Training Set Selection for Image Classification

John Koo

December 16, 2019

Background

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

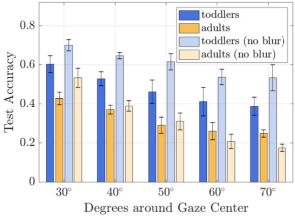
- Deep convolutional neural networks make use of the current wealth of curated image datasets and computational resources
- CNNs may fit poorly when there is insufficient data, and the data collection and labelling process can be expensive
- **Goal**: Determine a training set sampling method to select the best possible images under a fixed training set size

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
- **Goal**: Determine a training set sampling method to select the best possible images under a fixed training set size
- Largely based on "Toddler-Inspired Visual Object Learning" by Bambach, Crandall, Smith, and Yu (2018)

Background: "Toddler-Inspired Visual Object Learning"

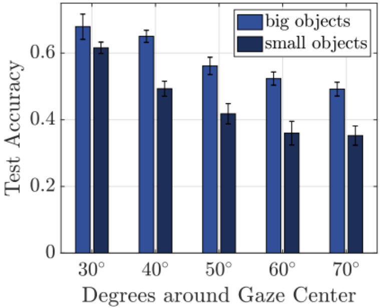
- Compared images taken by first-person cameras mounted on toddlers and parents
- Training VGG16 using toddler data resulted in higher test accuracy than training on parent data (same test set in both cases)

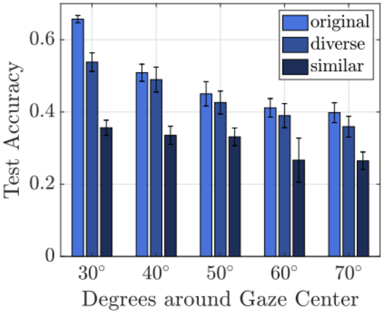


Background: “Toddler-Inspired Visual Object Learning”

- Objects in images from the toddler sample occupied more of the frame compared to objects in images from the adult sample
- Toddler
- Distilled the differences in the datasets into two components: object size (how much of the image does the object take up) and image “diversity” (hard to measure)
- Subsampled the images to obtain training subsets of:
    - big objects
    - small objects
    - diverse subset
    - similar subset
    - random subset
- Image similarity/distance based on an image embedding (GIST features)
- Found that when objects are larger or when images are more

Background: “Toddler-Inspired Visual Object Learning”





Outline and Summary

1. Applying the Toddler study to additional datasets

Outline and Summary

1. Applying the Toddler study to additional datasets
2. Some other ways to select training sets

Outline and Summary

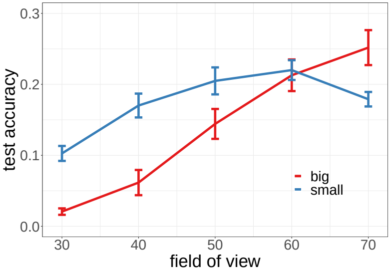
1. Applying the Toddler study to additional datasets
2. Some other ways to select training sets
3. Conclusions and future work

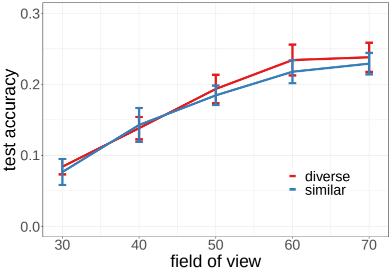
Reproduction Study: Stanford Dogs dataset

- ~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

- Some evidence that object size affects training
- No significant evidence that image diversity affects training





Reproduction Study: CIFAR-10

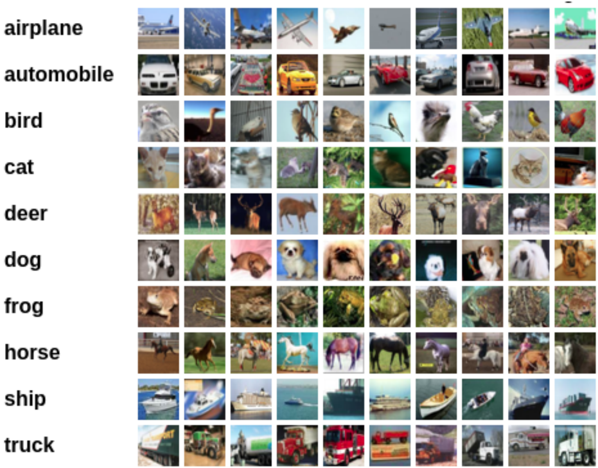
CIFAR-10

- 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

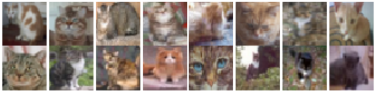
Sampling method

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
4. Fit VGG16 and assess accuracy on the test set

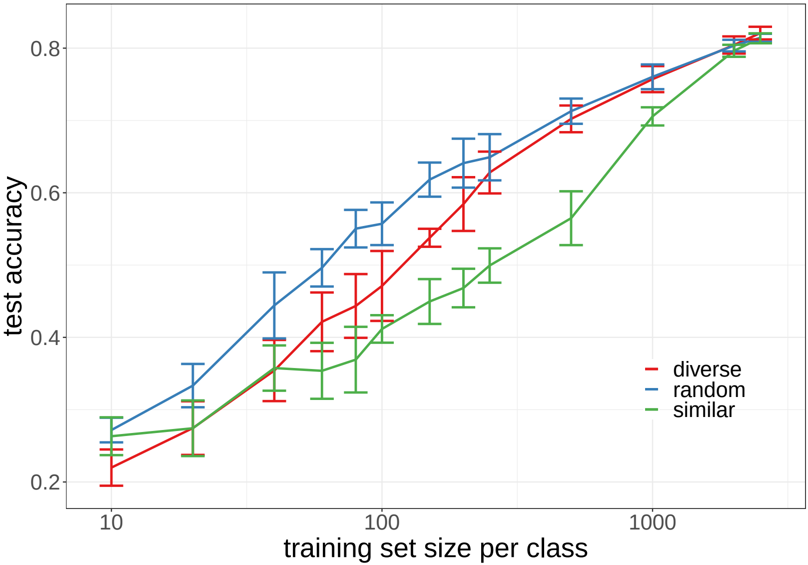
Reproduction Study: CIFAR-10



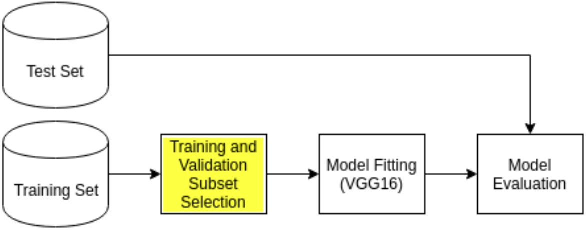
Diverse vs. Similar vs. Random Samples of CIFAR-10 Cats



Replication Study: CIFAR-10



New Approaches to Training Set Selection



Wilson Editing

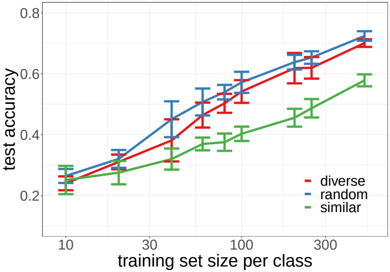
- Originally developed for k-nearest neighbors
- Algorithm
  1. Start with a sample $X_1, ..., X_n \in \mathbb{R}^p$ and corresponding discrete labels $Y_1, ..., Y_n \in \{1, ..., q\}$
  2. For $i = 1, ..., n$, determine $\hat{Y}_i$ using leave-one-out cross-validated k-nearest neighbors classification
  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

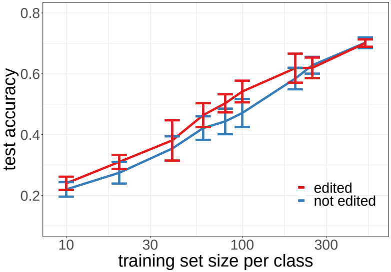
- Algorithm
  1. Use Wilson editing (using the GIST embedding) to reduce the training set
  2. For a training size n, draw a diverse sample from the edited training set
  3. Fit VGG16 on the diverse, edited training subset
  4. Assess model performance on the test set
- Idea: Draw a diverse sample while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

Wilson Editing

Editing done prior to drawing diverse samples



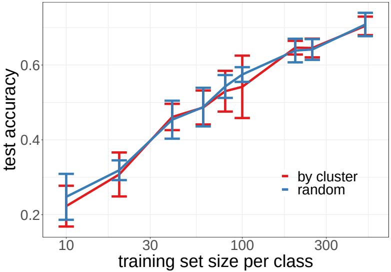
Effect of editing prior to drawing diverse samples



Clustering

Algorithm

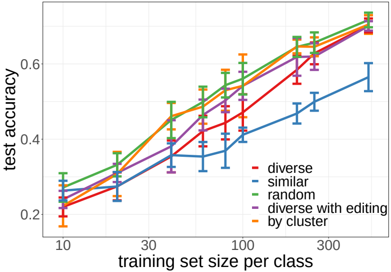
1. Use k-means clustering (using the GIST embedding) to split each class into "subclasses"
2. For a training size n, draw a cluster/subclass-stratified sample
3. Fit VGG16 on the stratified training subset
4. Assess model performance on the test set



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

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 ($\sim$80% accuracy on CIFAR-10)
  - Xception embedding results in $\sim$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