Modeling deep structures for using high performance images

Wanli Ouyang (欧阳万里)

The University of Sydney
Outline

Introduction

Effectively using high performance images

Feature fusion

Structured features

Structured samples

Pedestrian detection (CVPR’17)

Scene parsing and depth estimation (CVPR18)

3D human pose estimation (CVPR’18)

Back-bone model design

Conclusion
Outline

Introduction

Effectively using high performance images

Pedestrian detection (CVPR’17)

Scene parsing and depth estimation (CVPR’18)

3D human pose estimation (CVPR’18)

Back-bone model design

Image/video classification (CVPR’18, NIPS’18)

Conclusion
Simulate brain activities and employ **millions of neurons to fit billions of training samples**. Deep neural networks are trained with GPU clusters with **tens of thousands of processors**

Hinton won ImageNet competition
Classify 1.2 million images into 1,000 categories
Beating existing computer vision methods by 20+% 
Surpassing human performance

**Deep learning**

**REVOLUTIONARY**
Web-scale visual search, self-driving cars, surveillance, multimedia ...
Hold records on most of the computer vision problems
Performance vs practical need

Many other applications

Face recognition

Conventional model

Deep model

Very Deep model

Very deep structured learning

Face recognition applications
Structure in neurons

- Conventional neural networks
  - Neurons in the same layer have no connections
  - Neurons in adjacent layers are fully connected, at least within a local region

Structure exists in brain
Structure in data
Structure in data
Structure in data

Correlation
Outline

Introduction

Effectively using high performance images

Back-bone model design

Conclusion
Effectively using high performance imaging data

Training

Deployment

RGB only

Multi-modal data

Image from https://research.csiro.au/data61/high-performance-imaging/
Outline

Introduction

Effectively using high performance images

Feature fusion

Structured Features

Structured Samples

Pedestrian detection (CVPR’17)

Scene parsing and depth estimation (CVPR18)

3D human pose estimation (CVPR’18)
Outline

Introduction

Effectively using high performance images

Feature fusion

Pedestrian detection (CVPR’17)
Motivation

• Challenging open issues in pedestrian detection: illumination variation, shadows, background clutter, and low external light

• Exploiting thermal data in addition to RGB data for learning cross-modal representations

Hard positive samples

Hard negative samples

RGB
Motivation

• Challenging open issues in pedestrian detection: illumination variation, shadows, background clutter, and low external light

• Exploiting thermal data in addition to RGB data for learning cross-modal representations

• Can we transfer the learned cross-modal representations?
Approach

- RRN
  - RGB domain to Thermal domain
  - weakly supervised reconstruction
  - region-based instead of frame-level based

- MSDN
  - cross-modal multi-scale feature fusion
  - the parameters of subnetwork in yellow box are transferred from RRN
Results - Caltech

- Demonstrated the effectiveness of the learned cross-modal representations
- Achieved superior detection performance

<table>
<thead>
<tr>
<th></th>
<th>Average Miss rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMT-CNN-SA</td>
<td>13.76%</td>
</tr>
<tr>
<td>CMT-CNN</td>
<td>10.69%</td>
</tr>
</tbody>
</table>
Results - KAIST

CMT-CNN
- Night: 49.55%
- Day: 47.30%
- All: 54.78%

CMT-CNN-SA-SB(ImageNet)
- Night: 52.15%
- Day: 50.71%
- All: 57.65%

CMT-CNN-SA-SB(Random)
- Night: 56.76%
- Day: 54.83%
- All: 61.24%

CMT-CNN-SA
- Night: 54.26%
- Day: 52.44%
- All: 58.97%

- Demonstrated the effectiveness of the learned cross-modal representations
- Achieved superior detection performance
Qualitative results

ACF

w/o reconstruction network

With reconstruction network
Effectively using high performance images

Structured features
Motivation

Directly optimizing multiple tasks given input training data does not guarantee consistent gain on all the tasks.
Motivation

- Multi-modal input data improve training of deep networks
- Facilitate final tasks via leveraging intermediate multiple predictions while only one single modal data are required?
Illustration of the proposed multi-task distillation network for simultaneous depth estimation and scene parsing
Approach

- **Different multi-task distillation modules:**
  - Naive implementation via feature concatenation
  - Passing message between feature maps
  - Attention mechanism guided message passing module
Results for scene parsing on Cityscapes

Multi-task learning results in decrease of IOU accuracy
Distillation improves accuracy
Message passing with attention performs better
Results

- **Datasets:** NYUD-V2 and Cityscapes
- **Ablation study:**
  - (i) PAD-Net (Distillation A + DE): PAD-Net performing the DE task using the distillation module A
  - (ii) PAD-Net (Distillation B + DE): similar to (i) while using the distillation module B
  - (iii) PAD-Net (Distillation B + DE): similar to (i) while using the distillation module C
  - (iv) PAD-Net (Distillation C + DE + SP): performing DE and SP tasks simultaneously with the distillation module C
- **Effectiveness on both datasets**
- **Significant improvement over SOTA methods on joint prediction of both tasks**

Table 1. Diagnostic experiments for the depth estimation task on NYUD V2 dataset. Distillation A, B, C represents the proposed three multi-task distillation modules.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error (lower is better)</th>
<th>Accuracy (higher is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rel</td>
<td>log10</td>
</tr>
<tr>
<td>Front-end + DE (baseline)</td>
<td>0.265</td>
<td>0.120</td>
</tr>
<tr>
<td>Front-end + DE + SP (baseline)</td>
<td>0.260</td>
<td>0.117</td>
</tr>
<tr>
<td>PAD-Net (Distillation A + DE)</td>
<td>0.248</td>
<td>0.112</td>
</tr>
<tr>
<td>PAD-Net (Distillation B + DE)</td>
<td>0.230</td>
<td>0.099</td>
</tr>
<tr>
<td>PAD-Net (Distillation C + DE)</td>
<td>0.221</td>
<td>0.094</td>
</tr>
<tr>
<td>PAD-Net (Distillation C + DE + SP)</td>
<td><strong>0.214</strong></td>
<td><strong>0.091</strong></td>
</tr>
</tbody>
</table>

Table 2. Diagnostic experiments for the scene parsing task on the NYUD V2 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean IoU</th>
<th>Mean Accuracy</th>
<th>Pixel Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front-end + SP (baseline)</td>
<td>0.291</td>
<td>0.301</td>
<td>0.612</td>
</tr>
<tr>
<td>Front-end + SP + DE (baseline)</td>
<td>0.294</td>
<td>0.312</td>
<td>0.615</td>
</tr>
<tr>
<td>PAD-Net (Distillation A + SP)</td>
<td>0.308</td>
<td>0.365</td>
<td>0.628</td>
</tr>
<tr>
<td>PAD-Net (Distillation B + SP)</td>
<td>0.317</td>
<td>0.411</td>
<td>0.638</td>
</tr>
<tr>
<td>PAD-Net (Distillation C + SP)</td>
<td>0.325</td>
<td>0.432</td>
<td>0.645</td>
</tr>
<tr>
<td>PAD-Net (Distillation C + DE + SP)</td>
<td><strong>0.331</strong></td>
<td><strong>0.448</strong></td>
<td><strong>0.647</strong></td>
</tr>
</tbody>
</table>
Results

Qualitative results on NYUD-V2
Results

Qualitative results on Cityscapes
Effectively using high performance images

Feature fusion

Structured features

Structured Samples

+ + + + + +
- - - - - -
+ + + + + +
+ + + + + +
Challenges: No Annotation

Constrained scenes

In-the-wild scenes

Domain Discrepancy

Phone

No annotation
Which one is more plausible?
Weakly Supervised Adversarial Learning

$G$
3D Human Pose Estimator

$D$
Multi-source Discriminator

Images w/o GT

3D dataset

Prediction

Ground-truth

Real
Fake
Adversarial Learning

Generator

\( \text{Loss}_G \)

Euclidean Loss

Discriminator

\( \text{Loss}_D \)

Classification Loss

Fool

Tell
Generator

2D module

Depth module

Hourglass

2D score maps

3D Poses
Discriminator
Multi-Source Discriminator

Image $I$

Geometric descriptor

Raw poses

Real or Fake samples

CNN

Fully Connected layers

Real

Fake

Real or Fake samples

CNN

Concatenation
Effectiveness of Adversarial Learning

Initialization

60k iters

120k iters

Human3.6M

MPII
Ablation Study on H36M Dataset

Mean per joint position error

MPJPE (error in mm) on H36M

G + D

<table>
<thead>
<tr>
<th>Method</th>
<th>MPJPE (error in mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image + Pose + Geo</td>
<td>59.7</td>
</tr>
<tr>
<td>Image + Geo</td>
<td>60.3</td>
</tr>
<tr>
<td>Image + Pose</td>
<td>61.3</td>
</tr>
<tr>
<td>Jointly learn 2D + depth</td>
<td>64.8</td>
</tr>
<tr>
<td>Fix 2D, finetune depth</td>
<td>65.2</td>
</tr>
<tr>
<td>Zhou et al. ICCV’17</td>
<td>64.9</td>
</tr>
</tbody>
</table>

*Zhou et al. ICCV’17
Results on Images in the Wild

baseline

Ours
Multi-view Results
Outline

Introduction

Effectively using high performance images

Feature fusion

Structured Features

Structured Samples

Pedestrian detection (CVPR’17)

Scene parsing and depth estimation (CVPR18)

3D human pose estimation (CVPR’18)

Conclusion
Does structure only exist for specific task?
Outline

Introduction

Effectively using high performance images

Back-bone model design

FishNet (NIPS 18)

Optical guided feature (CVPR18)
Low-level and high-level features
Current CNN Structures

Image Classification: **Summarize** high-level semantic information of the whole image.

Detection/Segmentation: High-level semantic meaning with high spatial resolution.

Called U-Net, Hourglass, or Conv-deconv
Architectures designed for different granularities are **DIVERGING**
Unify the advantages of networks for pixel-level, region-level, and image-level tasks
Hourglass for Classification

Features with high-level semantics and high resolution is good

Directly applying hourglass for classification?

Poor performance.

So what is the problem?

• Different tasks require different resolutions of feature
• Down sample high-level features with high resolution
Hourglass for Classification

**Normal Res-Block**

- $1 \times 1, c_{in}$
- $3 \times 3, c_{in}$
- $1 \times 1, c_{out}$

**Res-Block for down/up sampling**

- $1 \times 1, c_{out}$
- $3 \times 3, c_{in}$
- $1 \times 1, c_{out}$ (Stride = 2)

**Our design**

- $1 \times 1, c_{in}$
- $3 \times 3, c_{in}$
- $1 \times 1, c_{out}$

- Low-level features $c_{out} - c_{in}$ up/down sample $c_{out}$

- Concat

- **Different tasks require different resolutions of feature**

- **Hourglass may bring more **isolated convolutions** than ResNet**

- The $1 \times 1$ convolution layer in yellow indicates the **Isolated convolution**.
Observation and design

Our observation

1. Diverged structures for tasks requiring different resolutions.
2. Isolated Conv blocks the direct back-propagation.
3. Features with different depths are not fully explored, or mixed but not preserved.

Solution

1. Unify the advantages of networks for pixel-level, region-level, and image-level tasks.
2. Design a network that does not need isolated convolution.
3. Features from varying depths are preserved and refined from each other.

Difference between mix and preserve and refine

High level

Low level

Mixed features

Message generation

High level

Low level

Preserve and refine
FishNet: Overview

- Features in the tail part
- Features in the body part
- Features in the head part

Residual Blocks
Concat
FishNet: Preservation & Refinement

Transferring Blocks $T$ (-) 

Regular Connection $s$

Down-sampling and Refinement (DR) Blocks

Up-sampling and Refinement (UR) Blocks

Feature from varying depth refines each other here

Sum up every $k$ adjacent channels
FishNet: Performance-ImageNet

Top-1 Error

Parameters, $\times 10^6$

FLOP, $\times 10^9$

Code
https://github.com/kevin-ssy/FishNet
FishNet: Performance-ImageNet

<table>
<thead>
<tr>
<th>Parameters, $\times 10^6$</th>
<th>FishNet</th>
<th>DenseNet</th>
<th>ResNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-1 Error</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.00%</td>
<td>21.25%</td>
<td>21.55%</td>
<td>21.00%</td>
</tr>
<tr>
<td>21.25% (5.76%)</td>
<td>21.20%</td>
<td>21.69% (5.94%)</td>
<td></td>
</tr>
<tr>
<td>21.55% (5.86%)</td>
<td>21.58% (6.35%)</td>
<td>22.15% (6.12%)</td>
<td></td>
</tr>
<tr>
<td>22.00%</td>
<td>22.15% (6.12%)</td>
<td>22.30% (6.20%)</td>
<td></td>
</tr>
<tr>
<td>22.59%</td>
<td>22.58% (6.35%)</td>
<td>23.78% (7.00%)</td>
<td></td>
</tr>
</tbody>
</table>

Code
https://github.com/kevin-ssy/FishNet
FishNet: Performance on COCO Detection

- **AP**:
  - R-50: 18.00%
  - RX-50: 18.20%
  - Fish-150: 18.40%
- **AP-small**: 19.00%
- **AP-medium**: 19.20%
- **AP-large**: 20.00%

Code: [https://github.com/kevin-ssy/FishNet](https://github.com/kevin-ssy/FishNet)
FishNet: Performance on COCO Instance Segmentation

AP
- R-50
- RX-50
- Fish-150

AP-small
- R-50
- RX-50
- Fish-150

AP-medium
- R-50
- RX-50
- Fish-150

AP-large
- R-50
- RX-50
- Fish-150

Code
https://github.com/kevin-ssy/FishNet
Winning COCO 2018 Instance Segmentation Task

<table>
<thead>
<tr>
<th>Method</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
<th>APs</th>
<th>APm</th>
<th>APl</th>
<th>ARI</th>
<th>AR10</th>
<th>AR100</th>
<th>ARS</th>
<th>ARM</th>
<th>ARl</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMDet</td>
<td>0.486</td>
<td>0.730</td>
<td>0.530</td>
<td>0.339</td>
<td>0.520</td>
<td>0.602</td>
<td>0.368</td>
<td>0.593</td>
<td>0.632</td>
<td>0.464</td>
<td>0.665</td>
<td>0.777</td>
<td>2018-08-18</td>
</tr>
<tr>
<td>Megvii (Face++)</td>
<td>0.485</td>
<td>0.737</td>
<td>0.532</td>
<td>0.298</td>
<td>0.507</td>
<td>0.641</td>
<td>0.369</td>
<td>0.594</td>
<td>0.630</td>
<td>0.474</td>
<td>0.659</td>
<td>0.767</td>
<td>2018-08-18</td>
</tr>
<tr>
<td>FirstShot</td>
<td>0.463</td>
<td>0.681</td>
<td>0.508</td>
<td>0.258</td>
<td>0.483</td>
<td>0.636</td>
<td>0.359</td>
<td>0.580</td>
<td>0.622</td>
<td>0.445</td>
<td>0.655</td>
<td>0.776</td>
<td>2018-08-17</td>
</tr>
</tbody>
</table>
Visualization
Visualization
Visualization
Visualization
Codebase

• **Comprehensive**
  - ✔ RPN
  - ✔ Mask R-CNN
  - ✔ Cascade R-CNN
  - ✔ More … … …
  - ✔ Fast/Faster R-CNN
  - ✔ FPN
  - ✔ RetinaNet

• **High performance**
  - ✔ Better performance
  - ✔ Optimized memory consumption
  - ✔ Faster speed

• **Handy to develop**
  - ✔ Written with PyTorch
  - ✔ Modular design

GitHub: mmdet
FishNet: Advantages

1. Better gradient flow to shallow layers
2. High-resolution features contain rich low-level and high-level semantics
3. Feature from varying depth are preserved and refined from each other
Outline

Introduction

Effectively using high performance images

Back-bone model design

FishNet (NIPS 18)

Optical guided feature (CVPR18)
Action Recognition

- Recognize action from videos
Optical flow in Action Recognition

- Motion is the important information
- Optical flow
  - Effective
  - Time consuming

We need a better motion representation

<table>
<thead>
<tr>
<th>Modality</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>85.5%</td>
</tr>
<tr>
<td>RGB+Optical Flow</td>
<td>94.0%</td>
</tr>
</tbody>
</table>
Optical flow guided feature
Optical flow guided feature

Optical flow:

\[ \frac{\partial I(x, y, t)}{\partial x} v_x + \frac{\partial I(x, y, t)}{\partial y} v_y + \frac{\partial I(x, y, t)}{\partial t} = 0 \]

\( \{v_x, v_y\} = \text{optical flow} \)

Coefficient for optical flow:

\[ \left\{ \frac{\partial I(x, y, t)}{\partial x}, \frac{\partial I(x, y, t)}{\partial y}, \frac{\partial I(x, y, t)}{\partial t} \right\} \]
Optical flow guided feature

Feature flow:

\[ f(I(x, y, t)) = f(I(x + \Delta x, y + \Delta y, t + \Delta t)) \]

\[ \frac{\partial f(I(x, y, t))}{\partial x} \tilde{v}_x + \frac{\partial f(I(x, y, t))}{\partial y} \tilde{v}_y + \frac{\partial f(I(x, y, t))}{\partial t} = 0 \]

\( \{\tilde{v}_x, \tilde{v}_y\} \) = feature flow

Optical flow guided feature (OFF):

\[ \left\{ \frac{\partial f(I(x, y, t); w)}{\partial x}, \frac{\partial f(I(x, y, t); w)}{\partial y}, \frac{\partial f(I(x, y, t); w)}{\partial t} \right\} \]
Optical flow guided feature
Optical Flow Guided Feature (OFF): Experimental results

1. OFF with only RGB inputs is comparable with the other state-of-the-art methods using optical flow as input.
Not only for action recognition

- Also effective for
  - Video object detection
  - Video compression artifact removal

![Graph showing detection (mAP) for different models.](graph)

- **Detection (mAP)**
  - resnet+rfcn+OFF
    - 71.5
  - resnet+rfcn
    - 72

![Graph showing compression artifact removal (PSNR) for different models.](graph)

- **Compression Artifact Removal (PSNR)**
  - DnCNN+OFF
    - 34.6
  - DnCNN
    - 35.2
Outline

Introduction

Effectively using high performance images

Back-bone model design

FishNet (NIPS 18)

Optical guided feature (CVPR18)
Take home message

• Structured deep learning is
  – effective

• Effectively using high performance imaging as the privileged information by exploring the structured information at
  – Sample level
  – Feature level

• End-to-end joint training bridges the gap between structure modeling and feature learning
Joint work

Xiaogang Wang
Nicu Sebe
Elisa Ricci
Wei Yang
Dan Xu
Shuyang Sun