Project 2: Deep Learning and Feature Visualization

CS 385

Datasets

- You may choose a single or multiple datasets
  - Learning an individual net using each dataset for a task

- MNIST
- CIFAR-10
- CUB200-2011
- Pascal VOC 2012
- Etc.

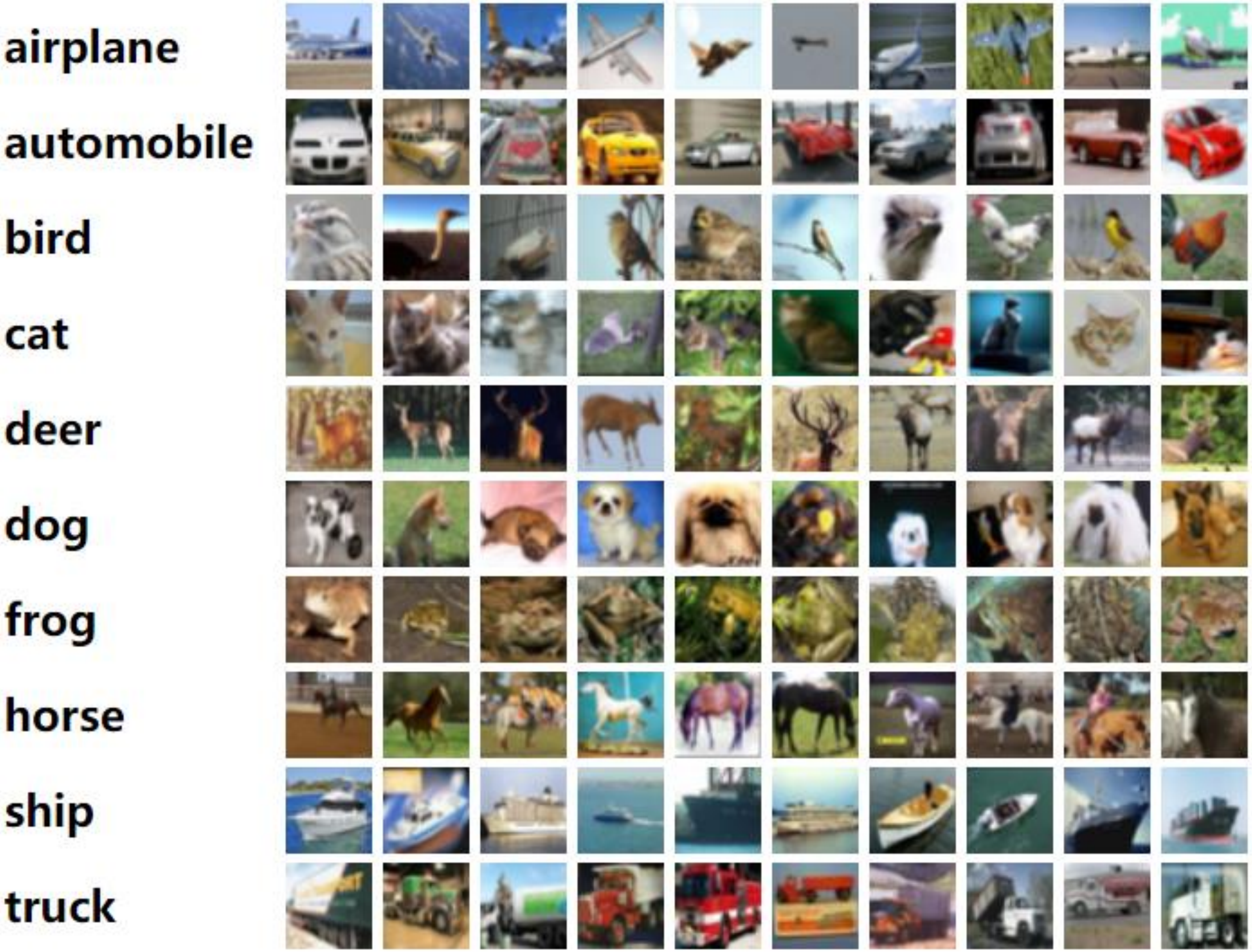
MNIST

- http://yann.lecun.com/exdb/mnist/
- 70,000 images
  - 10 classes
  - Each image has 28x28 pixels
  - 60,000 training images
  - 10,000 testing images

CIFAR-10

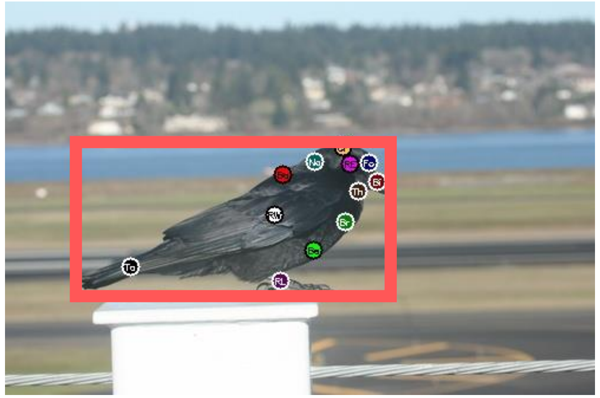
- http://www.cs.toronto.edu/~kriz/cifar.html
- 60000 32x32 colour images
  - 10 classes, with 6000 images per class
  - 50000 training images
  - 10000 test images.

CIFAR-10


airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

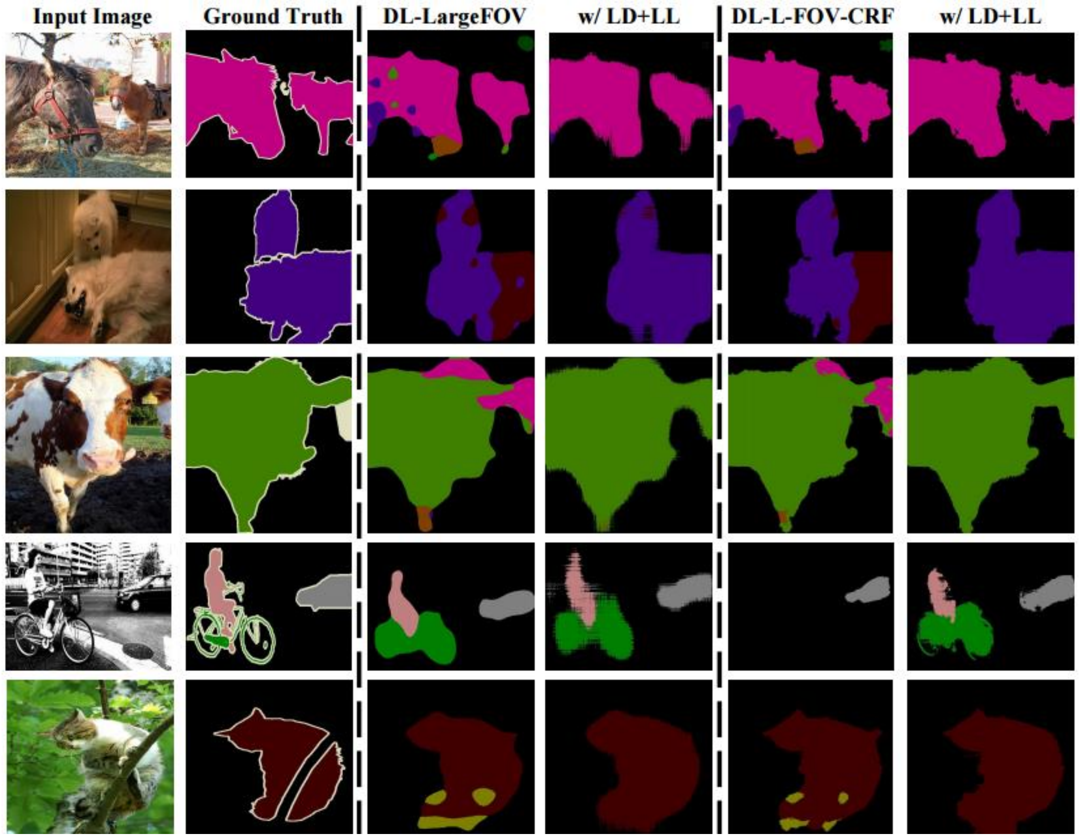
CUB200-2011

- http://www.vision.caltech.edu/visipedia/CUB-200-2011.html
- 11,788 Images
  - 200 bird categories (for fine-grained classification)
  - Each image contains a bounding box of a bird
  - About a half of the images are used for training, and the other half are used for testing.
  - Trick
    - To simplify the task, you may crop the bird image using bounding boxes, and use the cropped images for classification or image reconstruction.



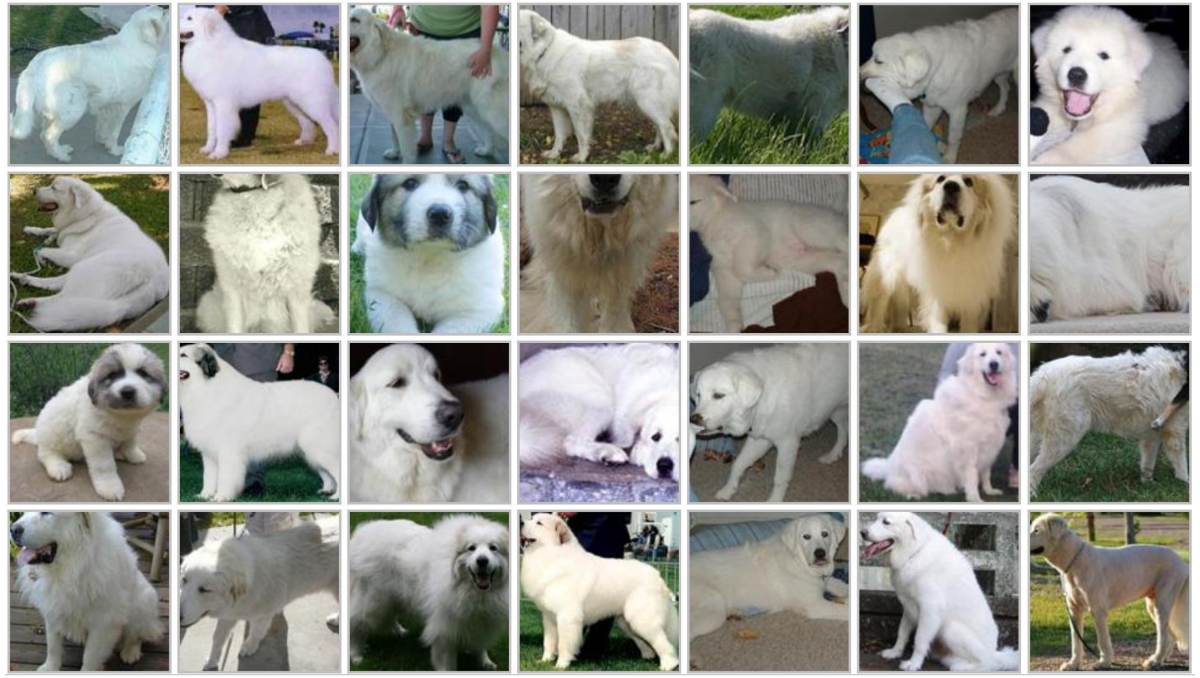
Pascal VOC 2012

- http://host.robots.ox.ac.uk/pascal/VOC/
- The train/val data has 11,530 images
  - 20 categories
  - 27,450 bounding boxes of objects
  - 6,929 segmentations.



Stanford dogs datasets

- http://vision.stanford.edu/aditya86/ImageNetDogs/
- 20,580 images
  - 120 categories with bounding boxes of dogs



Tasks

- Multi-category classification
  - Need ground-truth class labels for training
- Image reconstruction (variational auto-encoder)
- Image segmentation
  - Need ground-truth segmentation for training

	Classification	Reconstruction	Segmentation
MNIST	O	O	
CIFAR-10	O	O	
CUB200-2011	O	O	O
Pascal VOC 2012	O	O	O
Stanford Dogs	O	O	

Tasks

- Multi-category classification
  - The output is a n-dimensional vector for n-category classification.
- Image segmentation
  - The output is a $m*m*(n+1)$ tensor.
  - $m*m$ is the image size.
  - n is the number of category, plus a background category.
  - To classify the category label for each pixel.
- Image reconstruction
  - Loss=|| original image – generated image ||^2
  - Loss=crossEntropy(original image, generated image)

Tasks → Network structures

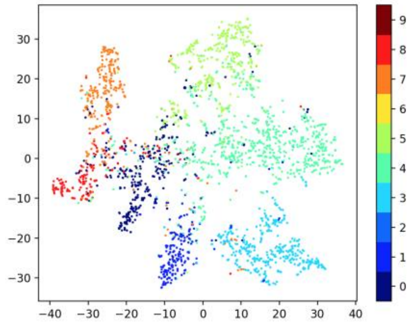
- Multi-category classification
  - AlexNet, VGG-16/19, ResNets
- Image segmentation
  - Modify the structure of the AlexNet/VGG/ResNet
  - The output is a tensor.
- Image reconstruction
  - Variational Auto-Encoder
    - Encoder
    - Decoder with transpose convolutional layers.

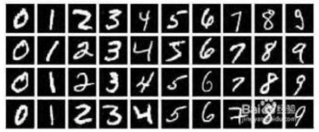
Neural Networks

- Please use one/multiple neural networks
  - AlexNet, and the modified network based on AlexNet
  - VGG-16/19, and the modified network based on VGG-16/19
  - ResNets, and the modified network based on ResNets
  - Variational Auto-Encoders
  - You may also choose other network structures

Feature analysis

- Principal component analysis (PCA)
  - Select an intermediate layer. The feature is f=g(x), transforming the tensor to a high-dimensional vector.
  - Dimension reduction: n-dimensional vector → m-dimensional vector (m<n)
  - Clustering using m-dimensional vectors
  - Show clustering results.
- t-SNE
  - Using t-SNE to visualize the intermediate-layer feature f=g(x)
  - Using t-SNE to visualize m-dimensional features of PCA
  - Put image samples on the 2-d plane.



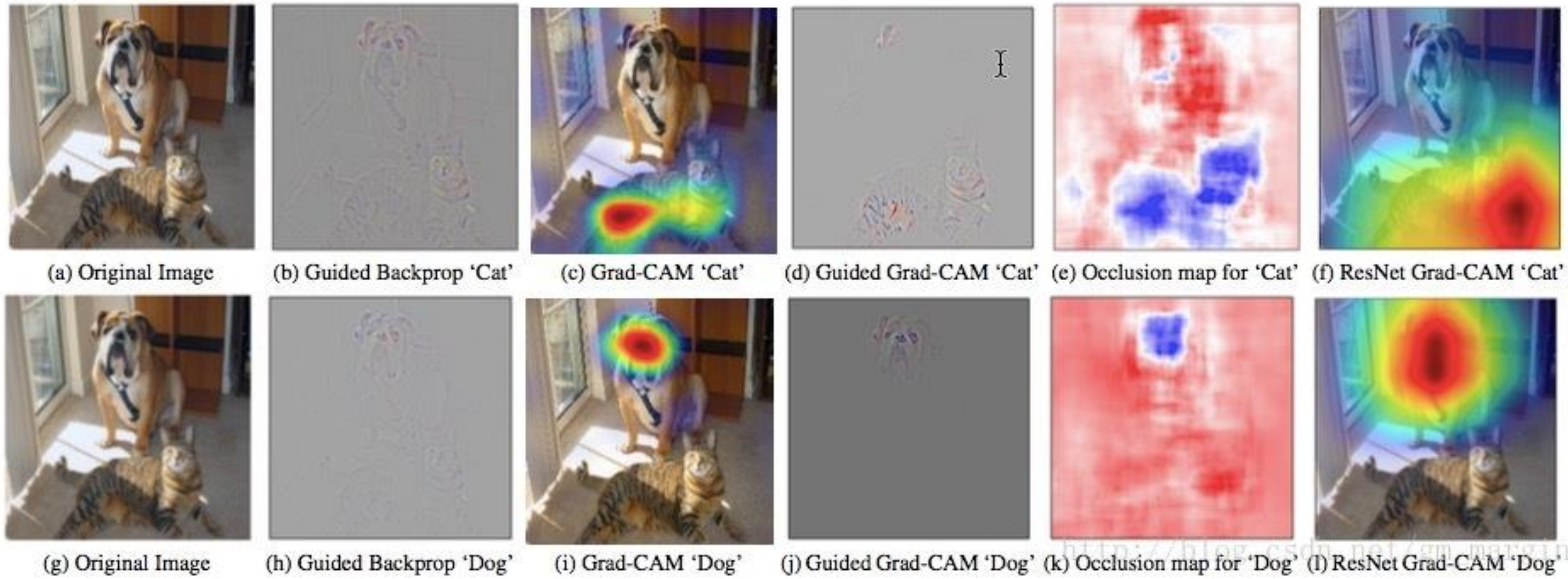


Grad-CAM

- Grad-CAM
  - Visualize the attention of different intermediate layers
- Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, in arXiv:1610.02391

$$\alpha_k^c = \frac{1}{Z} \sum_{i \in w} \sum_{j \in h} \frac{\partial y^c}{\partial A_{ij}^k}$$

$$L_{Grad-CAM}^c = ReLU(\alpha_k^c * A^k)$$



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

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog' (j) Guided Grad-CAM 'Dog' (k) Occlusion map for 'Dog' (l) ResNet Grad-CAM 'Dog'

If you do not have GPUs

- Trick 1
  - You may download a pre-trained net
  - Use CPUs to compute and save the intermediate-layer feature of each image.
  - Only finetune (or training from the initialized parameters) the upper layers.
    - Instead of finetuning/training the entire neural network
- Trick 2
  - Use object images cropped by bounding boxes
  - Resize the image into a smaller size
- Trick 3
  - Reduce the layer number of the neural network
  - Reduce the filter/kernel number of each layer
  - Reduce the kernel size of each filter
- Trick 4
  - Select your own training/testing datasets
  - For example, the first 30% samples of each category.
  - Do not carefully pick up easy samples

To enable a fair exam

- You do not compete the performance
  - Having GPUs → Complex and large neural networks → good performance
  - Therefore, I do not evaluate the project based on the performance.
  - However, you will be penalized if the performance is unusually bad concerning your network structure
    - The abnormal performance usually indicates bugs.

Your tasks

- Use different neural networks, conduct different tasks
    - Revise structures → obtain various networks
        - Layer number
        - Kernel number per layer
        - Kernel size
        - Add skip connections
    - Try different learning rates
        - Set a constant learning rate?
        - Set a descending learning rate?

Your tasks

- Use different neural networks, conduct different tasks
  - Apply different structural modification
    - Layer number
    - Kernel number per layer
    - Kernel size
  - Try different learning rates
    - Set a constant learning rate?
    - Set a descending learning rate?
  - Feature visualization and analysis
    - PCA+clustering
    - t-SNE
    - Grad-CAM

Your tasks

- You may decide the number of experiments that you implement
  - The more experiments, the better
  - The more analysis with deep insight, the better
- Summarize some experiences or conclusion by yourself.
  - Factors that affect the performance
    - Compare the performance of different networks, of different hyper-parameters, of different learning rates
  - Factors that affect the attention (grad-CAM) of a neural network
  - Factors that affect the clustering property of intermediate-layer features
    - Compare the t-SNE/PCA/grad-CAM visualization of different tasks, of different networks, of different layers.

Your tasks

- Build up your group within one week
  - Each group has four students (not three, not five)
  - We have 18-19 groups in total
- Prepare a poster to introduce your experiments in the 15^{th} and 16^{th} weeks (the last two classes).
- Write a detailed report about the experiments, and submit your in the 17^{th} week (I will determine the exact time later).