Lecture 19
Scene understanding
19. Scene Understanding

- Semantics
  - Semantic segmentation
  - Object detection
  - Instance segmentation
- Geometry
  - 3D in the deep learning era
  - Single view depth estimation
  - Unsupervised learning of monocular depth cues
  - 3D objects
A VIEW OF A PARK ON A NICE SPRING DAY
Do not feed the ducks sign

PEOPLE WALKING IN THE PARK

PERSON FEEDING DUCKS IN THE PARK

DUCKS LOOKING FOR FOOD
DUCKS ON TOP OF THE GRASS

PEOPLE UNDER THE SHADOW OF THE TREES
A bit of history...
So, let’s make the problem simpler: Block’s world
3D, compositional models

Binford and generalized cylinders

Recognition by components


Part based models

- Object as set of parts
  - Generative representation
- Model:
  - Relative locations between parts
  - Appearance of part
- Issues:
  - How to model location
  - How to represent appearance
  - Sparse or dense (pixels or regions)
  - How to handle occlusion/clutter
Scene models

Multiple levels of representation -- pixels > patches > regions > subimages > objects.
Neural Network-Based Face Detector

Train a set of multilayer perceptrons and arbitrate a decision among all outputs

Rowley, Baluja, and Kanade: Neural Network-Based Face Detection (PAMI, January 1998)
Histograms of oriented gradients (HOG)

1. Bin gradients from 8x8 pixel neighborhoods into 9 orientations
2. Linear SVM

Families of recognition algorithms

Bag of words models

- Csurka, Dance, Fan, Willamowski, and Bray 2004
- Sivic, Russell, Freeman, Zisserman, ICCV 2005

Voting models

- Viola and Jones, ICCV 2001
- Heisele, Poggio, et. al., NIPS 01
- Schneiderman, Kanade 2004
- Vidal-Naquet, Ullman 2003

Shape matching
- Berg, Berg, Malik, 2005
- Cootes, Edwards, Taylor, 2001

Deformable models

Constellation models

- Fischler and Elschlager, 1973
- Burl, Leung, and Perona, 1995
- Weber, Welling, and Perona, 2000
- Fergus, Perona, & Zisserman, CVPR 2003

Rigid template models

- Sirovich and Kirby 1987
- Turk, Pentland, 1991
- Dalal & Triggs, 2006

Neural networks

- Le Cun et al, 98
car
ImageNet classification and Neural nets

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.
Computation in a neural net

\[
f(x) = f_L(\ldots f_2(f_1(x)))
\]
Image classification

\[ f \rightarrow \text{“Birds”} \]
Semantic segmentation

\[ f \]

(Colors represent one-hot codes)
What's the object class of the center pixel?

Training data

\[
\begin{align*}
\mathbf{x} & \quad y \\
\text{\{} & \text{\} } & \text{\{} & \text{\} } & \text{\{} & \text{\} } & \text{\{} & \text{\} } \\
\text{\{} & \text{“Bird”} \quad \text{\} } & \text{\{} & \text{“Bird”} \quad \text{\} } & \text{\{} & \text{“Sky”} \quad \text{\} } & \text{\{} & \text{“Sky”} \quad \text{\} } \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
\end{align*}
\]

K-way classification problem

Solve with K-dimensional softmax regression:

\[
f_\theta : X \rightarrow \mathbb{R}^K
\]
Fully Convolutional Networks

Abstract

Convolutional networks are powerful neural models that yield impressive results in various computer vision tasks. However, they are typically used in a feedforward manner with a fixed input size, which limits their applicability to tasks requiring dynamic input sizes. In this work, we present a novel architecture for fully convolutional networks (FCNs) that enables end-to-end training of neural networks for semantic segmentation tasks. This architecture allows the network to learn end-to-end representations for semantic tasks, overcoming the limitations of traditional convolutional neural networks.

1. Introduction

Fully convolutional networks are driving advances in computer vision. Convolutional networks are not only improving for image classification [22, 34, 19], but also making progress in visual tasks with structured outputs. These include advances in bounding box object detection [21, 34, 19], part keypoint prediction [22], and local correspondence [21, 19].

The natural next step in the progression from coarse to fine inference is to make predictions on every pixel. Prior approaches have used convnets for semantic segmentation [33, 16, 15, 34, 14, 15], in which each pixel is labeled with the class of its enclosing object or region, with shortcomings that this work addresses.

Jonathan Long† Present Address: UC Berkeley
Evan Shelhamer‡ Present Address: UC Berkeley
Trevor Darrell§

† jlong@berkeley.edu
‡ eshelhamer@berkeley.edu
§ tdarrell@berkeley.edu

Figure 1: Fully convolutional networks can efficiently learn from pixel-to-pixel correspondence in semantic segmentation.

End-to-end training on FCN-21 with supervision on pixel-level loss. Fully convolutional networks (FCNs) trained end-to-end, pixel-to-pixel or per image can serve the same role as a deep neural network. Fully convolutional networks can serve the same role as a deep neural network.
Fully Convolutional Networks

96  256  384  384  256  4096  4096  1000

Upsampling

227x227  55x55
Fully Convolutional Networks
Encoder-decoder architectures

Encoder

Decoder

Convolutions

Deconvolutions

Skip connections
Encoder-decoder architectures

RGB Image

Convolutional Encoder-Decoder

Pooling Indices

Conv + Batch Normalisation + ReLU
Pooling
Upsampling
Softmax

Output

Segmentation

SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation
Semantic segmentation

“A bunch of bird stuff”
Object detection

Challenge: unbounded number of detections, possibly multiple detections per pixel

Each bounding box is: 

\[ [x, y, w, h] \]
PASCAL Visual Object Challenge

20 Object classes: aeroplane bike bird boat bottle bus car cat chair cow table dog horse motorbike person plant sheep sofa train tv
5000 training images
5000 testing images
Searching for objects

Scanning window approach & Image pyramids

Selective search

Input image

Candidate bounding boxes
We need translation invariance
We need translation and scale invariance
Image pyramids
The Gaussian pyramid

256×256

→ 128×128

→ 64×64

→ 32×32
The Gaussian pyramid

512×512  256×256  128×128  64×64  32×32

(original image)
Image and features pyramids

Gaussian image pyramid

Feature pyramid

Image and features pyramids

Each pooling reduces the resolution by a factor of 2

ConvNet architectures build:
- Multiscale feature hierarchies, but
- each layer builds a different representation
- first layers are low level, while
- last layers are high level.

A feature pyramid requires a uniform representations across scales.
Image and features pyramids

Gaussian image pyramid  Feature pyramid

1. Introduction

Recognizing objects at vastly different scales is a fundamental challenge in computer vision. Feature pyramids, built upon image pyramids (for short, we call them feature-based image pyramids) form the basis of a standard solution [19, 8]. These pyramids are scale invariant in the sense that an object's valid change is offset by shifting its level in this pyramid. Intuitively, this property enables a model to detect objects across a large range of scales by normalizing the models over both position and pyramid levels.

Pyramidal image pyramids were pioneered in the early days of computer vision (e.g., [19, 8]). They were so effective that object detectors like HOG [19] or deformable part models (e.g., [8]) are currently state-of-the-art approaches for object detection.

Pyramid-based features have been widely adopted in various computer vision tasks, such as object detection, segmentation, and image classification. However, there are several limitations to using feature pyramids. First, they are typically limited to a small range of scales due to the high-dimensional nature of feature pyramids. Second, they are computationally expensive, especially for high-resolution images. Third, they are not well-suited for tasks that require fine-grained discrimination between objects at different scales. To address these limitations, recent approaches have explored alternatives to feature pyramids, such as those based on convolutional neural networks (CNNs) and other deep learning techniques.
Image and features pyramids
Searching for objects

Scanning window approach & Image pyramids

Selective search

Input image

Candidate bounding boxes
Selective search

Stage 1: generate candidate bounding boxes

Input Image

Segmentation

Candidate objects

Stage 2: apply classifier to each candidate bounding box

Ground truth

Positive examples

Training Examples

Object hypotheses

Difficult negatives

if overlap with positive 20-50%

Model

SVM

(Histogram Intersection Kernel)

False Positives

Search for false positives

Add to training examples

Training Examples

Retrain
R-CNN

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

- aeroplane? no.
- person? yes.
- tvmonitor? no.
R-CNN

Selective search

AlexNet, ImageNet pretrained then fine-tuned on the 20 VOC classes

Non-maximal suppression
Making the structure end-to-end

Fast R-CNN

Faster R-CNN
Object detection renaissance (2013-present)

Source: Ross Girshick
Instance segmentation

Challenge: unbounded number of output instances
(can’t just do K-way classification)
Instance segmentation
### Approaches

InstanceCut, DWT, SAIS, DIN, FCIS, SGN, Mask-RCNN, PANet etc.

**DWT [Bai et al, CVPR’17]**

**Mask-RCNN [He et al, ICCV’17]**

**PANet [Liu et al, CVPR’18]**
Mask R-CNN


Mask R-CNN

Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick

Abstract

We present Mask R-CNN, a method that jointly performs object detection and instance segmentation. It extends R-CNN with a new branch that segments the instance that an object belongs to. Mask R-CNN uses a novel mask estimation module called ROIAlign, which learns to aggregate features that are most relevant to the object and its region of interest. We also propose a new loss term that encourages the object and segmentation head to learn similar representations. This allows Mask R-CNN to outperform its competitors on COCO by a large margin. Our model is also significantly faster, requiring only 21 FPS on the COCO test-dev set. To the best of our knowledge, this is the first end-to-end model that simultaneously performs object detection and instance segmentation.
Panoptic Segmentation

Panoptic segmentation: stuff and things are solved, instances distinguishable.
Unified Panoptic Segmentation Network (UPSNet)
Depth Perception

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & a & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]
Vision systems

One camera

Two cameras

N cameras
1 eye
Shadows
Learning based models

D. Hoiem, A.A. Efros, and M. Hebert,
SIGGRAPH 2005.

A. Saxena, M. Sun, A. Y. Ng. 2007.

Make3D
Ashutosh Saxena, Sung H. Chung, Andrew Y. Ng.
NeurIPS 18, 2005.

Karsch et al.
Ladicky et al.
...
3D scene understanding in the deep net era
3D in the deep learning era

Ground truth is collected by using traditional methods
Datasets

**KITTI**

"Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite", Geiger et al., CVPR’12

"The Cityscapes Dataset for Semantic Urban Scene Understanding", Cordts et al., CVPR’16

**Cityscapes**
Datasets

Cityscapes

KITTI
3D in the deep learning era

Data \( \{x_i, y_i\}_{i=1}^N \) → Learner

Objective
scale invariant MSE in log space

Hypothesis space
Deep Neural Network

Optimizer
SGD

\[ f^* = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i) \] → \( f \) Regular old supervised learning!
3D in the deep learning era

Teacher

Student

Loss

SGD
3D in the deep learning era

Input image

Predicted depth map

[Result of Eigen et al., NIPS, 2014]
Global Coarse-scale network

Input

Coarse network

Fine network

Coarse 7

Fine 4

304x228 Pixels

74x55 Pixels
Global Coarse-scale network

Coarse network
- Global scene view
- Vanishing points,
- Room layout
- Object locations
- Sees the entire image

Fine network
- Align with local details such as object and wall edges
- Each unit sees only a local window of 45x45 pixels
Global Coarse-scale network

- Conv Layers 1-5: Pretrained with ImageNet
- Layer 6: Dropout
- Layer 7: linear output
Global Coarse-scale network

Input

304x228 Pixels

Coarse 1
11x11 conv
4 stride
2x2 pool
5x5 conv
2x2 pool

Coarse 2
3x3 conv

Coarse 3
3x3 conv

Coarse 4
3x3 conv

Coarse 5
full

Coarse 6
full

Linear

Relus

Input

11x11 conv
4 stride
2x2 pool

Fine 1
9x9 conv
2 stride
2x2 pool

Fine 2
Concatenate

Fine 3
5x5 conv

Fine 4
74x55 Pixels

Layer
input

Coarse
1
2,3,4
5
6
7
1,2,3,4

Fine

Size (NYUDepth)
304x228
37x27
18x13
8x6
1x1
74x55
74x55

Ratio to input
/1
/8
/16
/32
–
/4
/4
Scale invariant error

With uncalibrated cameras (unknown $K$), the global scale of a scene is an “ambiguity” in depth prediction.

... you could learn estimate $K$ from a single image ...
Scale invariant error

Estimate log depth instead of depth. Defining \( y_i \) the ground truth depth on pixel \( i \), and \( y_i^* \) its estimated depth:

**Standard L2 error:**

\[
D_{L2}(y, y^*) = \frac{1}{n} \sum_{i=1}^{n} (\log y_i - \log y_i^*)^2
\]

**Scale invariant error:**

\[
D_{SI}(y, y^*) = \frac{1}{n} \sum_{i=1}^{n} (\log y_i - \log y_i^* + \alpha(y, y^*))^2
\]

with \( \alpha(y, y^*) = \frac{1}{n} \sum_{j=1}^{n} (\log y_j - \log y_j^*) \)
Training:

- **Training loss**: Mixture of both error measures (best $\lambda=0.5$):

\[
J = \lambda D_{L2}(y, y^*) + (1 - \lambda) D_{SI}(y, y^*)
\]

Depth contains missing values. Only evaluate on valid pixels.

- **Data augmentation**: flips, translations, scalings, color scalings, …
Results (best)
Results (worst)
Results

![Input](image1.png)  ![Prediction](image2.png)  ![Ground-truth](image3.png)

![Input](image4.png)  ![Prediction](image5.png)  ![Ground-truth](image6.png)

![Input](image7.png)  ![Prediction](image8.png)  ![Ground-truth](image9.png)
Kinect is a stereo active system

Ground truth is collected by using traditional methods.
3D in the deep learning era

Teacher

What else can we use as teacher?

Student

Loss

SGD
Learning Single-View Depth Prediction from Internet Photos

MegaDepth: Learning Single-View Depth Prediction from Internet Photos
Zhengwei Li, Noah Snavely
Department of Computer Science & Cornell Tech, Cornell University

1. Introduction
Predicting 3D shapes from a single image is an important capability of visual learning, with applications in robotics, graphics, and other vision tasks such as intrinsic images. While single-view depth estimation is a challenging, under-determined problem, deep learning methods have recently driven significant progress. Such methods deserve a thorough treatment with large amounts of data. Unfortunately, fully annotated training data in the form of (RGB image, depth map) pairs is difficult to come by. Currently, RGB-D sensors such as Kinect have been widely used for this purpose [13], but are limited in coverage. Laser scanners have enabled important datasets such as Make3D [22] and SfMLearner [24], but such devices are cumbersome to operate (in the case of industrial scanners), or produce sparse depth maps (in the case of LiDAR). Moreover, both Make3D and KITTI are collected in specific scenarios in urban campus, and map a car trajectory. Training data is thus generated through rendering, but this approach has so far been limited to capturing sparse visual relationships or surface normals [13, 4, 4].

In this paper we explore the use of a nearly infinite source of data for this problem: images from the Internet over millions of viewpoints, from which structure-from-motion (SfM) methods can recover 3D reconstructions. We propose to learn a function that maps images to depth through a single neural network. This is a highly under-determined problem, and thus requires a large amount of training data to converge. We propose the use of diverse image collections, as well as novel data augmentation methods, to address this challenge.

As training data, we use images from the Web, which have the advantage of being both large and diverse. The Internet is a rich source of images and provides us with a large amount of data for training. Although image data is highly variable, it also contains visual information that can be used for learning. We develop techniques for automatically generating training data from the Web, which includes images from various scenes and viewpoints.

The main contribution of this paper is the demonstration that it is possible to learn a function that maps images to depth from data collected on the Internet. We show that this approach is feasible and that the results are comparable to those obtained using traditional methods. This opens up new possibilities for visual learning, and provides a new source of training data that can be used to improve existing methods.

Figure 1: We use large Internet image collections, combined with 3D reconstruction and semantic labeling methods, to generate large amounts of training data for single-view depth prediction. (a) shows example input RGB images. (c), (d), (f) are depth maps produced by our MegaDepth model on the KITTI dataset. For these results, the network was not trained on Make3D or KITTI data.

Table 4: Results on Make3D for various training datasets and methods. The first column indicates the training dataset. Errors for “Ours” are averaged over four models trained independently on Make3D. Lower is better for all metrics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>NRE</th>
<th>Abs Rel</th>
<th>SqRE</th>
<th>PSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make3D</td>
<td>ours</td>
<td>0.66</td>
<td>0.051</td>
<td>0.293</td>
<td>0.120</td>
</tr>
<tr>
<td>Make3D</td>
<td>ours</td>
<td>0.72</td>
<td>0.050</td>
<td>0.288</td>
<td>0.118</td>
</tr>
<tr>
<td>Make3D</td>
<td>ours</td>
<td>0.81</td>
<td>0.049</td>
<td>0.276</td>
<td>0.104</td>
</tr>
<tr>
<td>Make3D</td>
<td>ours</td>
<td>0.89</td>
<td>0.048</td>
<td>0.265</td>
<td>0.099</td>
</tr>
<tr>
<td>Make3D</td>
<td>others</td>
<td>0.93</td>
<td>0.047</td>
<td>0.258</td>
<td>0.095</td>
</tr>
<tr>
<td>Make3D</td>
<td>others</td>
<td>0.96</td>
<td>0.045</td>
<td>0.252</td>
<td>0.090</td>
</tr>
<tr>
<td>Make3D</td>
<td>others</td>
<td>1.00</td>
<td>0.043</td>
<td>0.246</td>
<td>0.085</td>
</tr>
<tr>
<td>Make3D</td>
<td>others</td>
<td>1.07</td>
<td>0.041</td>
<td>0.240</td>
<td>0.080</td>
</tr>
</tbody>
</table>

Based on the split of [7], as with our Make3D experiments, we do not use images from KITTI during training. The KITTI dataset is very different from ours, consisting of driving sequences that include objects, such as sidewalks, cars, and people, that are difficult to reconstruct with SfM/MVS. Nevertheless, as shown in Table 5, our Make3D-trained model outperforms approaches trained on non-KITTI datasets. Finally, the last row of Table 4 shows that our model trained on Make3D achieves better performance than the state-of-the-art depth estimation methods.

KITTI Test: We evaluate our model on the KITTI test set based on the split of [7]. As with our Make3D experiments, we do not use images from KITTI during training. The KITTI dataset is very different from ours, consisting of driving sequences that include objects, such as sidewalks, cars, and people, that are difficult to reconstruct with SfM/MVS. Nevertheless, as shown in Table 5, our Make3D-trained model outperforms approaches trained on non-KITTI datasets. Finally, the last row of Table 4 shows that our model trained on Make3D achieves better performance than the state-of-the-art depth estimation methods.

Figure 6: Depth predictions on an MD test set. (Top row, left to right) For visualization, we mask out the detected sky region. In the columns marked “GT,” we apply the mask from the GT depth map (indicated with solid yellow/orange pixels) to the prediction map, to aid comparison with GT. (c) shows the MS MARC depth map GT; VGG* prediction using the loss and network of [46, 6]; GT-encoded version of VGG*; (d) Depth prediction from an homography [15] network; (e) GT-encoded version of (e); (f) GT-encoded version of (g).

Figure 7: Depth predictions on Make3D. The last four columns show results from the best model trained on non-Make3D datasets (final column in our result).

CVPR 2018
Structure from motion

The internet can be an unlimited source of 3D data
Structure from motion

Teacher

SFM

Loss

SGD

Student
MegaDepth dataset

200 locations, ~130k images

http://www.cs.cornell.edu/projects/megadepth
Stacked hourglass architecture
MegaDepth results

Source: http://www.cs.cornell.edu/projects/megadepth/
MegaDepth results

Source: http://www.cs.cornell.edu/projects/megadepth/
How else can we collect depth?
“Unsupervised” camera motion and monocular depth

Unsupervised Learning of Depth and Ego-Motion from Video
Tinghui Zhou*UC Berkeley
Matthew Brown
Google
David G. Lowe
Google

Abstract
We present an unsupervised learning framework for the task of monocular depth and camera motion estimation from unstructured video sequences. In common with recent work [1, 2, 5, 7, 6], we use an end-to-end learning approach with view synthesis as the supervisory signal. In contrast to the previous work, our method is completely unsupervised, requiring only monocular video sequences for training. We model the task as single-view depth and multiscale pose recovery, with a loss based on warping nearby views to the target using the computed depth and pose. The networks are then coupled by the loss during training, but can be applied independently at test time. Empirical evaluation on the KITTI dataset demonstrates the effectiveness of our approach: 1) monocular depth prediction performs comparably to supervised methods that see either ground-truth poses or depth for training, and 2) pose estimation performance is comparable to established EKF-based systems under comparable input settings.

1. Introduction
Humans are innately capable of inferring ego-motion and the 3D structure of a scene even over short timescales. For instance, in navigating along a street, we can easily locate obstacles and react quickly to avoid them. Years of research in geometric computer vision has failed to replicate similar modeling capabilities for real-world scenes (e.g., when non-rigidity, occlusion and lack of scene are present). So why do humans excel at this task? One hypothesis is that we develop a rich, internal understanding of the world through our past visual experience that has largely consisted of moving around and observing vast numbers of scenes and developing consistent modeling of our observations. From millions of such observations, we have learned about the regularity of the world—such as flat surfaces are straight, cars are supported by roads, etc., and we can apply this knowledge when processing new, even from a single monocular image.

The majority of the work was done while interning at Google.

Fig. 4. Network architecture for the off-line/positional-field prediction module. The width of each edge in the graph indicates the relative number of edge operations during the entire process at each time step. The number of edge operations is normalized to 100 for clarity.

Fig. 5. Qualitative results for the depth estimation module. (a) Target frame (first row) and corresponding depth prediction (second row). (b) Ground truth depth map (first row) and estimated depth map (second row). The color bar represents the range of depth, from light yellow to dark purple, indicating closer to the camera. (c) Corresponding ground truth egomotion graph (first row) and estimated egomotion graph (second row).

Fig. 6. Qualitative results for the view synthesis module. (a) Target frame (first row) and corresponding synthesized frame (second row). (b) Ground truth synthesized view (first row) and estimated synthesized view (second row). The color bar represents the range of depth, from light yellow to dark purple, indicating closer to the camera.

Fig. 7. Qualitative results for the pose recovery module. (a) Target frame (first row) and corresponding estimated pose (second row). (b) Ground truth ego-state (first row) and estimated ego-state (second row). The color bar represents the range of depth, from light yellow to dark purple, indicating closer to the camera.

Fig. 8. Example results for the egomotion graph. (a) Target frame (first row) and corresponding egomotion graph (second row). (b) Ground truth egomotion graph (first row) and estimated egomotion graph (second row). The color bar represents the range of depth, from light yellow to dark purple, indicating closer to the camera.
Related tasks

Structure from motion

New view synthesis

Deep3D [51] predicts a second stereo viewpoint

Learning 3D

Self-supervised learning from video

Teacher can be Kinect or any other standard stereo algorithm

design pretext tasks for learning generic visual features from video data that can later be re-purposed for other vision tasks such as object detection and semantic segmentation.
Training data

Assumptions:
• Many short sequences
• Scenes are rigid. Changes are due to camera motion.

No other metadata or annotations used.
Main idea: supervision by view synthesis
Main idea: supervision by view synthesis

Recover 3D structure and camera motion so that we can reconstruct the frame at time $t$ using the neighboring frames.

Hypothesis: geometric view synthesis system only performs consistently well when its intermediate predictions of the scene geometry and the camera poses correspond to the physical ground-truth.
For frame $t$, we want to estimate:

- Camera motion (6 degrees of freedom)
- Scene structure (pixel depth maps)
Training: View synthesis as supervision

Interpolated view

Hypothesis: geometric view synthesis system only performs consistently well when its intermediate predictions of the scene geometry and the camera poses correspond to the physical ground-truth.
System components

Depth estimation

Pose estimation

View synthesis
System components

Depth estimation

Pose estimation

View synthesis

$I_t$ → Depth CNN →
System components

Depth estimation

Pose estimation

View synthesis

\[ I_t \]

\[ I_{t-1} \]

\[ T_{t+1, t} \]
System components

Depth estimation

Pose estimation

View synthesis

$I_t$

$I_{t-1}$

$I_{t+1}$

Depth CNN

Pose CNN

$T_{t+1, t}$

$T_{t-1, t}$
System components

Depth estimation
Pose estimation
View synthesis

We want to synthesize the value of this pixel

\( I_t \)

\( \hat{p}_t \)

\( I_s \)

\( \hat{p}_s \) ?

So that it looks like:

But we can only use another frame
System components

Depth estimation

Pose estimation \( T_{t,s} \)

View synthesis

\[
p_s \sim K T_{t \rightarrow s} \hat{D}_t(p_t) K^{-1} p_t
\]

In this work, \( K \) is assumed to be known!
System components

Depth estimation

Pose estimation

View synthesis

Training Loss: \( \mathcal{L}_{vs} = \sum_{s \in \{\text{nearby frames}\}} \sum_{p} |I_t(p) - I_s(\hat{p}_s)| \)
Limitations

1) the scene is static without moving objects;

2) there is no occlusion/disocclusion between the target view and the source views;

3) the surface is Lambertian so that the photo-consistency error is meaningful.

If any of these assumptions are violated in a training sequence, the gradients could be corrupted and potentially inhibit training.
## Results

<table>
<thead>
<tr>
<th>Input</th>
<th>Ground-truth</th>
<th>Eigen <em>et al.</em> (depth sup.)</th>
<th>Garg <em>et al.</em> (pose sup.)</th>
<th>Ours (unsupervised)</th>
</tr>
</thead>
</table>
3D object modeling

[Kanazawa, Tulsiani, et al., ECCV 2018]
Learning object models from the internet

Angjoo Kanazawa

Learning Category-Specific Mesh Reconstruction from Image Collections
Angjoo Kanazawa*, Shihbum Tulsiani*, Alireza A. Efros, Jitendra Malik
University of California, Berkeley

Abstract. We present a learning framework for recovering the 3D shape, camera, and texture of an object from a single image. The shape is represented as a deformable 3D mesh model of an object category whose shape is parameterized by a learned mean shape and pose. We predict latent deformations. Our approach allows learning an annotated image collection for training, where the deformable model and the 3D prediction network are learned without relying on ground truth 3D multi-view supervision. Our representation enables us to go beyond existing 3D prediction approaches by incorporating semantic inference as prediction of an image in a canonical appearance space. Additionally, we show that semantic segmentation can be easily associated with the predicted shapes. We present qualitative and quantitative results of our approach on COB and PNS k-NN datasets and show that we can learn to predict these shapes and textures across objects using only annotated image collections. The project website can be found at https://akazawa.github.io/mesh/.

Fig. 3: Given an annotated image collection of an object category, we learn a predictor \(f \) that can map a novel image \(x\) to its 3D shape, camera pose, and texture.

1 Introduction
Consider the image of the bird in Figure 1. Even though this is a two-dimensional picture printed on a page may be the first time that we are seeing this particular bird, we can

* The first two authors contributed equally to this work.
Learning object models from the internet
Analysis by synthesis

Find a [shape, camera, texture] combination (analysis) that renders to the image (synthesis).

Shape

Camera

Texture
Funny looking autoencoder

Find a [shape, camera, texture] combination that renders to the image.

non-learned renderer
Training

Learn 3D only from 2D image-based annotations
Many images are only seen under a single view point
Approach

\begin{align*}
\text{Encoder} \rightarrow \text{Camera} \rightarrow \pi \rightarrow \text{Shape} \rightarrow \text{Texture} \rightarrow \text{3D Keypoints}
\end{align*}

\begin{align*}
\text{Losses:} & \quad \text{Predicted, GT} \\
\text{Texture:} \\
\text{Mask:} \\
\text{SfM Camera:} & \quad \pi \rightarrow \pi^{\text{sfm}} \\
\text{Keypoints:} & \quad \pi^{\text{sfm}}(\bullet) \rightarrow \mathbf{x}
\end{align*}
Shape Representation

Predicted Shape = Learned Mean Shape + Shape Deformation

A

3D keypoints
Texture Representation

\[ \Delta V_1 \]

\[ \Delta V_2 \]

\[ \bar{V} \]

\[ I^{uv} \]

\[ \phi, \theta \]
Texture Representation

\[ \Delta V_1 \quad \Rightarrow \quad \Delta V_2 \]

\[ \bar{V} \]

\[ \phi, \theta \]

\[ I^{uv} \]
Texture Representation

\[ \Delta V_1 \]

\[ \Delta V_2 \]

\[ \bar{V} \]

\[ I^{uv} \]

\[ \phi, \theta \]
Texture as UV Image Prediction

CNN

Texture Flow

UV Image
Results
Texture Transfer
Texture Transfer
Intuitive physics

[“Learning to See Physics via Visual De-animation”, Wu et al., NIPS 2017]
Intuitive physics

[“Learning to See Physics via Visual De-animation”, Wu et al., NIPS 2017]
Scene understanding

Input Image

Encoder

Semantic Decoder

Instance Decoder

Depth Decoder

Semantic Task Uncertainty

Instance Task Uncertainty

Depth Task Uncertainty

∑

Multi-Task Loss
Scene understanding is an integrated process

Mezzanotte & Biederman
Taskonomy: Disentangling Task Transfer Learning

Amir R. Zamir1,2, Alexander Sax1, William Shen1, Leonidas Guibas1, Jitendra Malik2 and Silvio Savarese3

1 Stanford University, 2 University of California, Berkeley

http://taskonomy.vision/

Abstract

We distill a common core of tasks shared by a family of visual and natural language tasks. Our approach is based on the observation that there is a natural hierarchy of tasks ordered by their difficulty. We propose a joint model that learns to transfer knowledge across tasks at different levels of the hierarchy. Our model is trained on a large dataset of semantically annotated images and is able to perform well on a variety of tasks, including object detection, image segmentation, and scene classification.

1. Introduction

Object recognition, depth estimation, edge detection, pose estimation, etc. are examples of common visual tasks deemed useful and studied by the research community. Some of these tasks have clear relationships: we understand that surface normals and depths are related (one is a derivative of the other), or that bounding boxes in a scene are useful for occlusion. Other relationships are less clear: how keypoints and the shape of a room can be used to perform pose estimation. Understanding these relationships is the first step towards developing more expressive and powerful models.

1x1

The field of computer vision has proved quite successful, with some of the most exciting recent advances in the field being driven by deep learning algorithms. However, most of these algorithms are trained in isolation and do not transfer knowledge across tasks. We believe that a better understanding of the relationships among tasks is needed to develop more expressive and powerful models.