A New Learning Paradigm - Green Learning

C.-C. Jay Kuo
University of Southern California
Concerns with Deep Learning

- May not be suitable for academic research
  - Demanding heavy resources
    Computing resource (GPU)
    Data collection/labeling cost
  - Engineering fine-tuning
    Blackbox tools – discouraging original thinking

- Previous examples
  - Computer graphics and SIGGRAPH
  - Image/video coding and standards
Green Learning as An Alternative

Green Machine Learning (or Green AI)

- Decouple “feature extraction” and “decision” again
  Feature extraction – unsupervised, statistics-based, signal processing (filter banks)
  Decision – classification, regression, etc.

- Unique characteristics
  Low power consumption in both training and testing
  Small model sizes
  Suitable for edge/mobile devices
  Also, beneficial to carbon footprint reduction in cloud servers
Outline

• Why Green Learning?
• Fundamentals of Green Learning
• Green Learning for Image Classification
• Green Learning for Fake Image Detection
• Green Learning for Point Cloud Classification and Registration
Why Green Learning?
How About Image Models?

Number of Model Parameters

<table>
<thead>
<tr>
<th>Number of Layers</th>
<th>Number of Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet 18</td>
<td>11.174M</td>
</tr>
<tr>
<td>ResNet 34</td>
<td>21.282M</td>
</tr>
<tr>
<td>ResNet 50</td>
<td>23.521M</td>
</tr>
<tr>
<td>ResNet 101</td>
<td>42.513M</td>
</tr>
<tr>
<td>ResNet 152</td>
<td>58.157M</td>
</tr>
</tbody>
</table>

• Transformer is much more expensive than CNN
  • Transformer: Multiple feed-forward layers (close to MLP)
  • CNN: shared filters
Development of Language Models

GPT 3 (2020) = 10 * Turing NLG

- Data hungry
- Huge model size
- Better performance

https://towardsdatascience.com/gpt-3-the-new-mighty-language-model-from-openai-a74ff35346fc
Environmental Problem
Carbon Footprint for DL in NLP

Common carbon footprint benchmarks

<table>
<thead>
<tr>
<th>Consumption</th>
<th>CO₂e (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air travel, 1 passenger, NY↔SF</td>
<td>1984</td>
</tr>
<tr>
<td>Human life, avg. 1 year</td>
<td>11,023</td>
</tr>
<tr>
<td>American life, avg. 1 year</td>
<td>36,156</td>
</tr>
<tr>
<td>Car, avg incl. fuel, 1 lifetime</td>
<td>126,000</td>
</tr>
</tbody>
</table>

Training one model (GPU)

<table>
<thead>
<tr>
<th>Consumption</th>
<th>CO₂e (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP pipeline (parsing, SRL) w/ tuning &amp; experimentation</td>
<td>78,468</td>
</tr>
<tr>
<td>Transformer (big) w/ neural architecture search</td>
<td>626,155</td>
</tr>
</tbody>
</table>

Table 1: Estimated CO₂ emissions from training common NLP models, compared to familiar consumption.

Chart: MIT Technology Review • Source: Strubell et al. • Created with Datawrapper

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? (Timnit Gebru, 2020)
What Green Learning Attempts to Achieve

Objectives:
- Low power consumption in training and inference (primary goal)
- Small model size
- High performance
- Weak supervision
- Interpretability
Fundamentals of Green Learning
Green Learning (GL)

Traditional Pattern Recognition Paradigm
• 1\textsuperscript{st} module (from data to features) - feature extraction
• 2\textsuperscript{nd} module (from features to decision) – classifier or regressor

Deep Learning Paradigm
• An integrated module (from data to decision)

Advantages of modular design
• Multi-tasking
• Unsupervised feature learning
Supervised Feature Learning in DL

Exemplary network: LeNet-5

2 convolutional layers + 2 FC layers + 1 output layer
Unsupervised Feature Learning in GL

Filter banks
- Multiple filters operating on local spatial patches
- Joint spatial-spectral representation

Filter kernel design
- Kernels form a base of a linear subspace
- Subspace approximation
One-Stage Transform with Filter Banks

Filter Banks: A set of filters operating in parallel on the input
Example: Laws’ 3x3 filters for texture analysis
Laws’ Filter Banks

• **Input & Output of Laws’ filter bank**
  • Input: an image of NxN pixels
  • Output (w/o padding): a 3D tensor of dimension (N-2)x(N-2)x9

• **Interpretation**
  • The response of each filter indicates the frequency components of a local neighborhood of size 3x3 (9 channels)

• **Limitations**
  • Filter coefficients are fixed (not adaptive to image contents)
  • Only one-stage transform (no information of mid- and long-range neighborhood)
New Transforms for Unsupervised Feature Learning

• **One-stage Transform**
  • Saab transform
    • Saab means “subspace approximation with adjusted bias”
    • Improved Laws’ filter banks
    • A variant of PCA

• **Multi-stage Transform**
  • channel-wise (c/w) Saab transform
Saab Transform

• Subspace decomposition

\[ S = S_{DC} \oplus S_{AC} \]

• DC subspace is spanned by constant-element vector \( d = (1, \ldots, 1) \)
• AC subspace is its orthogonal complement
• Conduct PCA on the AC subspace
Example of Saab Transform (1)

• **Gray-scale images: 3x3x1 Saab Transform**
  • 1 DC filter: constant-element filter (= mean of a 3x3 patch)
  • 8 AC filters (PCA analysis applied to AC components)
    • Covariance matrix of mean-removed 3x3 patches
    • The first 8 eigenvectors of the 9x9 covariance matrix
      • The last eigenvector has an eigenvalue close to 0
  • Output: (N-2)x(N-2)x9 three-D tensor
  • Why not apply PCA to 3x3 patches directly
    • Need to subtract the ensemble mean of these 3x3 patches, which is sensitive to the image input
    • The ensemble mean of residuals approximates to a zero vector
Example of Saab Transform (2)

• **Color images: 3x3x3 Saab Transform**
  - 1 DC filter: constant-element filter (= mean of a 3x3x3 patch)
  - 26 AC filters (PCA analysis applied to AC components)
    • Covariance matrix of mean-removed 3x3x3 patches
    • The first 26 eigenvectors of the 27x27 covariance matrix
      • The last eigenvector has an eigenvalue close to 0

• **Common filter sizes in spatial domain**
  - 2x2, 3x3, 4x4, 5x5, etc.
  - Should avoid the use of large filter sizes
    • Correlation between long-range pixels is weaker
    • The dimension of the output tensor would become too large
Lossless and Lossy Saab Transform

• **Example of Lossless Saab Transform**
  • Input: NxN (N even)
  • Filter size: 2x2
  • Stride: 2
  • Output: (N/2)x(N/2)x4

• **Redundant Saab Transform**
  • The above setting but with stride = 1
  • Output: (N-2)x(N-2)x4
  • Redundancy removal: (2x2) to (1x1) pooling

• **Lossy Saab Transform**
  • Discard channels with small responses – dimension reduction
Generalization to Multi-Stage Saab Transform

Neighborhood Construction \( \rightarrow \) Subspace Approximation via Saab Transform

\[ K_{i-1} \rightarrow 9 \times K_{i-1} \rightarrow \text{Saab Transform} \rightarrow K_i \]
Correlation Analysis of Saab Coefficients

Table 1. Averaged correlations of filtered AC outputs from the first to the third Pixelhop units with respect to the MNIST, Fashion MNIST and CIFAR-10 datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MNIST</th>
<th>Fashion MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial 1</td>
<td>0.48 ± 0.05</td>
<td>0.51 ± 0.03</td>
<td>0.53 ± 0.03</td>
</tr>
<tr>
<td>Spatial 2</td>
<td>0.22 ± 0.03</td>
<td>0.29 ± 0.05</td>
<td>0.27 ± 0.06</td>
</tr>
<tr>
<td>Spectral 1</td>
<td>0.33 ± 0.07</td>
<td>0.12 ± 0.02</td>
<td>0.0156 ± 0.0005</td>
</tr>
<tr>
<td>Spectral 2</td>
<td>0.18 ± 0.02</td>
<td>0.13 ± 0.01</td>
<td>0.0188 ± 0.0004</td>
</tr>
<tr>
<td>Spectral 3</td>
<td>0.0099 ± 0.0001</td>
<td>0.0082 ± 0.0001</td>
<td>0.0079 ± 0.0004</td>
</tr>
</tbody>
</table>
Comparison of Saab and c/w Saab Transforms

Saab Transform

\( S_t \times S_t \times K'_t \) \[ \text{Saab Transform} \] \( S_{t+1} \times S_{t+1} \times K'_{t+1} \)

Channel-wise (c/w) Saab Transform

\( (S_t \times S_t, K_t) \) \[ \text{Channel-wise Saab Transform} \] \( (S_{t+1} \times S_{t+1}, K_{t+1}) \)

\( K_{t+1} \)
3-Hop c/w Saab Transform

- Short-range correlation
- Mid-range correlation
- Long-range correlation
Frequently Asked Questions

• **Is the Saab transform linear?**
  • Yes. More precisely, it is an affine transform.

• **Can a linear transform yield powerful features?**
  • Nonlinear classifiers are important
  • Linear features may not be that bad
    • Easy to understand
    • Clustering can increase the power of Saab features
      • Clustering can be done after (or before) the Saab transform
Green Learning for Image Classification

Experiment Set-up

❖ Datasets:
  ➢ MNIST
    ■ Handwritten digits 0-9
    ■ Gray-scale images with size 32x32
    ■ Training set: 60k, Testing set: 10k
  ➢ Fashion-MNIST
    ■ Gray-scale fashion images with size 32 × 32
    ■ Training set: 60k, Testing set: 10k
  ➢ CIFAR-10
    ■ 10 classes of tiny RGB images with size 32 × 32
    ■ Training set: 50k, Testing set: 10k

❖ Evaluation:
  ➢ Top-1 classification accuracy
Performance Comparison

Table 8
Comparison of testing accuracy (%) of LeNet-5, feedforward-designed CNN (FF-CNN), PixelHop and PixelHop+ for MNIST, Fashion MNIST and CIFAR-10.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST</th>
<th>Fashion MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet-5</td>
<td>99.04</td>
<td>91.08</td>
<td>68.72</td>
</tr>
<tr>
<td>FF-CNN</td>
<td>97.52</td>
<td>86.90</td>
<td>62.13</td>
</tr>
<tr>
<td>PixelHop</td>
<td>98.90</td>
<td>91.30</td>
<td>71.37</td>
</tr>
<tr>
<td>PixelHop+</td>
<td>99.09</td>
<td>91.68</td>
<td>72.66</td>
</tr>
</tbody>
</table>

Table 9
Comparison of training time of the LeNet-5 and the PixelHop method on the MNIST, the Fashion MNIST and the CIFAR-10 datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST</th>
<th>Fashion MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet-5</td>
<td>~25 min</td>
<td>~25 min</td>
<td>~45 min</td>
</tr>
<tr>
<td>PixelHop</td>
<td>~15 min</td>
<td>~15 min</td>
<td>~30 min</td>
</tr>
</tbody>
</table>
Weak Supervision

- **MNIST**
  - Test ACC (%)
  - Number of Labeled Training Samples

- **Fashion-MNIST**
  - Test ACC (%)
  - Number of Labeled Training Samples

- **CIFAR-10**
  - Test ACC (%)
  - Number of Labeled Training Samples

Graphs show the performance of PixelHop and LeNet-5 models on different datasets with varying numbers of labeled training samples.
PixelHop++

Module 1

Module 2

Module 3

Classifier

Predicted Object Class
# Model Size and Test Accuracy Comparison

**Table 3.** Comparison of test accuracy (%) of LeNet-5 and PixelHop++ for MNIST, Fashion MNIST and CIFAR-10.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST</th>
<th>Fashion MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet-5</td>
<td>99.04</td>
<td>89.74</td>
<td>68.72</td>
</tr>
<tr>
<td>PixelHop++ (Large)</td>
<td>98.49</td>
<td>90.17</td>
<td>66.81</td>
</tr>
<tr>
<td>PixelHop++ (Small)</td>
<td>97.98</td>
<td>88.84</td>
<td>64.75</td>
</tr>
</tbody>
</table>

**Table 4.** Comparison of the model size (in terms of the total parameter numbers) of LeNet-5 and PixelHop++ for the MNIST, the Fashion MNIST and the CIFAR-10 datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST</th>
<th>Fashion MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet-5</td>
<td>61,706</td>
<td>194,558</td>
<td>395,006</td>
</tr>
<tr>
<td>PixelHop++ (Large)</td>
<td>111,981</td>
<td>127,186</td>
<td>115,623</td>
</tr>
<tr>
<td>PixelHop++ (Small)</td>
<td>29,514</td>
<td>33,017</td>
<td>62,150</td>
</tr>
</tbody>
</table>
Green Learning for Fake Image Detection

Introduction

• **Deepfake videos** are synthetic media in which a person in a video is replaced with someone else.

• **Deepfake videos** can be potentially harmful to society, from non-consensual explicit content creation to forged media by foreign adversaries used in disinformation campaigns.

• As the number of Deepfake video contents grows rapidly, an **automatic and effective Deepfake detection** mechanism is in urgent need.
Motivation

- Most state-of-the-art Deepfake detection methods are based upon deep learning (DL) technique.
- They can be mainly categorized into two types:
  - convolutional neural networks (CNNs)
  - integrate CNNs and recurrent neural networks (RNNs)
Motivation

• The size of DL-based methods is **large** -- containing hundreds of thousands or even millions of model parameters

• **Training** deep neural networks is **computationally expensive**

• There are also **non-DL-based Deepfake detection** methods, where handcrafted features are extracted and fed into classifiers

• The performance of non-DL-based methods is usually **inferior to that of DL-based ones**

• Our goal is to develop a **light-weight** non-DL-based methods and achieve a high-performance results
Face Preprocessing

Sampling Frames → Landmarks Extraction → Face Alignment → Regions Extraction
Defakehop Framework
Channel Wise Saab Transform
## Model Size

### Table 4. The number of parameters for various parts.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Number of Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixelhop++ Hop-1</td>
<td>270</td>
</tr>
<tr>
<td>Pixelhop++ Hop-2</td>
<td>90</td>
</tr>
<tr>
<td>Pixelhop++ Hop-3</td>
<td>90</td>
</tr>
<tr>
<td>PCA Hop-1</td>
<td>10,125</td>
</tr>
<tr>
<td>PCA Hop-2</td>
<td>1,225</td>
</tr>
<tr>
<td>PCA Hop-3</td>
<td>45</td>
</tr>
<tr>
<td>Channel-Wise XGBoost(s)</td>
<td>12,000</td>
</tr>
<tr>
<td>Final XGBoost</td>
<td>19,000</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>42,845</strong></td>
</tr>
</tbody>
</table>
Datasets

- We use two datasets from 1st generation dataset and two datasets from 2nd generation dataset.
- The numbers of real, fake, train and test video for each dataset are shown.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Real</th>
<th>Fake</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>UADFV</td>
<td>49</td>
<td>49</td>
<td>78</td>
<td>20</td>
</tr>
<tr>
<td>FaceForensics++</td>
<td>1000</td>
<td>1000</td>
<td>1440</td>
<td>280</td>
</tr>
<tr>
<td>Celeb-DF v1</td>
<td>408</td>
<td>795</td>
<td>1103</td>
<td>100</td>
</tr>
<tr>
<td>Celeb-DF v2</td>
<td>890</td>
<td>5639</td>
<td>6011</td>
<td>518</td>
</tr>
</tbody>
</table>
## Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>1st Generation datasets</th>
<th>2nd Generation datasets</th>
<th>Number of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UADFV</td>
<td>FF++ / DF</td>
<td>Celeb-DF v1</td>
</tr>
<tr>
<td>Zhou et al.(2017) [3]</td>
<td>InceptionV3&lt;sup&gt;a&lt;/sup&gt;</td>
<td>85.1%</td>
<td>70.1%</td>
</tr>
<tr>
<td>Afchar et al.(2018) [4]</td>
<td>Meso4&lt;sup&gt;a&lt;/sup&gt;</td>
<td>84.3%</td>
<td>84.7%</td>
</tr>
<tr>
<td>Li et al.(2018) [17]</td>
<td>FWA&lt;sup&gt;a&lt;/sup&gt; (ResNet-50)</td>
<td>97.4%</td>
<td>80.1</td>
</tr>
<tr>
<td>Yang et al.(2019) [9]</td>
<td>HeadPose&lt;sup&gt;b&lt;/sup&gt; (SVM)</td>
<td>89%</td>
<td>47.3%</td>
</tr>
<tr>
<td>Matern et al.(2019) [11]</td>
<td>VA-MLP&lt;sup&gt;b&lt;/sup&gt;</td>
<td>70.2%</td>
<td>66.4%</td>
</tr>
<tr>
<td>Ressler et al.(2019) [2]</td>
<td>Xception-raw&lt;sup&gt;a&lt;/sup&gt;</td>
<td>80.4%</td>
<td>99.7%</td>
</tr>
<tr>
<td>Nguyen et al.(2019) [5]</td>
<td>Multi-task&lt;sup&gt;a&lt;/sup&gt;</td>
<td>65.8%</td>
<td>76.3%</td>
</tr>
<tr>
<td>Nguyen et al.(2019) [6]</td>
<td>CapsuleNet&lt;sup&gt;a&lt;/sup&gt;</td>
<td>61.3%</td>
<td>96.6%</td>
</tr>
<tr>
<td>Sabir et al.(2019) [8]</td>
<td>DenseNet+RNN&lt;sup&gt;a&lt;/sup&gt;</td>
<td>-</td>
<td>99.6%</td>
</tr>
<tr>
<td>Li et al.(2020) [17]</td>
<td>DSP-FWA&lt;sup&gt;a&lt;/sup&gt; (SPPNet)</td>
<td>97.7%</td>
<td>93%</td>
</tr>
<tr>
<td>Tolosana et al.(2020) [1]</td>
<td>Xception&lt;sup&gt;a&lt;/sup&gt;</td>
<td>100%</td>
<td>99.4%</td>
</tr>
<tr>
<td>Ours</td>
<td>DefakeHop (Frame)</td>
<td>100%</td>
<td>95.95%</td>
</tr>
<tr>
<td></td>
<td>DefakeHop (Video)</td>
<td>100%</td>
<td>97.45%</td>
</tr>
</tbody>
</table>

Table 2. Comparison of the detection performance of benchmarking methods with the AUC value at the frame level as the evaluation metric. The **boldface** and the underbar indicate the best and the second-best results, respectively. The *italics* means it does not specify frame or video level AUC. The AUC results of DefakeHop is reported in both frame-level and video-level. The AUC results of benchmarking methods are taken from [19] and [20]. <sup>a</sup> deep learning method, <sup>b</sup> non deep learning method.
Experiments

- The ensemble of multiple facial regions can boost the AUC values by up to 5%. Each facial region has different strengths on various faces, and their ensemble gives the best result.

- The performance of DefakeHop degrades by 5% as video quality becomes worse.

<table>
<thead>
<tr>
<th></th>
<th>Left eye</th>
<th>Right eye</th>
<th>Mouth</th>
<th>Ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td>UADFV</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>FF++ / DF</td>
<td>94.37%</td>
<td>93.73%</td>
<td>94.25%</td>
<td>97.45%</td>
</tr>
<tr>
<td>Celeb-DF v1</td>
<td>89.69%</td>
<td>88.20%</td>
<td>92.66%</td>
<td>94.95%</td>
</tr>
<tr>
<td>Celeb-DF v2</td>
<td>85.17%</td>
<td>86.41%</td>
<td>89.66%</td>
<td>90.56%</td>
</tr>
</tbody>
</table>

Table 3. Comparison of Deepfake algorithms and qualities.

<table>
<thead>
<tr>
<th></th>
<th>FF++ with Deepfakes</th>
<th>FF++ with FaceSwap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HQ (c23)</td>
<td>LQ (c40)</td>
</tr>
<tr>
<td>Frame</td>
<td>95.95%</td>
<td>93.01%</td>
</tr>
<tr>
<td>Video</td>
<td>97.45%</td>
<td>95.80%</td>
</tr>
<tr>
<td></td>
<td>HQ (c23)</td>
<td>LQ (c40)</td>
</tr>
<tr>
<td>Frame</td>
<td>97.87%</td>
<td>89.14%</td>
</tr>
<tr>
<td>Video</td>
<td>98.78%</td>
<td>93.22%</td>
</tr>
</tbody>
</table>
Experiments

- DefakeHop can achieve about 85% AUC with less than 5% (250 videos) of the whole training data.

**Fig. 4.** The ROC curve of DefakeHop for different datasets.

**Fig. 5.** The plot of AUC values as a function of the training video number.
2nd generation dataset: Celeb-DF
Our codes are released in GitHub!

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Commits</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>data/UADFV</td>
<td>Delete .DS_Store</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>img</td>
<td>add images and data</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>preprocessing</td>
<td>add preprocessing</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>README.md</td>
<td>Update README.md</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>defakeHop.md</td>
<td>first version</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>model.py</td>
<td>Update model.py</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>multi_cwSaab.py</td>
<td>add images and data</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>saab.py</td>
<td>first version</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>utils.py</td>
<td>first version</td>
<td></td>
<td>2 months ago</td>
</tr>
</tbody>
</table>
Summary

• Defakehop has several advantages
  • a smaller model size
  • fast training procedure
  • high detection AUC
  • needs fewer training samples

• Extensive experiments were conducted to demonstrate its high detection performance
Green Learning for Point Cloud Classification and Registration

Background

**What**: A point cloud is a set of points in the 3D space

**How**: 3D scanning devices such as Lidar, measured by time of flight (ToF)

**Why**: With reduced cost of sensors, point cloud processing has become popular

Datasets and Performance Metrics

**ModelNet-40**
- 40 categories of objects (e.g., airplane, table, desk, sofa)
- Each object has 2048 points

**ShapeNet Part**
- 16 object categories
- 50 parts: each object is annotated with two to six parts
- Each shape has 2048 points

**Evaluation metric:**
- Classification - accuracy
- Segmentation – Intersection over Union (IoU)
- Registration - Mean Square Error (MSE)
Two Topics

Classification: label each object

Registration: align two point clouds
PointHop – A Successive-Subspace-Learning-based (SSL-based) or Green Learning (GL) method

- **Black Box** (e.g., PointNet, PointNet++)
  - High cost
  - Slow speed
  - Hard to interpret
  - GPU

- **White Box** (e.g., PointHop)
  - Low complexity
  - Fast training
  - Interpretable
  - CPU

PointHop

cascade of PointHop units + classifier
spatial sampling scheme: farthest point sampling (FPS)
• reduce computational complexity
• speed up the coverage rate

Constructing a local descriptor with attributes of one-hop neighbors
- Solve both unordered and disturbance problem
- The attributes of a point grow from a low dimension one to a high dimension one

Using the Saab transform to reduce the dimension of the local descriptor
- the dimension grows at a slower rate

**New features:**

- Reduce model complexity – c/w Saab transform
- Order discriminant features automatically based on the cross-entropy criterion
## Performance Evaluation for ModelNet-40: Accuracy and Sparser Models

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervised</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>class-avg</td>
<td>overall</td>
</tr>
<tr>
<td>PointNet [10]</td>
<td>86.2</td>
<td>89.2</td>
</tr>
<tr>
<td>PointNet++ [11]</td>
<td>-</td>
<td>90.7</td>
</tr>
<tr>
<td>PointCNN [12]</td>
<td>88.1</td>
<td>92.2</td>
</tr>
<tr>
<td>DGCNN [13]</td>
<td>90.2</td>
<td>92.2</td>
</tr>
<tr>
<td>LFD-GAN [28]</td>
<td>-</td>
<td>85.7</td>
</tr>
<tr>
<td>FoldingNet [29]</td>
<td>-</td>
<td>88.4</td>
</tr>
<tr>
<td>PointHop [15]</td>
<td>84.4</td>
<td>89.1</td>
</tr>
<tr>
<td>PointHop++ (baseline)</td>
<td>85.6</td>
<td>90.3</td>
</tr>
<tr>
<td>PointHop++ (FS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PointHop++ (FS+ES)</td>
<td>87</td>
<td>91.1</td>
</tr>
</tbody>
</table>
# Performance Evaluation for ModelNet-40: Complexity and Model Size

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (Hours)</th>
<th>Parameter No. (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
<td>Inference</td>
</tr>
<tr>
<td>GPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PointNet [10]</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>PointNet++ [11]</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>DGCNN [13]</td>
<td>21</td>
<td>154</td>
</tr>
<tr>
<td>CPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PointHop [15]</td>
<td>0.33</td>
<td>108</td>
</tr>
<tr>
<td>PointHop++</td>
<td>0.42</td>
<td>97</td>
</tr>
</tbody>
</table>

Zhang, Min, et al. "PointHop++: A Lightweight Learning Model on Point Sets for 3D Classification." ICIP 2020
POINT CLOUD REGISTRATION

• Registration is the process of finding a spatial transformation that optimally aligns two point clouds.

• Register point clouds to merge multiple point cloud scans to get a globally consistent view.

• Registration acts as a pre-processing step before other tasks.
PROBLEM STATEMENT

• We are interested in finding a rigid transformation (rotation and translation) that optimally registers the two point clouds

• **Input** – 1. target point cloud \((N_t \times 3)\)
  
  2. source point cloud \((N_s \times 3)\)

• **Goal** – Align source to target \((N_s \times 3 \rightarrow N_s \times 3)\)

• **Output** – 3D Rotation matrix \((R)\) and translation vector \((t)\) that minimizes point-wise MSE

\[
R = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta_x & -\sin \theta_x \\
0 & \sin \theta_x & \cos \theta_x
\end{bmatrix}
\begin{bmatrix}
\cos \theta_y & 0 & -\sin \theta_y \\
0 & 1 & 0 \\
\sin \theta_y & 0 & \cos \theta_y
\end{bmatrix}
\begin{bmatrix}
\cos \theta_z & -\sin \theta_z & 0 \\
\sin \theta_z & \cos \theta_z & 0 \\
0 & 0 & 1
\end{bmatrix},
\begin{bmatrix}
t_x \\
t_y \\
t_z
\end{bmatrix}
\]

\[
\text{source} = R \ast \text{target} + t
\]
R-POINTHOP

Feature learning

Downsampling

1st hop

R-PointHop unit

N₁ x 24

c/w Saab transform

R-PointHop unit

N₂ x 8

c/w Saab transform

R-PointHop unit

N₃ x 8

c/w Saab transform

R-PointHop unit

concatenate leaf nodes

Nᵢ points

N₁ x 24

N₂ points

N₃ points

Apply transformation

Rᵣ t

SVD of covariance matrix

Mᵢ x D

Transformation estimation

Point correspondence

top M₁ correspondences

Ratio test

top M₁ correspondences

source

Feature distance matrix

(N₁ x Nᵢ)

1st neighbor

2nd neighbor

Nᵢ x D

1st node to next hop

x : discarded node
LOCAL REFERENCE FRAME (LRF)

- Consider a local patch around every point – find $K$ nearest points
- Take a PCA of the $K \times 3$ data matrix
- The three principal components gives orthogonal axes ranked in order of decreasing variance
- These axes are invariant under any rigid transformation (rotation and translation)
POINT ATTRIBUTE CONSTRUCTION

- Find $K$ nearest neighbors of a point
- Project neighbors onto local reference frame (LRF) – XYZ coordinates to local coordinates
- Divide 3D space into 8 octants based on local coordinates – space partitioning
- Compute mean of points in each octant and concatenate 8 means, total 24D attribute
**SIGN RESOLUTION**

- Every eigen vector comes with a sign ambiguity of +/-
- Choice of sign affects space partitioning, hence a consistency is desired
- **Solution** – project the neighboring points onto a principal axis to get 1D local coordinates
- Order points in ascending order and calculate moment about median point
- If left moment > right moment, flip the sign
- This can be formulated as multiplying local coordinates with reflection matrix whose diagonal entries are 1 / -1
MULTI-HOP FEATURE LEARNING

1. Channel-wise Saab transform – discard nodes with energy less than threshold
2. Downsample the point cloud
3. Repeat attribute building step (R-PointHop unit)
4. Collect all leaf nodes at end of fourth hop (point feature)
FEATURE LEARNING

1. Use of LRF makes features invariant to rotation and translation – robust correspondence
2. Hierarchical multi-hop approach helps learn short-, mid- and long-range point relations
3. No class label, no pairs of point clouds with ground truth transformations to learn Saab kernels
4. One pass feedforward
5. Independent of correspondence and transformation estimation modules
POINT CORRESPONDENCES

• For every point in the source, find its matching point in target using nearest neighbor in feature space

• Select a subset of good correspondences (NEW!!)

1. Smaller l2 distance in feature space

2. Smaller ratio of distance to first neighbor by distance to second neighbor
POINT CORRESPONDENCES

- Rich feature information to filter out correspondences against use of spatial information in SPA
- Favors partial point cloud registration –
  1. overlapping points have smaller l2 distance in feature space
  2. Cannot comment from their spatial coordinates
TRANSFORMATION ESTIMATION – SVD

- Given corresponding points \((f_i, g_i)\) solve the orthogonal procrustes problem

1. Find means \(\bar{f} = \frac{1}{N} \sum_{i=1}^{N} f_i\) and \(\bar{g} = \frac{1}{N} \sum_{i=1}^{N} g_i\)

2. Compute covariance matrix \(Cov(F, G) = \sum_{i=1}^{N} (f_i - \bar{f})(g_i - \bar{g})^T\)

3. Find SVD of covariance matrix \(Cov(F, G) = USV^T\)

4. Optimal R and t are given by \(R^* = VU^T\) and \(t^* = -R^*\bar{f} + \bar{g}\)

MODELNET40 DATASET

- 40 classes of CAD models
- 2048 points in each point cloud
- 9840 training samples

EXPERIMENTAL SETUP

• For training – use the target point clouds (without rotations)
• For testing – apply a random rotation and translation along three coordinate axes to get source
• Rotation – uniformly sample rotation angle in \([0, 45^\circ]\)
• Translation – uniformly sample in \([-0.5,0.5]\)
• Comparisons with – ICP, Go-ICP, FGR (model free)
  - PointNetLK, Deep Closest Point (DCP), PR-Net (deep learning)
  - Salient Points Analysis (SPA)
• Evaluation metric – MSE, RMSE, MAE between ground truth and predicted angles
TEST ON UNSEEN OBJECTS AND UNSEEN CLASSES

- Best performance among benchmark methods
- Moreover, R-PointHop can better generalize on unseen classes than DCP and PointNetLK

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE (R)</th>
<th>RMSE (R)</th>
<th>MAE (R)</th>
<th>MSE (t)</th>
<th>RMSE (t)</th>
<th>MAE (t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICP [8]</td>
<td>451.11</td>
<td>21.24</td>
<td>17.69</td>
<td>0.049701</td>
<td>0.222937</td>
<td>0.184111</td>
</tr>
<tr>
<td>Go-ICP [31]</td>
<td>140.47</td>
<td>11.85</td>
<td>2.59</td>
<td>0.00659</td>
<td>0.025665</td>
<td>0.007092</td>
</tr>
<tr>
<td>FGR [33]</td>
<td>87.66</td>
<td>9.36</td>
<td>1.99</td>
<td>0.000194</td>
<td>0.013939</td>
<td>0.002839</td>
</tr>
<tr>
<td>PointNetLK [25]</td>
<td>227.87</td>
<td>15.09</td>
<td>4.23</td>
<td>0.000487</td>
<td>0.022065</td>
<td>0.005405</td>
</tr>
<tr>
<td>DCP [23]</td>
<td>1.31</td>
<td>0.14</td>
<td>0.77</td>
<td>0.000003</td>
<td>0.001786</td>
<td>0.001195</td>
</tr>
<tr>
<td>SPA [26]</td>
<td>318.41</td>
<td>17.84</td>
<td>5.43</td>
<td>0.000022</td>
<td>0.004690</td>
<td>0.003261</td>
</tr>
<tr>
<td>R-PointHop</td>
<td>0.12</td>
<td>0.34</td>
<td>0.24</td>
<td>0.000000</td>
<td>0.000374</td>
<td>0.000295</td>
</tr>
</tbody>
</table>
TEST ON NOISY DATA

- Robust to noise
- Refinement using ICP further reduces error

### TABLE III
**REGISTRATION ON NOISY POINT CLOUDS**

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE (R)</th>
<th>RMSE (R)</th>
<th>MAE (R)</th>
<th>MSE (t)</th>
<th>RMSE (t)</th>
<th>MAE (t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICP [8]</td>
<td>558.38</td>
<td>23.63</td>
<td>19.12</td>
<td>0.058166</td>
<td>0.241178</td>
<td>0.206283</td>
</tr>
<tr>
<td>Go-ICP [31]</td>
<td>131.18</td>
<td>11.45</td>
<td>2.53</td>
<td>0.000531</td>
<td>0.023051</td>
<td>0.004192</td>
</tr>
<tr>
<td>FGR [33]</td>
<td>607.69</td>
<td>24.65</td>
<td>10.05</td>
<td>0.011876</td>
<td>0.108977</td>
<td>0.027393</td>
</tr>
<tr>
<td>PointNetLK [25]</td>
<td>256.15</td>
<td>16.00</td>
<td>4.59</td>
<td>0.000465</td>
<td>0.021558</td>
<td>0.005652</td>
</tr>
<tr>
<td>DCP [23]</td>
<td>1.17</td>
<td>1.08</td>
<td>0.74</td>
<td>0.000002</td>
<td>0.001500</td>
<td>0.001053</td>
</tr>
<tr>
<td>SPA [26]</td>
<td>331.73</td>
<td>18.21</td>
<td>6.28</td>
<td>0.000462</td>
<td>0.021511</td>
<td>0.004100</td>
</tr>
<tr>
<td>R-PointHop</td>
<td>7.73</td>
<td>2.78</td>
<td>0.98</td>
<td>0.000001</td>
<td>0.000874</td>
<td>0.003748</td>
</tr>
<tr>
<td>R-PointHop + ICP</td>
<td>1.16</td>
<td>1.08</td>
<td>0.21</td>
<td>0.000001</td>
<td>0.000744</td>
<td>0.001002</td>
</tr>
</tbody>
</table>

Add Gaussian noise of zero mean and 0.01 variance to source
TEST ON PARTIAL DATA

- Only a part of the source and target point cloud visible

**TABLE IV**

**REGISTRATION ON PARTIAL POINT CLOUDS (R-PointHop® INDICATES CHOOSING CORRESPONDENCES WITHOUT THE RATIO TEST).**

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE (R)</th>
<th>RMSE (R)</th>
<th>MAE (R)</th>
<th>MSE (t)</th>
<th>RMSE (t)</th>
<th>MAE (t)</th>
<th>MSE (R)</th>
<th>RMSE (R)</th>
<th>MAE (R)</th>
<th>MSE (t)</th>
<th>RMSE (t)</th>
<th>MAE (t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICP [8]</td>
<td>1134.55</td>
<td>33.68</td>
<td>25.05</td>
<td>0.0856</td>
<td>0.2930</td>
<td>0.2500</td>
<td>1217.62</td>
<td>34.89</td>
<td>25.46</td>
<td>0.0860</td>
<td>0.293</td>
<td>0.251</td>
</tr>
<tr>
<td>Go-ICP [31]</td>
<td>195.99</td>
<td>13.99</td>
<td>3.17</td>
<td>0.0011</td>
<td>0.0330</td>
<td>0.0120</td>
<td>157.07</td>
<td>12.53</td>
<td>2.94</td>
<td>0.0009</td>
<td>0.031</td>
<td>0.010</td>
</tr>
<tr>
<td>FGR [33]</td>
<td>126.29</td>
<td>11.24</td>
<td>2.83</td>
<td>0.0009</td>
<td>0.0300</td>
<td>0.0080</td>
<td>98.64</td>
<td>9.93</td>
<td>1.95</td>
<td>0.0014</td>
<td>0.038</td>
<td>0.007</td>
</tr>
<tr>
<td>PointNetLK [25]</td>
<td>280.04</td>
<td>16.74</td>
<td>7.55</td>
<td>0.0020</td>
<td>0.0450</td>
<td>0.0250</td>
<td>526.40</td>
<td>22.94</td>
<td>9.66</td>
<td>0.0037</td>
<td>0.061</td>
<td>0.033</td>
</tr>
<tr>
<td>DCP [23]</td>
<td>45.01</td>
<td>6.71</td>
<td>4.45</td>
<td>0.0007</td>
<td>0.0270</td>
<td>0.0200</td>
<td>95.43</td>
<td>9.77</td>
<td>6.95</td>
<td>0.0010</td>
<td>0.034</td>
<td>0.025</td>
</tr>
<tr>
<td>PR-Net [38]</td>
<td>10.24</td>
<td>3.12</td>
<td>1.45</td>
<td>0.0003</td>
<td>0.0160</td>
<td>0.0100</td>
<td>15.62</td>
<td>3.95</td>
<td>1.71</td>
<td>0.0003</td>
<td>0.017</td>
<td>0.011</td>
</tr>
<tr>
<td>R-PointHop®</td>
<td>3.58</td>
<td>1.89</td>
<td>0.11</td>
<td>0.0002</td>
<td>0.0150</td>
<td>0.0008</td>
<td>3.75</td>
<td>1.94</td>
<td>0.12</td>
<td>0.0002</td>
<td>0.0151</td>
<td>0.0008</td>
</tr>
<tr>
<td>R-PointHop</td>
<td>2.75</td>
<td>1.66</td>
<td>0.09</td>
<td>0.0002</td>
<td>0.0149</td>
<td>0.0008</td>
<td>2.53</td>
<td>1.59</td>
<td>0.08</td>
<td>0.0002</td>
<td>0.0148</td>
<td>0.0008</td>
</tr>
</tbody>
</table>
**TEST ON REAL WORLD DATA**

**STANFORD BUNNY**

- Real scans of bunny – 10 scans, ~50k points in each point cloud
- The model trained on ModelNet40 is reused
- Surprisingly, DCP does not do so well for this dataset

### TABLE V

**Registration on the Stanford Bunny Dataset**

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE (R)</th>
<th>RMSE (R)</th>
<th>MAE (R)</th>
<th>MSE (t)</th>
<th>RMSE (t)</th>
<th>MAE (t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICP [8]</td>
<td>177.35</td>
<td>13.32</td>
<td>10.72</td>
<td>0.0024</td>
<td>0.0492</td>
<td>0.0242</td>
</tr>
<tr>
<td>Go-ICP [31]</td>
<td>166.85</td>
<td>12.92</td>
<td>4.52</td>
<td>0.0018</td>
<td>0.0429</td>
<td>0.0282</td>
</tr>
<tr>
<td>FGR [33]</td>
<td>3.98</td>
<td>1.99</td>
<td>1.49</td>
<td>0.0397</td>
<td>0.1993</td>
<td>0.1658</td>
</tr>
<tr>
<td>DCP [23]</td>
<td>41.45</td>
<td>6.44</td>
<td>4.78</td>
<td>0.0016</td>
<td>0.0406</td>
<td>0.0374</td>
</tr>
<tr>
<td>R-PointHop</td>
<td>2.21</td>
<td>1.49</td>
<td>1.09</td>
<td>0.0013</td>
<td>0.0361</td>
<td>0.0269</td>
</tr>
</tbody>
</table>

TEST ON REAL WORLD DATA
STANFORD 3D SCANNING REPOSITORY

• Registration on some more scans – drill, armadillo, Buddha, dragon
LOCAL VS GLOBAL REGISTRATION

- ICP and its variants work well only in presence of a good initial alignment (local in nature)
- Rotation invariant features make R-PointHop useful for global registration
- Can be used as an initialization for ICP
TOWARDS GREEN LEARNING

• Deep learning has reformulated registration as a **supervised learning problem** – large number of pairs of point clouds used along with ground truth transformations
• Large model size, longer training times, expensive GPUs – **large carbon footprint**
• Classical model-free methods are favorable in this respect, but poor performance

<table>
<thead>
<tr>
<th>Method</th>
<th>Training time</th>
<th>Model size</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointNetLK</td>
<td>40 minutes / epoch (200 epochs)</td>
<td>630kB</td>
<td>1 GPU</td>
</tr>
<tr>
<td>Deep Closest Point (DCP)</td>
<td>27 hours</td>
<td>21MB</td>
<td>8 GPUs</td>
</tr>
<tr>
<td>R-PointHop</td>
<td>40 minutes</td>
<td>200kB</td>
<td>CPU + multithreading</td>
</tr>
</tbody>
</table>

**Green learning – low cost and high performance!**
Conclusion
## Similarities of GL and DL

<table>
<thead>
<tr>
<th></th>
<th>GL</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Information collection</strong></td>
<td>Successively growing neighborhoods</td>
<td>Gradually enlarged receptive fields</td>
</tr>
<tr>
<td><strong>Information processing</strong></td>
<td>Trade spatial dimension for spectral dimension</td>
<td>Trade spatial dimension for spectral dimension</td>
</tr>
<tr>
<td><strong>Spatial information reduction</strong></td>
<td>Spatial pooling</td>
<td>Spatial pooling</td>
</tr>
</tbody>
</table>
### Differences between GL and DL

<table>
<thead>
<tr>
<th></th>
<th>GL</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model expandability</td>
<td>Non-parametric model</td>
<td>Parametric model</td>
</tr>
<tr>
<td>Incremental learning</td>
<td>Easy</td>
<td>Difficult</td>
</tr>
<tr>
<td>Model architecture</td>
<td>Flexible</td>
<td>Networks</td>
</tr>
<tr>
<td>Model interpretability</td>
<td>Easy</td>
<td>Difficult</td>
</tr>
<tr>
<td>Model parameter search</td>
<td>Feedforward design</td>
<td>Backpropagation</td>
</tr>
<tr>
<td>Training/testing complexity</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Spectral dim. reduction</td>
<td>Subspace approximation</td>
<td>Number of filters</td>
</tr>
<tr>
<td>Task-independent features</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Multi-tasking</td>
<td>Easy</td>
<td>Difficult</td>
</tr>
<tr>
<td>Incorporation of priors and constraints</td>
<td>Easy</td>
<td>Difficult</td>
</tr>
<tr>
<td>Weak supervision</td>
<td>Easy</td>
<td>Difficult</td>
</tr>
<tr>
<td>Adversarial Attacks</td>
<td>Difficult</td>
<td>Easy</td>
</tr>
</tbody>
</table>