Real-time Monocular Depth Estimation with Extremely Light-Weight Neural Network

Paper ID: 1954

Mian-Jhong Chiu  Wei-Chen Chiu  Hua-Tsung Chen  Jen-Hui Chuang

National Chiao Tung University
Introduction

Real-time Depth Estimation

- Autonomous Driving
- Augmented Reality
- Robotics
- 3D Modeling
Introduction

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elkerdawy et al.</td>
<td>5.891</td>
<td>5.9 M</td>
</tr>
<tr>
<td>Poggi et al.</td>
<td>6.030</td>
<td>1.9 M</td>
</tr>
<tr>
<td>Nekrasov et al.</td>
<td>3.453</td>
<td>2.99 M</td>
</tr>
<tr>
<td>Ours</td>
<td>3.871</td>
<td>0.32 M</td>
</tr>
</tbody>
</table>

our depth prediction results
Pipeline Overview

Training

Input Image → Light-weight neural network → Depth estimation

Inference

Input Image → Light-weight neural network → Depth Estimation
Teacher Models (Semantic Segmentation)

- Semantic segmentation
  - Using DeepLabV3 [11] to generate semantic segmentation as training ground truth

[11] Chen et al., Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs., TPAMI 2018
Teacher Models (Depth Estimation)

- Depth Estimation
  - Use Pyramid Stereo Matching Network (PSMNet) [12] proposed by Chang et al. to generate dense disparity map

[12] Chang et al., Pyramid Stereo Matching Network. CVPR 2018
Teacher Models (Depth Estimation)

- Depth Estimation
  - Use Pyramid Stereo Matching Network (PSMNet) [12] proposed by Chang et al. to generate dense disparity map

### Table

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth value</td>
<td>5.64</td>
<td>69.67</td>
<td>0.96</td>
</tr>
<tr>
<td>Compensated Value</td>
<td>5.67</td>
<td>87.32</td>
<td>0.41</td>
</tr>
</tbody>
</table>
Network Architecture

**Encoder**
(feature extraction)

**Decoder**
(reconstruction from features)
Network Architecture

Encoder
(Feature extraction)

Decoder
(Reconstruction from features)

Introduction
Method
Experiment
Conclusion
Network Architecture

Encoder
(Feature extraction)

MobileNetV2

Decoder
(Reconstruction from features)

D1
D2
D3
D4

Decoder Block
Network Architecture

Encoder
(feature extraction)

MobileNetV2

Decoder
(reconstruction from features)

D1  D2  D3  D4

PixelShuffle Layer

Bottleneck, 16, s1

1x1 Conv

1x1 Conv

Softmax

PixelShuffle [13]

Input 2x2x4

Output 4x4x1

Network Architecture

Encoder
(feature extraction)

Decoder
(reconstruction from features)

Introduction
Method
Experiment
Conclusion
Loss Function

Depth loss

\[ L_{D_i} = \frac{1}{W_i H_i} \sum_{x=1}^{W_i} \sum_{y=1}^{H_i} F(G(\tilde{d}_{x,y}) - G(d_{x,y})) \]

\[ F(x) = \begin{cases} 
|x| & |x| \leq \alpha \\
\frac{x^2 + \alpha^2}{2\alpha} & |x| > \alpha.
\end{cases} \]

\[ \alpha = \frac{1}{5} \max_i |G(\tilde{d}_{x,y}) - G(d_{x,y})| \]

Semantic segmentation loss

\[ L_{S_i} = -\frac{1}{W_i H_i} \sum_{x=1}^{W_i} \sum_{y=1}^{H_i} \sum_{c=1}^{C} s_{x,y}^c \log \hat{s}_{x,y}^c \]

(W_i, H_i): resolution of the depth map
\( d \): predicted depth map
\( d \): ground truth depth map
\( s \): predicted semantic segmentation
\( \hat{s} \): ground truth semantic segmentation
\( C \): number of classes of semantic segmentation
Ablation Study

Evaluation of models trained with differently pre-processed training data.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Training data</th>
<th>RMSE (meter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Sparse depth map</td>
<td>4.922</td>
</tr>
<tr>
<td>Pre-processed</td>
<td>PSMNet (origin)</td>
<td>4.716</td>
</tr>
<tr>
<td></td>
<td>PSMNet (compensated)</td>
<td>3.945</td>
</tr>
</tbody>
</table>

Evaluation of models trained with differently pre-processed training data

<table>
<thead>
<tr>
<th>Improvement</th>
<th>RMSE (meter)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Depth Teacher</strong></td>
<td></td>
</tr>
<tr>
<td>Log Depth</td>
<td></td>
</tr>
<tr>
<td><strong>Segment Teacher</strong></td>
<td></td>
</tr>
<tr>
<td>[✓]</td>
<td>3.945</td>
</tr>
<tr>
<td>[✓] [✓]</td>
<td>3.884</td>
</tr>
<tr>
<td>[✓] [✓] [✓]</td>
<td>3.871</td>
</tr>
</tbody>
</table>
Performance Evaluation (Model Size)

Model Size vs. Accuracy

![Graph showing model size vs. accuracy comparison with different methods.]

- **Elkerdawy et al.**
- **Poggi et al.**
- **Nekrasov et al.**
- **Ours**
### Performance Evaluation (Computation Speed)

**Processing Speed Evaluation**

Evaluation of our model on GTX 1060 and Jetson TX2

<table>
<thead>
<tr>
<th>Model</th>
<th>Output dim.</th>
<th>1060GPU (FPS)</th>
<th>Jetson TX2 (FPS)</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Before TRT</td>
<td>After TRT</td>
<td>Before TRT</td>
</tr>
<tr>
<td>L</td>
<td>(240, 160)</td>
<td>100</td>
<td>121.6</td>
<td>21</td>
</tr>
<tr>
<td>M</td>
<td>(120, 80)</td>
<td>126.5</td>
<td>148.7</td>
<td>31.5</td>
</tr>
<tr>
<td>S</td>
<td>(60, 40)</td>
<td>148.6</td>
<td>174.8</td>
<td>36.4</td>
</tr>
<tr>
<td>XS</td>
<td>(30, 20)</td>
<td>151.2</td>
<td>179.4</td>
<td>45.4</td>
</tr>
</tbody>
</table>

TRT: TensorRT
Conclusion

• Design an efficient CNN for depth estimation with only 2.1 GFLOPs computations and 0.3M parameters.

• Propose effective training strategies for such extremely small model:
  (i) joint-training
  (ii) data generation by complex teacher model
  (iii) using a multi-resolution log depth loss

• The detachable structure enables model customization, offering the trade-off between output resolution and computation cost (speed).
Thank you for your attention

Q&A