Training Set Selection for Image Classification

John Koo
Advisors: David Crandall, Michael Trosset

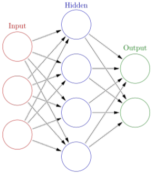
December 16, 2019

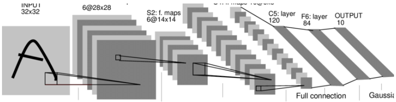
Background: Image Classification

- Most predictive modeling involves some sort of $n \times p$ data matrix
  - Observations typically thought of as points in $\mathbb{R}^k$
- Image data are harder to think of in terms of data matrices and Euclidean space
- Maybe use pixel values as features
  - Extremely high dimensional
    - Maybe can exploit spatial correlation to reduce number of dimensions
  - Doesn't account for image transformations such as translation or rotation
- Image classification (object/scene detection) doesn't fit well into traditional predictive modeling frameworks

Background: Convolutional Neural Networks

- Most neural networks are “fully connected”, i.e., each node of a layer is connected to every node of the previous layer and every node of the next layer
- Convolutional layers are locally connected, i.e., nodes point to small, spatially connected groups of nodes
- Accounts for localized features (e.g., edge detection)
- CNNs often outperform other image classification methods (but their complexity requires large training sets)





Sources: Wikipedia, LeCun et al.

Problem Statement and Objective

- Deep convolutional neural networks make use of the current wealth of curated image datasets and computational resources

Problem Statement and Objective

- Deep convolutional neural networks make use of the current wealth of curated image datasets and computational resources
- CNNs may fit poorly when there is insufficient data, and the data collection and labelling process can be expensive

Problem Statement and Objective

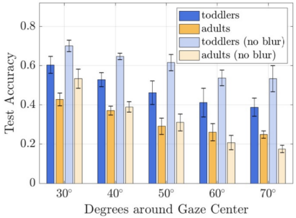
- Deep convolutional neural networks make use of the current wealth of curated image datasets and computational resources
- CNNs may fit poorly when there is insufficient data, and the data collection and labelling process can be expensive
- **Main question**: Is it possible to successfully train a neural network with a small number of carefully selected images?

Problem Statement and Objective

- Deep convolutional neural networks make use of the current wealth of curated image datasets and computational resources
- CNNs may fit poorly when there is insufficient data, and the data collection and labelling process can be expensive
- **Main question**: Is it possible to successfully train a neural network with a small number of carefully selected images?
- Largely based on "Toddler-Inspired Visual Object Learning" by Bambach, Crandall, Smith, and Yu (2018)

Background: "Toddler-Inspired Visual Object Learning"

- Collected two samples of images:
  - Taken from first-person cameras mounted on toddlers
  - Taken from first-person cameras mounted on adults
- Task: Identify which objects (toys) are in the images
- Training VGG16 using toddler data resulted in higher test accuracy than training on parent data (same test set in both cases)



Figures from Toddler paper

Background: "Toddler-Inspired Visual Object Learning"

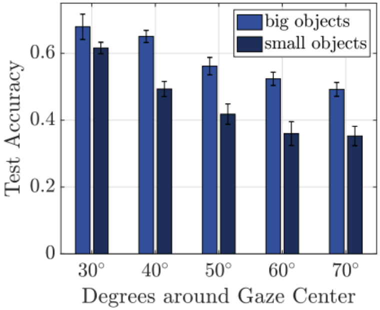
- Identified two characteristics of images from the toddler set vs. the adult set:
  - Objects in images from the toddler sample typically occupied more of the frame compared to objects in images from the adult sample
  - The toddler sample tended to be more "diverse" than the adult sample

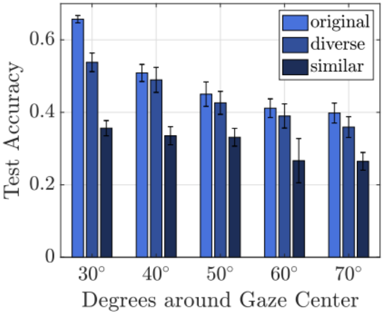
Background: “Toddler-Inspired Visual Object Learning”

- Two experiments performed based on object size and sample diversity
  - Find a subsample of big objects and a subsample of small objects, and train VGG16 on each. Determine which model attains a higher test accuracy.
    - Object “size” defined by determining how much of the image a bounding box drawn around the object takes up
  - Find a heterogeneous subsample and a homogeneous subsample, and train VGG16 on each. Determine which model attains a higher test accuracy.
    - Subsample “diversity” determined by pairwise distances of an image embedding (GIST features)

Background: "Toddler-Inspired Visual Object Learning"

- Results consistent with intuition
  - Subsample of big objects resulted in better models than subsample of small objects
  - Heterogeneous subsample resulted in better models than homogeneous subsample





Figures from Toddler paper

Outline and Summary

1. Applying the Toddler study to additional datasets

Outline and Summary

1. Applying the Toddler study to additional datasets
2. Overview of some other ways to select training sets

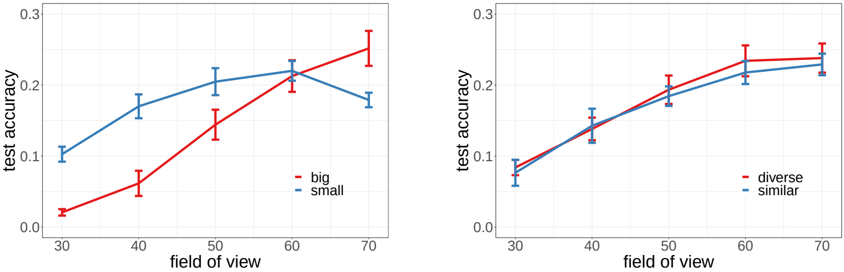
Outline and Summary

1. Applying the Toddler study to additional datasets
2. Overview of some other ways to select training sets
3. Conclusions and future work

Reproduction Study: Stanford Dogs dataset

- ~20,000 images of 120 dog breeds
- 100 images per breed set aside for training
  - further subdivided into 50-50 big/small or diverse/similar
- 25 images per breed set aside for validation
- Remainder for testing

Reproduction Study: Stanford Dogs dataset



Reproduction Study: CIFAR-10

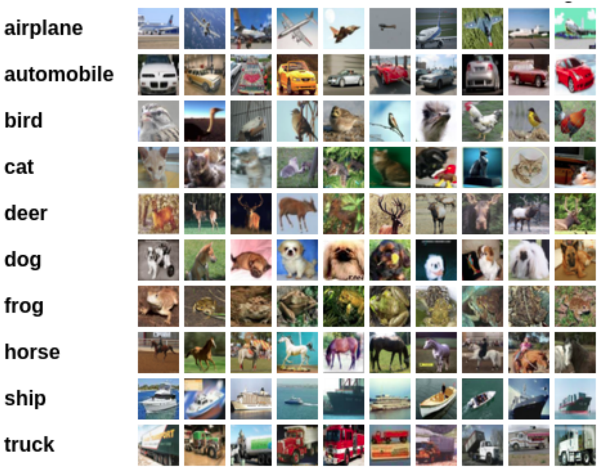
CIFAR-10

- 32×32 RGB images of 10 different object classes
- 5,000 training and 1,000 testing images per class
- No bounding box information
  - Diversity experiment only
  - No adjusting field of view

Sampling method

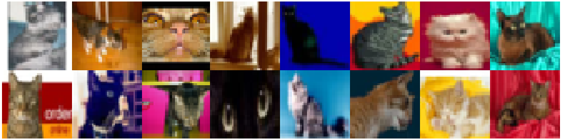
1. Choose training size n
2. Draw $2n$ images using diverse, similar, or random sampling
3. Split the data in half for training and validation
   - Validation set used to determine when to stop training
4. Fit VGG16 and assess accuracy on the test set

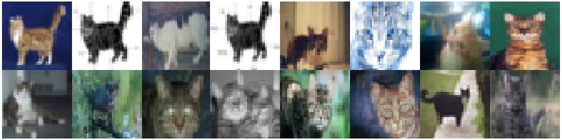
Reproduction Study: CIFAR-10



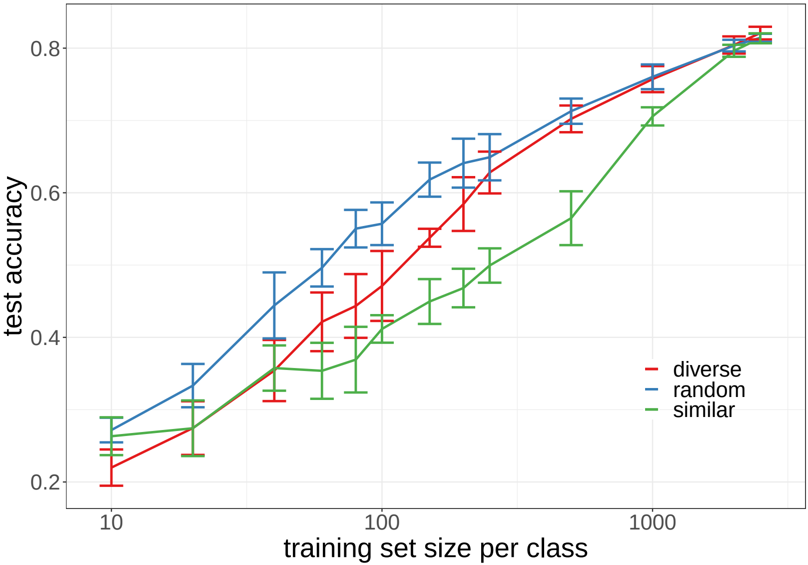
Source: https://www.cs.toronto.edu/~kriz/cifar.html

Diverse vs. Similar vs. Random Samples of CIFAR-10 Cats

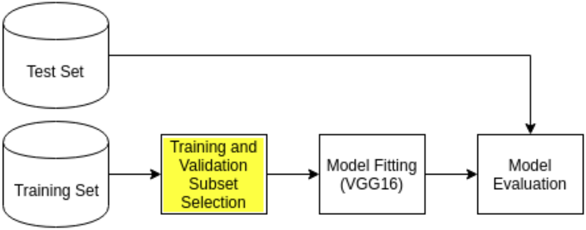




Replication Study: CIFAR-10



New Approaches to Training Set Selection



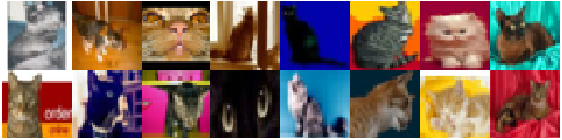
Wilson Editing

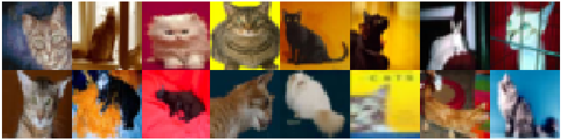
- Originally developed for k-nearest neighbors
- Algorithm
  1. Start with a sample $X_1, ..., X_n \in \mathbb{R}^p$ and corresponding discrete labels $Y_1, ..., Y_n \in \{1, ..., q\}$
  2. For $i = 1, ..., n$, determine $\hat{Y}_i$ using leave-one-out cross-validated k-nearest neighbors classification
  3. Discard $i \in \{1, ..., n\}$ where $Y_i \neq \hat{Y}_i$ to construct a reduced, "edited" training set
  4. Use the edited training set to fit a new k-nearest neighbors model
- Outperforms "unedited" k-nearest neighbors (comparing risk on a held-out test set)

Wilson Editing

- Method
  1. Use Wilson editing (using the GIST embedding) to reduce the training set
  2. For a training size n, draw a diverse sample from the edited training set
  3. Fit VGG16 on the diverse, edited training subset
  4. Assess model performance on the test set
- Idea: Draw a diverse sample while excluding "outliers"

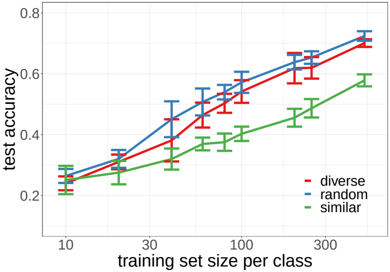
Diverse Sample vs. Diverse Sample After Wilson Editing



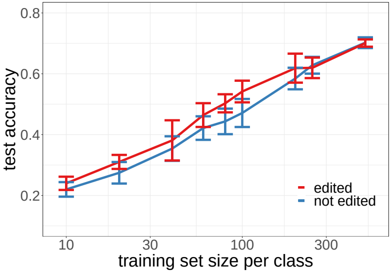


Wilson Editing

Editing done prior to drawing diverse samples



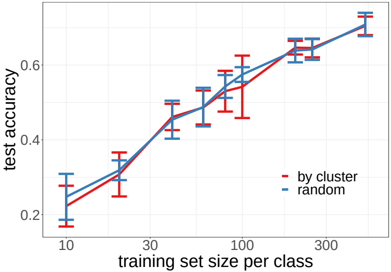
Effect of editing prior to drawing diverse samples



Clustering

Algorithm

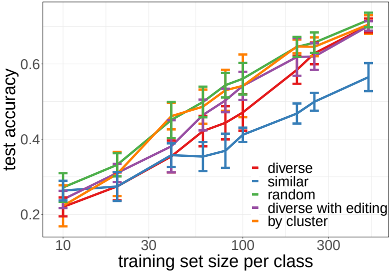
1. Use k-means clustering (using the GIST embedding) to split each class into "subclasses"
2. For a training size n, draw a cluster/subclass-stratified sample
3. Fit VGG16 on the stratified training subset
4. Assess model performance on the test set



Summary and Conclusions

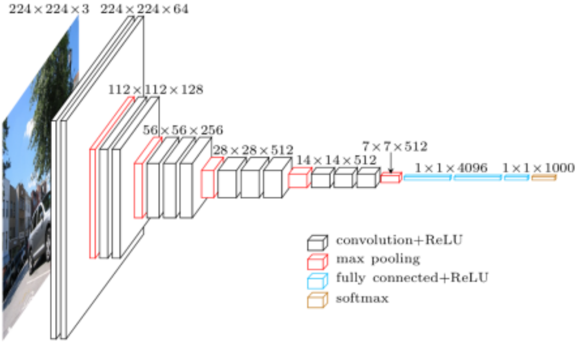
- Diverse sampling outperforms similar sampling but fails to improve upon uniform random sampling
- Removing outlier images prior to drawing a diverse sample seems to improve model performance
- Cluster-stratified sampling resulted in equivalent model performance as uniform random sampling
- Future work
  - More sophisticated clustering methods
  - Active learning based approaches

Comparison of sampling methods



Supplemental Slides

VGG16



Source: Simonyan and Zisserman

GIST Embedding

Autoencoders

Transfer Learning

- CNNs can be thought of as supervised image embedding methods
  - Second to last layer should be a linearly separable embedding
- Pretrained Xception model (~80% accuracy on CIFAR-10)
  - Xception embedding results in ~80% accuracy using k-NN
  - Can be thought of as an ideal case for embedding CIFAR-10
- Diverse sampling on Xception embedding still results in worse performance than random sampling