Deep Inside Convolutional Networks: Visualizing Image Classification Models and Saliency Maps

Figure 1: Numerically computed images, illustrating the class appearance models, learnt by a ConvNet, trained on ILSVRC-2013. Note how different aspects of class appearance are captured in a single image. Better viewed in colour.

Figure 2: Image-specific class saliency maps for the top-1 predicted class in ILSVRC-2013 test images. The maps were extracted using a single back-propagation pass through a classification ConvNet. No additional annotation (except for the image labels) was used in training.

Both images reproduced under fair use from https://arxiv.org/pdf/1312.6034.pdf
Overview

• Visualization of image classification models
  • convolutional neural nets (CNN)

• Focus on gradient of the output w.r.t the input.

• 2 Ideas:
  • Generate an image maximizing the output
  • Computes a saliency map
    • specific to a given image and label
    • using a single backward pass

• Lateral outcome: object segmentation
  • Weakly supervised
Problem Definition

• CNNs are the de facto architecture for image classification

• How do we understand
  • the aspects of visual appearance
  • captured inside a deep model?

• Earlier approach solved this
  • Found an input image which maximizes the output
  • using gradient search
    • in the image space.
Focus on CNNs for ImageNet

• A single “deep” CNN

• The ImageNet dataset
  • 1.2M training images
  • 1000 classes

• Data Augmentation
  • zeroing-out random parts of an image

• Top-1/top-5 classification error of 39.7%/17.7%
Model Visualization

• Given
  • a CNN
  • and a label c of interest,

• visualization *generates* an image numerically
  • Representing label c as learned by the CNN
Model Visualization - II

• Formally,
• Given input $I$ and class label $c$,
• let $F_c(I)$ be the output $F$ of the CNN for the label $c$
• Find an image such that the output $F_c(I)$ is high

$$\arg \max_I F_c(I)$$

• How do you solve it? Your favorite optimizer for a local optima.
Model Visualization - III

• Formally,
• Given input $I$ and class label $c$,
• let $F_c(I)$ be the output $F$ of the CNN for the label $c$
• Find an image such that the output logit $F_c(I)$ is high

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

• How do you solve it?
Model Visualization - IV

• Formally,
• Given input $I$ and class label $c$,
• let $F_c(I)$ be the output $F$ of the CNN for the label $c$
• Find an image such that the output logit $F_c(I)$ is high

$$ \arg \max_I F_c(I) - \lambda \| I \|^2_2 $$

• How do you solve it? Fix the model weights; vary the input
More on optimization

• Backpropagation finds an $I$
  • Locally optimal
• Similar to model training
  • Optimizes model weights.
• In model visualization
  • Weights are fixed
  • Optimizes the input
• Implementation details:
  • Initialize with 0 image
  • Add training set mean to result.

Reproduced under fair use from https://arxiv.org/pdf/1312.6034.pdf
Why not use class posteriors?

• Uses output $F_c$ and not class posteriors (soft-max):

$$P_c = \frac{\exp F_c}{\sum_c \exp F_c}$$

• Posterior can be maximized by minimizing scores of other classes
  • Helpful?

• Maximizing $F_c$ ensures focus on the label of interest – class $c$. 
Image-Specific Class Saliency Map

• Goal: Query CNN about spatial support of a label in a given image.
  • Attribution analysis

• Given
  • an image $I_0$,
  • a label of class $c$, and
  • a CNN with the output function $F_c(I)$,

• Rank the pixels of $I_0$ based on their influence on the score $S_c(I_0)$. 
Saliency Map - II

- Motivating example: a linear score model for the label $c$

$$F_c(I) = w_c^TI + b_c,$$

- Here,
  - the image $I$ is represented as a vector
  - $w_c$ is the weight vector,
  - and $b_c$ is the bias vector.

- Intuitively,
  - elements in $w_c$ define the *influence* of corresponding pixels
  - So, $w_c$ is an attribution vector!
Saliency Map - III

• Motivating example: a linear score model for the label $c$:
  \[ F_c(I) = b_c + w_c^T I \]

• Intuitively,
  • elements in $w_c$ define the influence of corresponding pixels
  • So, $w_c$ is an attribution vector!

• Now, consider a CNN with output $F_c(I)$

• Use Taylor expansion around an input image $I$
  • and drop second as well as higher order terms:
    • $F_c(I) \approx F_c(0) + \frac{\partial F_c}{\partial I} I$
Saliency Map - IV

• Motivating example: a linear score model for the label c:

\[ F_c(I) = b_c + w_c^T I \]

• Intuitively,
  • elements in \( w_c \) define the \textit{influence} of corresponding pixels
  • So, \( w_c \) is an attribution vector!

• Now, consider a CNN with output \( F_c(I) \)

• Use Taylor expansion around an input image \( I \)
  • and drop second as well as higher order terms:

\[ F_c(I) \approx F_c(0) + \frac{\partial F_c}{\partial I} I \]

The derivative shows which pixels have the most influence on the output
Saliency Map Results

Implementation Details
- No additional annotation beyond labels
- Single backpropagation
- Color images:
  - Maximum across all channels
- Used 10 cropped and reflected images
  - Took average of all of them

Both images reproduced under fair use from https://arxiv.org/pdf/1312.6034.pdf
Object Localization using Saliency Maps

• Given an image and its saliency map,
• the object segmentation mask computed using the GraphCut colour segmentation.
• Simple saliency thresholding may not highlight the whole object
  • Hence, use colour continuity cues
• GMMs for foreground and background color models
  • pixels with the saliency higher than 95% quantile => foreground
  • pixels with the saliency smaller than the 30% quantile => background
• Object segmentation = largest connected component of foreground
• 46.4% top-5 error on ImageNet localization challenge

GraphCut is discussed in Boykov YY, Jolly MP. Interactive graph cuts for optimal boundary and region segmentation of objects in N-d images. In Proc. ICCV, volume 2, pages 105–112, 2001.
Object Localization using Saliency Maps - II

- Given an image and its saliency map,
- the object segmentation mask computed using the GraphCut colour segmentation.
- Object segmentation = largest connected component of foreground
- 46.4% top-5 error on ImageNet localization challenge
Conclusions

• 2 visualization techniques for deep CNNs
• Synthesize an image of a given class from a trained deep CNN.
• Computes the saliency map for a given image and a given label.
• Employ in GraphCut based object segmentation.
• Future Work:
  • “Incorporate image-specific saliency maps into learning formulations in a more principled manner”

Both images reproduced under fair use from https://arxiv.org/pdf/1312.6034.pdf