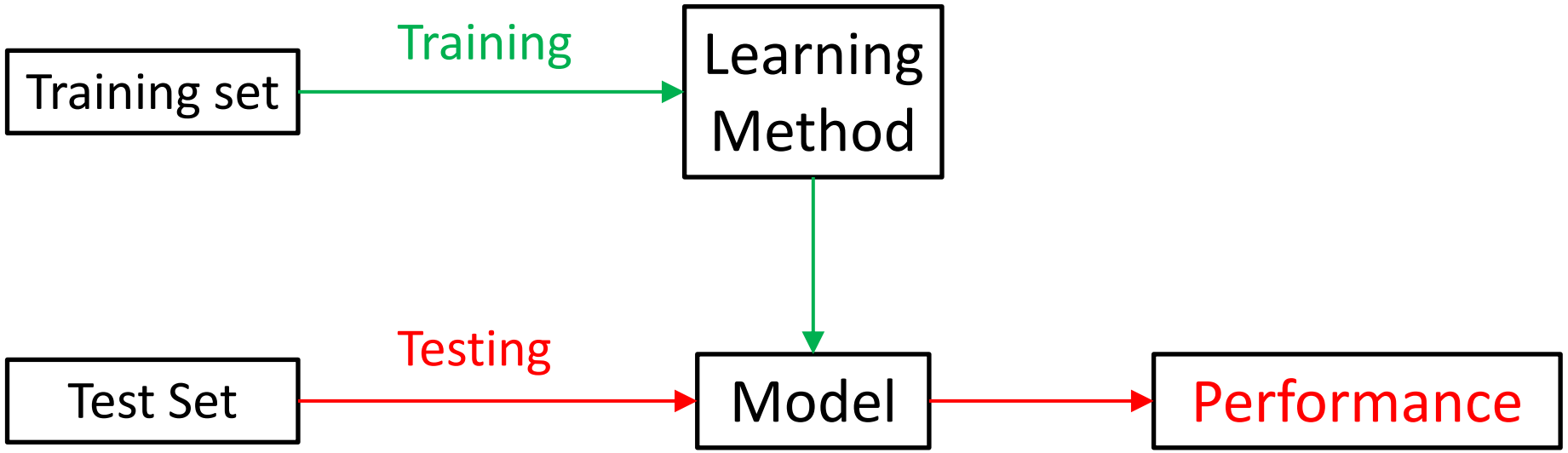
Australian National University

Are Labels Always Necessary for Classifier Accuracy Evaluation?

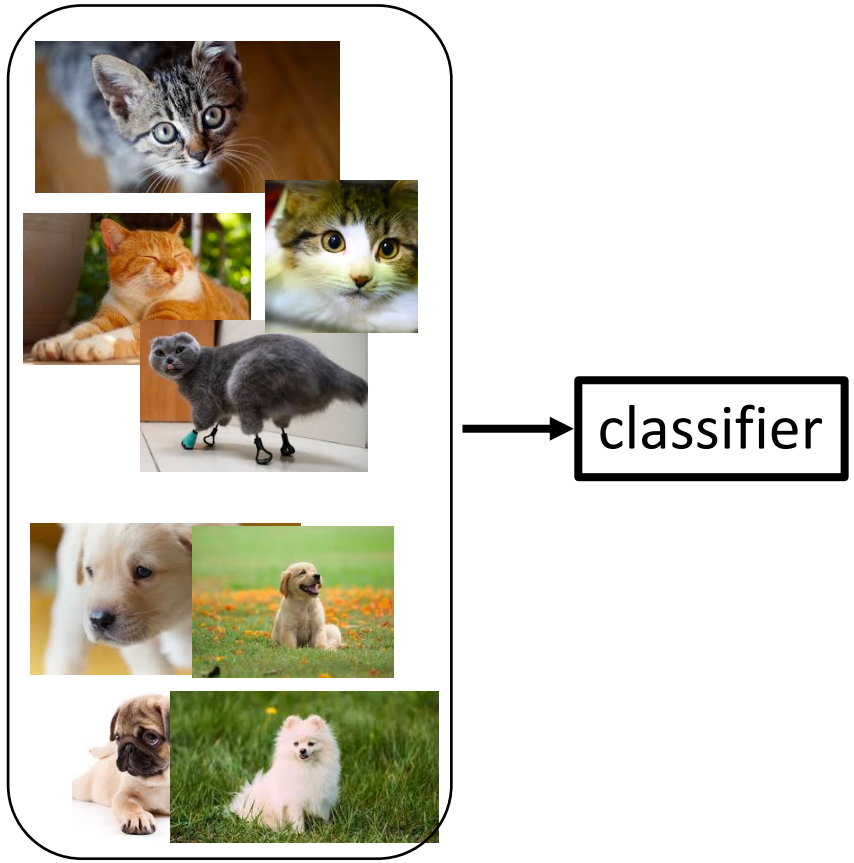
Weijian Deng and Liang Zheng

Australian National University

Pillars in machine learning



We start with training a classifier



classifier

Training data

We do a bit testing …

Correct prediction

Testing image → classifier → Dog 0.05, Cat 0.95 → Cat

Prediction result = Ground truth

Wrong prediction

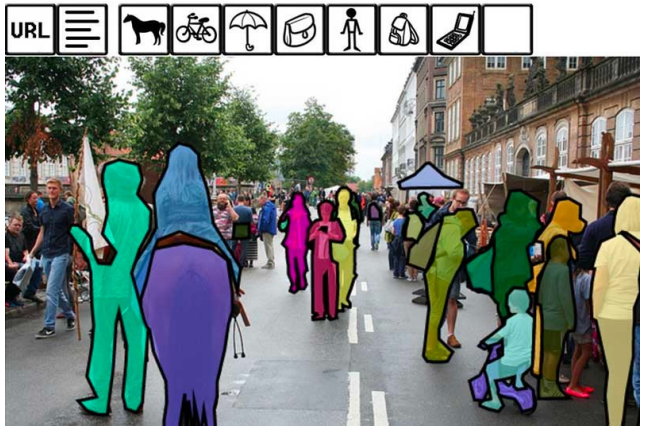
Testing image → classifier → Dog 0.85, Cat 0.15 → Cat

Prediction result ≠ Ground truth

Is this way of evaluation feasible?

- Yes

ImageNet



MSCOCO

Ground truths provided

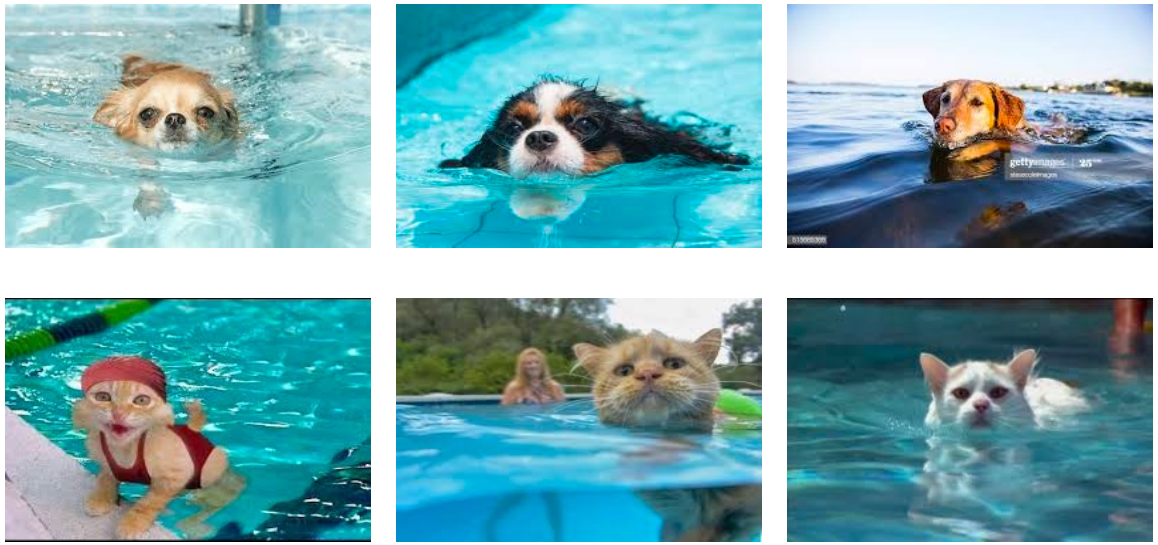
LFW

Is this way of evaluation feasible?

- No

Suppose we deploy our cat-dog classifier to a swimming pool

We can’t calculate a classifier accuracy!



Ground truths are NOT provided

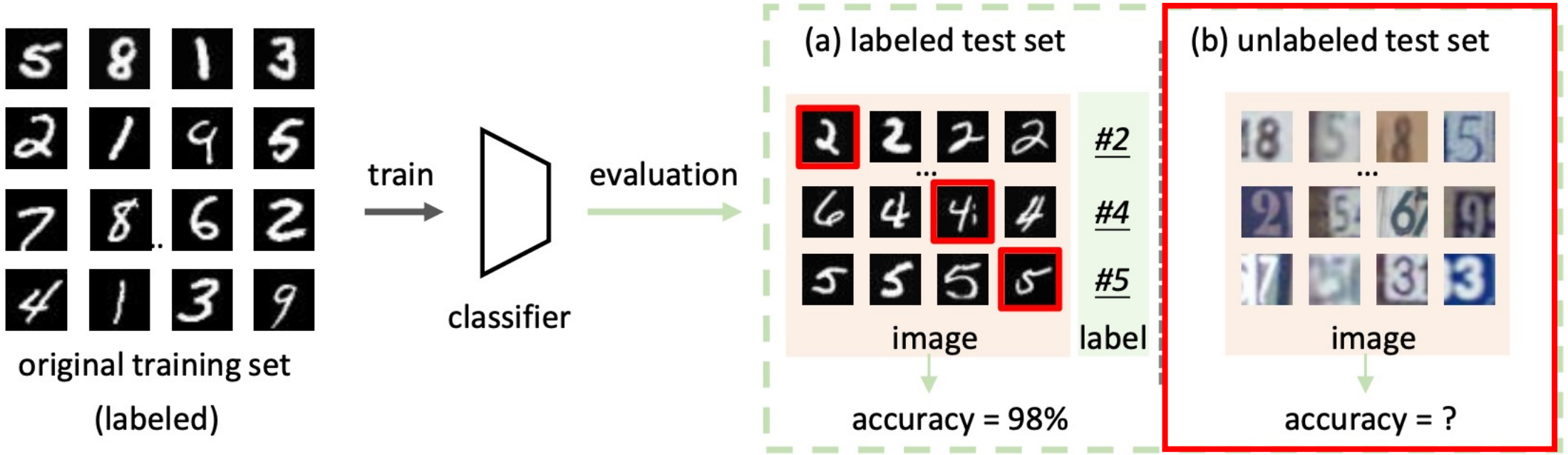
We encounter this problem too many times in CV applications

- Deploy face recognition in an airport
- Deploy a 3D object detection system to a new city
- ……

We can’t quantitatively measure the model accuracy like we usually do!!

Unless we annotate the test data, but environments keep changing. We need to annotate test data again

Formally, we want to solve:

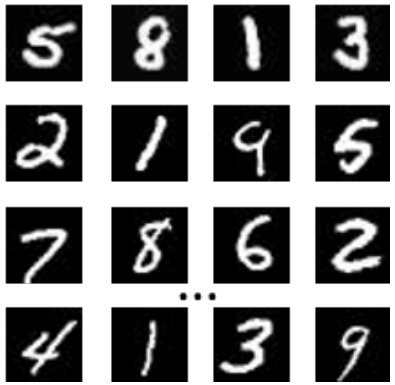


Given
- A training dataset
- A classifier trained on this dataset
- A test set without labels

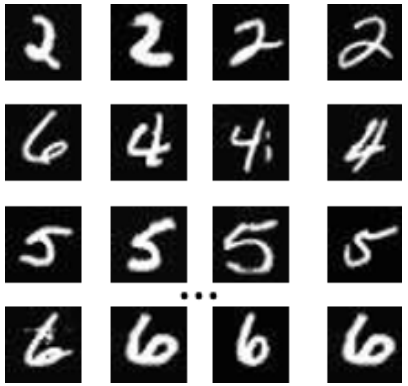
We want to ***estimate***:
Classification accuracy on the test set

Our idea

Training set



Test set 1



Test set 2

Test set 3

97%

80%

60%

domain gap:

recognition accuracy:

***Larger* domain gap -> *lower* recognition accuracy**

Our idea

Known (from existing literature)

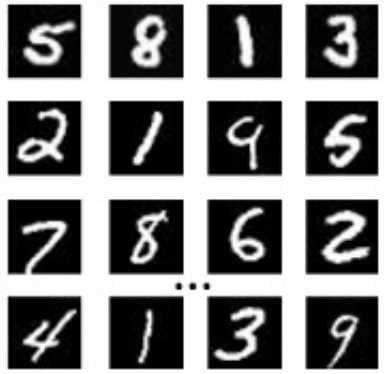
Larger domain gap -> lower recognition accuracy

Unknown

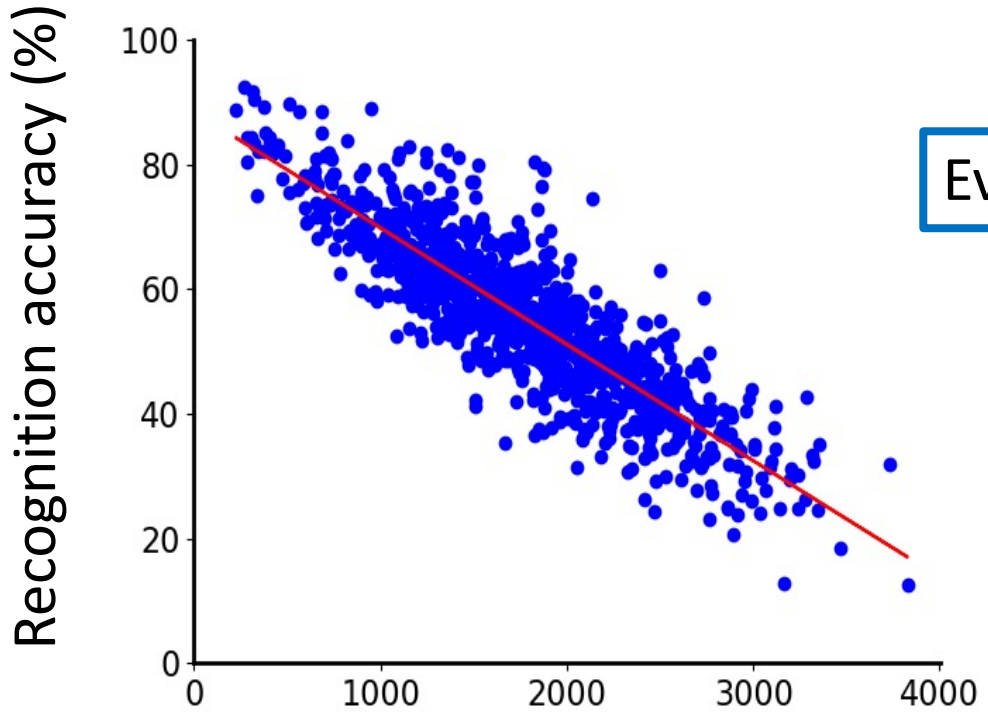
Can we quantify this relationship?

A regression problem!

Some experiments



digit classification



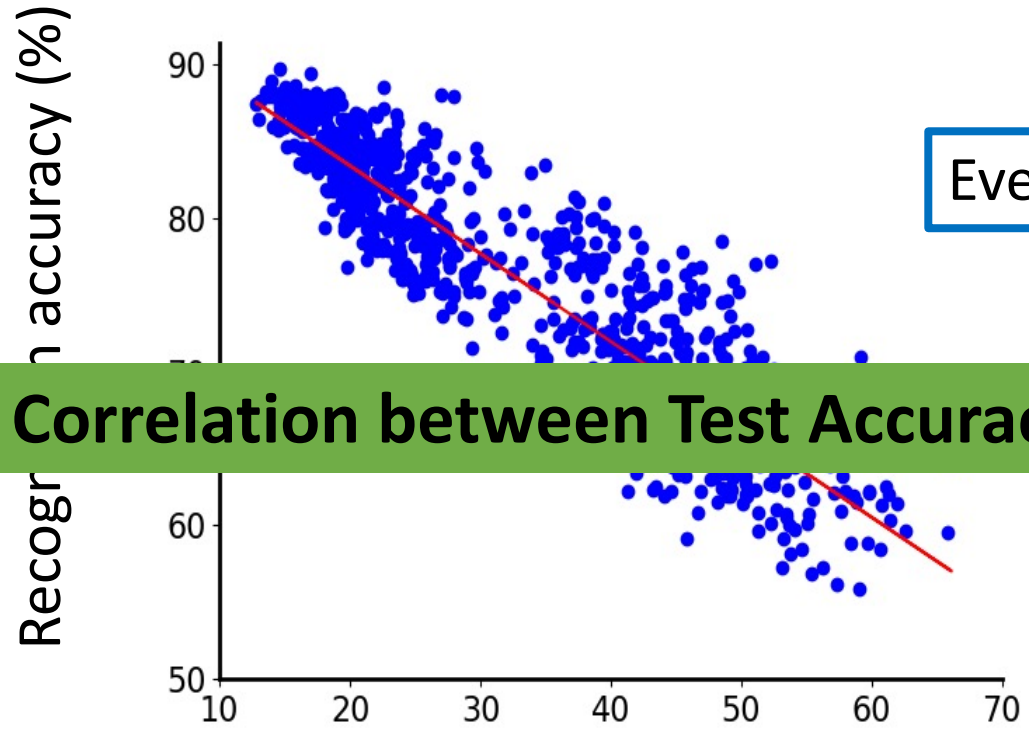
Every point is a dataset

Fréchet distance

Domain gap between a training set and test sets

Some experiments

natural image classification

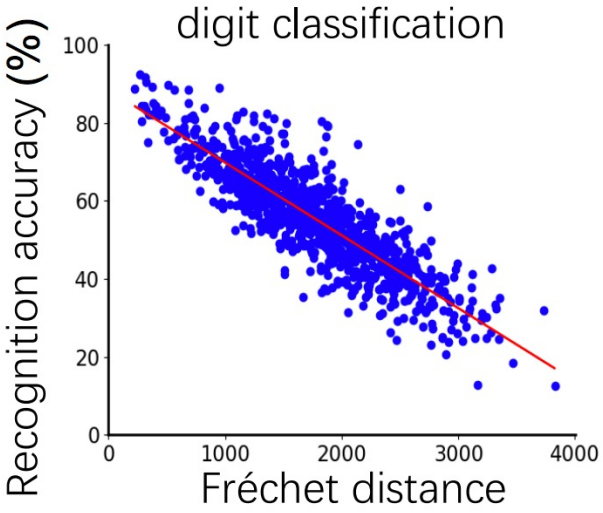


Every point is a dataset

Negative Linear Correlation between Test Accuracy and Distribution Shift

Domain gap between a training set and test sets

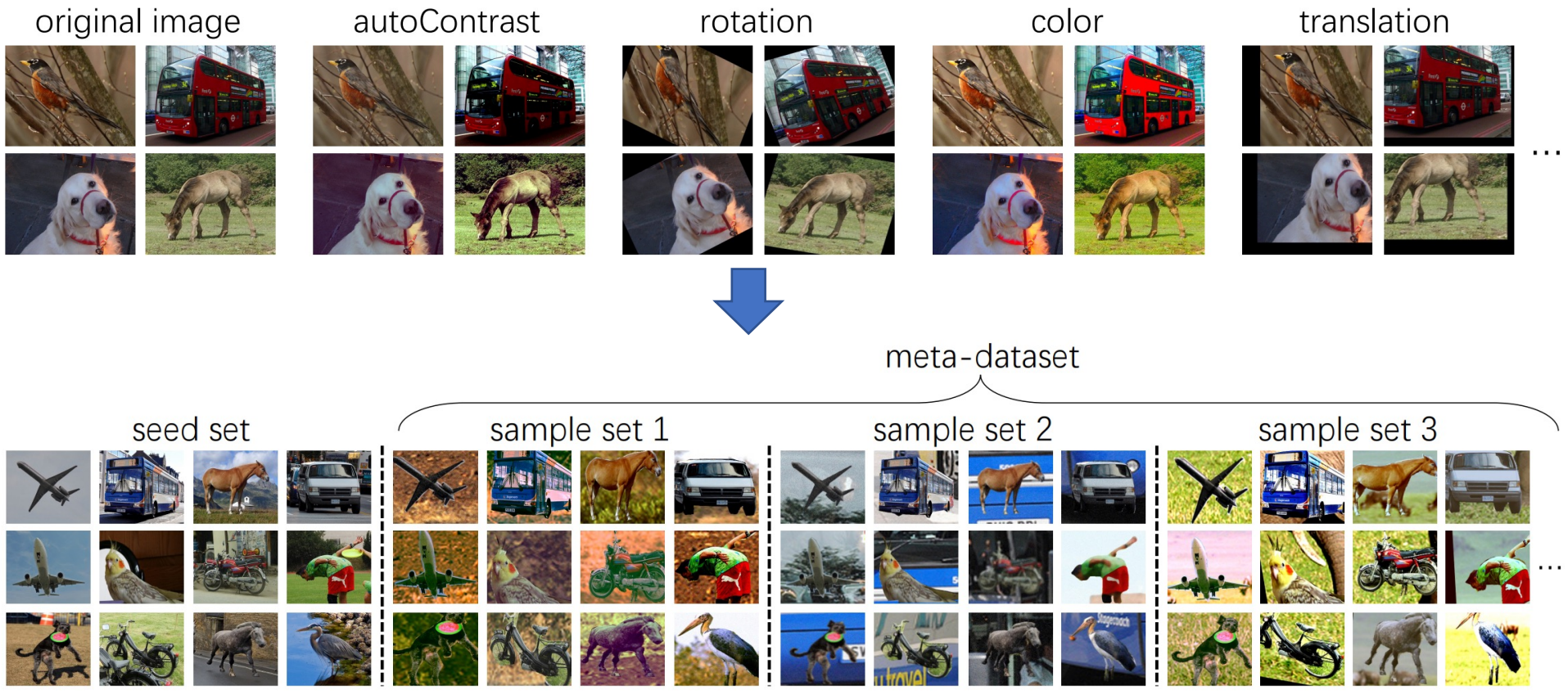
Method key points



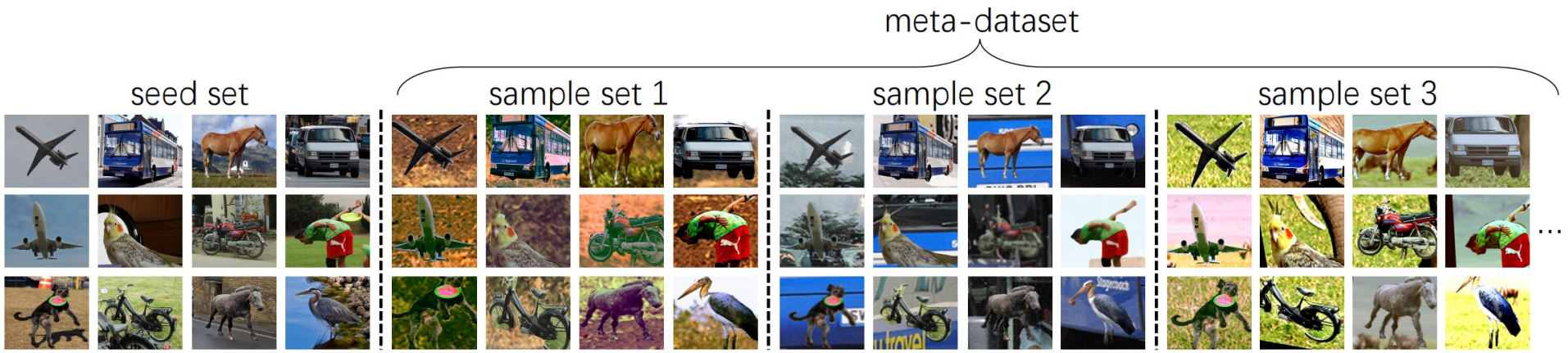
- How can we have many datasets?
- How to obtain the recognition accuracy for each dataset?
- Dataset representation
  - Fréchet distance?
  - Other representations?
- We use regression to relate dataset representation with accuracy

How can we have many datasets?

- Using image transformations



How to obtain the accuracy for each dataset?



Labels of the sample sets are inherited from the seed set

Given a classifier, the accuracy on each sample set can be easily calculated

Dataset representation

- **Method 1:** Fréchet distance (FD) between a sample set and the original training set

$$f_{linear} = \mathrm{FD}(\mathcal{D}_{ori}, \mathcal{D}) = \|\boldsymbol{\mu}_{ori} - \boldsymbol{\mu}\|_2^2 + Tr(\boldsymbol{\Sigma}_{ori} + \boldsymbol{\Sigma} - 2(\boldsymbol{\Sigma}_{ori}\boldsymbol{\Sigma}))^{\frac{1}{2}}$$

- FD: distribution difference between two domains
- Including mean and covariance
- Dimension of f_{linear}: 1

- We thus can use linear regression to predict accuracy

$$a_{linear} = A_{linear}(\boldsymbol{f}) = w_1 f_{linear} + w_0$$

Dataset representation

- **Method 2:** FD + mean + covariance (sum)

$$\boldsymbol{f}_{neural} = [f_{linear}; \boldsymbol{\mu}; \boldsymbol{\sigma}]$$

- We calculate $\boldsymbol{\sigma}$ by taking a weighted summation of each row of $\boldsymbol{\Sigma}$ to produce a single vector
- Dimension of f_{linear}: $2d + 1$

  (d is the dimension of an image feature)

- We use neural network regression

$$a_{neural} = A_{neural}(\boldsymbol{f}_{neural})$$

Experiment

Settings	Training set	Seed set	Test sets
Digit classification	MNIST training set	MNIST test set	SVHN and USPS
Natural image classification	COCO training set	COCO validation set	PASCAL, ImageNet, and Caltech

We use root mean squared error (RMSE) to evaluate the accuracy of recognition accuracy prediction.

Experiment

Train Set	Digits			Natural images			
Unseen Test Set	SVHN	USPS	RMSE↓	Pascal	Caltech	ImageNet	RMSE↓
Ground-truth accuracy	25.46	64.08	-	86.13	93.40	88.83	-
Predicted score ($\tau = 0.7$)	10.09	43.60	18.11	88.34	93.28	90.17	1.49
Predicted score ($\tau = 0.8$)	7.97	37.22	22.66	84.32	90.78	86.50	2.28
Predicted score ($\tau = 0.9$)	7.03	32.94	25.59	78.61	87.71	81.33	6.96

“Predicted Score”: a simple pseudo label method.

If the maximum value of the softmax outputs is greater than τ, we view this sample as correctly classified.

Experiment

Train Set	Digits			Natural images			
Unseen Test Set	SVHN	USPS	RMSE↓	Pascal	Caltech	ImageNet	RMSE↓
Ground-truth accuracy	25.46	64.08	-	86.13	93.40	88.83	-
Predicted score ($\tau = 0.7$)	10.09	43.60	18.11	88.34	93.28	90.17	1.49
Predicted score ($\tau = 0.8$)	7.97	37.22	22.66	84.32	90.78	86.50	2.28
Predicted score ($\tau = 0.9$)	7.03	32.94	25.59	78.61	87.71	81.33	6.96
Linear reg.	26.28	50.14	9.87	83.87	79.77	83.19	8.62

Linear regression achieves good estimations on some test sets

Experiment

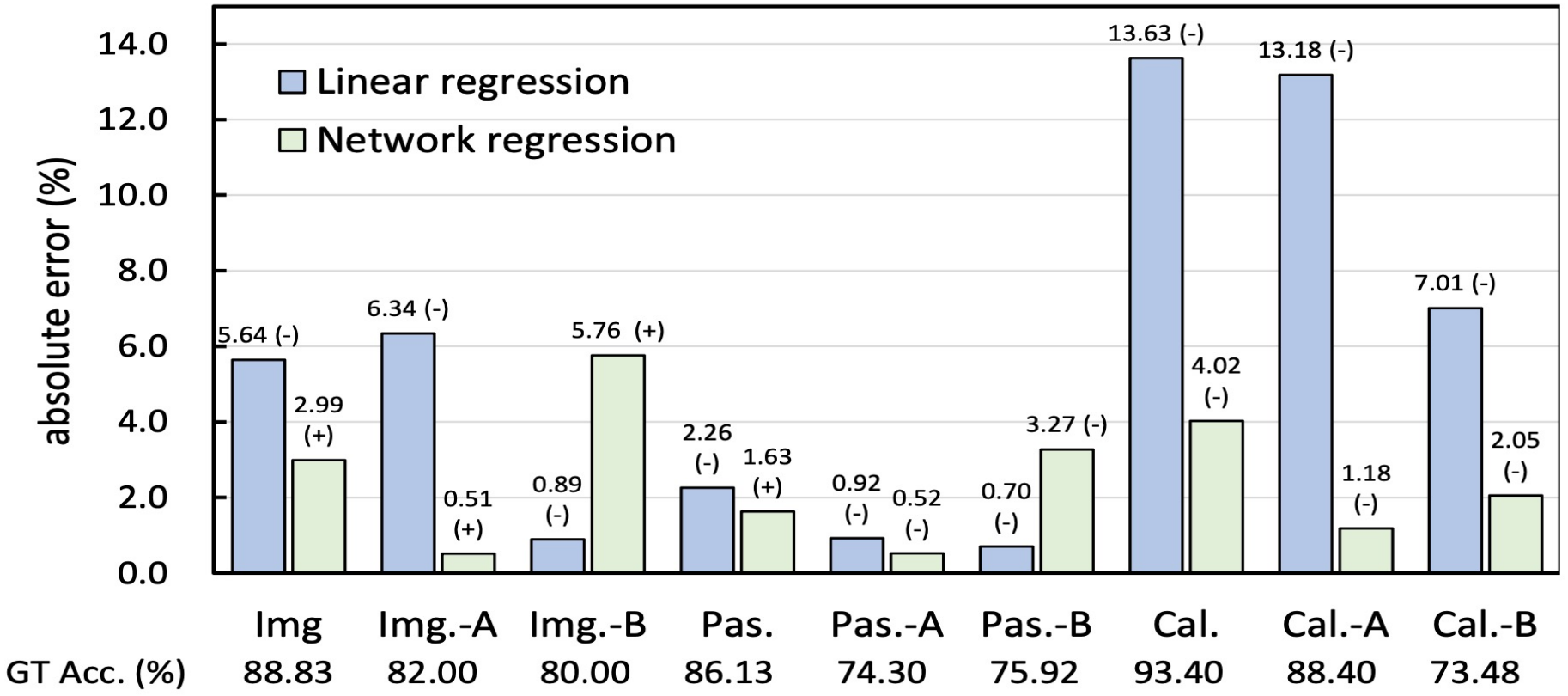
Train Set	Digits			Natural images			
Unseen Test Set	SVHN	USPS	RMSE↓	Pascal	Caltech	ImageNet	RMSE↓
Ground-truth accuracy	25.46	64.08	-	86.13	93.40	88.83	-
Predicted score ($\tau = 0.7$)	10.09	43.60	18.11	88.34	93.28	90.17	1.49
Predicted score ($\tau = 0.8$)	7.97	37.22	22.66	84.32	90.78	86.50	2.28
Predicted score ($\tau = 0.9$)	7.03	32.94	25.59	78.61	87.71	81.33	6.96
Linear reg.	26.28	50.14	9.87	83.87	79.77	83.19	8.62
Neural network reg.	27.52	64.11	1.46	87.76	89.39	91.82	3.04

The two regression methods are stable

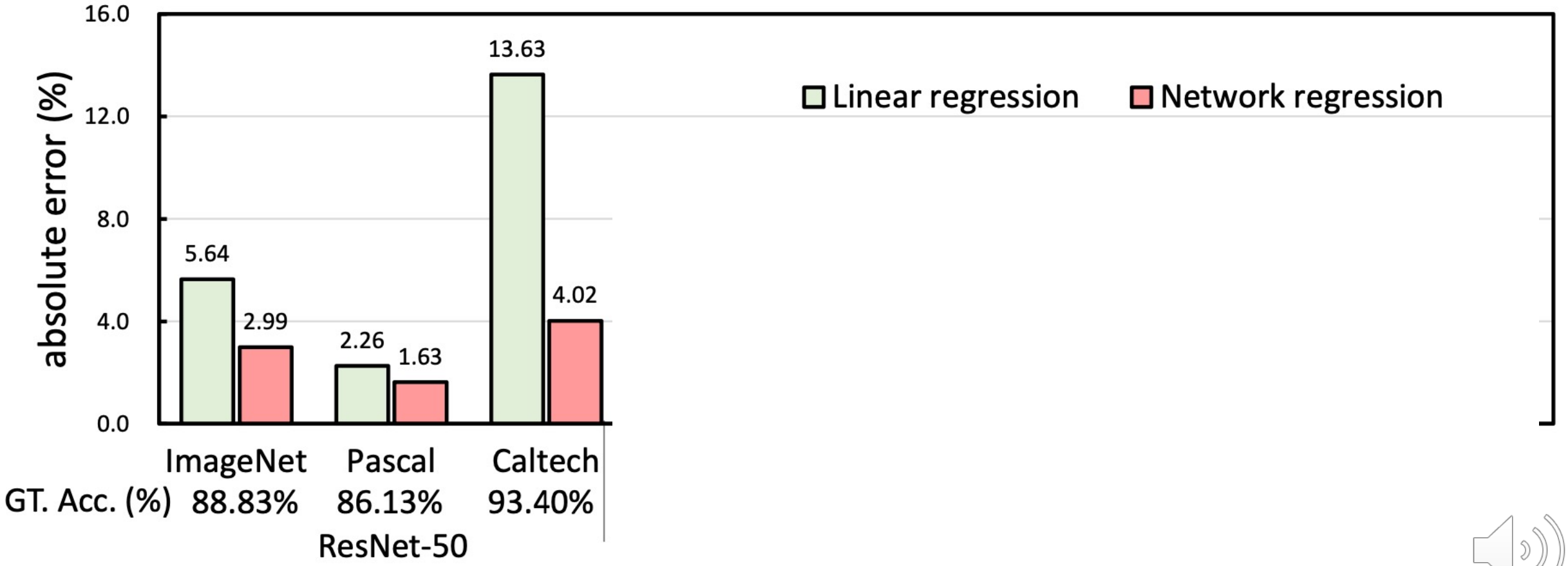
Network regression produces promising estimations

Test sets undergo new transformations

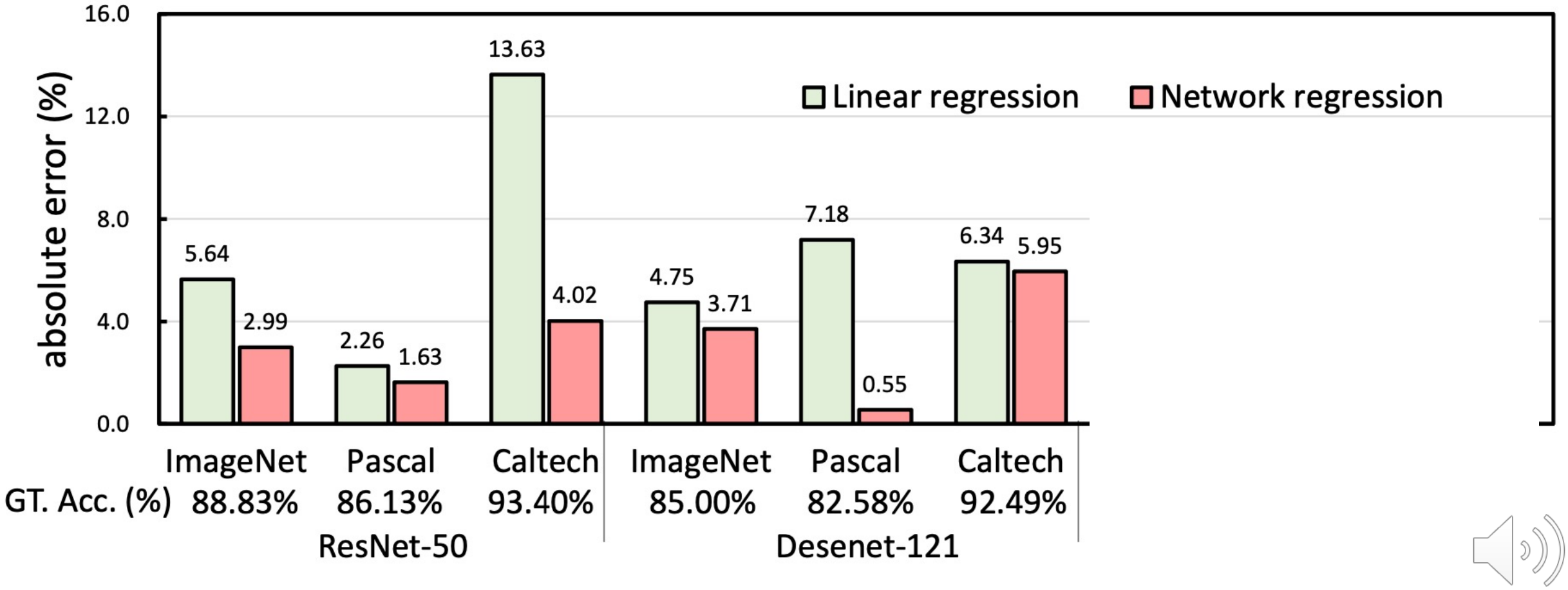
- We add new image transformations to the test sets.
- Random erasing / cutout, Shear, Equalize and ColorTemperature



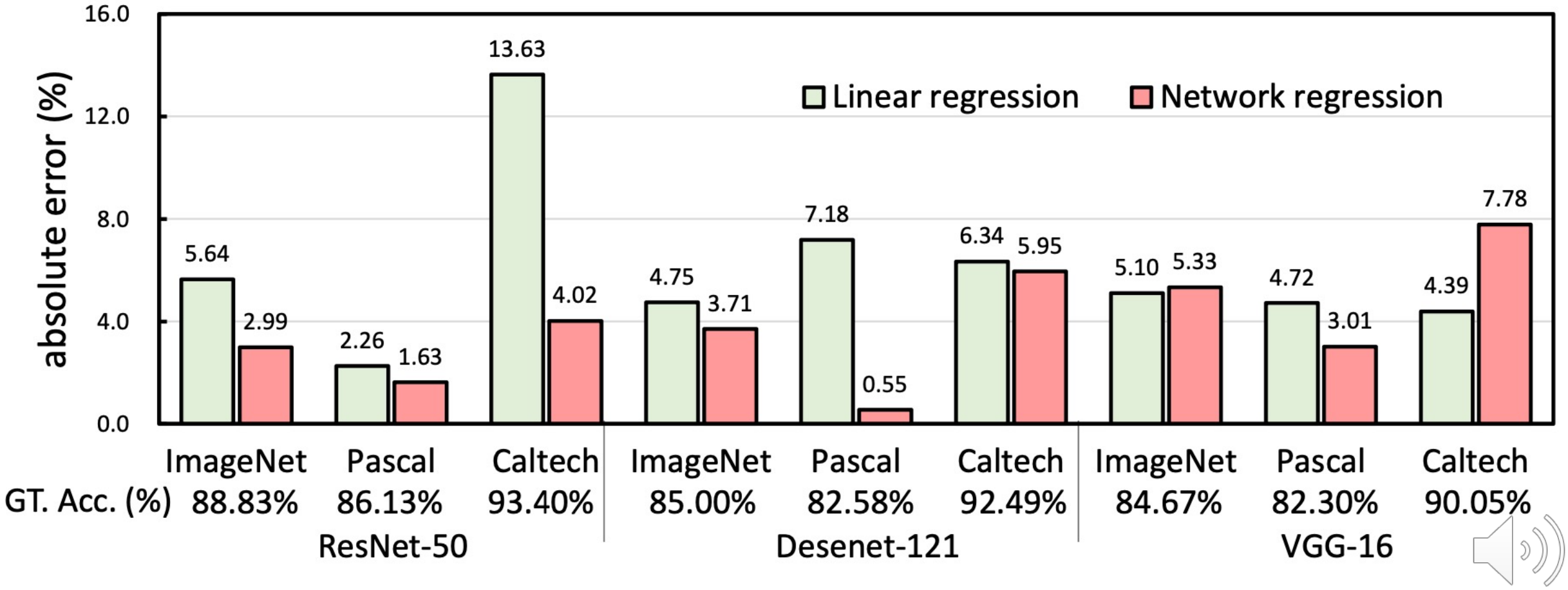
Predicting the accuracy of various classifiers



Predicting the accuracy of various classifiers



Predicting the accuracy of various classifiers



Conclusions and insights

- We study a very interesting problem:

  Evaluating model performance *without* ground truths

- We use a very simple method:

  Dataset-level regression (Linear regression and Neural network regression)

- Potential Applications:

  Object recognition, detection, segmentation, re-identification, etc.

Conclusions and insights

- Application scope
  - The space spanned by the sample sets ***(Meta-dataset) should cover*** the test sets
  - If not, there will be failure cases
- Dataset representation
  - A less studied problem
  - We use ***first- and second-order*** feature statistics and ***FD***
  - Better representations?
- Dataset similarity
  - We use FD score
  - Better similarity estimation? (*JS* …)

Thank you!

The code is available at
https://weijiandeng.xyz/AutoEval