Deep Data and Big Learning: More quality data for better knowledge

Francisco Herrera

Data Science and Computational Intelligence
Andalusian Research Institute
Dept. of Computer Science and A.I.
University of Granada, Spain

Email: herrera@decsai.ugr.es
http://sci2s.ugr.es
Deep Data and Big Learning: More quality data for better knowledge

Outline

1. Deep Data: Towards quality data
2. Big Learning: CNNs with quality data
3. Case of study 1: MNIST
4. Case of study 2: Whale detection
5. Case of study 3: Knife detection
6. Concluding Remarks: More quality data for better knowledge
Deep Data and Big Learning: More quality data for better knowledge

Outline

1. **Deep Data:** Towards quality data
2. **Big Learning:** CNNs with quality data
3. Case of study 1: **MNIST**
4. Case of study 2: **Whale detection**
5. Case of study 3: **Knife detection**
6. Concluding Remarks: More quality data for better knowledge
Towards quality data

Quality decisions ("quality models/patterns/rules") are based on Quality Data!
Towards quality data

Quality decisions ("quality patterns/rules") are based on Quality Data!

More quality data for better knowledge
Towards quality data

Big data preprocessing is the key to transform raw big data into quality and smart data.

Transforming big data into Smart data: An insight on the use of k-Nearest Neighbours algorithm to obtain quality data
I. Triguero, J. Maillo, D. Garcia, S. Garcia, F. Herrera
Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2019. Open access
Towards quality data

Big Data

Preprocessing

Knowledge Extraction

Unreliable
Not scalable
Slow

Smart Data: Focusing on value in Big Data

Bridge between Big Data and Smart Data:
Big Data Preprocessing

More quality data for better knowledge
Towards quality data
Towards quality data

Big Data

Volume
Variety
Velocity

Quality Data

Smart Data
Big Data Preprocessing

The necessary binomial in big data beyond technology and approaching data

Big Data

Technology
Cloud/clusters
Hadoop Ecosystem:
HDFS, Spark, Flink, ...

Algorithms (Scalable)

Machine Learning and Artificial Intelligence
Scalable, efficient and effective algorithms
MLlib and other software libraries
Exact vs approximate Algorithms
Towards quality data
Towards quality data

More quality data for better knowledge

Smart Data

Knowledge
Towards quality data

Data Preprocessing → Smart Data

Applications

Smart Data + Machine Learning (AI) → Knowledge

More quality data for better knowledge
Outline

1. Deep Data: Towards quality data

2. Big Learning: CNNs with quality data

3. Case of study 1: MNIST

4. Case of study 2: Whale detection

5. Case of study 3: Knife detection

6. Concluding Remarks: More quality data for better knowledge
CNNs and quality data

Artificial Neural Networks

Learn and predict on data
Convolutional Neural Networks

Ada Lovelace?
A CNN automatically learns the values of its filters based on the task you want to perform.
By the way, what is image classification?

**Classification**

- CAT
- Single object

**Object Detection**

- CAT, DOG, DUCK

**Instance Segmentation**

- CAT, DOG, DUCK
- Multiple objects
Limitations: CNNs with quality data

CNNs require large amount of data to get better accuracies

Practical solutions: Smart data (deep data) as quality artificial data and quality original data together with Transfer learning

Quality artificial data: Data preprocessing as data augmentation

Data augmentation replicates the instances of the training set by introducing various types of transformations, e.g., translation, rotation, several types of symmetries, etc. Such techniques decrease the sensitivity of the training to noise and overfitting.

Deep Data: Smart data, Quality Data
(original and artificial data for Deep Learning)
Limitations: CNNs with quality data

**Data augmentation** replicates the instances of the training set by introducing various types of transformations, e.g., translation, rotation, several types of symmetries, etc. Such techniques decrease the sensitivity of the training to noise and overfitting.

Data preprocessing is very important to create quality artificial data.
Limitations: CNNs with quality data

CNNs require large amount of data to get better accuracies

Practical solutions: Smart data (deep data) as quality artificial data and quality original data together with Transfer learning

Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task, applying a fine tuning process in deep learning.

Fine tuning is a process to take a network model that has already been trained for a given task, and make it perform a second similar task.

Big Learning: Deep Learning + Transfer Learning
Limitations: CNNs with quality data

Deep Data: Smart data, Quality Data
(original and artificial quality data for Deep Learning)

Big Learning: Deep Learning + Transfer Learning

Fundamental Idea
Deep Data and Big Learning:
More quality data for better knowledge
Deep Data and Big Learning: More quality data for better knowledge

Outline

1. Deep Data: Towards quality data
2. Big Learning: CNNs with quality data
3. Case of study 1: MNIST
4. Case of study 2: Whale detection
5. Case of study 3: Knife detection
6. Concluding Remarks: More quality data for better knowledge
Case of study: MNIST

Handwriting recognition (60,000 training, 10,000 test)
Assign a digit from 0 to 9.
Case study: Data-augmentation for CNNs using MNIST

- **Objective:** Analyze the benefit of data-augmentation and ensembles on CNNs

- **Methodology:**
  - MNIST (60,000 train + 10,000 test) & 10 classes
  - Three CNNs: LeNet, Network3, Dropconnect

- **Results**

---

Data augmentation

- Increase the training dataset volume artificially using transformations (tackling the large amount of data limitations)

- Objective: Improve model robustness

Original Scaling Centering translation rotation .... etc
# Case study: Data-augmentation for CNNs using MNIST

Lenet-5 like CNNs: LeNet, Network3, DropConnect

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Combination</th>
<th># of training instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Original</td>
<td>60,000</td>
</tr>
<tr>
<td>2</td>
<td>Centering</td>
<td>60,000</td>
</tr>
<tr>
<td>3</td>
<td>Elastic</td>
<td>300,000</td>
</tr>
<tr>
<td>4</td>
<td>Translation</td>
<td>300,000</td>
</tr>
<tr>
<td>5</td>
<td>Rotation</td>
<td>300,000</td>
</tr>
<tr>
<td>6</td>
<td>Elastic-centering</td>
<td>300,000</td>
</tr>
<tr>
<td>7</td>
<td>Rotation-centering</td>
<td>300,000</td>
</tr>
<tr>
<td>8</td>
<td>Translation-elastic</td>
<td>1,500,000</td>
</tr>
<tr>
<td>9</td>
<td>Translation-rotation</td>
<td>1,500,000</td>
</tr>
<tr>
<td>10</td>
<td>Rotation-elastic</td>
<td>1,500,000</td>
</tr>
<tr>
<td>11</td>
<td>Rotation-elastic-centering</td>
<td>1,500,000</td>
</tr>
<tr>
<td>12</td>
<td>Elastic-elastic</td>
<td>1,500,000</td>
</tr>
</tbody>
</table>

Data-augmentation techniques
## Case study: Data-augmentation for CNNs using MNIST

### Test-set accuracies

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Average</th>
<th>Best</th>
<th>Epochs</th>
<th>Time(s)</th>
<th>Average</th>
<th>Best</th>
<th>Epochs</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>99.08%</td>
<td>99.18%</td>
<td>10.67</td>
<td>267.91</td>
<td>99.05%</td>
<td>99.21%</td>
<td>213.33</td>
<td>1070.29</td>
</tr>
<tr>
<td>Centered</td>
<td>98.85%</td>
<td>99.06%</td>
<td>10.67</td>
<td>203.52</td>
<td>98.95%</td>
<td>98.09%</td>
<td>213.33</td>
<td>926.38</td>
</tr>
<tr>
<td>Elastic</td>
<td>99.09%</td>
<td>99.19%</td>
<td>2.13</td>
<td>232.75</td>
<td>99.36%</td>
<td>99.44%</td>
<td>42.67</td>
<td>1065.38</td>
</tr>
<tr>
<td>Translation</td>
<td>99.09%</td>
<td>99.32%</td>
<td>2.13</td>
<td>268.75</td>
<td>99.30%</td>
<td>99.41%</td>
<td>42.67</td>
<td>1065.38</td>
</tr>
<tr>
<td>Rotation</td>
<td>99.05%</td>
<td>99.10%</td>
<td>2.13</td>
<td>268.03</td>
<td>99.25%</td>
<td>99.37%</td>
<td>42.67</td>
<td>1065.38</td>
</tr>
<tr>
<td>Elastic-centered</td>
<td>99.17%</td>
<td>99.26%</td>
<td>2.13</td>
<td>267.20</td>
<td>99.27%</td>
<td>99.36%</td>
<td>42.67</td>
<td>925.51</td>
</tr>
<tr>
<td>Rotation-centered</td>
<td>98.90%</td>
<td>99.07%</td>
<td>2.13</td>
<td>232.73</td>
<td>99.19%</td>
<td>99.33%</td>
<td>42.67</td>
<td>950.38</td>
</tr>
<tr>
<td>Translation-elastic</td>
<td>99.18%</td>
<td>99.32%</td>
<td>0.43</td>
<td>267.43</td>
<td>99.39%</td>
<td>99.54%</td>
<td>8.53</td>
<td>1050.38</td>
</tr>
<tr>
<td>Translation-rotation</td>
<td>99.16%</td>
<td>99.40%</td>
<td>0.43</td>
<td>267.41</td>
<td>99.40%</td>
<td>97.55%</td>
<td>8.53</td>
<td>1045.38</td>
</tr>
<tr>
<td>Rotation-elastic</td>
<td>100.00%</td>
<td>99.39%</td>
<td>0.43</td>
<td>268.14</td>
<td><strong>99.47 %</strong></td>
<td><strong>99.57 %</strong></td>
<td>8.53</td>
<td>1046.25</td>
</tr>
<tr>
<td>Rotation-elastic-centered</td>
<td>99.19%</td>
<td>99.24%</td>
<td>0.43</td>
<td>232.30</td>
<td>99.43%</td>
<td>99.52%</td>
<td>8.53</td>
<td>925.68</td>
</tr>
<tr>
<td>Elastic-elastic</td>
<td>99.27%</td>
<td><strong>99.45 %</strong></td>
<td>0.43</td>
<td>268.10</td>
<td>99.40%</td>
<td>99.50%</td>
<td>8.53</td>
<td>1047.64</td>
</tr>
</tbody>
</table>
### Case study: Data-augmentation for CNNs using MNIST

#### Test-set accuracies

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Network3(10 epochs)</th>
<th>Network3(20 epochs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Best</td>
</tr>
<tr>
<td>Original</td>
<td>99.01%</td>
<td>99.07%</td>
</tr>
<tr>
<td>Centered</td>
<td>98.73%</td>
<td>98.80%</td>
</tr>
<tr>
<td>Elastic</td>
<td>99.49%</td>
<td>99.54%</td>
</tr>
<tr>
<td>Translation</td>
<td>599.49%</td>
<td>99.55%</td>
</tr>
<tr>
<td>Rotation</td>
<td>99.44%</td>
<td>99.50%</td>
</tr>
<tr>
<td>Elastic-centered</td>
<td>99.32%</td>
<td>99.39%</td>
</tr>
<tr>
<td>Rotation-centered</td>
<td>98.88%</td>
<td>98.94%</td>
</tr>
<tr>
<td>Translation-elastic</td>
<td>499.54%</td>
<td>99.57%</td>
</tr>
<tr>
<td>Translation-rotation</td>
<td>399.57%</td>
<td>99.61%</td>
</tr>
<tr>
<td>Rotation-elastic</td>
<td>299.62%</td>
<td>99.67%</td>
</tr>
<tr>
<td>Rotation-elastic-centered</td>
<td>99.43%</td>
<td>99.51%</td>
</tr>
<tr>
<td>Elastic-elastic</td>
<td>199.65%</td>
<td>99.66%</td>
</tr>
</tbody>
</table>
Case study: Data-augmentation for CNNs using MNIST

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DropConnet (100 epochs)</th>
<th></th>
<th></th>
<th>DropConnet (200 epochs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Best</td>
<td>Time(s)</td>
<td>Average</td>
</tr>
<tr>
<td>Original</td>
<td>98.32%</td>
<td>98.83%</td>
<td>7803.43</td>
<td>98.98%</td>
</tr>
<tr>
<td>Centered</td>
<td>95.35%</td>
<td>94.46%</td>
<td>6659.31</td>
<td>95.13%</td>
</tr>
<tr>
<td>Elastic</td>
<td>99.33%</td>
<td>99.35%</td>
<td>7512.25</td>
<td>99.36%</td>
</tr>
<tr>
<td>Translation</td>
<td>5 99.43%</td>
<td>99.46%</td>
<td>7736.41</td>
<td>5 99.47%</td>
</tr>
<tr>
<td>Rotation</td>
<td>99.18%</td>
<td>99.29%</td>
<td>7151.73</td>
<td>99.37%</td>
</tr>
<tr>
<td>Elastic-centered</td>
<td>96.58%</td>
<td>96.69%</td>
<td>6969.89</td>
<td>97.08%</td>
</tr>
<tr>
<td>Rotation-centered</td>
<td>98.30%</td>
<td>98.41%</td>
<td>6974.23</td>
<td>98.55%</td>
</tr>
<tr>
<td>Translation-elastic</td>
<td>99.40%</td>
<td>99.57%</td>
<td>7162.37</td>
<td>3 99.58%</td>
</tr>
<tr>
<td>Translation-rotation</td>
<td>2 99.57%</td>
<td>99.59%</td>
<td>7410.32</td>
<td>1 99.69%</td>
</tr>
<tr>
<td>Rotation-elastic</td>
<td>3 99.54%</td>
<td>99.60%</td>
<td>7397.40</td>
<td>4 99.56%</td>
</tr>
<tr>
<td>Rotation-elastic-centered</td>
<td>4 99.47%</td>
<td>99.49%</td>
<td>7803.73</td>
<td>99.44%</td>
</tr>
<tr>
<td>Elastic-elastic</td>
<td>1 99.58%</td>
<td>99.59%</td>
<td>7911.30</td>
<td>2 99.59%</td>
</tr>
</tbody>
</table>
Case study: Data-augmentation for CNNs using MNIST

Results

<table>
<thead>
<tr>
<th></th>
<th>LeNet (500 neurons)</th>
<th>Network3</th>
<th>DropConnect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10,000 iter</td>
<td>50,000 iter</td>
<td>10 epochs</td>
</tr>
<tr>
<td>Ensemble-5</td>
<td>99.55%</td>
<td>99.57%</td>
<td>99.72%</td>
</tr>
<tr>
<td>Ensemble-3</td>
<td>99.43%</td>
<td>99.54%</td>
<td>99.69%</td>
</tr>
</tbody>
</table>

Error: 0.28% versus state of the art ensemble 0.16%
Case study: Data-augmentation for CNNs using MNIST

Results

The 28 misclassified characters (15 different)

ensemble-5 (Network3)  
ensemble-5 (DropConnect)
The 13 handwritten digits misclassified by ensemble-5 of DropConnet and Network3

Case study: Data-augmentation for CNNs using MNIST
Case study: Data-augmentation for CNNs using MNIST

Deep Learning: MNIST data (10,000 test) 
Ensemble with different top CNN models

The digit between () represents the correct class.

May 2019, Granada team (12 errors). World RECORD
1. Deep Data: Towards quality data
2. Big Learning: CNNs with quality data
3. Case of study 1: MNIST
4. Case of study 2: Whale detection
5. Case of study 3: Knife detection
6. Concluding Remarks: More quality data for better knowledge
<table>
<thead>
<tr>
<th>Problem</th>
<th>Database</th>
<th>Model</th>
<th>Results</th>
</tr>
</thead>
</table>

- **Training dataset**: 700 (with 976 whales) *aerial* images extracted from: Google Earth, free Arkive, NOAA Photo Library y NWPU-RESISC45 dataset.

- **Test dataset**: *Satellite* images of ten whale watching hotspots
<table>
<thead>
<tr>
<th>Problem</th>
<th>Database</th>
<th>Model</th>
<th>Results</th>
</tr>
</thead>
</table>

The first step detects whale presence with an F1_score of 84%

- **Trimming**
- **Scaling up and down** -> the key for generalizing from higher to lower resolution
- **Rotations and translations**
- **Different illumination conditions**

The second step counts whales with F1_score 97%
<table>
<thead>
<tr>
<th>Problem</th>
<th>Database</th>
<th>Model</th>
<th>Results</th>
</tr>
</thead>
</table>

Impact of whale postures and movements on the performance of the model
Deep Data and Big Learning:
More quality data for better knowledge

Outline

1. **Deep Data**: Towards quality data
2. **Big Learning**: CNNs with quality data
3. Case of study 1: **MNIST**
4. Case of study 2: **Whale detection**
5. Case of study 3: **Knife detection**
6. Concluding Remarks: **More quality data for better knowledge**
Project: Weapon Detection Alarm in Video Surveillance

The future of smart security

One of the ways to reduce the threat of violence generated by weapons is the early detection of their presence with enough time for agents or watchmen to act.

A novel solution could integrate an automatic weapon detection system with video surveillance system.
No published work, patent, or commercial product addresses the problem of gun detection in real-time video using Deep Learning.

Our publication (February 2017) was the first work that use Deep Learning to detect weapons in video surveillance.

https://arxiv.org/abs/1702.05147

Roberto Olmos\textsuperscript{1}, Siham Tabik\textsuperscript{1}, and Francisco Herrera\textsuperscript{1,2}

A roundup of the most interesting papers from the arXiv: Automatic Handgun Detection Alarm in Videos Using Deep Learning

MIT Technology Review
The Best of the Physics arXiv (week ending March 4, 2017)

This week's most thought-provoking papers from the Physics arXiv.

by Emerging Technology from the arXiv  March 4, 2017

R Olmos, S Tabik, F Herrera
Automatic handgun detection alarm in videos using deep learning
Neurocomputing 275, 67-72, 2018
Case of study: Knife detection

**Project: Weapon Detection Alarm in Video Surveillance**

- **Objective:** Develop a fast and accurate arms detection model in videos
- **Methodology:** To create a database (knife / no knife) + To develop a Deep learning model
Case of study: Knife detection

VGG16  ❌ Fine-tuning

CNN learns from scratch fitting weights through Backpropagation
Test: 178 knives and 138 not knives

<table>
<thead>
<tr>
<th></th>
<th>Knife</th>
<th>no knife</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knife 178</td>
<td>147</td>
<td>31</td>
</tr>
<tr>
<td>No knife 138</td>
<td>37</td>
<td>101</td>
</tr>
</tbody>
</table>

Accuracy = 0.799
Recall = 0.826
F1 score = 0.812

FP

<table>
<thead>
<tr>
<th></th>
<th>Knife</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Knife</td>
<td>49.6%</td>
<td>69.8%</td>
<td>73.7%</td>
<td>50.1%</td>
</tr>
</tbody>
</table>
Case of study: Knife detection

VGG16  Previous FP  ✔  Fine-tuning

• Fine-tuning improve classification because of pre-training

Class | Probability
--- | ---
No-Knife | 99.99%
Knife | 0%

Class | Probability
--- | ---
No-knife | 98.77%
Knife | 0%

Class | Probability
--- | ---
No-knife | 99.99%
Knife | 0%

Big Learning
Case of study: Knife detection

**VGG16** ✓ Fine-tuning

- CNN knows to extract key features and learns to classify
- Backpropagation fit fully connected layers only
- Test: 178 knifes and 138 not knifes

<table>
<thead>
<tr>
<th></th>
<th>Knife</th>
<th>no knife</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Knife 178</strong></td>
<td>168</td>
<td>10</td>
</tr>
<tr>
<td><strong>No knife 142</strong></td>
<td>7</td>
<td>135</td>
</tr>
</tbody>
</table>

**Accuracy** = 0.96 (± 0.16)

**Recall** = 0.944

**F1 score** = 0.952
Case of study: Knife detection

**Challenge:** Brightness conditions may deteriorate image quality
Case of study: Knife detection

**Challenge:** Brightness conditions may deteriorate image quality

*Fig. 7. An example of the detection results in two similar situations with different brightness conditions.*
Case of study: Knife detection

**Challenge:** Brightness conditions may deteriorate image quality

<table>
<thead>
<tr>
<th>Table 6</th>
<th>Detection performance obtained on videos recorded in different brightness conditions.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Brightness</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>Large</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>Small</td>
</tr>
<tr>
<td></td>
<td><strong>Average</strong></td>
</tr>
<tr>
<td>Medium</td>
<td>Large</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>Small</td>
</tr>
<tr>
<td></td>
<td><strong>Average</strong></td>
</tr>
<tr>
<td>Low</td>
<td>Large</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>Small</td>
</tr>
<tr>
<td></td>
<td><strong>Average</strong></td>
</tr>
<tr>
<td>Artificial</td>
<td>Large</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>Small</td>
</tr>
<tr>
<td></td>
<td><strong>Average</strong></td>
</tr>
</tbody>
</table>
Case of study: Knife detection

Movement also generates noise and distortion
Case of study: Knife detection

**DaCOLT:** Darkening and Contrast at Learning and Test stages

Enhance robustness through data-augmentation and Preprocessing

Tackling brightness via darkening and contrast.

SSD(InceptionV2)
R-FCN(ResNet101)
Faster R-CNN (Inception-ResNet-V2, ResNet50, ResNet101, and Inception)

---

Castillo, S Tabik, F Pérez, R Olmos, F Herrera

Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning

Neurocomputing 330, 151-161, 2019

*Fig. 8. An illustration of DaCoLT procedure applied at both, learning and test time.*
Case of study: Knife detection

**Darkening and Contrast at Learning and Test stages**

**DaCoLT** procedure consists of two stages:

- Training the detection model on a selected range of brightness conditions using data-augmentation
- Achieving the ideal brightness condition by adjusting the darkening of the frames and improving their visual quality using a preprocessing approach (CLAHE) before analyzing them with the detection model.

![Diagram of DaCoLT procedure](image)

*Fig. 8. An illustration of DaCoLT procedure applied at both, learning and test time.*
Case of study: Knife detection
DaCoLT image sample

Original  | Preprocessing  | Preprocessing + data aug.
No detection | Detection: 71% | Detection: 99%
Case of study: Knife detection

Detection in original condition

- The knife Surface reflects and make difficult the detection

- Sometimes the knife even dissapear
Case of study: Knife detection
Detection applying DaCoT

- Preprocessing applied at inference stage
- Improve the reflectance in brighter areas
- Slight True Positives rise
Case of study: Knife detection

Detection applying DaCoLT

- The preprocessing technique is applied for data-augmentation

- High True Positives rate rises
Case of study: Knife detection

DaCoLT results

High brightness conditions with different knife sizes

<table>
<thead>
<tr>
<th>Knife size</th>
<th>#frames</th>
<th>#GT_P</th>
<th>#TP</th>
<th>#FP</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td>121</td>
<td>112</td>
<td>78</td>
<td>0</td>
<td>100%</td>
<td>69.64%</td>
<td>82.11%</td>
</tr>
<tr>
<td>High Brightness</td>
<td>Medium</td>
<td>107</td>
<td>90</td>
<td>44</td>
<td>0</td>
<td>100%</td>
<td>48.89%</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>137</td>
<td>103</td>
<td>53</td>
<td>0</td>
<td>100%</td>
<td>51.46%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100% Average</td>
<td>56.66%</td>
<td>71.91%</td>
</tr>
<tr>
<td>Guided brightness</td>
<td>Large</td>
<td>121</td>
<td>112</td>
<td>85</td>
<td>0</td>
<td>100%</td>
<td>75.89%</td>
</tr>
<tr>
<td>DaCoLT (Test time)</td>
<td>Medium</td>
<td>107</td>
<td>90</td>
<td>56</td>
<td>0</td>
<td>100%</td>
<td>62.22%</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>137</td>
<td>103</td>
<td>53</td>
<td>0</td>
<td>100%</td>
<td>51.46%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100% Average</td>
<td>63.19%</td>
<td>76.98%</td>
</tr>
<tr>
<td>Guided brightness</td>
<td>Large</td>
<td>121</td>
<td>112</td>
<td>84</td>
<td>0</td>
<td>100%</td>
<td>75%</td>
</tr>
<tr>
<td>DaCoLT (Learning+Test)</td>
<td>Medium</td>
<td>107</td>
<td>90</td>
<td>64</td>
<td>0</td>
<td>100%</td>
<td>71.11%</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>137</td>
<td>103</td>
<td>74</td>
<td>0</td>
<td>100%</td>
<td>71.84%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100% Average</td>
<td>72.65%</td>
<td>84.15%</td>
</tr>
</tbody>
</table>
Case of study: Knife detection

Simulation applying DaCoLT: **Brightness guided preprocessing for knife detection**
Deep Data and Big Learning:
More quality data for better knowledge

Outline

1. Deep Data: Towards quality data
2. Big Learning: CNNs with quality data
3. Case of study 1: MNIST
4. Case of study 2: Whale detection
5. Case of study 3: Knife detection
6. Concluding Remarks: More quality data for better knowledge
Concluding Remarks

In contrast to the classical classification models, the high abstraction capacity of CNNs allows them to work on the original high dimensional space, which reduces the need for manually preparing the input.

However, a suitable preprocessing is still important to improve the quality of the result (including image preprocessing, data augmentation, …)

The supervised Deep Learning depends a lot on that phase of human annotation/labeling/selection.
Concluding Remarks

• Quality data preprocessing techniques adapt the data to fulfill the input demands of each data algorithm.

• Quality data preprocessing is an essential part of any automatic learning process.

• Quality data preprocessing techniques (including Data Augmentation) are very important for Deep Learning

Central idea. Deep Data and Big Learning: More quality data for better knowledge
Concluding Remarks

Limitations and reflection

Focused in image analysis, the creation of the "Smart data" level databases in the context of supervised deep learning always goes through the manual revision of the expert notebook.

There are still no automatic methods that create "Smart data" for Deep Learning without the help of the human annotation".
Limitations and reflection

Remember: There are still no automatic methods that create "Smart data" for the Deep Learning without the help of the human annotation.

There are open research studies towards quality data

Imbalanced classification: It needs preprocessing for the minority class

What is the meaning of noise data in deep learning?

Difficult instances for classification, selection and filtering

What is the correspondence with data reduction for getting quality small data for deep learning?
Concluding Remarks

Ending as we began

Quality decisions ("quality models/patterns/rules") are based on Quality Data!

More quality data for better knowledge

Quality Data to drive Deep Learning Applications
Thanks!!!

Deep Data and Big Learning:
More quality data for better knowledge