Emerging Directions in Deep Learning

Introduction.
"Extrinsic" approaches: learning algorithms relying on regular structures

Mihai-Sorin Stupariu, 2024-2025

Motivation - why do we need Geometric Machine Learning?

3D world; around us it is a lot of 3D geometry!

Source: https://unibuc.ro/studii/facultati/

Motivation - why do we need Geometric Machine Learning?

Practical problems - Facial expressions:

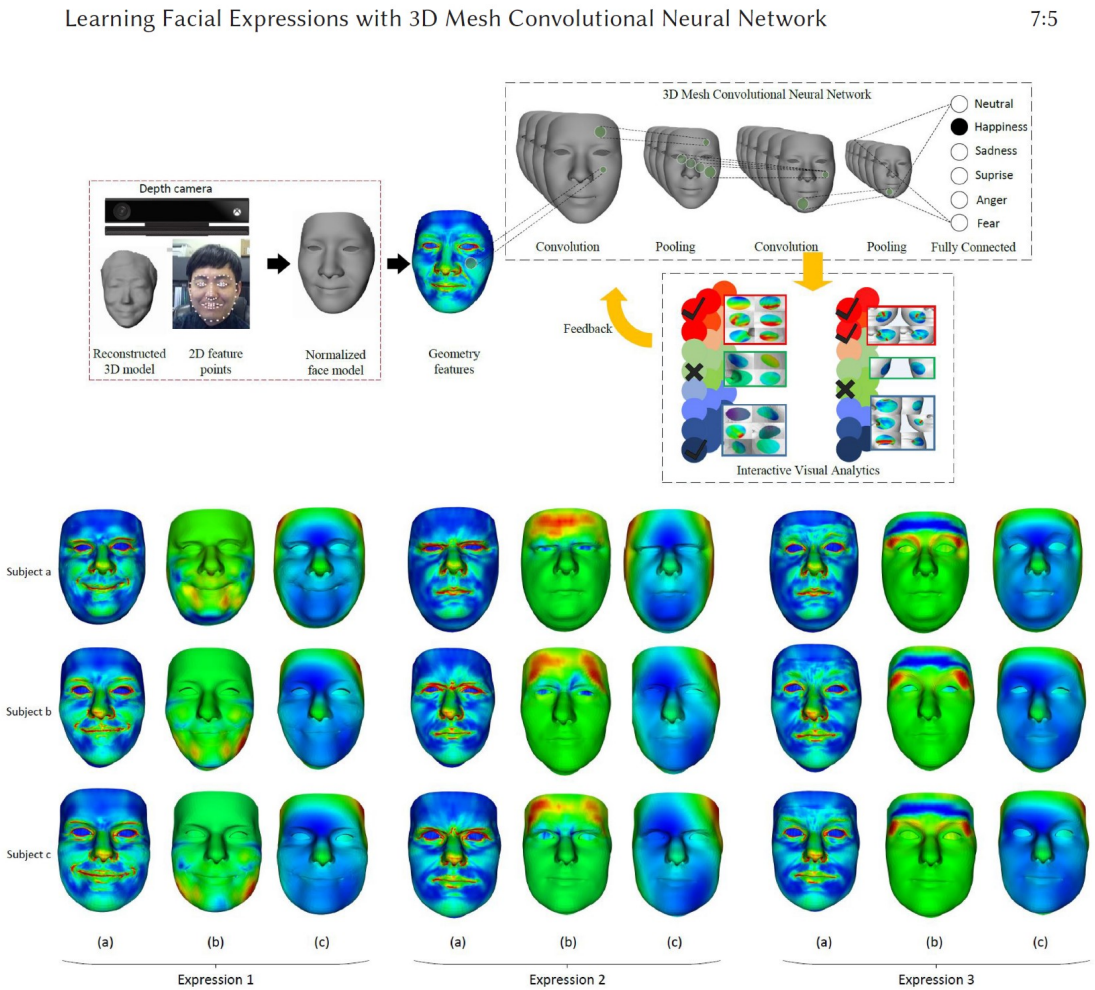


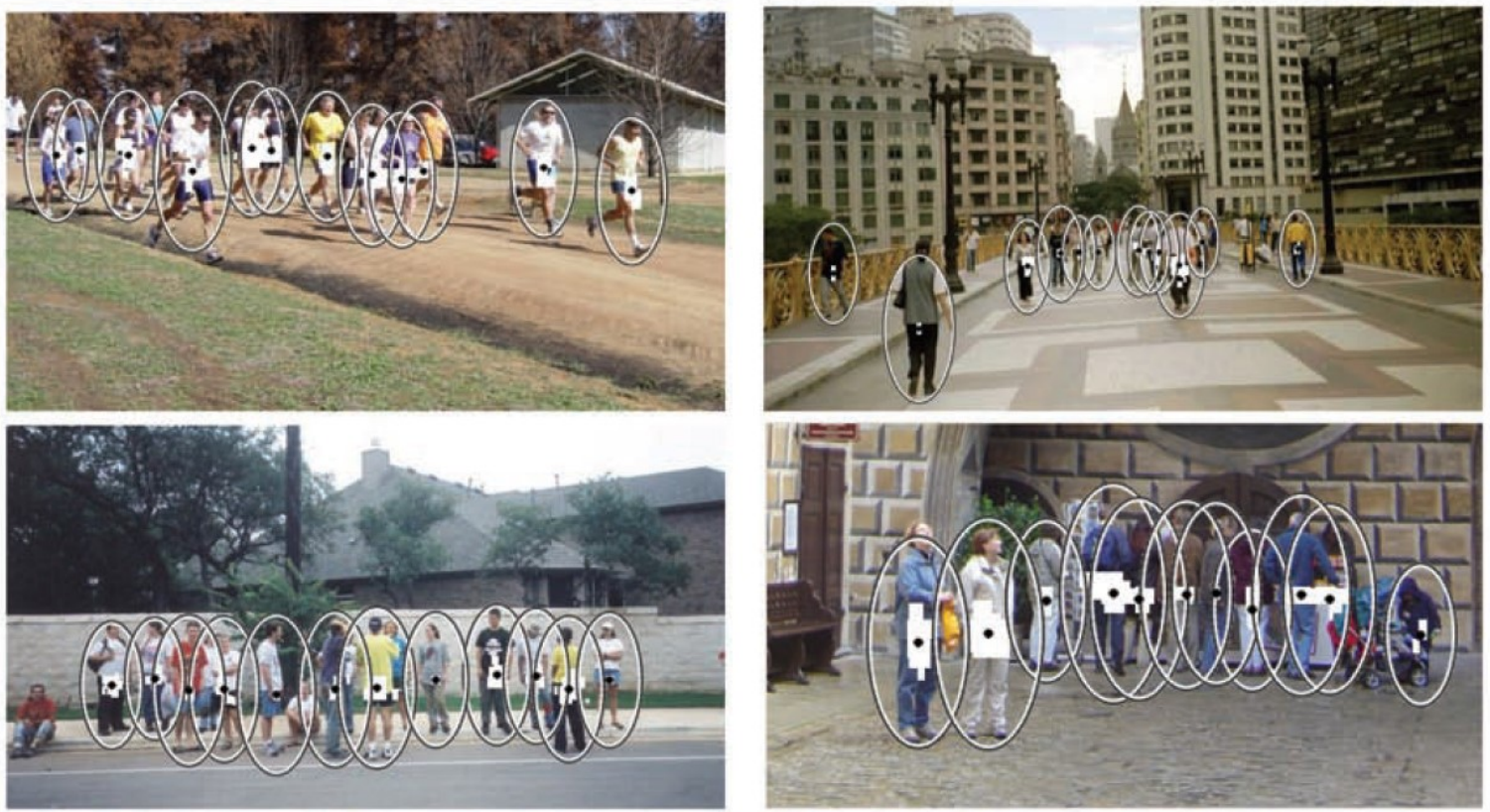
Fig. 11. The geometry signatures on the 3D face. Each row shows different expressions of the same subject.

[Jin et al., 2018]

Motivation - why do we need Geometric Machine Learning?

Practical problems - Pedestrian detection:

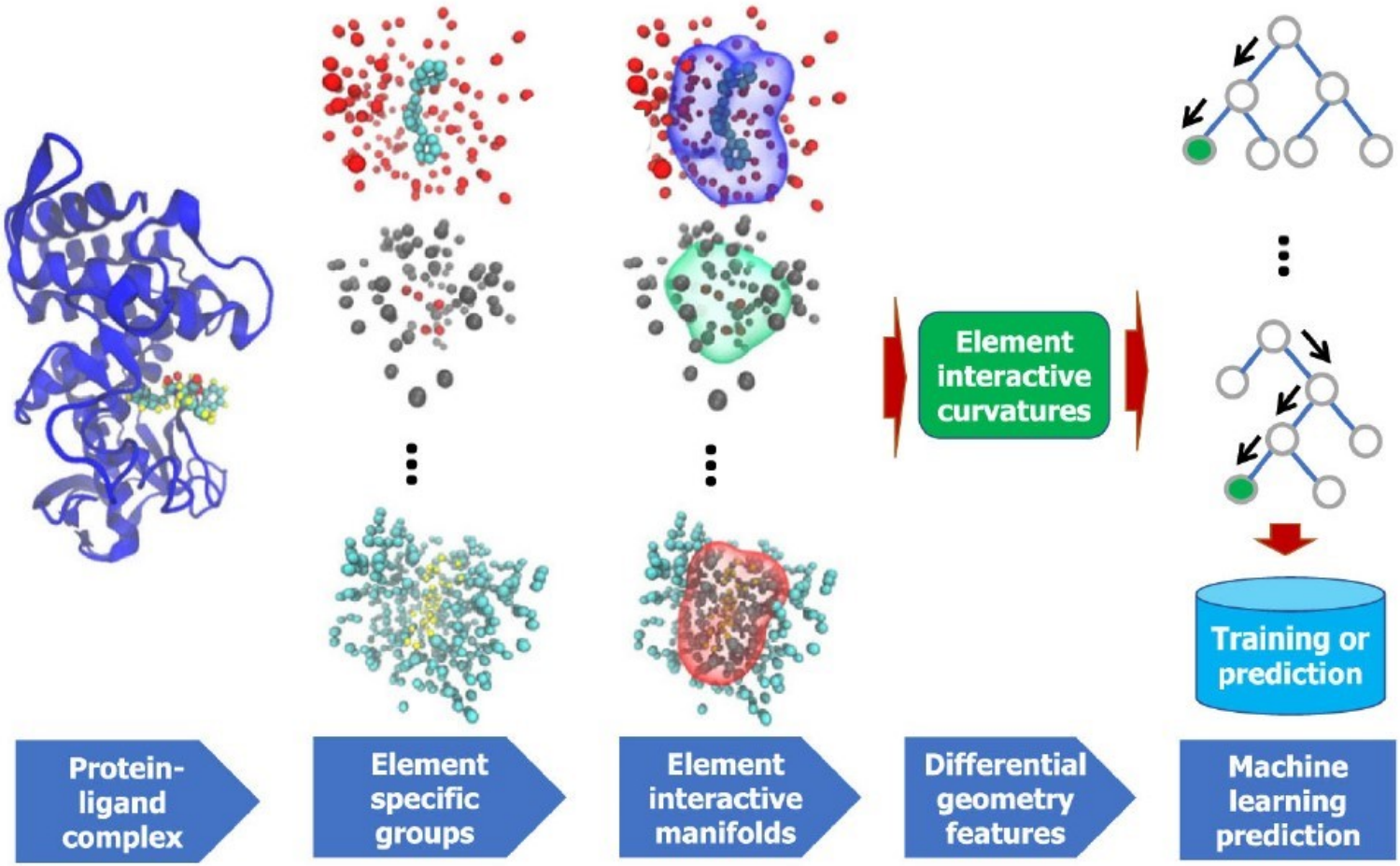
TUZEL ET AL.: PEDESTRIAN DETECTION VIA CLASSIFICATION ON RIEMANNIAN MANIFOLDS



[Tuzel, 2008]

Motivation - why do we need Geometric Machine Learning?

Practical problems - Molecular Biology



[Nguyen & Wei, 2019]

Applications: medicine

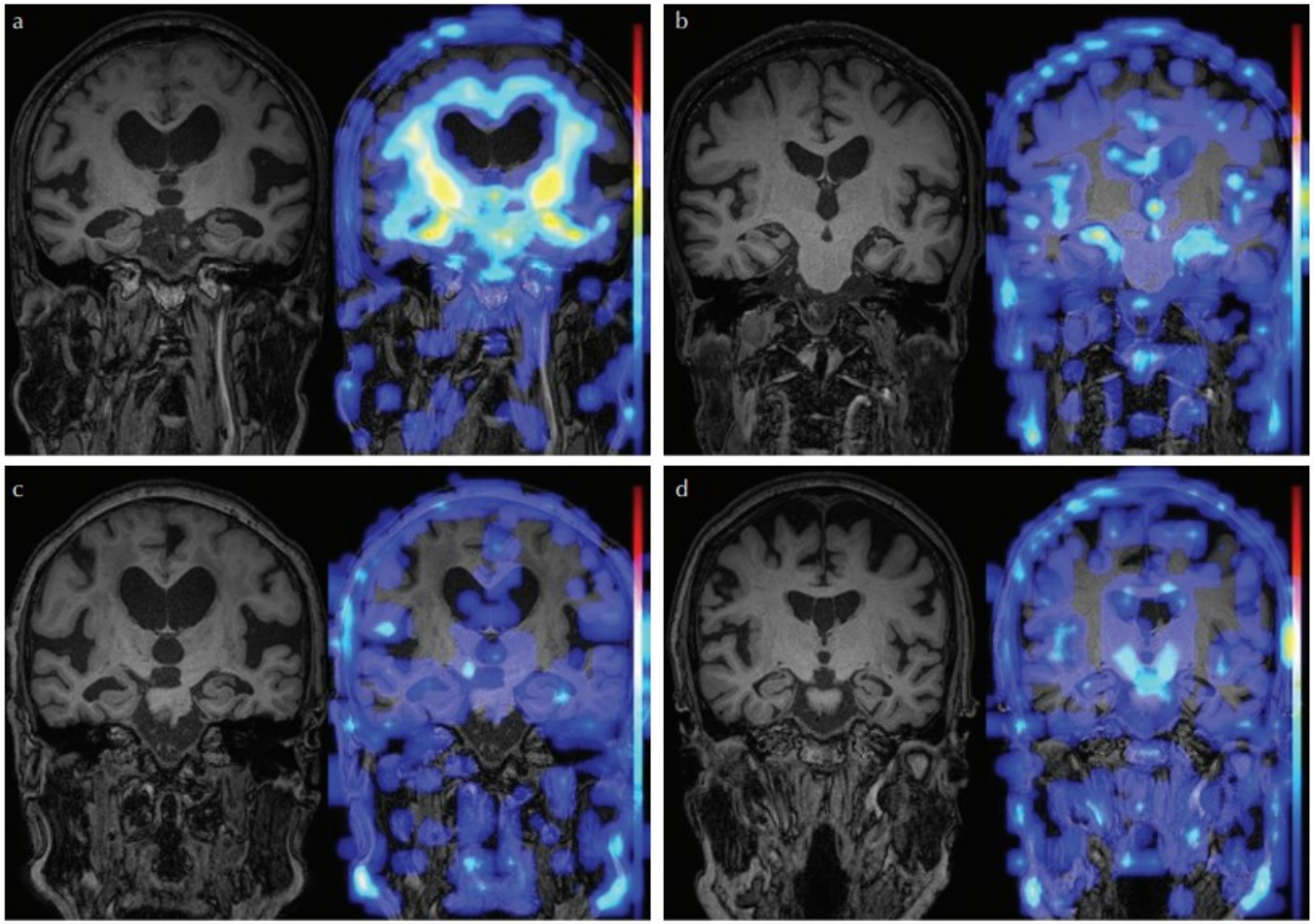


Fig. 3 Representative images of successfully diagnosed cases. The 3D T_1-weighted image is on the left and the Gradient-weighted Class Activation Mapping heat map overlaid on the 3D T_1-weighted image is on the right in each case. Brain parenchyma surrounding the lateral ventricle is highlighted in an idiopathic normal pressure hydrocephalus (iNPH) case (**a**). Medial temporal lobe or inferior horn of the lateral ventricle is highlighted in an AD case (**b**). About half of the successful cases show nonspecific heat maps (**c**: iNPH, **d**: AD).

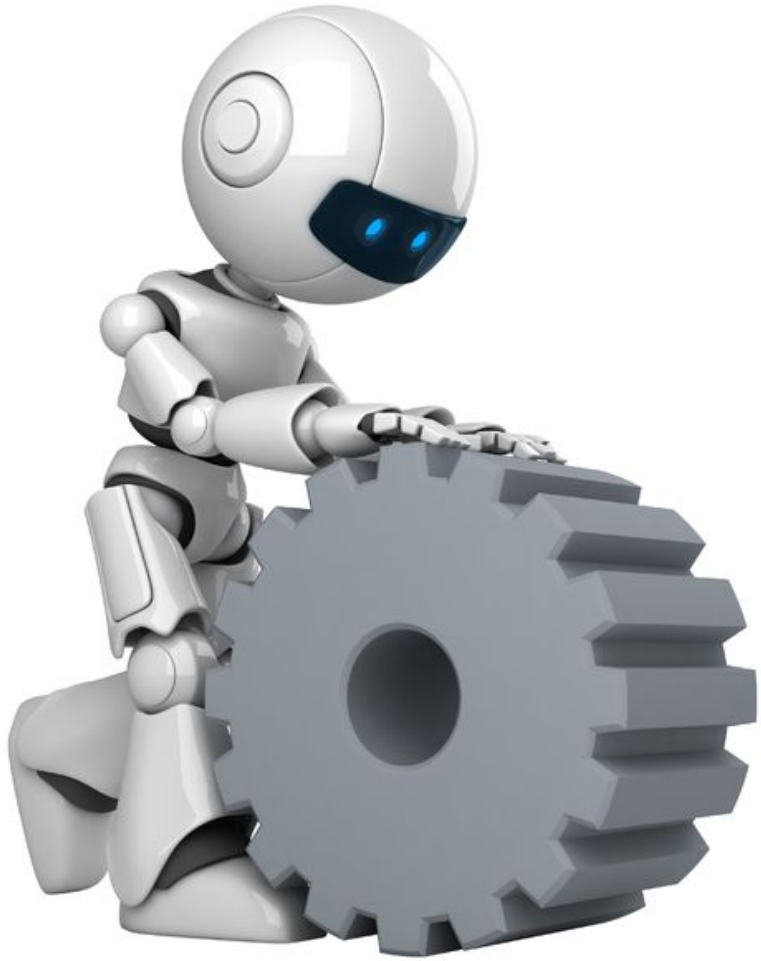
[Irie et al., 2020]

Applications: 3D visual computing for cultural heritage

Figure 5: *Small detail of a cuneiform tablet in 3D having a bounding box of* $8 \times 5.5 \times 1.8$*mm described by 67 000 vertices: (a) virtual high-contrast illumination, (b) manually annotated wedges including a damaged wedge, shown in red color with hatches, (c) correlation of feature vectors, combined with (e) auto-correlation as suggested in [MKJB10], (d,f) results for 30 repetitions of the mean filter* $\bar{f}_{\times 30}(\mathbf{p}_i)$ *for the respective filters on the left-hand side in (c,e).*

Source: [Mara & Krömker, 2017]

Applications: robotics



Source: https://pacetoday.com.au/wp-content/uploads/2016/02/Applications-robot.jpg

Applications: autonomous driving

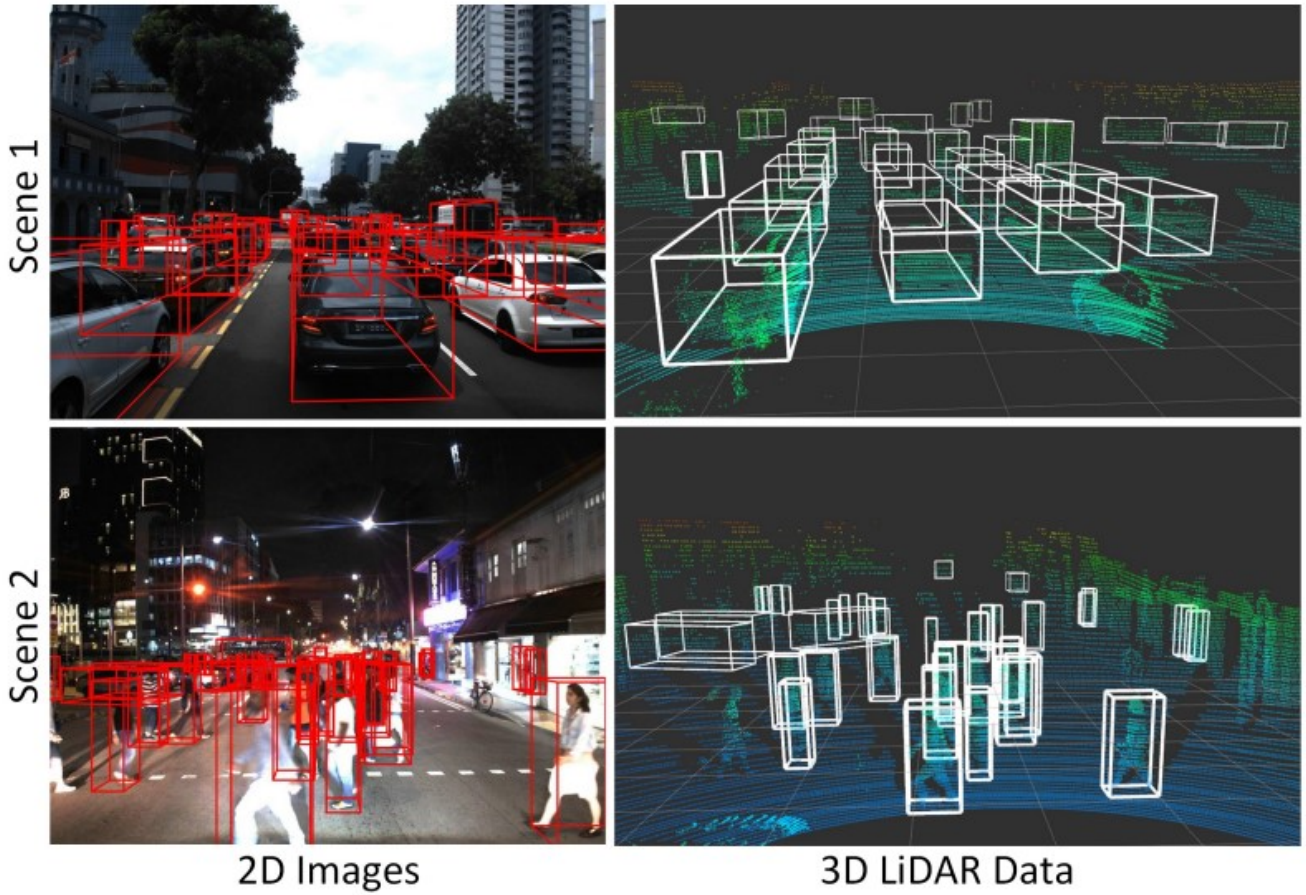
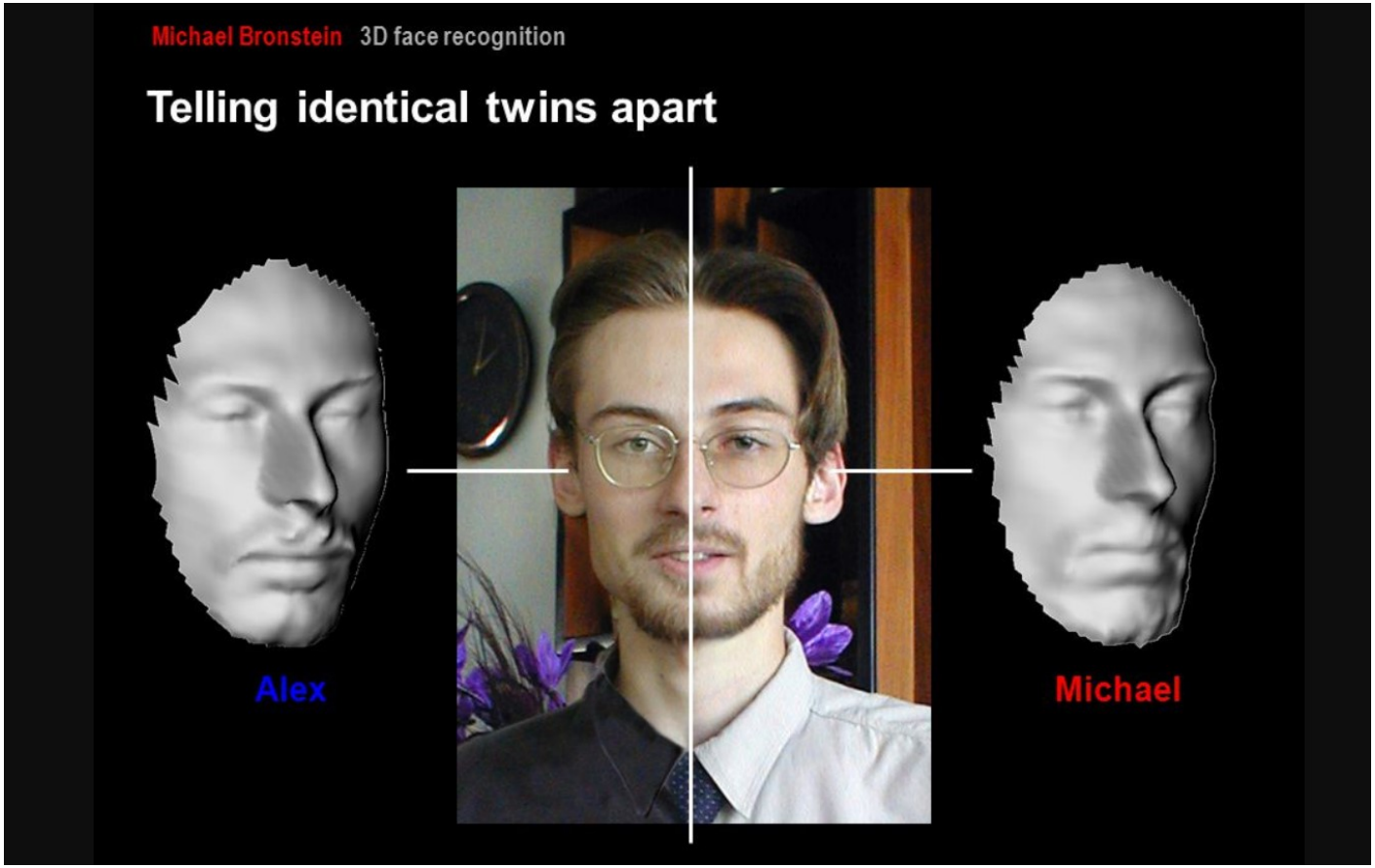


Fig. 1: Samples from the proposed A*3D dataset with RGB images and their corresponding LiDAR data. The two scenes captured in the evening and at night demonstrate high object-density in the environment.

Source: Pham et al., 2020

Once again, why do we need GDL?

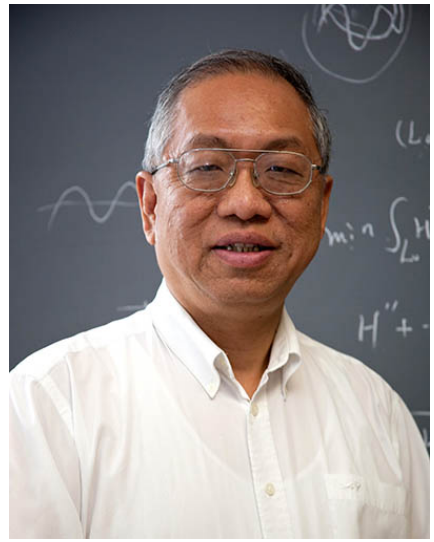


Source: M. Bronstein: Face recognition. New technologies, new challenges.

Is GDL a topic of interest?

Leonidas Guibas

Stephane Mallat



Shing-Tung Yau

.... and many others

DL in the 2D-framework

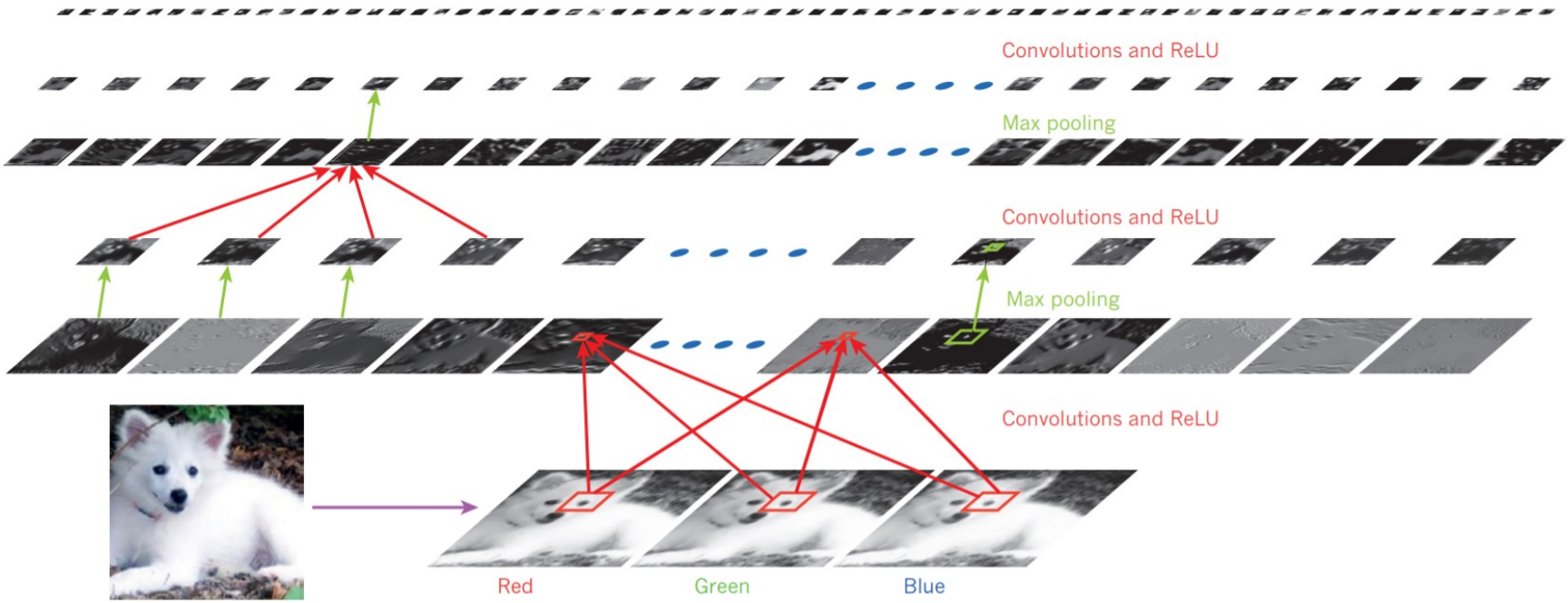


Figure 2 | Inside a convolutional network. The outputs (not the filters) of each layer (horizontally) of a typical convolutional network architecture applied to the image of a Samoyed dog (bottom left; and RGB (red, green, blue) inputs, bottom right). Each rectangular image is a feature map corresponding to the output for one of the learned features, detected at each of the image positions. Information flows bottom up, with lower-level features acting as oriented edge detectors, and a score is computed for each image class in output. ReLU, rectified linear unit.

[LeCun et al., 2015]

Passing to the 3D context

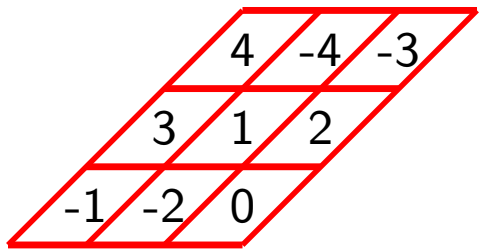
- **Aim:** To extend to the 3D-framework ML algorithms and techniques that work in the 2D-framework.

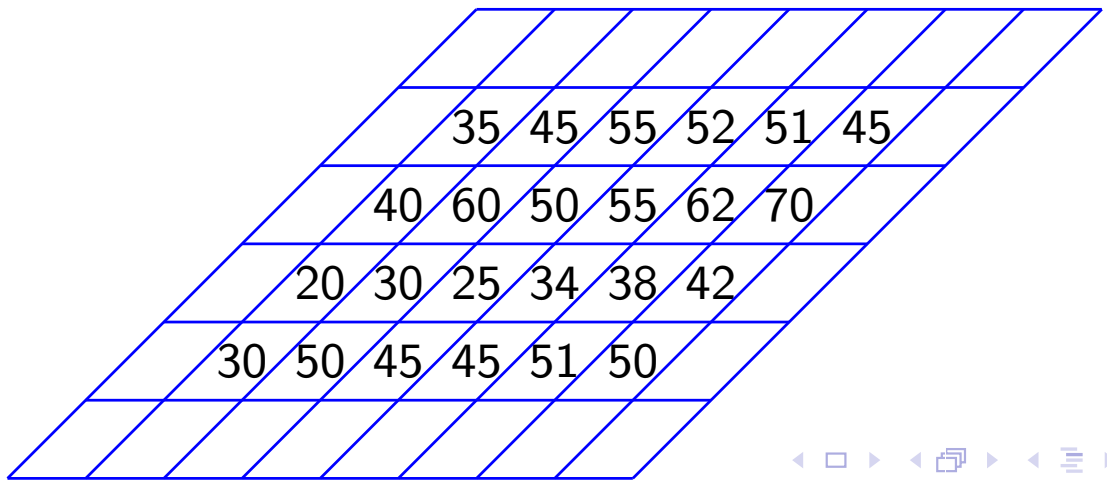
Passing to the 3D context

- **Aim:** To extend to the 3D-framework ML algorithms and techniques that work in the 2D-framework.
- Need to understand the key ingredients: (i) data structure; (ii) tasks; (iii) main operations (e.g., for a CNN: the grid (regular) structure, the convolution operation).

Passing to the 3D context

- **Aim:** To extend to the 3D-framework ML algorithms and techniques that work in the 2D-framework.
- Need to understand the key ingredients: (i) data structure; (ii) tasks; (iii) main operations (e.g., for a CNN: the grid (regular) structure, the convolution operation).





(i) About the data

- How to store the data? What is the meaningful information?

(i) About the data

- How to store the data? What is the meaningful information?
- How to represent the data during the analyses? In which format?

(i) About the data: a little bit of CS/CG history

The Utah teapot (1975). Have also a loot at Martin Newell's drawing of the Utah Teapot.

Source: Wikipedia, image loaded by Marshall Astor (http://www.marshallastor.com/)

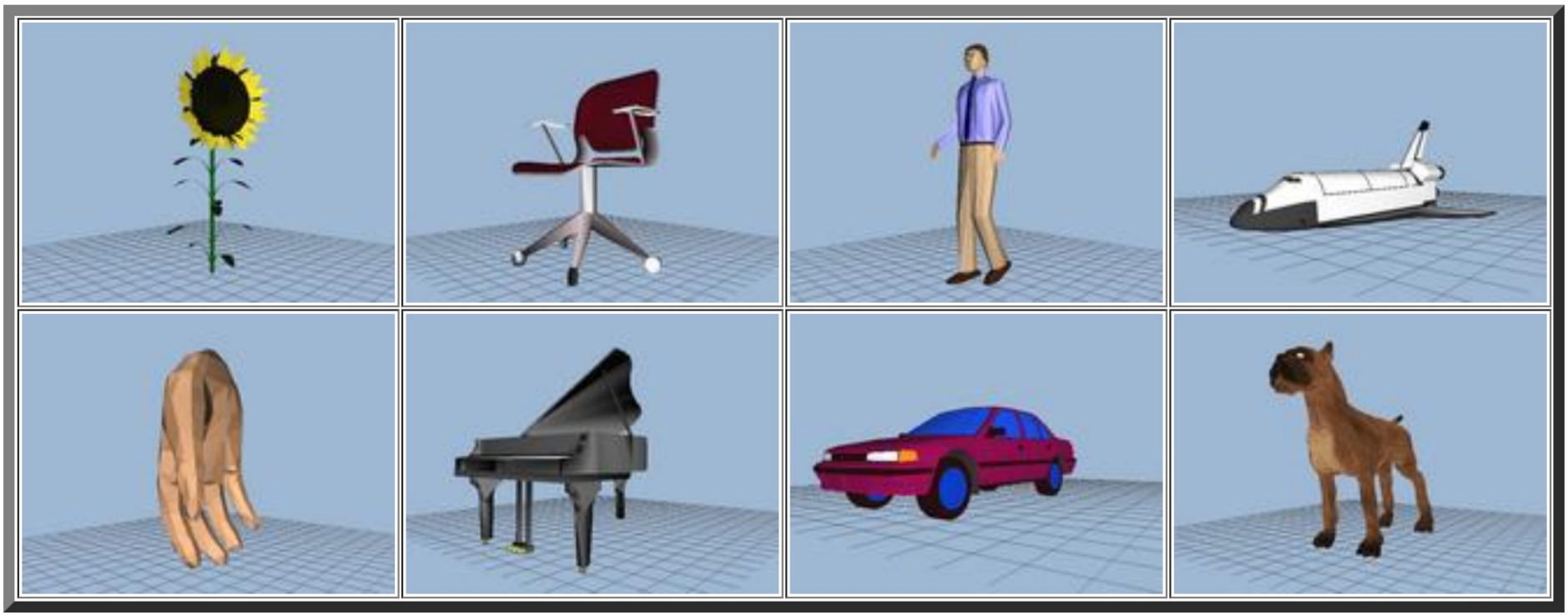
(i) About the data: a little bit of CS/CG history

The Stanford bunny (1994) - the Stanford repository.

Source: Stanford scanning repository

(i) About the data: a little bit of CS/CG history

The Princeton shape benchmark ($\sim$ 2005)



Source: Princeton shape benchmark

(i) About the data: a little bit of CS/CG history

ModelNet (2015)

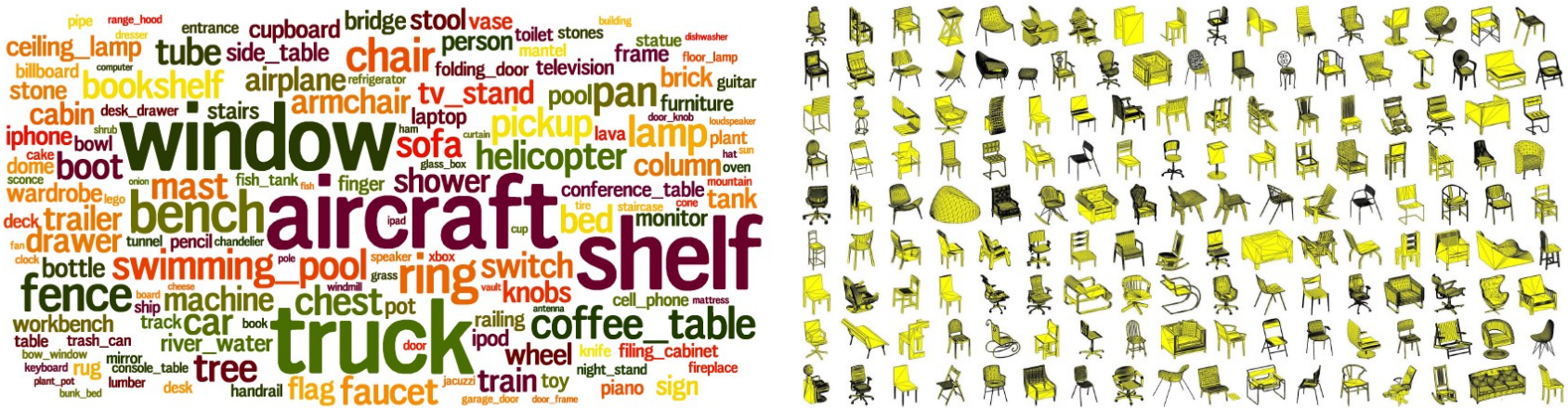
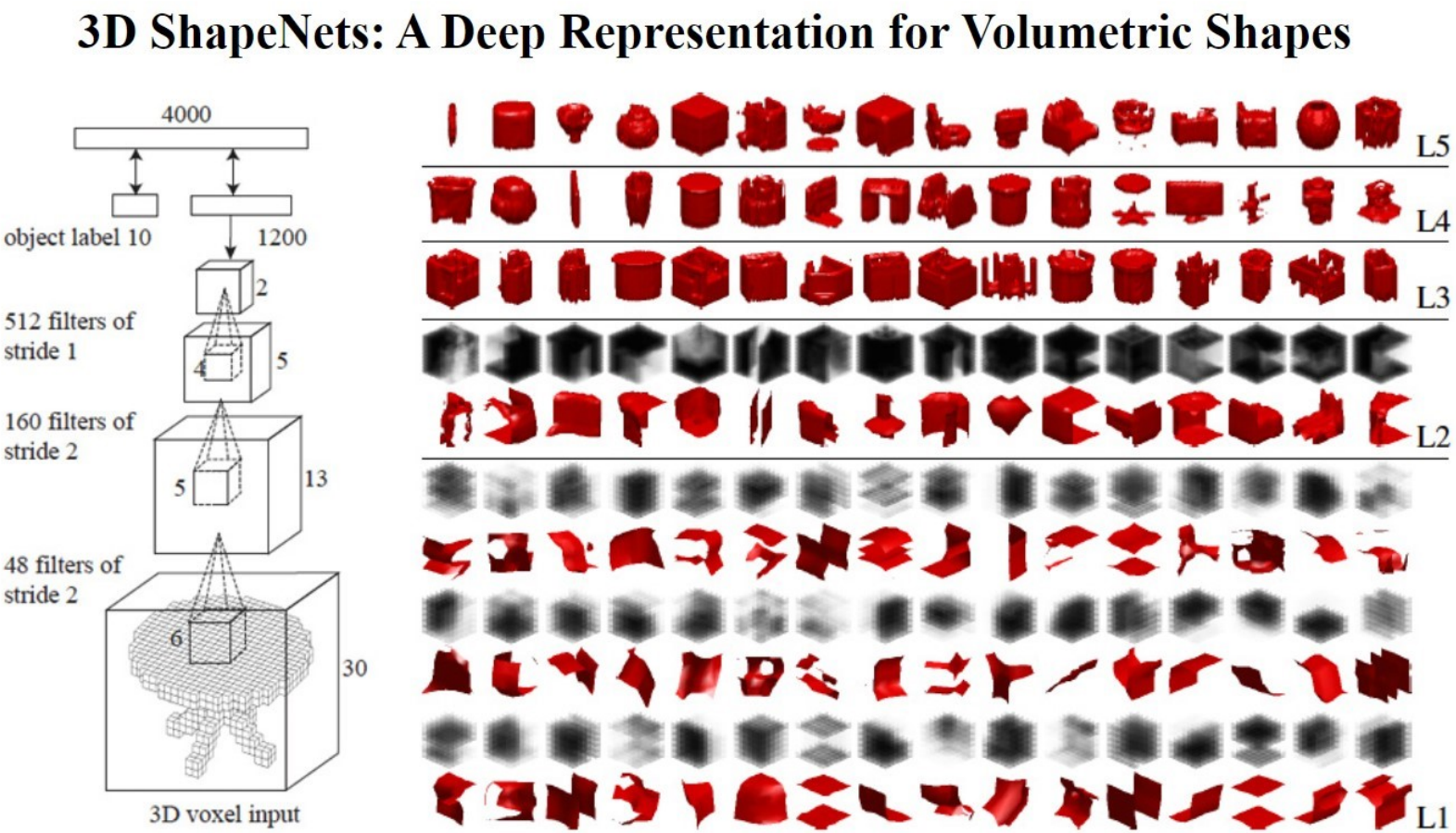


Figure 5: **ModelNet Dataset.** Left: word cloud visualization of the ModelNet dataset based on the number of 3D models in each category. Larger font size indicates more instances in the category. Right: Examples of 3D chair models.

Source: [Wu et al., 2015]

(i) About the data: a little bit of CS/CG history

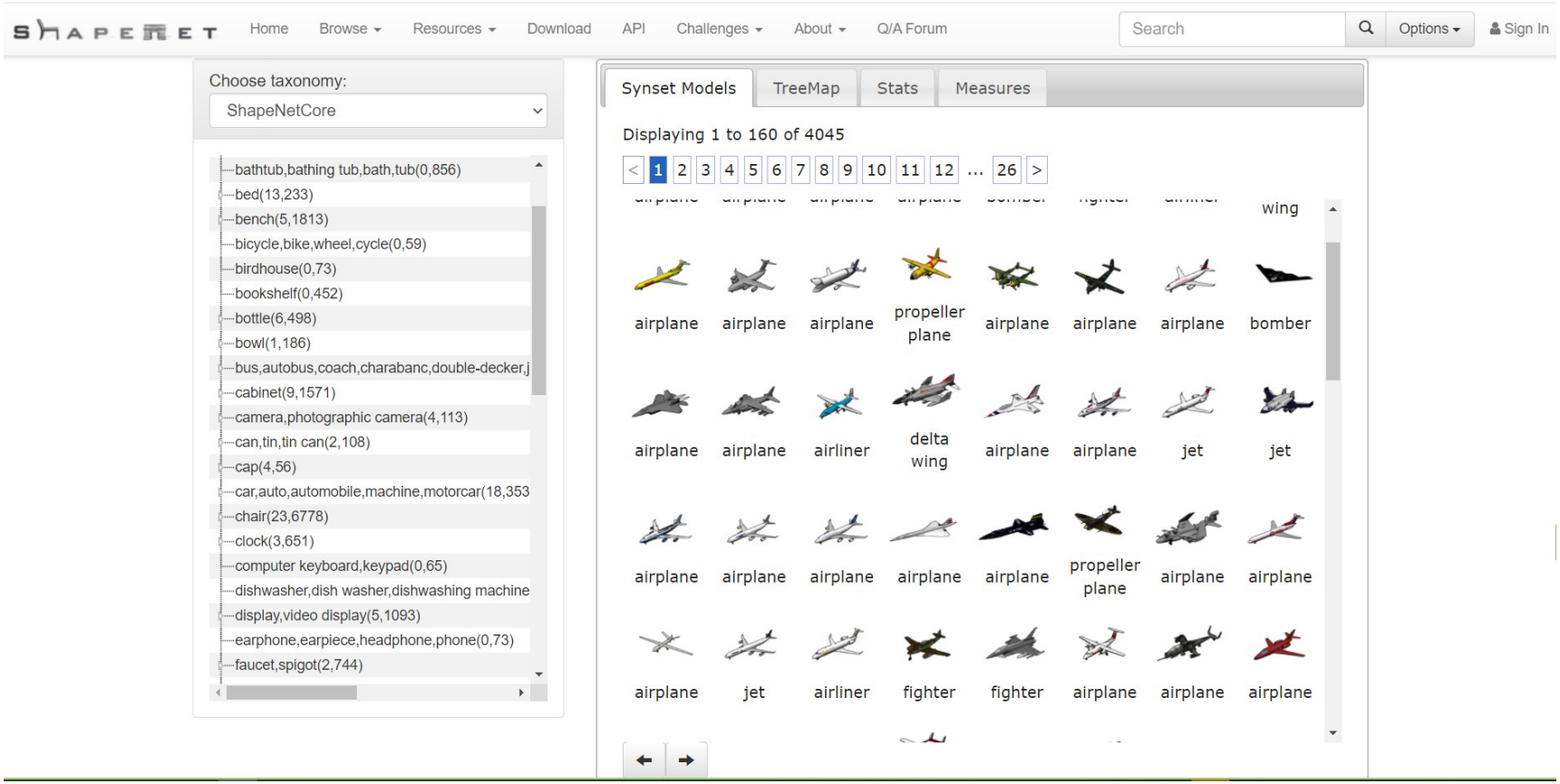
ModelNet (2015)



Source: ModelNet / 3D ShapeNets

(i) About the data: a little bit of CS/CG history

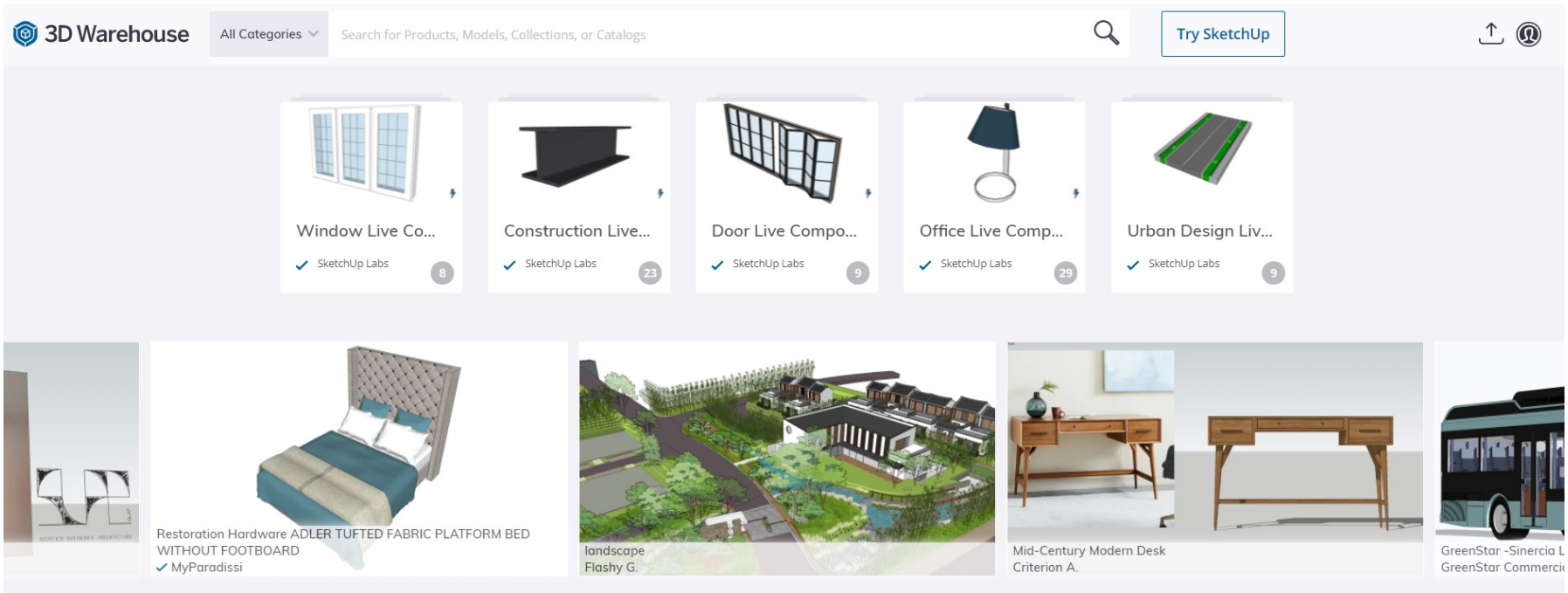
ShapeNet (2015)



Source: The ShapeNet repository

(i) About the data: a little bit of CS/CG history

Many resources are nowadays available!



Source: 3D warehouse

(ii) Tasks: analysis - classification

3D Warehouse

Source: 3D warehouse, lounge chair

This is a lounge chair!

(ii) Tasks: analysis - segmentation

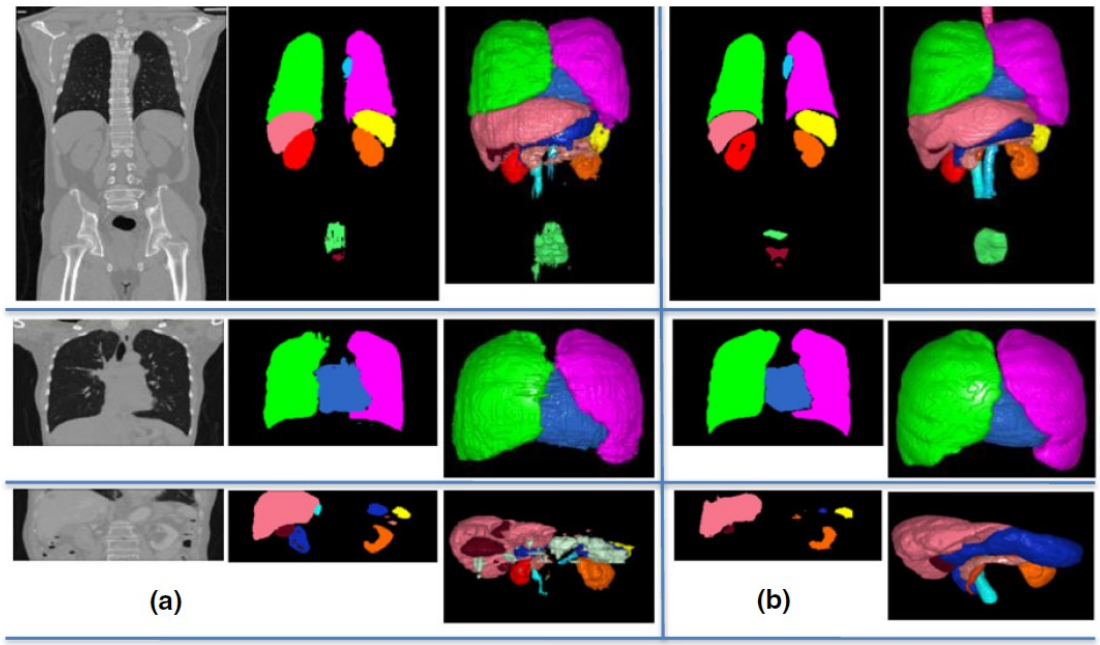


FIG. 5. (a): Three examples of segmentation in 3D CT case covering torso (upper), chest (middle), and abdomen (lower) regions along with segmented regions labeled with different colors for one 2D coronal CT slice (middle column) and 3D visualization based on surface-rendering method (right column). (b): Corresponding ground-truth segmentations for three cases.

Source: [Zhou et al., 2017]

(ii) Tasks: analysis - parsing

Fig. 1: *SIZER* dataset of people with clothing size variation. (*Left*): 3D Scans of people captured in different clothing styles and *sizes*. (*Right*): T-shirt and short pants for sizes small and large, which are registered to a common template.

Source: [Tiwari et al., 2020]

(ii) Tasks: analysis - correspondence

Figure 3: Pointwise geodesic error (in % of geodesic diameter) of different correspondence methods (top to bottom: Blended Intrinsic Maps, GCNN, ACNN) on the FAUST dataset. Error values are saturated at 10% of the geodesic diameter. Hot colors correspond to large errors.

Source: [Boscaini et al., 2016]

(ii) Tasks: synthesis - shape completion

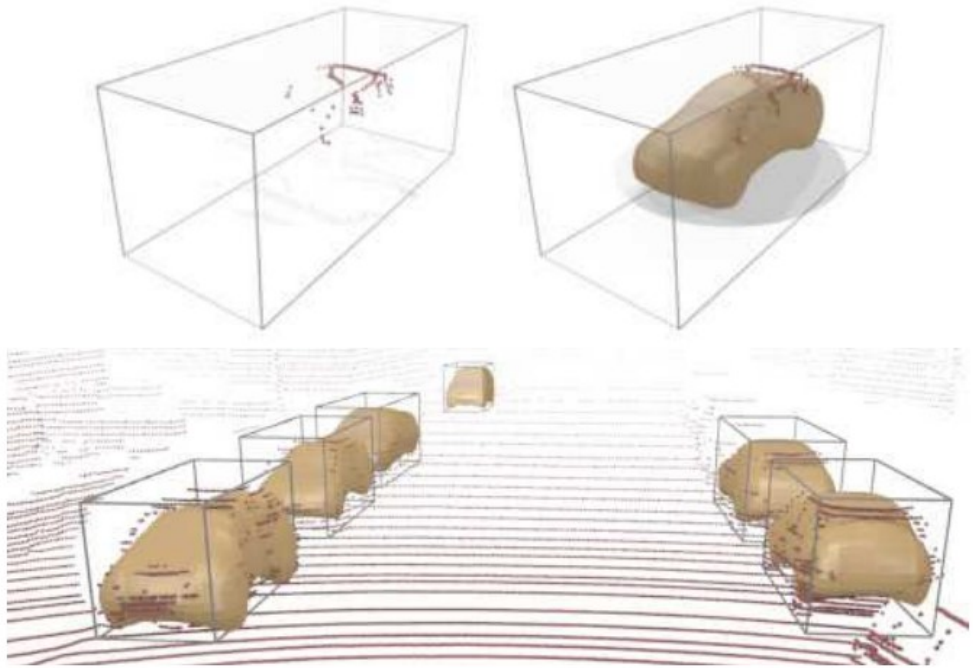


Figure 1: **Illustration of the 3D Shape Completion Problem.** Top: Given a 3D bounding box and an incomplete point cloud (left, red), our goal is to predict the complete shape of the object (right, beige). Bottom: Shape completion results on a street scene from KITTI [18]. Learning shape completion on real-world data is challenging due to sparse / noisy observations and missing ground truth.

Source: [Stutz & Geiger, 2018]

(iii) Operations: (if possible) take advantage of the 2D mechanisms!

- **Regular data:**

(iii) Operations: (if possible) take advantage of the 2D mechanisms!

- **Regular data:**
  - "Decompose" the 3D data in 2D layers (multiview), apply standard CNNs and then "aggregate" back the information.

(iii) Operations: (if possible) take advantage of the 2D mechanisms!

- **Regular data:**
  - "Decompose" the 3D data in 2D layers (multiview), apply standard CNNs and then "aggregate" back the information.
  - Replace pixels with voxels ("volumetric pixels") and the apply 3D-analogous of the convolution.

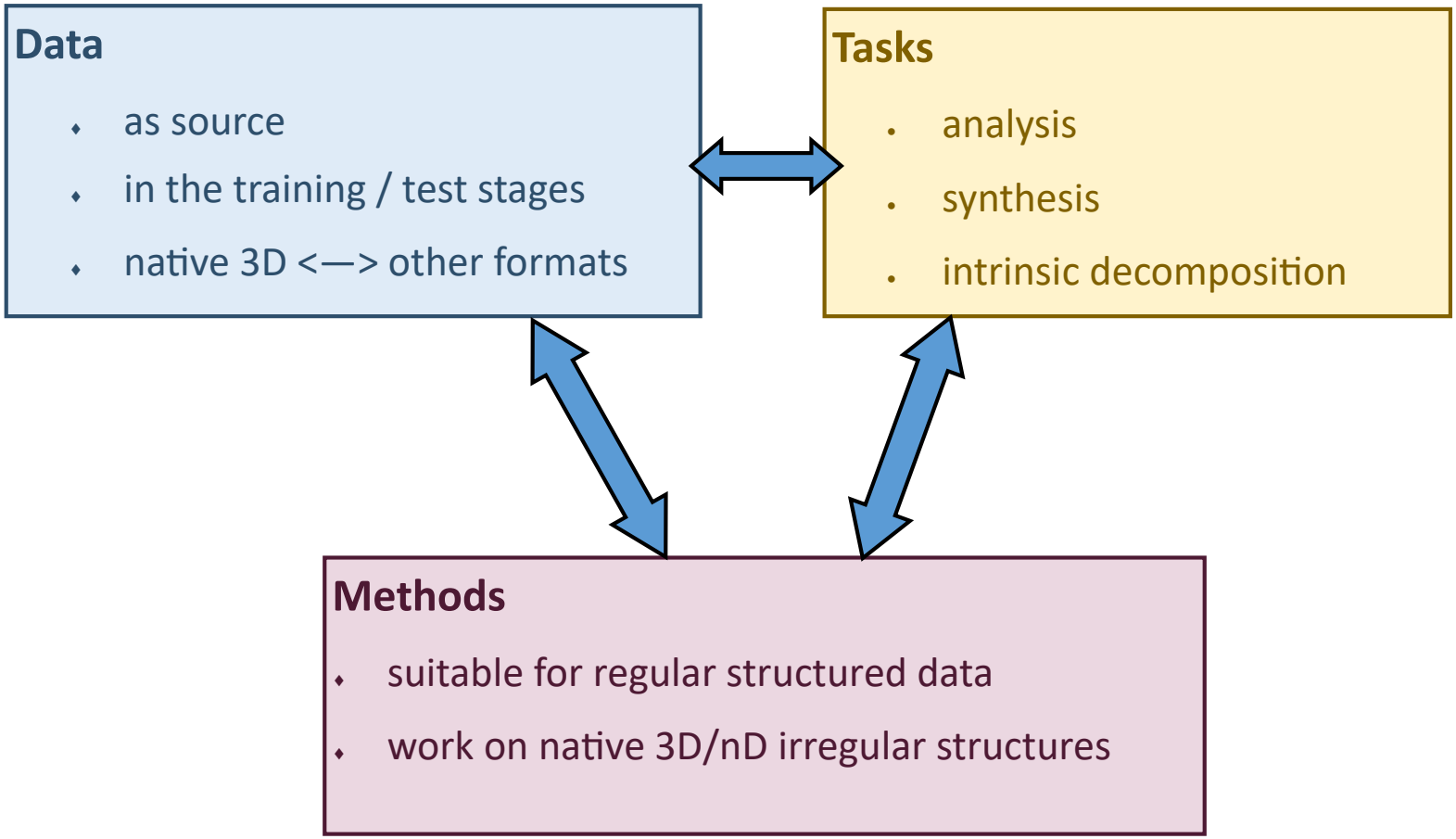
(iii) Operations: (if possible) take advantage of the 2D mechanisms!

- **Regular data:**
  - "Decompose" the 3D data in 2D layers (multiview), apply standard CNNs and then "aggregate" back the information.
  - Replace pixels with voxels ("volumetric pixels") and the apply 3D-analogous of the convolution.
- **Irregular data:**
  - Extend / generalize the key operations of a DL/ML pipeline for other native 3D formats (point clouds, triangle meshes, CAD models, mixed representations).

(iii) Operations: (if possible) take advantage of the 2D mechanisms!

- **Regular data:**
  - "Decompose" the 3D data in 2D layers (multiview), apply standard CNNs and then "aggregate" back the information.
  - Replace pixels with voxels ("volumetric pixels") and the apply 3D-analogous of the convolution.
- **Irregular data:**
  - Extend / generalize the key operations of a DL/ML pipeline for other native 3D formats (point clouds, triangle meshes, CAD models, mixed representations).
  - **Challenge 1:** cannot apply the standard 2D-methods on irregular data structures! **Challenge 2:** seek for methods that can be generalized to higher dimensions!

In a nutshell: brief overview



Multiview CNNs: overview of the approach

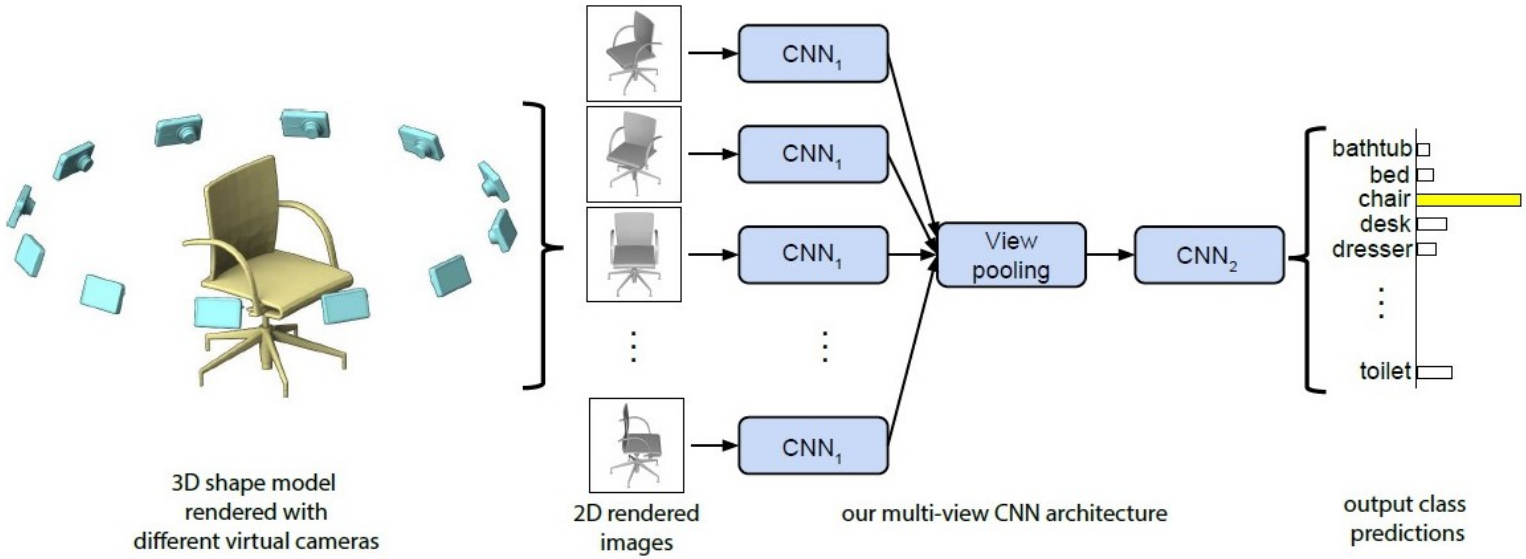


Figure 1. Multi-view CNN for 3D shape recognition (illustrated using the 1st camera setup). At test time a 3D shape is rendered from 12 different views and are passed thorough CNN_1 to extract view based features. These are then pooled across views and passed through CNN_2 to obtain a compact shape descriptor.

[Su et al., 2015]

Multiview CNNs: overview of the approach

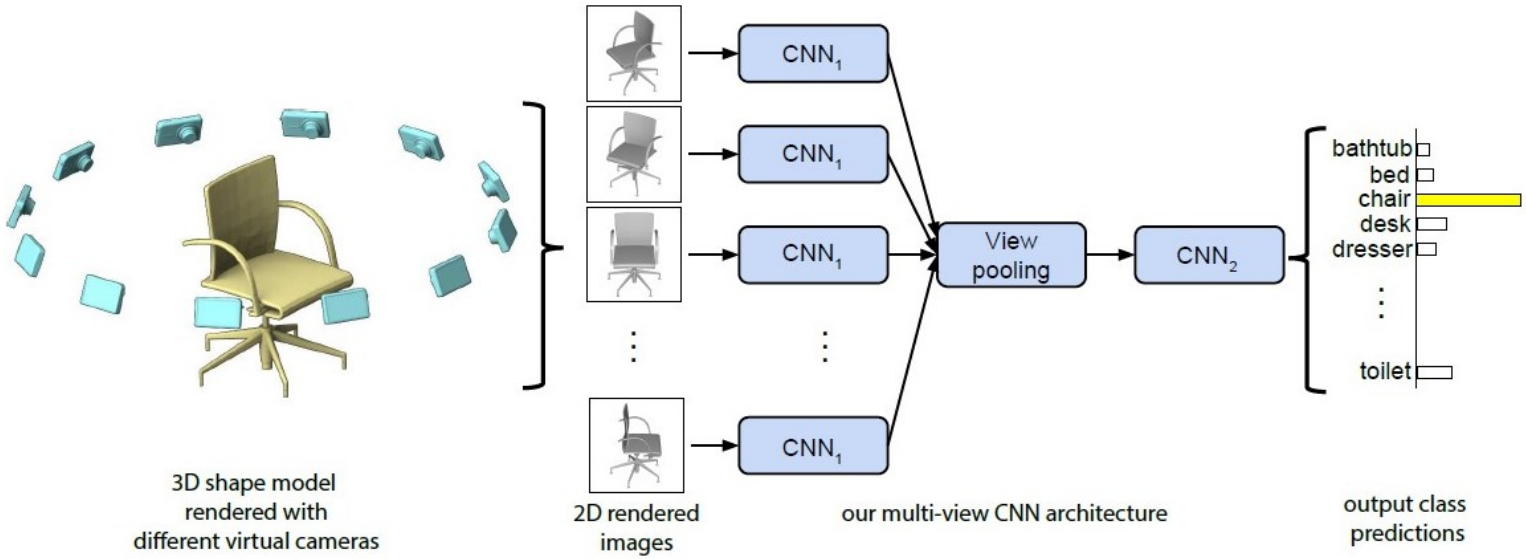


Figure 1. Multi-view CNN for 3D shape recognition (illustrated using the 1st camera setup). At test time a 3D shape is rendered from 12 different views and are passed thorough CNN_1 to extract view based features. These are then pooled across views and passed through CNN_2 to obtain a compact shape descriptor.

[Su et al., 2015]

- **Key of the approach:** Aim to get a single compact descriptor (alternative: pairwise comparisons between single-view representations).

Multiview CNNs: overview of the approach

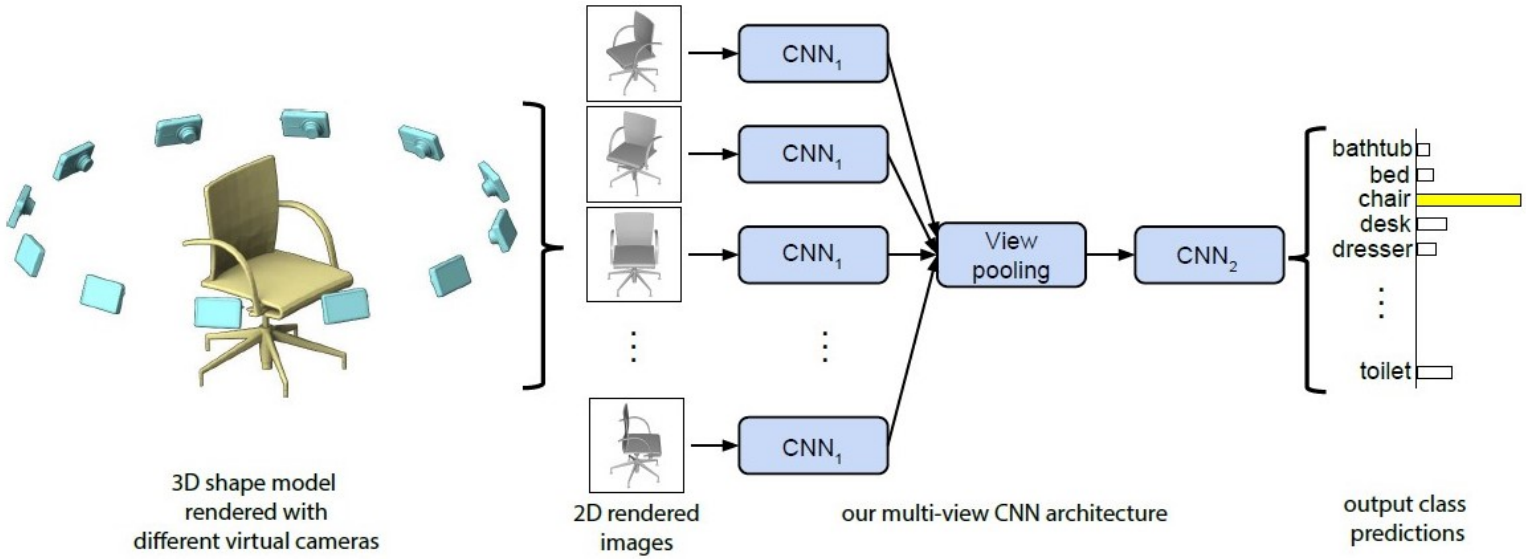


Figure 1. Multi-view CNN for 3D shape recognition (illustrated using the 1st camera setup). At test time a 3D shape is rendered from 12 different views and are passed thorough CNN_1 to extract view based features. These are then pooled across views and passed through CNN_2 to obtain a compact shape descriptor.

[Su et al., 2015]

- **Key of the approach:** Aim to get a single compact descriptor (alternative: pairwise comparisons between single-view representations).
- Try to find similarities with the human view!

Multiview CNNs: overview of the approach

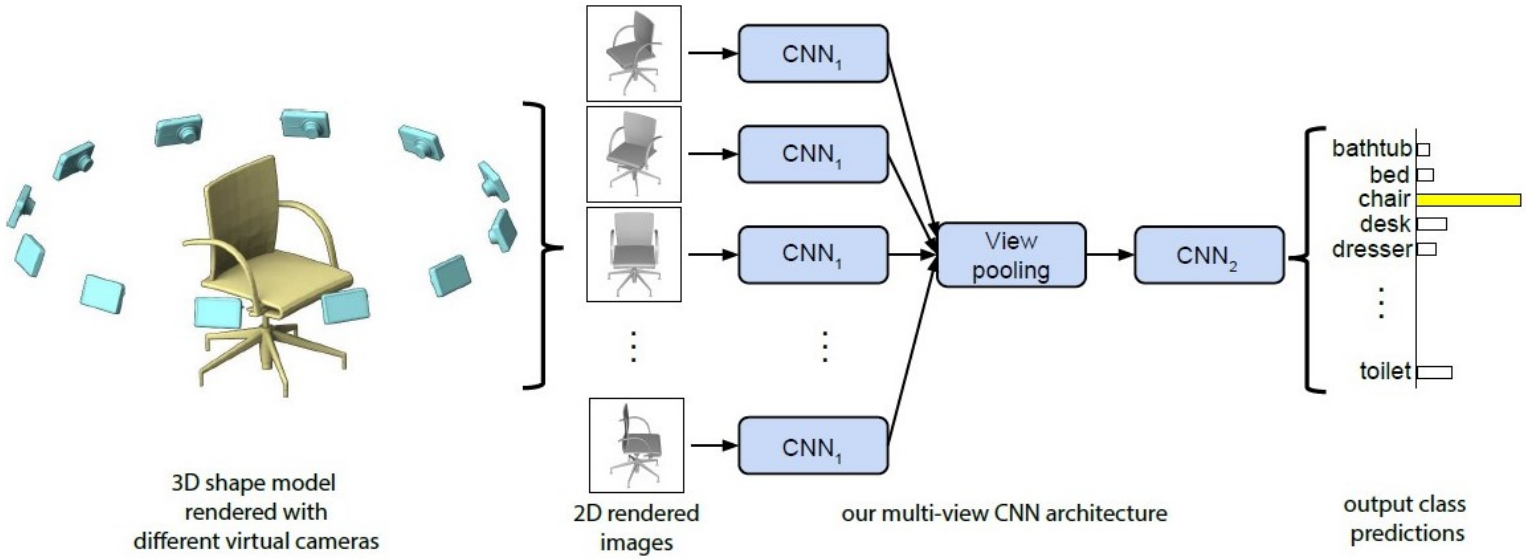


Figure 1. Multi-view CNN for 3D shape recognition (illustrated using the 1st camera setup). At test time a 3D shape is rendered from 12 different views and are passed thorough CNN_1 to extract view based features. These are then pooled across views and passed through CNN_2 to obtain a compact shape descriptor.

[Su et al., 2015]

- **Key of the approach:** Aim to get a single compact descriptor (alternative: pairwise comparisons between single-view representations).
- Try to find similarities with the human view!
- Take advantage of the available 2D Neural Networks techniques.

Multiview CNNs: comment on data handling

- **Data:** Polygon meshes; apply the Phong illumination model. Generate rendered views by applying a perspective projection.

Multiview CNNs: comment on data handling

- **Data:** Polygon meshes; apply the Phong illumination model. Generate rendered views by applying a perspective projection.

Source: http://maxwell.cs.umass.edu/mvcnn-data/modelnet40v1png/

Multiview CNNs: comment on data handling

- **Data:** Polygon meshes; apply the Phong illumination model. Generate rendered views by applying a perspective projection.

Source: http://maxwell.cs.umass.edu/mvcnn-data/modelnet40v1png/

- In brief: (i) pass from irregular to regular data; (ii) keyword: projection!; (ii) get rid of artifacts.

Multiview CNNs: results - saliency across views

Aim: rank pixels in the 2D views with respect to their influence on the output score F_c of the network (taken from $\mathbf{fc}_8$ layer) for its ground truth class c.

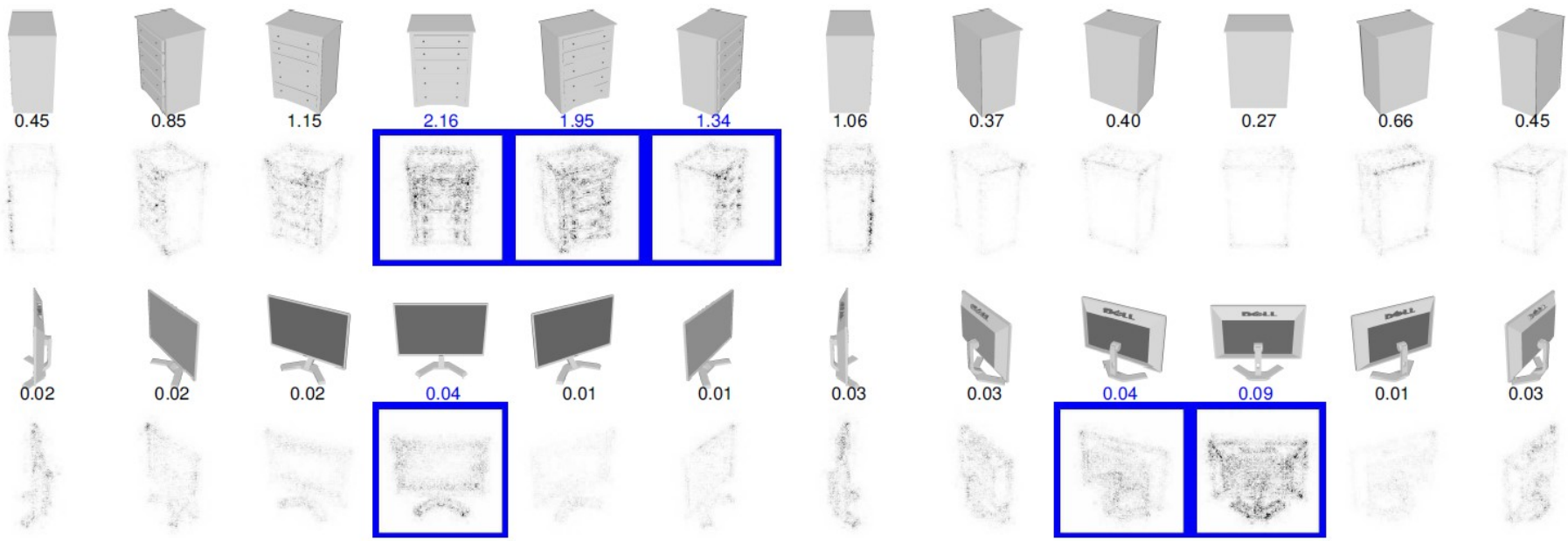


Figure 3. Top three views with the highest saliency are highlighted in blue and the relative magnitudes of gradient energy for each view is shown on top. The saliency maps are computed by back-propagating the gradients of the class score onto the image via the view-pooling layer. Notice that the handles of the dresser and of the desk are the most discriminative features. (Figures are enhanced for visibility.)

[Su et al., 2015; see also http://vis-www.cs.umass.edu/mvcnn/]

Multiview CNNs: results - sketch based 3D shape retrieval

Motivation: most online repositories provide only text-based search engines or hierarchical catalogs. Shape retrieval (e.g. [Eitz et al., 2012]) has been proposed as an alternative for users to retrieve shapes with an approximate sketch of the desired 3D shape in mind.

Method	Aug.	Accuracy
(1) FV [30]	-	79.0%
(2) CNN M	-	77.3%
(3) CNN M, fine-tuned	-	84.0%
(4) CNN M, fine-tuned	6×	85.5%
(5) MVCNN M, fine-tuned	6×	86.3%
(6) CNN VD	-	69.3%
(7) CNN VD, fine-tuned	-	86.3%
(8) CNN VD, fine-tuned	6×	86.0%
(9) MVCNN VD, fine-tuned	6×	**87.2%**
(10) Human performance	n/a	93.0%

Table 2. Classification results on SketchClean. Fine-tuned CNN models significantly outperform Fisher vectors [30] by a significant margin. MVCNNs are better than CNN trained with data jittering. The results are shown with two different CNN architectures – VGG-M (row 2-5) and VGG-VD (row 6-9).

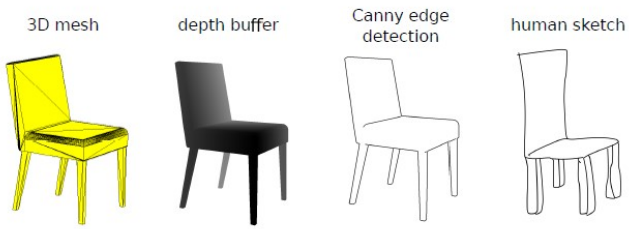


Figure 4. Line-drawing style rendering from 3D shapes.

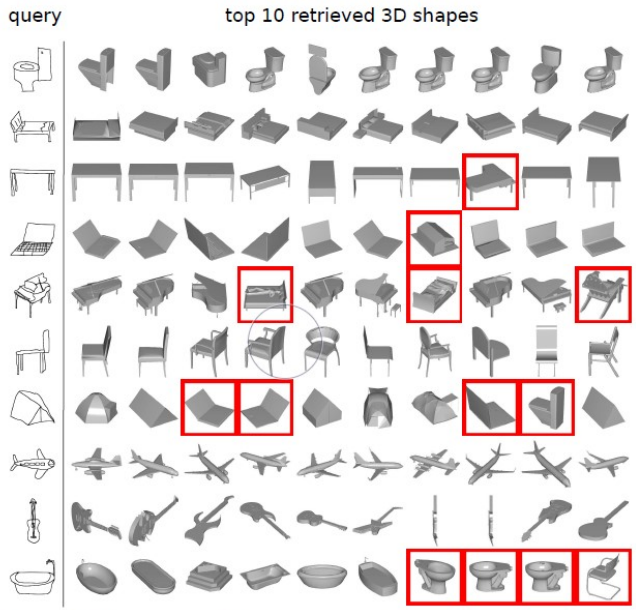


Figure 5. Sketch-based 3D shape retrieval examples. Top matches are shown for each query, with mistakes highlighted in red.

[Su et al., 2015; see also http://vis-www.cs.umass.edu/mvcnn/]

Related approaches: Segmentation with multi-view CNNs

Deep architecture for segmenting 3D objects into their labeled semantic parts.

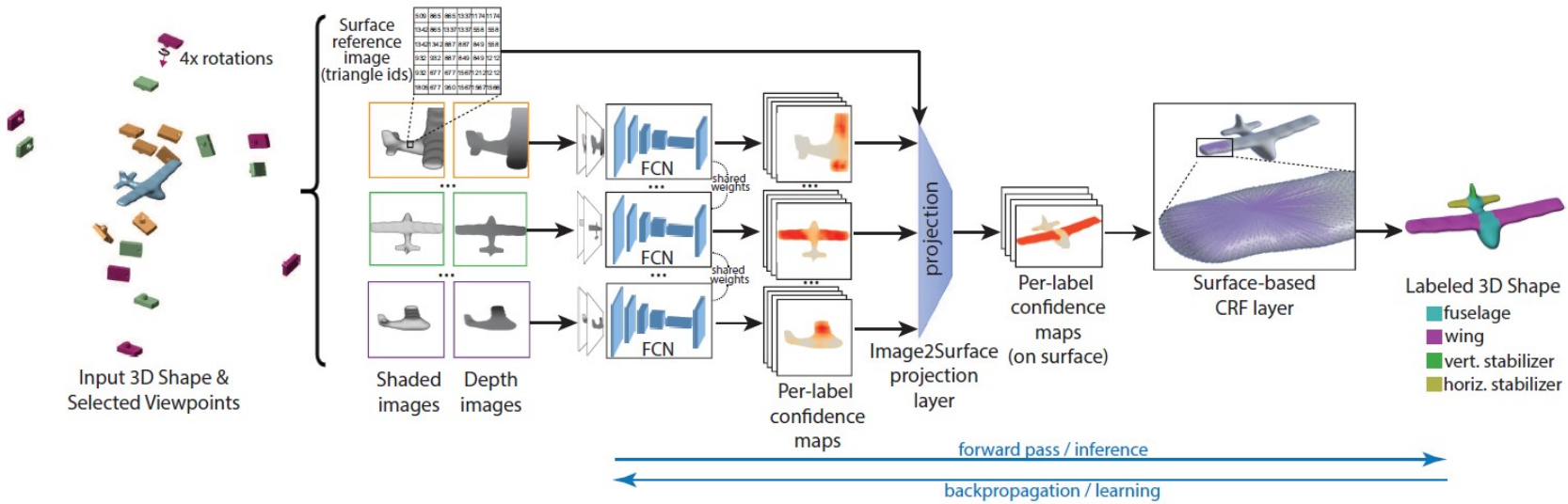


Figure 1. Pipeline and architecture of our method for 3D shape segmentation and labeling. Given an input shape, a set of viewpoints are computed at different scales such that the viewed shape surface is maximally covered (left). Shaded and depth images from these viewpoints are processed through our architecture (here we show images for three viewpoints, corresponding to 3 different scales). Our architecture employs image-based Fully Convolutional Network (FCN) modules with shared parameters to process the input images. The modules output image-based part label confidences per view. Here we show confidence maps for the wing label (the redder the color, the higher the confidence). The confidences are aggregated and projected on the shape surface through a special projection layer. Then they are further processed through a surface-based CRF that promotes consistent labeling of the entire surface (right).

[Kalogerakis et al., 2017]

Related approaches: Segmentation with multi-view CNNs

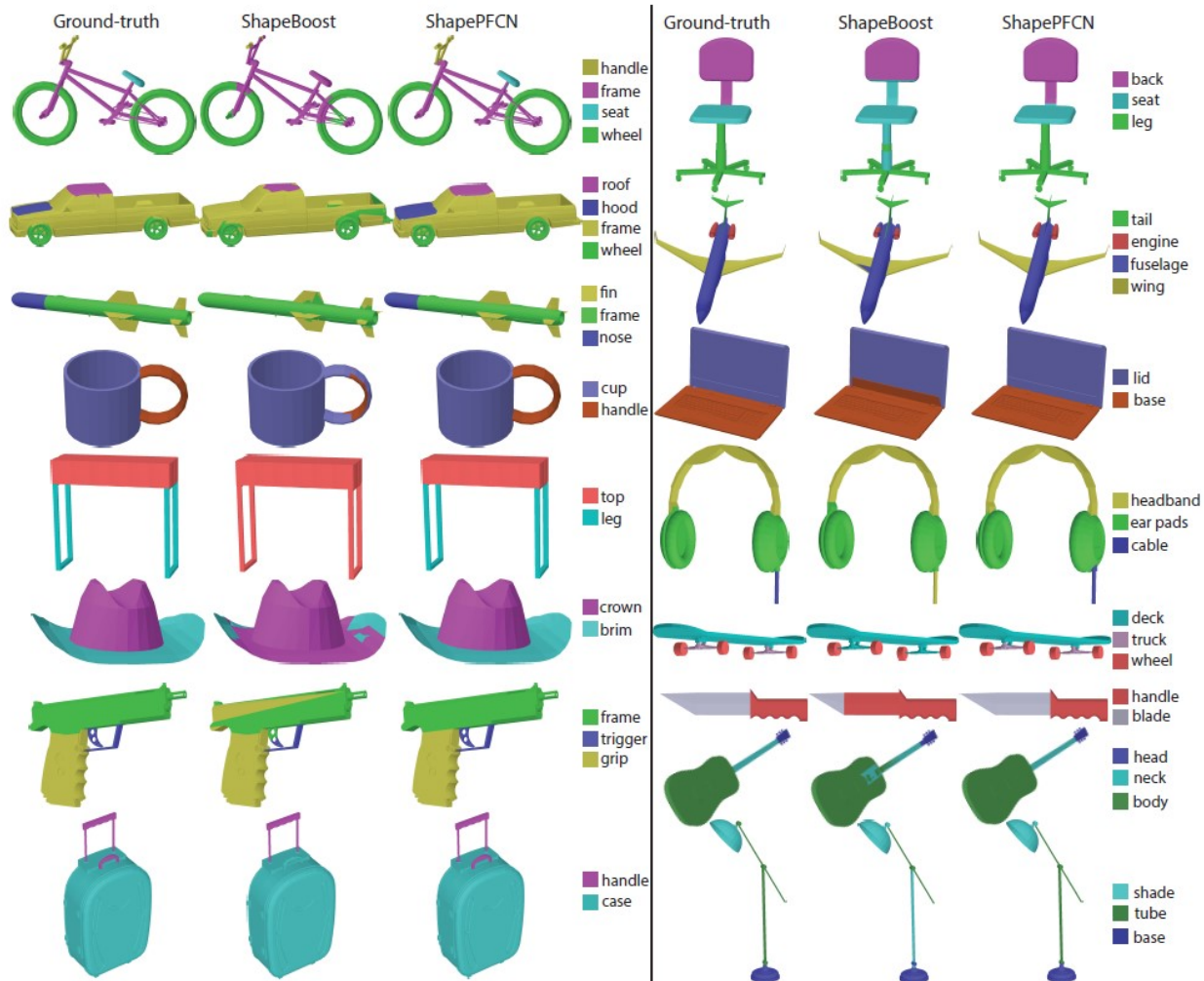


Figure 3. Ground-truth (human) labeled segmentations of ShapeNet shapes, along with segmentations produced by ShapeBoost [26] and our method (ShapePFCN) for test shapes originating from the ShapeNetCore dataset (best viewed in color).

[Kalogerakis et al., 2017]

Reconstruction with multi-view CNNs

Reconstruction of 3D shapes from 2D sketches in the form of line drawings ("ShapeMVD").

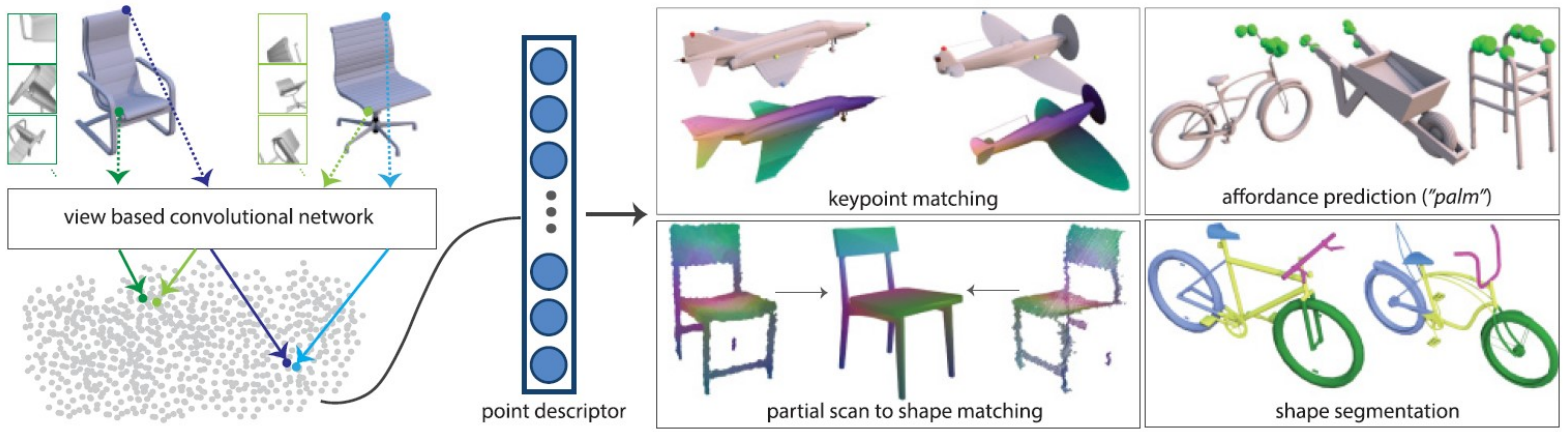


Fig. 1. We present a view-based convolutional network that produces local, point-based shape descriptors. The network is trained such that geometrically and semantically similar points across different 3D shapes are embedded close to each other in descriptor space (left). Our produced descriptors are quite generic—they can be used in a variety of shape analysis applications, including dense matching, prediction of human affordance regions, partial scan-to-shape matching, and shape segmentation (right).

[Lun et al., 2017]

Reconstruction with multi-view CNNs

Local descriptor for 3D shapes, directly applicable to a wide range of shape analysis problems.

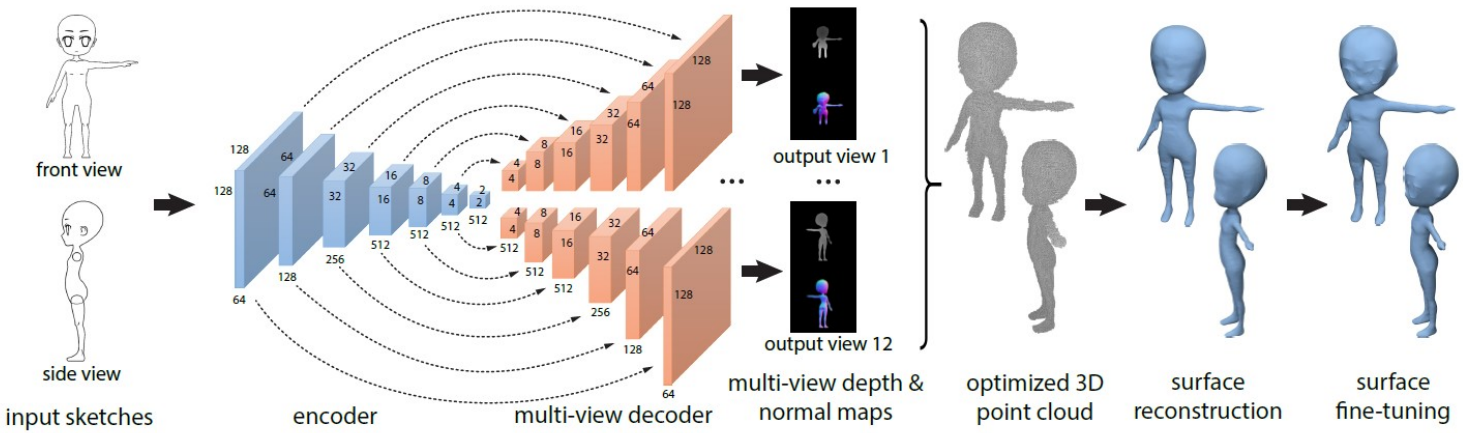
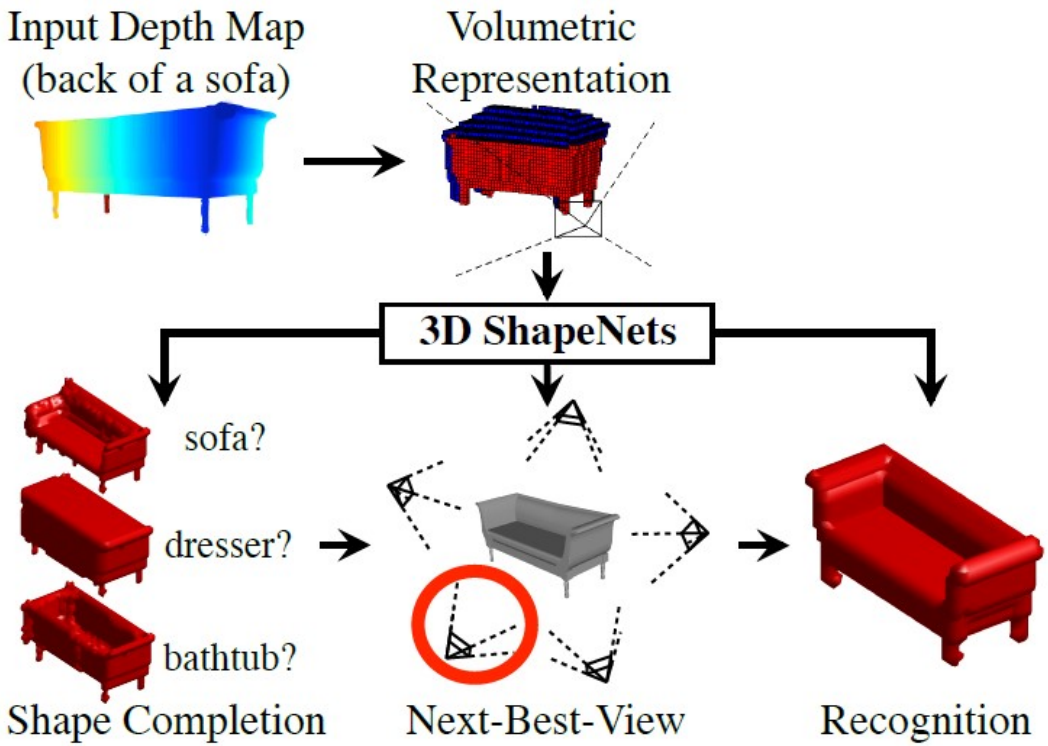


Figure 1. Our method takes line drawings as input and converts them into multi-view surface depth and normals maps from several output viewpoints via an encoder-multi-view-decoder architecture. The maps are fused into a coherent 3D point cloud, which is then converted into a surface mesh. Finally, the mesh can be further fine-tuned to match the input drawings more precisely through geometric deformations.

[Huang et al., 2017]

3D ShapeNets



http://3DShapeNets.cs.princeton.edu

Figure 1: **Usages of 3D ShapeNets.** Given a depth map of an object, we convert it into a volumetric representation and identify the observed surface, free space and occluded space. 3D ShapeNets can recognize object category, complete full 3D shape, and predict the next best view if the initial recognition is uncertain. Finally, 3D ShapeNets can integrate new views to recognize object jointly with all views.

[Wu et al., 2015]

VoxNet

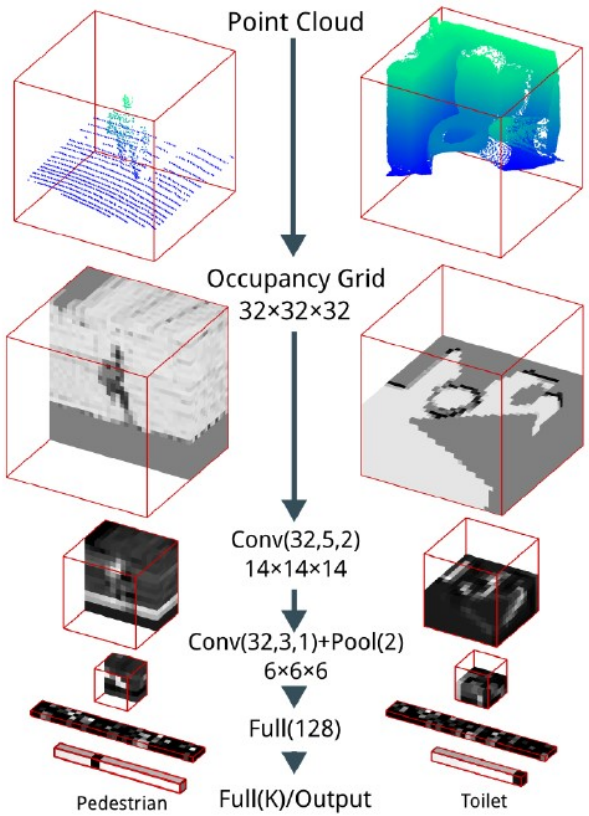


Fig. 1. The VoxNet Architecture. $Conv(f, d, s)$ indicates f filters of size d and at stride s, $Pool(m)$ indicates pooling with area m, and $Full(n)$ indicates fully connected layer with n outputs. We show inputs, example feature maps, and predicted outputs for two instances from our experiments. The point cloud on the left is from LiDAR and is part of the Sydney Urban Objects dataset [4]. The point cloud on the right is from RGBD and is part of NYUv2 [5]. We use cross sections for visualization purposes.

[Maturana & Scherer, 2015]