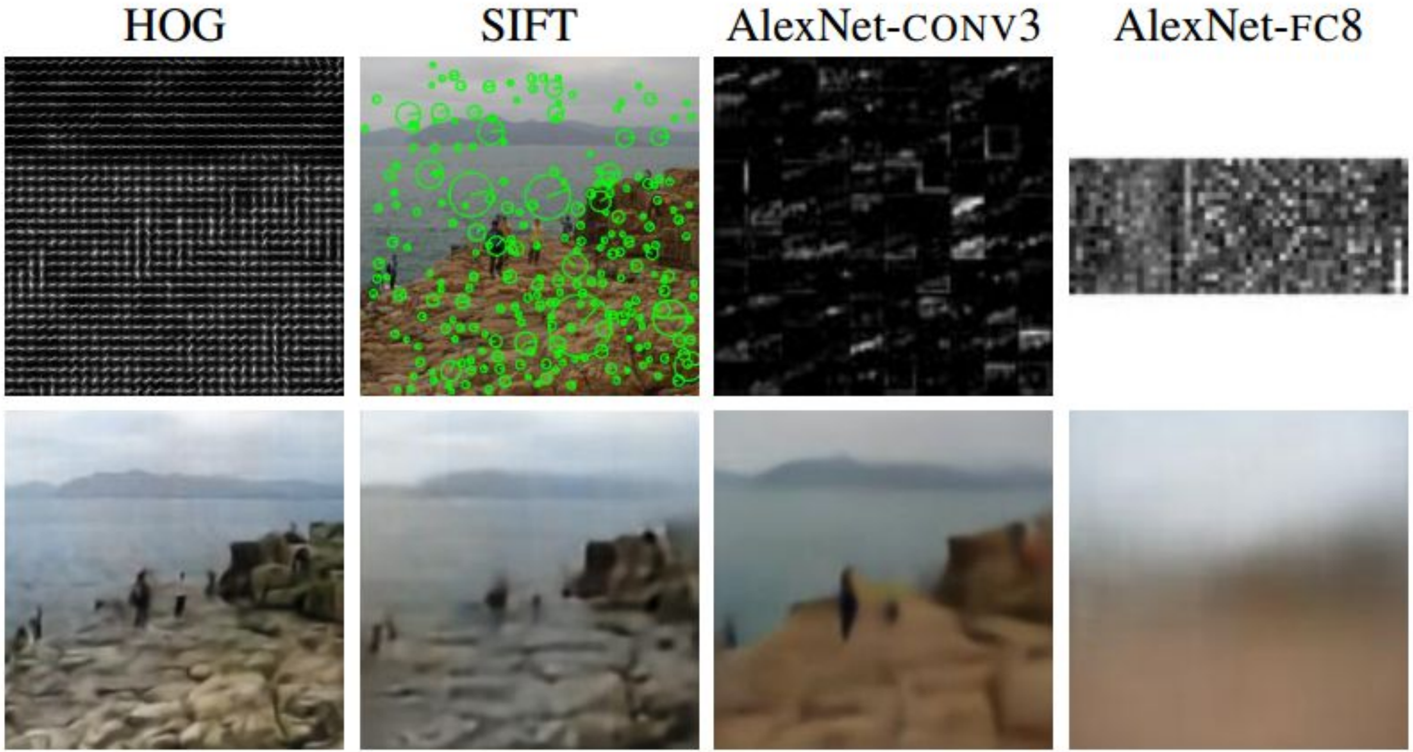
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Inverting using CNNs

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

- Train a CNN to invert image representations
    - SIFT, HOG, CNN representation etc.

Sample results



Network

- Training set of images and their features $\{x_i , \phi_i\}$

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_i ||\mathbf{x}_i - f(\boldsymbol{\phi}_i, \mathbf{w})||_2^2.$$

Results

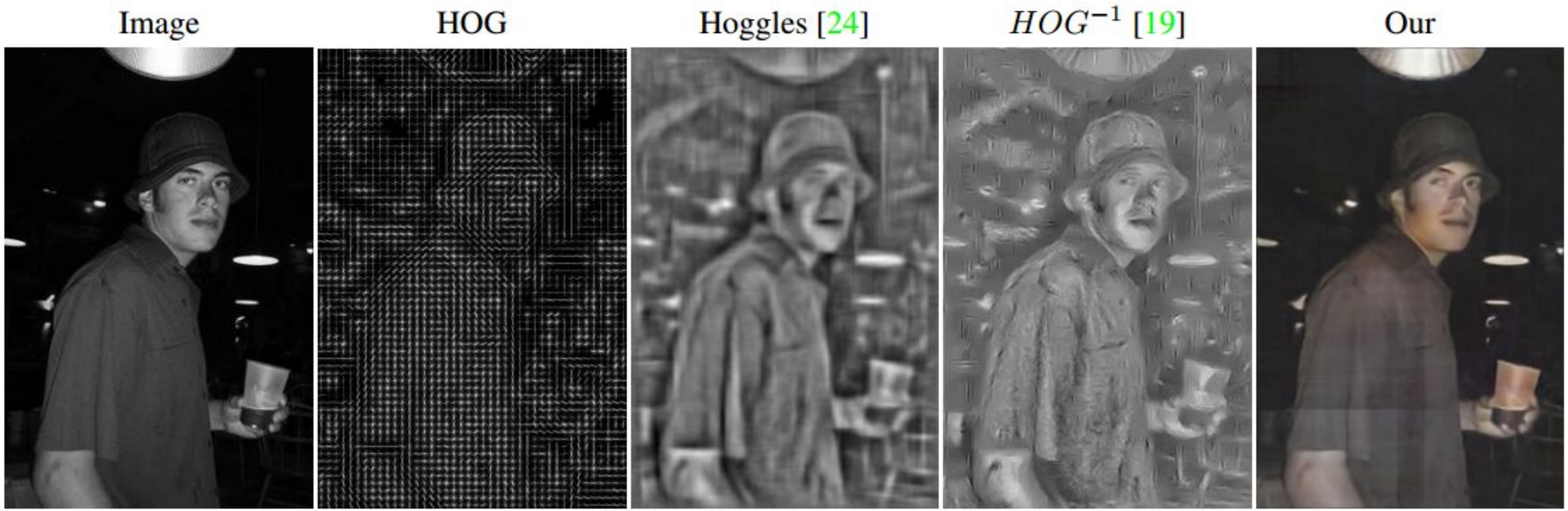


Figure 2: Reconstructing an image from its HOG descriptors with different methods.

What next ?

Future challenges/directions

- Transparency is useful at three different stages of Artificial Intelligence (AI) evolution
- AI is significantly weaker than humans - not yet reliably 'deployable' (e.g, VQA)
  - Identify the failure modes
- AI is on par with humans - reliably 'deployable' (e.g, recognition)
  - To establish appropriate trust and confidence in users
- AI is significantly stronger than humans - e.g, Chess/Go
  - Machine teaching

Appendix

Guided Backprop

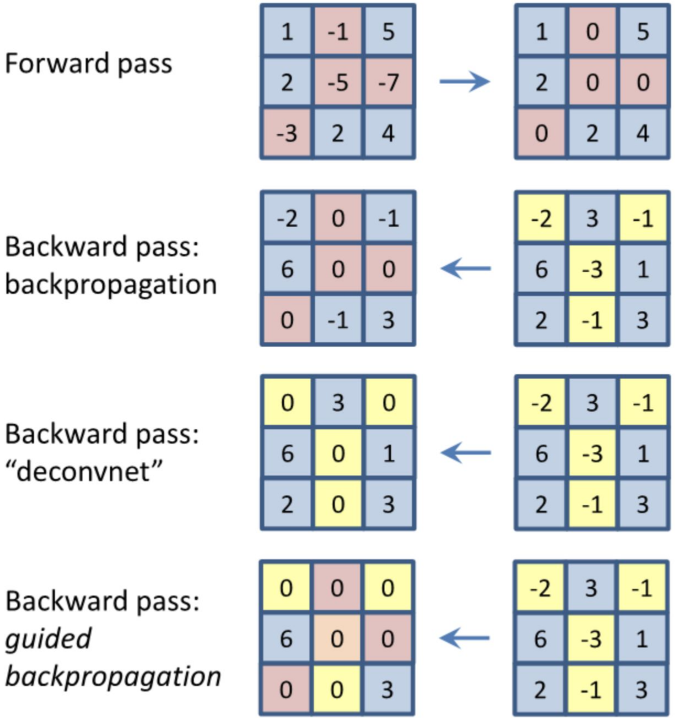


Figure Springenberg et al.

Texture synthesis

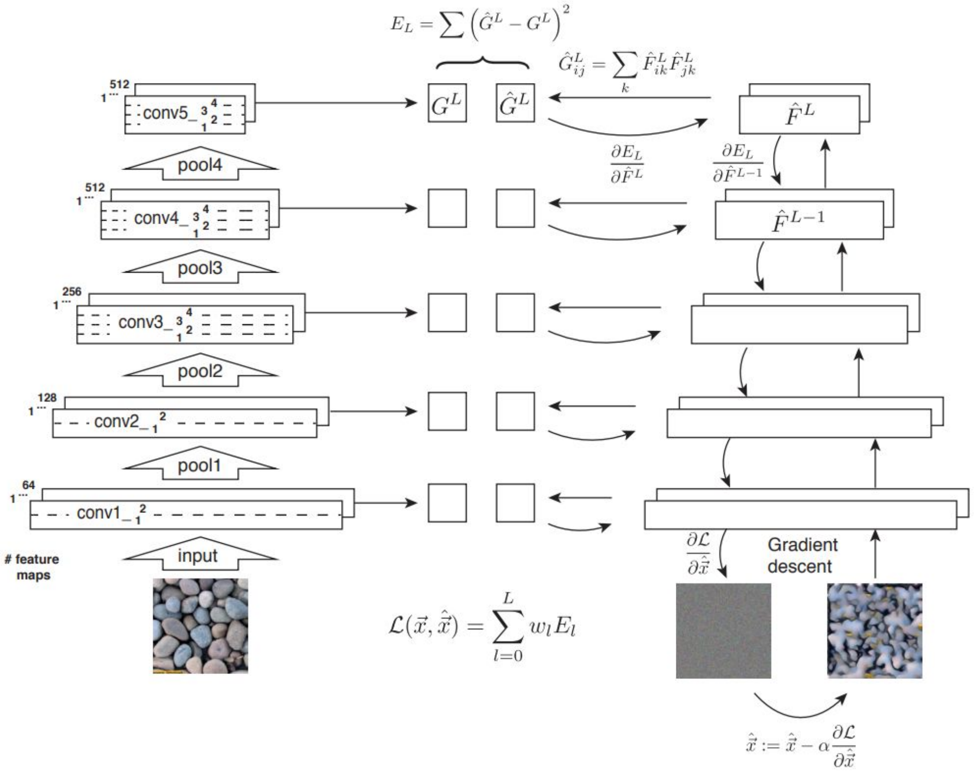


Figure I, Gatys et al. 2015

Texture synthesis

Figure I, Gatys et al. 2015

Style Transfer

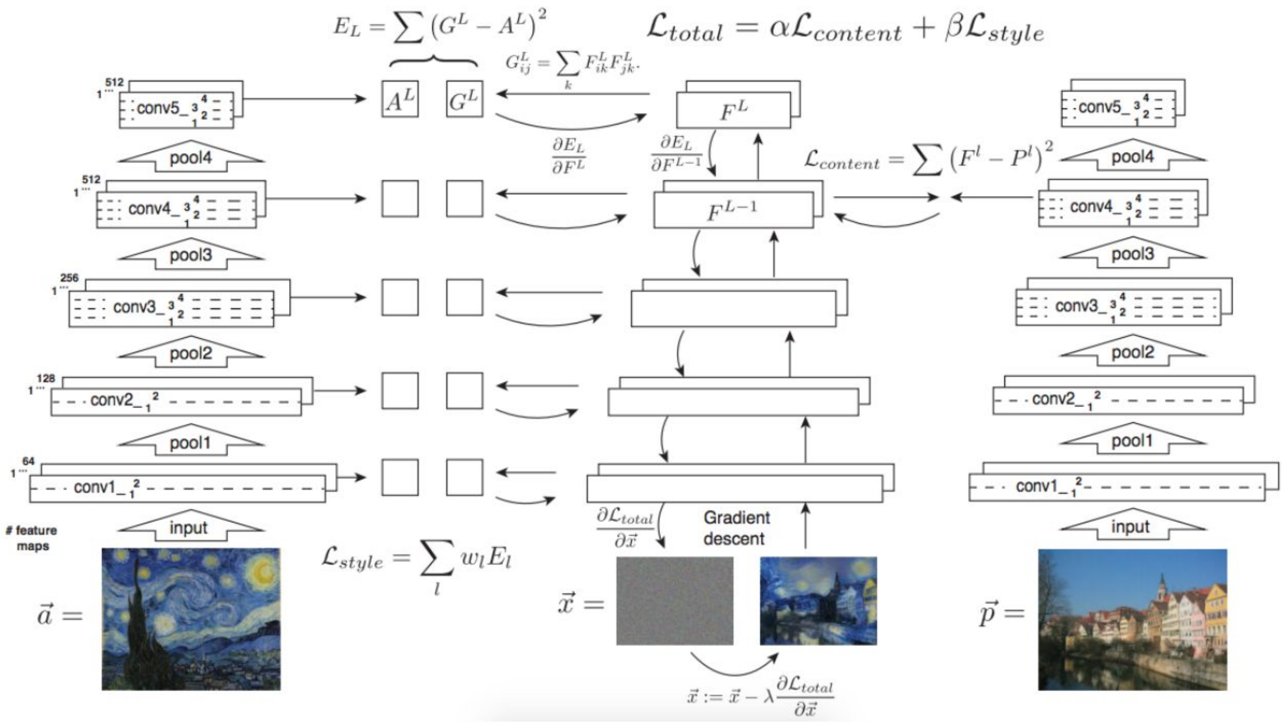


Figure from L Gatys et al. 2016

Style Transfer

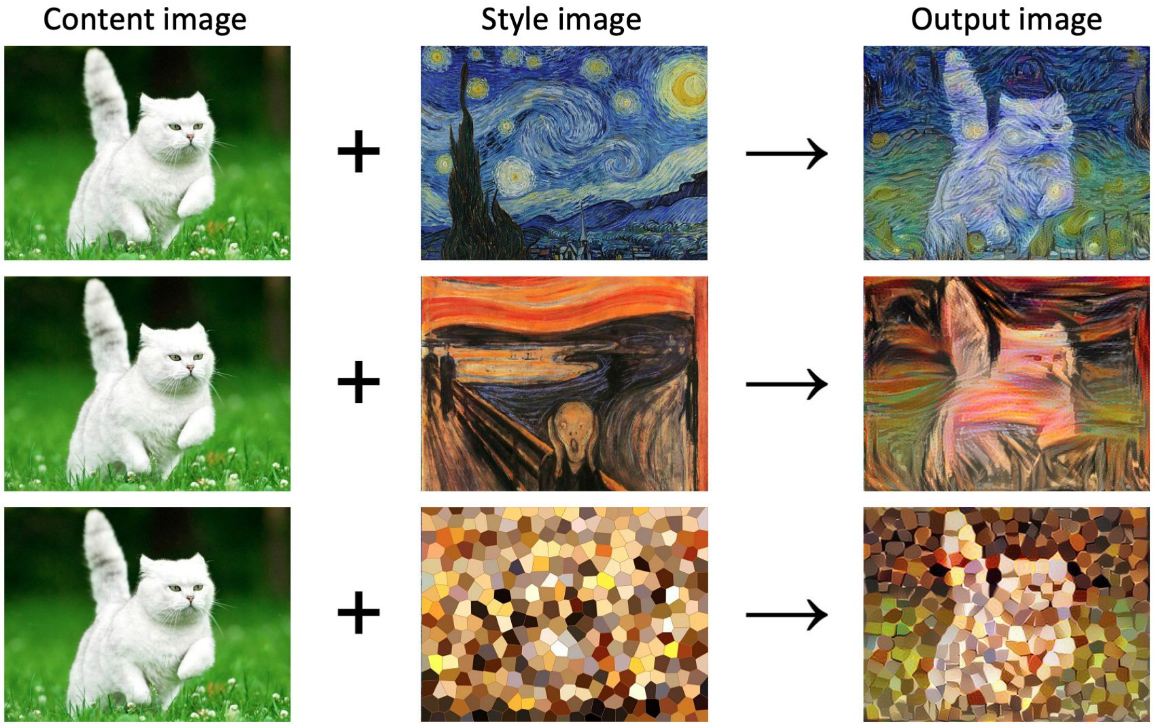


Figure from godatadriven.com