Deep Supervision with Intermediate Concepts

Chi Li, M. Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D. Hager, *Fellow, IEEE*, and Manmohan Chandraker

Abstract—Recent data-driven approaches to scene interpretation predominantly pose inference as an end-to-end black-box mapping, commonly performed by a Convolutional Neural Network (CNN). However, decades of work on perceptual organization in both human and machine vision suggest that there are often intermediate representations that are intrinsic to an inference task, and which provide essential structure to improve generalization. In this work, we explore an approach for injecting prior domain structure into neural network training by supervising hidden layers of a CNN with intermediate concepts that normally are not observed in practice. We formulate a probabilistic framework which formalizes these notions and predicts improved generalization via this deep supervision method. One advantage of this approach is that we are able to train only from synthetic CAD renderings of cluttered scenes, where concept values can be extracted, but apply the results to real images. Our implementation achieves the state-of-the-art performance of 2D/3D keypoint localization and image classification on real image benchmarks including KITTI, PASCAL VOC, PASCAL3D+, IKEA, and CIFAR100. We provide additional evidence that our approach outperforms alternative forms of supervision, such as multi-task networks.

Index Terms—Deep learning, multi-task learning, single image 3D structure prediction, object pose estimation

1 Introduction

Our visual world is rich in structural regularity. Studies in perception show that the human visual system imposes structure to reason about stimuli [1]. Consequently, early work in computer vision studied perceptual organization as a fundamental precept for recognition and reconstruction [2], [3]. However, algorithms designed on these principles relied on hand-crafted features (e.g., corners or edges) and hard-coded rules (e.g., junctions or parallelism) to hierarchically reason about abstract concepts such as shape [4], [5]. Such approaches suffered from limitations in the face of real-world complexities. In contrast, convolutional neural networks (CNNs), as end-to-end learning machines, ignore inherent perceptual structures encoded by task-related intermediate concepts and attempt to directly map from input to the label space.

Abu-Mostafa [6] proposes “hints” as a middle ground, where a task-related hint derived from prior domain knowledge regularizes the training of neural networks by either constraining the parameter space or generating more training data. In this work, we revisit and extend this idea by exploring a specific type of hint, which we refer to as an “intermediate concept”, that encodes a sub-goal to achieve the main task of interest. For instance, knowing object orientation is a prerequisite to correctly infer object part visibility which in turn constrains the 3D locations of semantic object parts. We present a generic learning architecture where intermediate concepts sequentially supervise hidden layers of a deep neural network to learn a specific inference sequence for predicting a final task.

We implement this deep supervision framework with a novel CNN architecture for predicting 2D and 3D object skeletons given a single test image. Our approach is in the spirit of [2], [3] that exploit object pose as an auxiliary shape concept to aid shape interpretation and mental rotation. We combine this early intuition with the discriminative power of modern CNNs by deeply supervising for multiple shape concepts such as object pose. As such, deep supervision teaches the CNN to sequentially model intermediate goals to parse 2D or 3D object skeletons across large intra-class appearance variations and occlusion.

An earlier version of this work has been presented in a conference paper [7]. In this extended version, we formalize a probabilistic notion of intermediate concepts that predicts improved generalization performance by deeply supervising intermediate concepts (Section 3). Further, we add new experiments including a new object class (bed) (Section 5.2.4) and image classification results on CIFAR100 [8] (Section 5.1). This motivates our network architecture in which we supervise convolutional layers at different depths with the available intermediate shape concepts.

Due to the scarcity of 3D annotated images, we render 3D CAD models to create synthetic images with concept labels as training data. In addition, we simulate challenging occlusion configurations between objects to enable robust data-driven occlusion reasoning (in contrast to earlier model-driven

- *C. Li is with the Computer Science, Johns Hopkins University, Baltimore, MD 21218. E-mail: chi_li@jhu.edu.*
- *M. Zeeshan Zia is with the Hololens, Microsoft, Redmond, WA. E-mail: zeeshan.zia@microsoft.com.*
- *Q.-H. Tran, X. Yu, and M. Chandraker are with the Media Analytics, NEC Laboratories America Inc, Cupertino, CA 95014. E-mail: {qhtran, xiangyu, manu}@nec-labs.com.*
- *G. D. Hager is with the Department of Computer Science, The Johns Hopkins University, Baltimore, MD 21218. E-mail: hager@cs.jhu.edu.*

Manuscript received 20 Dec. 2017; revised 18 July 2018; accepted 25 July 2018. Date of publication 12 Aug. 2018; date of current version 11 July 2019. (Corresponding author: Chi Li.)
Recommended for acceptance by A. Geiger.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2018.2863285

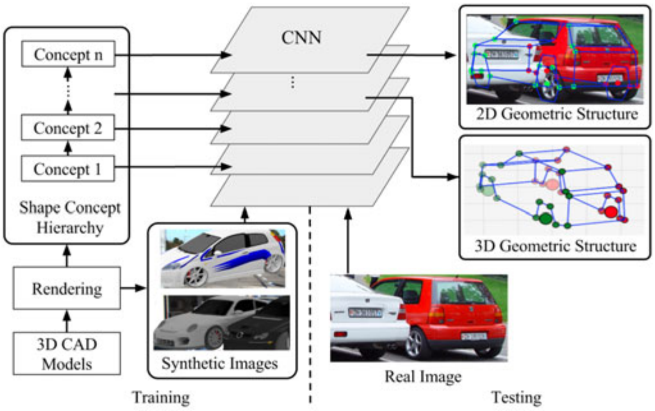


Fig. 1. Overview of our approach. We use synthetic training images with intermediate shape concepts to deeply supervise the hidden layers of a CNN. At test time, given a single real image of an object, we demonstrate accurate localization of semantic parts in 2D and 3D, while being robust to intra-class appearance variations and occlusions.

attempts [9], [10]). Fig. 1 introduces our framework and Fig. 4 illustrates an instance of a CNN deeply supervised by intermediate shape concepts for 2D/3D keypoint localization. We denote our network as "DISCO" short for Deep supervision with Intermediate Shape COncepts.

Most existing approaches [10], [11], [12], [13], [14] estimate 3D geometry by comparing projections of parameterized shape models with separately predicted 2D patterns, such as keypoint locations or heat maps. This makes prior methods sensitive to partial view ambiguity [15] and incorrect 2D structure prediction. Moreover, scarce 3D annotation of real image further limits their performance. In contrast, our method is trained on synthetic data only and generalizes well to real images. We find deep supervision with intermediate concepts to be a critical element to bridge the synthetic and real world. In particular, our deep supervision scheme empirically outperforms the single-task architecture, and multi-task networks which supervise all the concepts at the final layer. Further, we quantitatively demonstrate significant improvements over prior state-of-the-art for 2D/3D keypoint prediction on PASCAL VOC, PASCAL3D+ [16], IKEA [17] and KITTI-3D where we add 3D annotation for part of KITTI [18] data. These observations confirm that intermediate concepts regularize the learning of 3D shape in the absence of photorealism in rendered training data.

Additionally, we show another application of our generic deep supervision framework for image classification on CIFAR100 [8]. As such, coarse-grained class labels used as intermediate concepts are able to improve fine-grained recognition performance, which further validates our deep supervision strategy.

In summary, we make the following contributions in this work:

- We present a CNN architecture where its hidden layers are supervised by a sequence of intermediate shape concepts for the main task of 2D and 3D object geometry estimation.
- We formulate a probabilistic framework to explain why deep supervision may be effective in certain cases. Our proposed framework is a generalization of conventional supervision schemes employed in CNNs, including multi-task supervision and Deeply Supervised Nets [19].
- We show the utility of rendered data with access to intermediate shape concepts. We model occlusions by rendering multiple object configurations, which presents a novel route to exploiting 3D CAD data for parsing cluttered scenes.
- We empirically demonstrate state-of-the-art performance on 2D/3D semantic part localization and object classification on several public benchmarks. In some experiments, the proposed approach even outperforms the state-of-the-art methods trained on real images. We also demonstrate superior performance to baselines including the conventional multi-task supervision and different orders of intermediate concepts.

In the following, we review the related work in Section 2 and introduce the probabilistic framework and algorithm of deep supervision in Section 3. Details of network architecture and data simulation are discussed in Section 4. We discuss experiment results in Section 5 and conclude the paper in Section 6.

2 Related Work

We present a deep supervision scheme with intermediate concepts for deep neural networks. One application of our deep supervision is 3D object structure inference which is linked to recent advances including reconstruction, alignment and pose estimation. We review related work on these problems in the following:

Multi-task Learning. In neural networks, multi-task learning architectures exploit multiple task-related concepts to jointly supervise a network at the last layer. Caruana [20] empirically demonstrates its advantage over a single-task neural architecture on various learning problems. Recently, multi-task learning has been applied to a number of vision tasks including face landmark detection [21] and viewpoint estimation [22]. Hierarchy and Exclusion (HEX) graph [23] is proposed to capture hierarchical relationships among object attributes for improved image classification. In addition, some theories [24], [25] attempt to investigate how shared hidden layers reduce required training data by jointly learning multiple tasks. However, to our knowledge, no study has been conducted on quantifying the performance boost to a main task. It is also unclear whether a design choice meets the assumption of conducive task relationships used in these theories. This may explain that some task combinations for multi-task networks yield worse performance compared with single-task networks [20].

Deep Supervision. Deeply Supervised Nets (DSN) [19] uses a single task label to supervise the hidden layers of a CNN, speeding up convergence and addressing the vanishing gradient problem. However, DSN assumes that optimal local filters at shallow layers are building blocks for optimal global filters at deep layers, which is probably not true for a complex task. Recently a two-level supervision is proposed [26] for counting objects in binary images. One hidden layer is hard-coded to output object detection responses at fixed image locations. This work can be seen as a preliminary study to leverage task-related cues that assist the final task by deep supervision. We advance this idea further to a more general setting for deep learning without hard-coded internal representations.

3D Skeleton Estimation. Many works model 3D shape as a linear combination of shape bases and optimize basis coefficients to fit computed image evidence such as heat maps [14] and object part detections [10]. A prominent recent approach called single image 3D INterpreter Network (3D-INN) [27] is a sophisticated CNN architecture to estimate a 3D skeleton based only on detected visible 2D joints. However, in contrast to our approach, the training of 3D-INN does not jointly optimize for 2D and 3D keypoint localization. This decoupling of 3D structure from object appearance leads to partial view ambiguity and thus 3D prediction error.

3D Reconstruction. A generative inverse graphics model is formulated in [12] for 3D mesh reconstruction by matching mesh proposals to extracted 2D contours. Recently, given a single image, autoencoders have been exploited for 2D image rendering [28], multi-view mesh reconstruction [29] and 3D shape regression under occlusion [30]. The encoder network learns to invert the rendering process to recognize 3D attributes such as object pose. However, methods such as [29], [30] are quantitatively evaluated only on synthetic data and seem to achieve limited generalization to real images. Other works such as [11] formulate an energy-based optimization framework involving appearance, keypoint and normal consistency for dense 3D mesh reconstruction, but require both 2D keypoint and object segmentation annotations on real images for training. Volumetric frameworks using either discriminative [31] or generative [32] modeling infer a 3D shape distribution on voxel grids given image(s) of an object, limited to low-resolutions. Lastly, 3D voxel examplars [33] jointly recognize 3D shape and occlusion patterns by template matching, which is not scalable.

3D Model Retrieval and Alignment. This line of work estimates 3D object structure by retrieving the closest object CAD model and performing alignment, using 2D images [16], [34], [35] and RGB-D data [36], [37]. Unfortunately, a limited number of CAD models can not represent all instances in one object category. Further, the retrieval step is slow for a large CAD dataset and alignment is sensitive to error in estimated pose.

Pose Estimation and 2D Keypoint Detection. "Render for CNN" [22] renders 3D CAD models as additional training data besides real images for object viewpoint estimation. We extend this rendering pipeline to support object keypoint prediction and cluttered scene rendering to learn occlusions from data. Viewpoint prediction is utilized in [38] to boost the performance of 2D landmark localization. Recent work such as DDN [39] optimizes deformation coefficients based on the PCA representation of 2D keypoints to achieve state-of-the-art performance on face and human body. Dense feature matching approaches which exploit top-down object category knowledge [14], [40] are recent successes, but our method yields superior results while being able to transfer knowledge from rich CAD data.

Occlusion Modeling. Most work on occlusion invariant recognition relies on explicit occluder modeling [10], [41]. However, as it is hard to explicitly model object appearance, the variation in occluder appearance is also too broad to be captured effectively by model-driven approaches. This is why recent work has demonstrated gains by learning occlusions patterns from data [33], [42]. Thanks to deep supervision, which enables effective generalization from CAD renderings to real images, we are able to leverage a significantly larger array of synthetic occlusion configurations.

3 Deep Supervision with Intermediate Concepts

In this section, we introduce a novel CNN architecture with deep supervision. Our approach draws inspiration from Deeply Supervised Nets [19]. DSN supervises each layer by the main task label to accelerate training convergence. Our method differs from DSN in that we sequentially apply deep supervision on intermediate concepts intrinsic to the ultimate task, in order to regularize the network for better generalization. We employ this enhanced generalization ability to transfer knowledge from richly annotated synthetic data to the domain of real images.

Toy Example. To motivate the idea of supervising intermediate concepts, consider a very simple network with 2 layers: $y = \sigma(w_2\sigma(w_1 x + b_1) + b_2)$ where σ is ReLU activation $\sigma(x) = \max(x, 0)$. Provided that the true model for a phenomenon is $(w_1, w_2, b_1, b_2) = (3, 1, -2, -7)$ and the training data $\{(x, y)\}$ is $\{(1, 0), (2, 0), (3, 0)\}$. A learning algorithm may obtain a different model $(w_1, w_2, b_1, b_2) = (1, 3, -1, -10)$ which still achieves zero loss over training data but fails to generalize to the case when $x = 4$ or 5. However, if we have additional cues that tell us the value of intermediate layer activations, $\sigma(w_1 x + b_1)$ for each (x, y), we can achieve better generalization. For example, suppose we have training examples with an additional intermediate cue $\{(x, y', y)\} = \{(1, 0, 0), (2, 0, 0), (3, 1, 0)\}$ where $y' = \sigma(w_1 x + b_1)$. We find that the incorrect solution above that works for $\{x, y\}$ is removed because it does not agree with $\{x, y', y\}$. While simple, this example illustrates that deep supervision with intermediate concepts can regularize network training and reduce overfitting.

In the following, we formalize the notion of intermediate concept in Section 3.1, introduce our supervision approach which exploits intermediate concepts in Section 3.2, and discuss the improved generalization of deep supervision in Section 3.3.

3.1 Intermediate Concepts

We consider a supervised learning task to predict y_m from x. We have a training set $S = \{(x, (y_1, \ldots, y_m))\}$ sampled from an unknown distribution $\mathcal{D}$, where each training tuple consists of multiple task labels: $(y_1, \ldots, y_m)$. Without the loss of generality, we analyze the ith concept y_i in the following, where $1 < i \leq m$. Here, y_{i-k} is regarded as an intermediate concept to estimate y_i, where $k > 0$ and $i - k > 0$. Intuitively, knowledge of y_{i-k} constrains the solution space of y_i, as in our simple example above.

Formally, we define an intermediate concept y_{i-k} of y_i as a strict necessary condition such that there exists a deterministic function T which maps y_i to y_{i-k}: $y_{i-k} = T(y_i)$. In general, there is no inverse function T' that maps y_{i-k} to y_i because multiple y_i may map to the same y_{i-k}. In the context of multi-

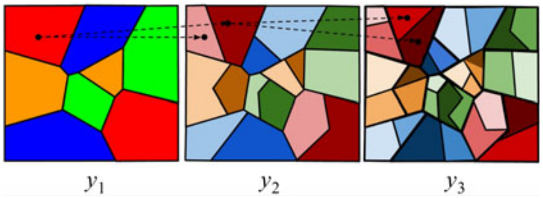


Fig. 2. Illustration of a concept hierarchy with three concepts $\mathcal{Y} = \{y_1, y_2, y_3\}$ on 2D input space. Black arrows indicate the finer decomposition within the previous concept in the hierarchy. Each color represents one individual class defined by the concept.

class classification where task y_i and y_{i-k} both contain discrete class labels, task y_i induces a finer partition over the input space $\mathcal{X} = \{x\}$ than task y_{i-k} by further partitioning each class in y_{i-k}. Fig. 2 illustrates a fictitious example of hierarchical partitioning over 2D input space created by three intermediate concepts $\{y_1, y_2, y_3\}$. As we can see in Fig. 2, a sequence of intermediate concepts hierarchically decompose the input space from coarse to fine granularity. Concretely, we denote a concept hierarchy as $\mathcal{Y} = (y_1, \ldots, y_m)$ where y_{i-k} is a strict necessary condition of y_i for all $i > 1$.

In many vision problems, we can find concepts that approximate a concept hierarchy $\mathcal{Y}$. As mentioned above, non-overlapping coarse-grained class labels constitute strict necessary conditions for a fine-grained classification task. In addition, object pose and keypoint visibility are both strict necessary conditions for 3D object keypoint location, because the former can be unambiguously determined by the latter.

3.2 Algorithm

Given a concept hierarchy $\mathcal{Y}$ and the corresponding training set S, we formulate a new deeply supervised architecture to jointly learn the main task along with its intermediate concepts. Consider a multi-layer convolutional neural network with N hidden layers that receives input x and outputs m predictions for $y_1, \ldots, y_m$. The ith concept y_i is applied to supervise the intermediate hidden layer at depth d_i by adding a side output branch at d_ith hidden layer. We denote the function represented by the kth hidden layer as $h_k(x, W_k)$, with parameters W_k. The output branch at depth d_i constructs a function $g_{d_i}(\cdot, V_{d_i})$ with parameters V_{d_i}. Further, we denote f_{y_i} as the function for predicting concept y_i such that $f_{y_i} = g_{d_i} \circ h_{d_i} \circ \cdots \circ h_1$. Fig. 3 shows a schematic diagram of our deep supervision framework. In Section 4, we concretely instantiate each h_k as a convolutional layer followed by batch normalization and ReLU layers and each g_k as global average pooling followed by fully connected layers. However, we emphasize that our algorithm is not limited to this particular layer configuration.

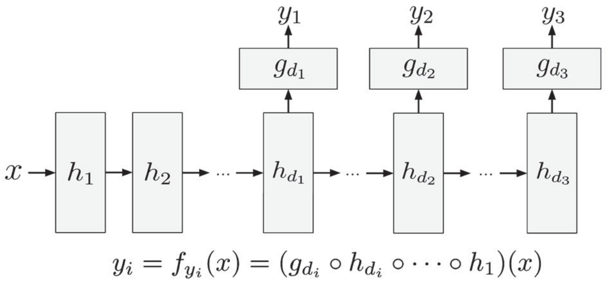


Fig. 3. Schematic diagram of deep supervision framework.

TABLE 1
Notation Table

Notation	Meaning
y_i	The ith concept
y_{i-k}	The intermediate concept of y_i
d_i	The supervision depth of y_i
f_{y_i}	A function that predicts y_i given input x
$R(f_{y_i})$	True risk of f_{y_i}
$R_S(f_{y_i})$	Empirical risk of f_{y_i} given a training set S
$\mathcal{H}_{y_i}$	A set of f_{y_i} with low empirical risk
$\mathcal{F}_{y_i}$	A set of f_{y_i} with low empirical and true risk
P_{y_i}	Generalization probability of y_i
$\mathcal{H}_{y_i \mid y_{i-k}}$	Subset of $\mathcal{H}_{y_i}$ that achieves low empirical risk on y_{i-k}
$\mathcal{F}_{y_i \mid y_{i-k}}$	Subset of $\mathcal{F}_{y_i}$ that achieves low empirical risk on y_{i-k}
$\mathrm{P}_{y_i \mid y_{i-k}}$	Generalization probability y_i constrained by y_{i-k}

We formulate the following objective function to encapsulate these ideas:

$$W^*, V^* = \underset{W,V}{\operatorname{argmin}} \sum_{(x,\{y_i\}) \in S} \sum_{i=1}^{m} \lambda_i l_i(y_i, f_{y_i}(x\ ; W_{1:d_i}, V_{d_i})), \quad (1)$$

where $W_{1:d_i} = \{W_1, \ldots, W_{d_i}\}$, $W = W_{1:d_m}$ and $V = \{V_{d_1}, \ldots, V_{d_m}\}$. In addition, l_i is the loss for task y_i scaled by the loss weight λ_i. We optimize Equation (1) over S by simultaneously backpropagating the loss of each supervisory signal all the way back to the first layer.

We note that Equation (1) is a generic supervision framework which represents many existing supervision schemes. For example, the standard CNN with a single task supervision is a special case when $m = 1$. Additionally, the multi-task learning [20] places all supervision on the last hidden layer: $d_i = N$ for all i. DSN [19] framework is obtained when $m = N$ and $y_i = y_m$ for all i. In this work, we propose to apply m different concepts $\{y_i\}$ in a concept hierarchy $\mathcal{Y}$ at locations with growing depths: $d_{i-k} < d_i$ where $k > 0$ and $i - k > 0$.

3.3 Generalization Analysis

In this section, we present a generalization metric and subsequently show how deep supervision with intermediate concepts can improve the generalization of a deep neural network with respect to this metric, compared to other standard supervision methods. We also discuss the limitations of this analysis. For clarity, we summarize our notation in Table 1.

3.3.1 Generalization Metric

Deep neural networks are function approximators that learn mappings from an input space x to an output space y. For a network with a fixed structure, there usually exists a set of functions $\mathcal{H}$ (equivalently a set of parameters) where each element $f \in \mathcal{H}$ achieves a low empirical loss on a training set S. In the following, we define a generalization metric to measure the probability that a function $f \in \mathcal{H}$ is a "true" solution for a supervised learning task.

Recall that f_{y_i} represents the function composed by the first d_i hidden layers and an output branch for predicting concept y_i. The true risk $R(f_{y_i})$ is defined based on random variables x and y_i where $(x, y_i) \sim D$

$$R(f_{y_i}) = \mathbb{E}\ [l_i(f_{y_i}(x), y_i)\ ]. \tag{2}$$

Given a training set S, the empirical risk $R_S(f_{y_i})$ of f_{y_i} is

$$R_S(f_{y_i}) = \frac{1}{|S|} \sum_{(x,y_i)\in S} l_i(f_{y_i}(x), y_i). \tag{3}$$

Given limited training data S, a deep neural network is optimized to find a solution f_{y_i} with low empirical loss. We consider empirical loss to be "low" when $R_S(f_{y_i}) < \delta$. δ is the risk threshold which indicates "good" performance for a task. Next, we define the function set $\mathcal{H}_{y_i}$ in which each function achieves low empirical risk

$$\mathcal{H}_{y_i} = \{f_{y_i}\,|\,R_S(f_{y_i})\ <\ \delta\}. \tag{4}$$

Similarly, we also define the function set $\mathcal{F}_{y_i}$ where each function achieves risks less than δ for both $R(f_{y_i})$ and $R_S(f_{y_i})$

$$\mathcal{F}_{y_i} = \{f_{y_i}\,|\,R_S(f_{y_i})\ <\ \delta\ \wedge\ R(f_{y_i})\ <\ \delta\}. \tag{5}$$

By definition, we know $\mathcal{F}_{y_i} \subseteq \mathcal{H}_{y_i}$. Given a training set and network structure, the generalization capability of the outcome of network training depends upon the likelihood that $f_{y_i} \in \mathcal{H}_{y_i}$ is also a member of $\mathcal{F}_{y_i}$.

We consider f_{y_i} to be a random variable as it is the outcome of a stochastic optimization process such as stochastic gradient descent. We assume that the optimization algorithm is unbiased within $\mathcal{H}_{y_i}$, such that apriori probability of converging to any $f_{y_i} \in \mathcal{H}_{y_i}$ is uniformly distributed. We formalize a generalization metric for a CNN for predicting y_i by defining a probability measure P_{y_i} based on the function sets $\mathcal{F}_{y_i}$ and $\mathcal{H}_{y_i}$

$$\begin{aligned}\mathrm{P}_{y_i} &= \mathrm{P}(R(f_{y_i})\ <\ \delta\,|\,R_S(f_{y_i})\ <\ \delta)\\ &= \begin{cases} \frac{\mu(\mathcal{F}_{y_i})}{\mu(\mathcal{H}_{y_i})} & : \mathcal{H}_{y_i} \neq \emptyset\\ 0 & : \mathcal{H}_{y_i} = \emptyset,\end{cases}\end{aligned} \tag{6}$$

where $\mu(A)$ is the Lebesgue measure [43] of set A indicating the "volume" or "size" of set A.[1] Moreover, $\mu(\mathcal{F}_{y_i}) \leq \mu(\mathcal{H}_{y_i})$ due to $\mathcal{F}_{y_i} \subseteq \mathcal{H}_{y_i}$. The equality $\mu(\mathcal{F}_{y_i}) = \mu(\mathcal{H}_{y_i})$ is achieved when $\mathcal{F}_{y_i} = \mathcal{H}_{y_i}$. It follows that the higher the P_{y_i}, the better the generalization.

When an intermediate concept y_{i-k} of y_i is available, we insert one output branch $g_{d_{i-k}}$ at depth d_{i-k} of CNN to predict y_{i-k}. Then, our deep supervision algorithm in Section 3.2 aims to minimize empirical risk on both y_{i-k} and y_i. Recall that $f_{y_i} = g_{d_i} \circ f_{d_i} \circ \cdots \circ f_1$. As a consequence, f_{y_i} does not contain any output branch $g_{d_{i-k}}$ for the intermediate concept y_{i-k}. However, we note that f_{y_i} shares some hidden layers with $f_{y_{i-k}}$. Similar to P_{y_i}, we can define the generalization probability $\mathrm{P}_{y_i\,|\,y_{i-k}}$ of f_{y_i} given the supervision of its intermediate concept y_{i-k}

$$\begin{aligned}\mathrm{P}_{y_i\,|\,y_{i-k}} &= \mathrm{P}(R(f_{y_i})\ <\ \delta\,|\,R_S(f_{y_i})\ <\ \delta, R_S(f_{y_{i-k}})\ <\ \delta')\\ &= \begin{cases} \frac{\mu(\mathcal{F}_{y_i\,|\,y_{i-k}})}{\mu(\mathcal{H}_{y_i\,|\,y_{i-k}})} & : \mathcal{H}_{y_i\,|\,y_{i-k}} \neq \emptyset\\ 0 & : \mathcal{H}_{y_i\,|\,y_{i-k}} = \emptyset,\end{cases}\end{aligned} \tag{7}$$

where the function set $\mathcal{H}_{y_i\,|\,y_{i-k}}$ is a subset of $\mathcal{H}_{y_i}$

$$\mathcal{H}_{y_i\,|\,y_{i-k}} = \{f_{y_i}|R_S(f_{y_i})\ <\ \delta\ \wedge\ R_S(f_{y_{i-k}})\ <\ \delta'\}, \tag{8}$$

and the function set $\mathcal{F}_{y_i\,|\,y_{i-k}}$ is a subset of $\mathcal{F}_{y_i}$

$$\mathcal{F}_{y_i\,|\,y_{i-k}} = \{f_{y_i}|R(f_{y_i})\ <\ \delta \wedge R_S(f_{y_i})\ <\ \delta \wedge R_S(f_{y_{i-k}})\ <\ \delta'\}. \tag{9}$$

Note that we use a different threshold δ' for $R_S(f_{y_{i-k}})$ in order to account for the difference between loss functions l_{i-k} and l_i. We do not require the true risk of intermediate concept $R(y_{i-k})$ to be lower than δ' because the objective is to analyze the achievable generalization with respect to predicting y_i.

3.3.2 Improved Generalization through Deep Supervision

A machine learning model for predicting y_i suffers from overfitting when the solution f_{y_i} achieves low empirical risk $R_S(f_{y_i})$ over S but high true risk $R(f_{y_i})$. In other words, the higher the probability P_{y_i}, the lower the chance that the trained model f_{y_i} overfits S. One general strategy to reduce the overfitting is to increase the diversity and size of training set S. In this case, the denominator $\mu(\mathcal{H}_{y_i})$ of Equation (6) decreases because fewer functions achieve low loss on more diverse data. In the following, we show that supervising an intermediate concept y_{i-k} of y_i at some hidden layer is similarly capable of removing some incorrect solutions in $\mathcal{H}_{y_i} \setminus \mathcal{F}_{y_i}$ and thus improves the generalization because $\mathrm{P}_{y_i\,|\,y_{i-k}} \geq \mathrm{P}_{y_i}$.

First, given an intermediate concept y_{i-k} of y_i where $y_{i-k} = T(y_i)$, we specify the following assumptions for our analysis.

(1) The neural network underlying our analysis is large enough to satisfy the universal approximation theorem [44] for the concepts of interest, that is, its hidden layers have sufficient learning capacity to approximate arbitrary functions.

(2) For a concept hierarchy $\mathcal{Y} = \{y_1, \ldots, y_m\}$, if y_i' is a reasonable estimate of y_i, then $T(y_i')$ should also be a reasonable estimate of the corresponding intermediate concept y_{i-k}. Formally, we assume

$$\forall y_i, y_i' \in Q_i:\quad l_i(y_i, y_i') \leq \delta \Rightarrow l_{i-k}(T(y_i), T(y_i')) \leq \delta', \tag{10}$$

where Q_i is the value space of concept y_i.

(3) Based on Assumption 1 and 2, it follows that if $f_{y_i} \in \mathcal{F}_{y_i}$, there exists a $d_{i-k} < d_i$ such that the first d_{i-k} layers of f_{y_i} can be used to construct a $f_{y_{i-k}} \in \mathcal{F}_{y_{i-k}}$.

In practice, one may identify many tasks and relevant intermediate concepts satisfying Assumption 2 when using common loss functions and $\delta = \delta'$. We discuss this further in Section 3.3.3. To obtain Assumption 3 above, we take the following two steps. First, with $k > 0$, Assumption 1 allows

[1] Each function f_{y_i} has a one-to-one mapping to a parameter W in $\mathbb{R}^n$ where n is the dimension of the parameter. We know that any subset of $\mathbb{R}^n$ is Lebesgue measurable.

us to find a $g_{d_{i-k}} = T \circ g_{d_i} \circ h_{d_i} \circ \cdots h_{d_{i-k+1}}$. As a consequence, we can always construct a $f_{y_{i-k}}$ from f_{y_i} through T using the first d_{i-k} layers: $T \circ f_{y_i} = T \circ g_{d_i} \circ h_{d_i} \circ \cdots \circ h_1 = g_{d_{i-k}} \circ h_{d_{i-k}} \circ \cdots \circ h_1 = f_{y_{i-k}}$. Second, Assumption 2 further extends that for any $f_{y_i} \in \mathcal{F}_{y_i}$, its first d_{i-k} layers can be used to obtain a $f_{y_{i-k}} \in \mathcal{F}_{y_{i-k}}$.

Given an intermediate concept y_{i-k} that satisfies the above assumptions, the following two propositions discuss how d_{i-k} (the supervision depth of y_{i-k}) affects the generalization ability of y_i in terms of $\mathrm{P}_{y_i \,|\, y_{i-k}}$. First, we show that supervising intermediate concepts in the wrong order has no effect on improving generalization.

Proposition 1. *If $d_{i-k} \geq d_i$, the generalization performance of y_i is not guaranteed to improve*

$$\forall d_{i-k} \geq d_i, \quad \mathrm{P}_{y_i \,|\, y_{i-k}} = \mathrm{P}_{y_i}. \tag{11}$$

Proof. We first consider the case when y_i and y_{i-k} both supervise the same hidden layer: $d_i = d_{i-k}$. Given a sample set $(x, y_{i-k}, y_i) \sim \mathcal{D}$ and a function f_{y_i} which correctly predicts y_i for x: $y_i = f_{y_i}(x)$, we can construct $f_{y_{i-k}} = T \circ f_{y_i}$ to yield the correct prediction for y_{i-k}. Based on Assumption 1, a multi-layer perceptron (i.e., fully connected layers) is able to represent any mapping function T. Therefore, to approximate $f_{y_{i-k}} = T \circ f_{y_i}$, we can append fully connected layers which implement T to g_{d_i}: $g_{d_{i-k}} = T \circ g_{d_i}$. Based on Assumption 2, for any function f_{y_i} in $\mathcal{F}_{y_i}$, there exists a corresponding function $f_{y_{i-k}} = T \circ f_{y_i}$ which satisfies $R_S(f_{y_{i-k}}) \leq \delta'$. This indicates that $\mathcal{H}_{y_i \,|\, y_{i-k}} = \mathcal{H}_{y_i}$ which in turn implies $\mathcal{F}_{y_i \,|\, y_{i-k}} = \mathcal{F}_{y_i}$. When $d_{i-k} > d_i$, hidden layers from d_i to d_{i-k} can be implemented to achieve an identity mapping and then follow the same analysis for the case $d_i = d_{i-k}$. As a consequence, Proposition 1 holds. □

Proposition 2. *There exists a d_{i-k} such that $d_{i-k} < d_i$ and the generalization performance of y_i is improved*

$$\exists d_{i-k} < d_i, \quad \mathrm{P}_{y_i \,|\, y_{i-k}} \geq \mathrm{P}_{y_i}. \tag{12}$$

Proof. From Equations (4) and (8), we observe that $\mathcal{H}_{y_i \,|\, y_{i-k}} \subset \mathcal{H}_{y_i}$ and $\mu(\mathcal{H}_{y_i \,|\, y_{i-k}}) < \mu(\mathcal{H}_{y_{i-k}})$. Thus, we obtain

$$\mu(\mathcal{H}_{y_i \,|\, y_{i-k}}) \leq \min(\mu(\mathcal{H}_{y_i}), \mu(\mathcal{H}_{y_{i-k}})). \tag{13}$$

Given a training set S, Equation (13) essentially means that the number of functions that simultaneously fit both y_i and y_{i-k} is not more than the number of functions that fit each of them individually. Intuitively, as the toy example earlier, the hidden layers of some network solutions for y_i yield incorrect predictions of the intermediate concept y_{i-k}. This implies that $\mu(\mathcal{H}_{y_i \,|\, y_{i-k}}) \ll \min(\mu(\mathcal{H}_{y_i}), \mu(\mathcal{H}_{y_{i-k}}))$ in practice. Subsequently, Assumption 3 suggest that there exists one or multiple d_{i-k}'s such that the first d_{i-k} layers of each solution $f_{y_i} \in \mathcal{F}_{y_i}$ are contained in $f_{y_{i-k}} \in \mathcal{F}_{y_{i-k}}$. In other words, we can find a supervision depth d_{i-k} for y_{i-k} which satisfies

$$\exists\ d_{i-k} < d_i, \quad \mu(\mathcal{F}_{y_i}) = \mu(\mathcal{F}_{y_i|y_{i-k}}). \tag{14}$$

As a result, Proposition 2 is proved by Equations (13) and (14). □

To this end, we can improve the generalization of y_i via y_{i-k} by inserting the supervision of y_{i-k} before y_i. As a consequence, given a concept hierarchy $\mathcal{Y}_0 = (y_1, \ldots, y_m)$, the supervision depths of concepts $\{d_1, \ldots, d_m\}$ should be monotonically increasing: $1 \leq d_1 < \cdots < d_m$. We then extend Equation (13) to incorporate all available intermediate concepts of y_m

$$\mu(\mathcal{H}_{y_m \,|\, y_{m-1},\ldots,y_1}) \leq \min_{y_i} \mu(\mathcal{H}_{y_i}) \;\; s.t. \;\; \forall i < j, d_i < d_j. \tag{15}$$

As we report in Section 5, the empirical evidence shows that more intermediate concepts often greatly improves the generalization performance of the main task, which implies a large gap between two sides of Equation (15). Similar to Equation (14), we still have

$$\exists\ d_1 < \cdots < d_m, \quad \mu(\mathcal{F}_{y_m}) = \mu(\mathcal{F}_{y_m|y_{m-1},\ldots,y_1}). \tag{16}$$

As a consequence, the generalization performance of y_m given its necessary conditions $y_1, \ldots, y_{m-1}$ can be improved if we supervise each of them at appropriate depths $d_1, \ldots, d_{m-1}$ where $d_1 < \cdots < d_{m-1} < d_m$

$$\exists d_1 < \cdots < d_m, \quad \mathrm{P}_{y_m \,|\, y_{m-1},\ldots,y_1} \geq \mathrm{P}_{y_m}. \tag{17}$$

Furthermore, $\mathrm{P}_{y_m \,|\, y_{m-1},\ldots,y_1}$ is monotonically decreasing by removing intermediate concepts: $\mathrm{P}_{y_m \,|\, y_{m-1},\ldots,y_1} \geq \mathrm{P}_{y_m \,|\, y_{m-2},\ldots,y_1} \geq \cdots \geq \mathrm{P}_{y_m \,|\, y_1} \geq \mathrm{P}_{y_m}$. The more concepts applied, the better chance that the generalization is improved. In conclusion, deep supervision with intermediate concepts regularizes the network training by decreasing the number of incorrect solutions that generalize poorly to the test set.

3.3.3 Discussion

Generalization of Intermediate Concept. We generalize the notion of intermediate concept, using conditional probabilities, with y_{i-k} being the ϵ-error necessary condition of y_i if y_{i-k} and y_i for any sample $(x, (y_{i-k}, y_i)) \sim \mathcal{D}$ satisfy

$$\forall c, \quad \max \mathrm{P}(y_{i-k} \,|\, y_i = c) \geq 1 - \epsilon, \tag{18}$$

where $0 \leq \epsilon \leq 1$. The strict necessary condition defined in Section 3.1 holds when $\epsilon = 0$. When $\epsilon > 0$, the monotonically increasing supervision order indicated by Equation (17) is no longer ensured. However, the architecture design suggested by our generalization analysis in Section 3.3.2 achieves the best performance in our empirical studies in Section 5. We believe that the generalization analysis in Section 3.3.2 is a good approximation for case with small ϵ in real applications. We leave the analytic quantification of how ϵ affects deep supervision to future work.

Assumption 2. If Assumption 2 does not hold, both the numerator and denominator in Equation (7) decrease by different amounts. As a consequence, we cannot obtain Proposition 1 for all cases. However, many commonly used loss functions satisfy this assumption when $\delta = \delta'$. One simple example is when l_i and l_{i-k} are indicator functions (i.e., $l_i(y, y') = \mathbf{1}(y = y')$) for all i.[2] As such, $l_i(y, y') = l_{i-k}(T(y), T(y'))$ when $\epsilon = 0$ and thus Assumption 2 is satisfied. Another

[2] Note that the indicator function can be applied to discrete and continuous values of y and y'.

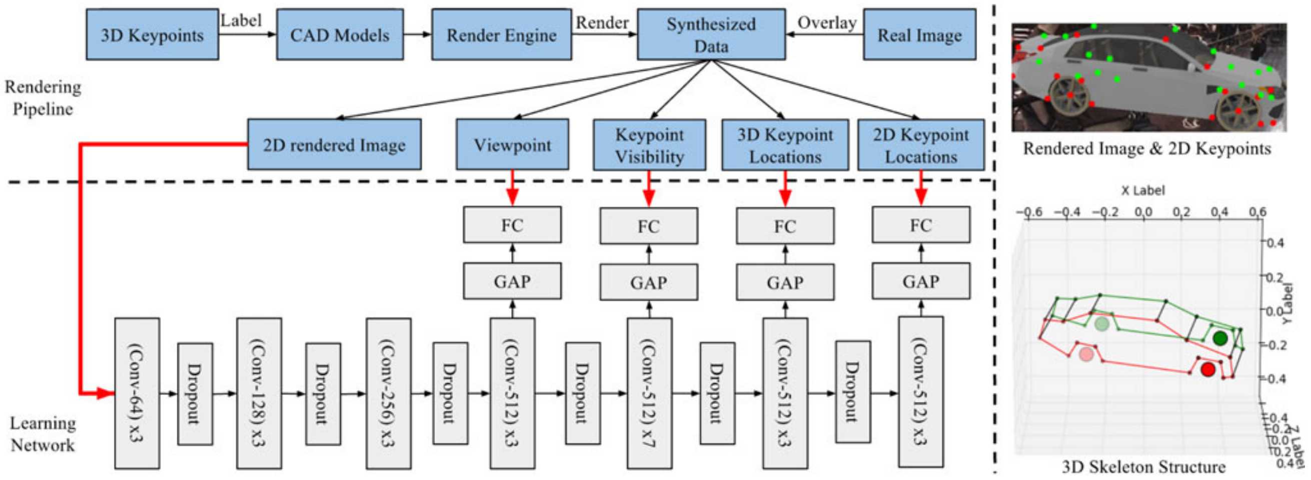


Fig. 4. Visualization of our rendering pipeline (top-left), DISCO network (bottom-left), an example of rendered image and its annotations of 2D keypoints (top-right) as well as 3D skeleton (bottom-right).

example can be that l_i and l_{i-k} are both L2 loss (i.e., $l_i(y, y') = \|y = y'\|^2$) and T is a projection function where $T(y) = Py$ and P is a projection (i.e., $P^2 = P$). In this case, $l_i(y, y') = \|y - y'\|^2 \geq \|P(y - y')\|^2 = l_{i-k}(T(y), T(y'))$.

Uniform Probability of $f_{y_i} \in \mathcal{H}_{y_i}$. In practice, this assumption may seem to contradict some empirical studies like [45] where common CNNs generalize well after overfitting to large-scale training data (e.g., Imagenet [46]). This phenomenon actually demonstrates another dimension of improving generalization: training models on a large training set S so that $\mathcal{H}_{y_i}$ is shrinking and converging to $\mathcal{F}_{y_i}$. Our work results shows that with deep supervision is an alternative route to achieve generalization given limited training data or data from a different domain, compared with standard supervision methods.

DSN as a Special Case. Since a task is also a necessary condition of itself, our deep supervision framework actually contains DSN[19] as a special case where each intermediate concept y_i is the main task itself. To illustrate the distinction enabled by our framework, we mimic DSN by setting the first intermediate concept $y_1 = y_m$. Thus, the first d_1 hidden layers are forced to directly predict y_m. Each $f_{d_1} \in \mathcal{F}_{y_1}$ can be trivially used to construct $f_{d_m} \in \mathcal{F}_{y_m}$ by forcing an identity function for layers d_1 to d_m. This suggests that $\mathcal{F}_{y_m}$ is mainly constrained by $\mathcal{F}_{y_1}$. Therefore, even though larger spatial supports from deeper layers between d_1 and d_m reduce empirical risk in DSN, the learning capacity is restricted by supervision for y_m at the first d_1 layers.

4 Implementation and Data

We apply our method to both object classification and key point localization. For object classification, we use the semantic hierarchy of labels to define intermediate concepts. For example, container is an intermediate concept (a generalization) of cup. For key point localization, we specify a 3D skeleton for each object class where nodes or keypoints represent semantic parts, and their connections define 3D object geometry. Given a single real RGB image of an object, our goal is to predict the keypoint locations in image coordinates as well as normalized 3D coordinates while inferring their visibility states. X and Y coordinates of 2D keypoint locations are normalized to $[0, 1]$ along the image width and height, respectively. 3D keypoint coordinates are centered at origin and scaled to set the longest dimension along X, Y, Z to unit length. Note that 2D/3D keypoint locations and their visibility all depend on the specific object pose with respect to the camera viewpoint.

To set up the concept hierarchy for 2D/3D keypoint localization, we have chosen in order, object orientation y_1, which is needed to predict keypoint visibility y_2, which roughly depicts the 3D structure prediction y_3, which finally leads to 2D keypoint locations y_4 including ones that are not visible in the current viewpoint. We impose the supervision of the concept hierarchy $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$ into a CNN as shown in Fig. 4 and minimize Equation (1) to compute the network parameters.

We emphasize that the above $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$ is not a 0-error concept hierarchy because object pose (y_1), and 3D keypoint location (y_3) are not strict necessary conditions for visibility (y_2), and 2D keypoint location (y_4), respectively. However, we posit that the corresponding residuals (ϵ's) of $\mathcal{Y}$ are small. First, knowing object pose constrains keypoint visibilities to such an extent, that prior work has chosen to use ensembles of 2D templates for visual object parsing [42], [47]. Second, there is a long and fruitful tradition in computer vision, starting from Marr's seminal ideas [3] to leverage 3D object representations as a tool for 2D recognition. In sum, our present choice of $\mathcal{Y}$ is an approximate realization of a 0-error concept hierarchy which nonetheless draws inspiration from our analysis, and works well in practice.

4.1 Network Architecture

In this section, we detail the network structure for keypoint localization. Our network resembles the VGG network [48] and consists of deeply stacked 3×3 convolutional layers. Unlike VGG, we remove local spatial pooling between convolutional layers. This is motivated by the intuition that spatial pooling leads to the loss of spatial information. Further, we couple each convolutional layer with batch normalization [49] and ReLU, which defines $h_{d_i}(x, W_{d_i})$. The output layer $g_{d_i}(\cdot, V_{d_i})$ at depth d_i for task y_i is constructed with one global average pooling (GAP) layer followed by one fully connected (FC) layer with 512 neurons, which is different from stacked FC layers in VGG. The GAP layer averages

TABLE 2
Classification Error of Different Methods on CIFAR100

Methods	Error(%)
DSN [19]	34.57
FitNet, LSUV [52]	27.66
ResNet-1001 [53]	27.82
pre-act ResNet-1001 [54]	22.71
plain-single	23.31
plain-all	23.26
DISCO-random	27.53
DISCO	**22.46**

The first four are previous methods and "pre-act ResNet-1001" is the current state-of-the-art. The remaining four are results of DISCO and its variants.

filter responses over all spatial locations within the feature map. From Table 3 in Section 5.2.1, we empirically show that these two changes are critical to significantly improve the performance of VGG-like networks for 2D/3D landmark localization.

We follow the common practice of employing dropout [50] layers between the convolutional layers, as an additional means of regularization. At layers 4,8,12, we perform the downsampling using convolution layers with stride 2. The bottom-left of Fig. 4 illustrates the details of our network architecture. "(Conv-A)xB" means A stacked convolutional layers with filters of size BxB. We deploy 25 convolutional layers in total.

We use L2 loss at all points of supervision. In practice, we only consider the azimuth angle of the object viewpoint with respect to a canonical pose. We further discretize the azimuth angle into K bins and regress it to a one-hot encoding (the entry corresponding to the predicted discretized pose is set to 1 and all others to 0). Keypoint visibility is also represented by a binary vector with 1 indicating occluded state of a keypoint. During training, each loss is backpropagated to train the network jointly.

4.2 Synthetic Data Generation

Our approach needs a large amount of training data because it is based on deep CNNs. It also requests finer grained labels than many visual tasks such as object detection. Furthermore, we aim for the method to work for heavily cluttered scenes. Therefore, we generate synthetic images that simulate realistic occlusion configurations involving multiple objects in close proximity. To our knowledge, rendering cluttered scenes that comprise of multiple CAD models is a novelty of our approach, although earlier work [33], [42] used real image cut-outs for bounding box level localization.

An overview of the rendering process is shown in the upper-left of Fig. 4. We pick a small subset of CAD models from ShapeNet [51] for a given object category and manually annotate 3D keypoints on each CAD model. Next, we render each CAD model via Blender with randomly sampled graphics parameters including camera viewpoint, number/strength of light sources, and surface gloss reflection. Finally, we follow [22] to overlay the rendered images on real backgrounds to avoid over-fitting. We crop the object from each rendered image and extract the object viewpoint, 2D/3D keypoint locations and their visibility states from Blender as the training labels. In Fig. 4 (right), we show an example of rendering and its 2D/3D annotations.

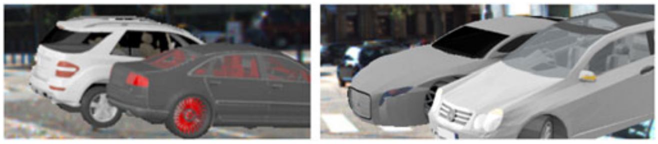


Fig. 5. Examples of synthesized training images for simulating the multi-car occlusion.

To model multi-object occlusion, we randomly select two different object instances and place them close to each other without overlapping in 3D space. During rendering, we compute the occlusion ratio of each instance by calculating the fraction of visible 2D area versus the complete 2D projection of CAD model. Keypoint visibility is computed by ray-tracing. We select instances with occlusion ratios ranging from 0.4 to 0.9. Fig. 5 shows two training examples where cars are occluded by other nearby cars. For truncation, we randomly select two image boundaries (left, right, top, or bottom) of the object and shift them by $[0, 0.3]$ of the image size along that dimension.

5 Experiments

We first present an empirical study of image classification problem on CIFAR100 [8] where a strict concept hierarchy is applied to boost the fine-grained object classification performance. Subsequently, we extensively demonstrate competitive or superior performance for 2D/3D keypoint localization over several state-of-the-art methods, on multiple datasets: KITTI-3D, PASCAL VOC, PASCAL3D+ [16] and IKEA [17].

5.1 CIFAR100

The image classification problem has a natural concept hierarchy where object categories can be progressively partitioned from coarse to fine granularity. In this section, we exploit coarse-grained class labels (20-classes) from CIFAR100 [8] to assist fine-grained recognition into 100 classes. Most existing methods directly learn a model for fine-grained classification task while ignoring coarse-grained labels. In contrast, we leverage coarse-grained labels as an intermediate concept in our formulation. We use the same network architecture shown in Section 4.1 but with only 20 layers. The number of filters are 128, 256 and 512 for layers of 1-5, 6-10 and 10-20 respectively. Downsampling is performed at layer 6 and 11 and the coarse-grained label supervises layer 16.

Table 2 compares the error of DISCO with state-of-the-art and variants of DISCO. We use plain-single and plain-all to denote the networks with supervisions of single fine-grained label, and both labels at last layer, respectively. DISCO-random uses a (fixed) random coarse-grained class label for each training image. We observe that plain-all achieves roughly the same performance as plain-single, which replicates our earlier finding (Section 5.2.1) that intermediate supervision signal applied at the same layer as the main task helps relatively little in generalization. However, DISCO is able to reduce the error of plain-single by roughly 0.6 percent using the intermediate supervision signal. These results support our derivation of Proposition 1 and Proposition 2 in Section 3.3. Further, DISCO-random is significantly inferior to DISCO as a random intermediate concept makes the training more difficult. Finally, DISCO slightly outperforms the current state-of-the-art "pre-act ResNet-1001 [54]" on image classification but with only half of the network parameters compared with [54].

5.2 2D and 3D Keypoint Localization

In this Section, we demonstrate the performance of the deep supervision network (Fig. 4) for predicting the locations of object keypoints on 2D image and 3D space.

Dataset. For data synthesis, we sample CAD models of 472 cars, 100 sofas, 100 chairs and 62 beds from ShapeNet [51]. Each car model is annotated with 36 keypoints [10] and each furniture model (chair, sofa or bed) with 14 keypoints [16].[3] We synthesize 600 k car images including occluded instances and 300 k images of fully visible furniture (chair+sofa+bed). We pick rendered images of 5 CAD models from each object category as validation set.

We introduce KITTI-3D with annotations of 3D keypoint and occlusion type on 2040 car images from [18]. We label car images with one of four occlusion types: no occlusion (or fully visible cars), truncation, multi-car occlusion (target car is occluded by other cars) and occlusion cause by other objects. The number of images for each type is 788, 436, 696 and 120, respectively.

To obtain 3D groundtruth for these car images, we fit a PCA model trained on 3D keypoint annotation on CAD data, by minimizing the 2D projection error for known 2D landmarks provided by Zia et al. [10] and object pose from KITTI [18]. First, we compute the mean shape M and 5 principal components $P_1, \ldots, P_5$ from 3D skeletons of our annotated CAD models. M and P_i $(1 \leq i \leq 5)$ are 3×36 matrices where each column contains 3D coordinates of a keypoint. Thus, the 3D object structure X is represented as $X = M + \sum_{i=1}^{5} \alpha_i P_i$, where α_i is the weight for P_i. To avoid distorted shapes caused by large α_i, we constrain α_i to lie within $-2.7\sigma_i \leq \alpha_i \leq 2.7\sigma_i$ where σ_i is the standard deviation along the ith principal component direction. Next, given the groundtruth pose T, we compute 3D structure coefficients $\alpha = \{\alpha_i\}$ that minimize the projection error with respect to 2D ground truth Y

$$\begin{aligned} \alpha^* = & \arg\min_{\alpha} \min_{s,\beta} \|s\Pr(T(M + \sum_{i=1}^{N} \alpha_i P_i)) + \beta - Y\|_2^2 \\ & s.t. \; -2.7\sigma_i \leq \alpha_i \leq 2.7\sigma_i, \end{aligned} \tag{19}$$

where the camera intrinsic matrix is $K = [s_x, 0, \beta_x; 0, s_y, \beta_y; 0, 0, 1]$ with the scaling $s = [s_x; s_y]$ and shifting $\beta = [\beta_x; \beta_y]$. $\Pr(x)$ computes the 2D image coordinate from 2D homogeneous coordinate x. In practice, to obtain the ground truth with even higher quality, we densely sample object poses $\{T_j\}$ in the neighborhood of T and solve (19) by optimizing $\{\alpha_i\}, \beta, s$ given a fixed T_j and then search for the lowest error among all sampled T_j. We only provide 3D keypoint labels for fully visible cars because we do not have enough visible 2D keypoints for most of the occluded or truncated cars and thus obtain rather crude 3D estimates for such cases.

Evaluation Metric. We use PCK and APK metrics [56] to evaluate the performance of 2D keypoint localization. A 2D keypoint prediction is correct when it lies within the radius αL of the ground truth, where L is the maximum of image height and width and $0 < \alpha < 1$. PCK is the percentage of correct keypoint predictions given the object location and keypoint visibility. APK is the mean average precision of keypoint detection computed by associating each estimated keypoint with a confidence score. In our experiments, we use the regressed values of keypoint visibility as confidence scores. We extend 2D PCK and APK metrics to 3D by defining a correct 3D keypoint prediction whose euclidean distance to the ground truth is less than α in normalized coordinates.

Training Details. We set loss weights of visibility, 3D and 2D keypoint locations $\{\lambda_i\}$ to 1 and object pose to 0.1. We use stochastic gradient descent with momentum 0.9 to train the proposed CNN from scratch. Our learning rate starts at 0.01 and decreases by one-tenth when the validation error reaches a plateau. We set the weight decay to 0.0001, resize all input images to 64×64 and use batch size of 100. We initialize all weights using Glorot and Bengio [57]. For car model training, we form each batch using a mixture of fully visible, truncated and occluded cars, numbering 50, 20 and 30, respectively. For the furniture, each batch consists of 70 fully visible and 30 truncated objects randomly sampled from the joint synthetic image set of chair, sofa and bed.

5.2.1 *KITTI-3D*

We compare our method with DDN [39] and WarpNet [40] for 2D keypoint localization and Zia et al. [10] for 3D structure prediction. We use the original source codes for these methods. However, WarpNet is a siamese archtecture which warps a reference image to a test image benefiting from class-aware training. In order to use it for landmark transfer task, we need a reference image to be warped. Thus, we retrieve 30 labeled synthetic car images with the same pose as test image for landmark transfer using the CNN architecture proposed in [40] (WN-gt-yaw), and then compute the median of predicted landmark locations as the final result. The network is trained to warp pairs of synthetic car images in similar poses. Additionally, we perform an ablative analysis of DISCO. First, we replace all intermediate supervisions with the final labels, as DSN [19] does, for 2D (DSN-2D) and 3D (DSN-3D) structure prediction. Next, we incrementally remove the deep supervision used in DISCO one by one. DISCO-vis-3D-2D, DISCO-3D-2D, plain-3D, and plain-2D are networks without pose, pose+visibility, pose+visibility +2D and pose+visibility+3D, respectively. Further, we change the locations of the intermediate supervision signals. plain-all shifts supervision signals to the final convolutional layer. DISCO-(3D-vis) switches 3D and visibility in DISCO, and DISCO-reverse reverses the entire order of supervisions in DISCO. Finally, DISCO-VGG replaces stride-based downsampling and GAP in DISCO with non-overlapping spatial pooling (2×2) and a fully connected layer with 512 neurons, respectively. All methods are trained on the same set of synthetic training images and tested on real cropped cars on ground truth locations in KITTI-3D.

[3] We use 10 keypoints which are consistent with [27] to evaluate chair and bed on IKEA.

TABLE 3
PCK[$\alpha = 0.1$] Accuracies (%) of Different Methods for 2D and 3D Keypoint Localization on KITTI-3D Dataset

Method	2D					3D	3D-yaw
	Full	Truncation	Multi-Car Occ	Other Occ	All	Full	Full
DDN [39]	67.6	27.2	40.7	45.0	45.1	NA	
WN-gt-yaw* [40]	88.0	76.0	81.0	82.7	82.0	NA	
Zia et al. [10]	73.6	NA				73.5	7.3
DSN-2D	45.2	48.4	31.7	24.8	37.5	NA	
DSN-3D	NA					68.3	12.5
plain-2D	88.4	62.6	72.4	71.3	73.7	NA	
plain-3D	NA					90.6	6.5
plain-all	90.8	72.6	78.9	80.2	80.6	92.9	3.9
DISCO-3D-2D	90.1	71.3	79.4	82.0	80.7	94.3	3.1
DISCO-vis-3D-2D	92.3	75.7	81.0	83.4	83.4	95.2	2.3
DISCO-(3D-vis)	91.9	77.6	82.2	**86.1**	84.5	94.2	2.3
DISCO-reverse	30.4	29.7	22.8	19.6	25.6	54.8	13.0
DISCO-Vgg	83.5	59.4	70.1	63.1	69.0	89.7	6.8
DISCO	**93.1**	**78.5**	**82.9**	85.3	**85.0**	**95.3**	**2.2**
DISCO(Det)	95.9	78.9	87.7	90.5	88.3	95.5	2.1

Last column represents angular error in degrees. WN-gt-yaw [40] uses groundtruth pose of the test car. The bold numbers indicates the best result on groundtruth object bounding boxes. The last row presents the accuracies of DISCO on detection results from RCNN [55].

In Table 3, we report PCK accuracies for various methods[4] and the mean error of estimated yaw angles "3D-yaw" over all fully visible cars. This object-centric yaw angle is computed by projecting all 3D keypoints onto the ground plane and averaging the directions of lines connecting correspondences between left and right sides of a car. In turn, the 3D-yaw error is the average of absolute error between the estimated yaw and the ground truth.

We observe that DISCO outperforms competitors in both 2D and 3D keypoint localization across all occlusion types. Moreover, we observe a monotonic increase in 2D and 3D accuracy with increasing supervision: plain-2D or plain-3D $<$ DISCO-3D-2D $<$ DISCO-vis-3D-2D $<$ DISCO. Further, plain-all is superior to plain-2d and plain-3d, while DISCO exceeds plain-all by 4.4 percent on 2D-All and 2.4 percent on 3D-Full. These experiments confirm that joint modeling of 3D shape concepts is better than independent modeling. Moreover, alternative supervision orders (DISCO-reverse, DISCO-(3D-vis)) are found to be inferior to the proposed order which captures underlying structure between shape concepts. Last, DISCO-VGG performs significantly worse than DISCO by 16.0 percent on 2D-All and 5.6 percent on 3D-Full, which validates our removal of local spatial pooling and adopt global average pooling. In conclusion, the proposed deep supervision architecture coupled with intermediate shape concepts improves the generalization ability of CNN. As more concepts are introduced in the "correct" order, we observe improvement in performance.

We also conduct an ablative study of training data with different occlusion types. Table 4 demonstrates 2D keypoint localization accuracies over different occlusion categories on KITTI-3D given various combination of training data. "Occ." stands for test examples with multi-object occlusions where the occluder is either another car or a different object such as a pedestrian. As we can see, DISCO trained on fully visible cars alone achieves much worse performance on truncated and occluded test data than when trained on data with simulated truncation and multi-car occlusion. We observe that multi-car occlusion data is also helpful in modeling truncation cases, and the network trained by multi-car data obtains the second best result on truncated cars. The best overall performance is obtained by including all three types of examples (no occlusion, multi-car occlusion, truncation), emphasizing the efficacy of our data generation strategy.

Finally, we evaluate DISCO on detection bounding boxes computed from RCNN[55] with IoU> 0.7 to the groundtruth of KITTI-3D. "DISCO-Det" in the last row of Table 3 shows PCK accuracies of DISCO using detection results. The 2D/3D keypoint localization accuracies even exceeds the performance of DISCO using groundtruth bounding boxes by 3.3 percent on 2D-All and 0.2 percent on 3D-All.

5.2.2 PASCAL VOC

We evaluate DISCO on the PASCAL VOC 2012 dataset for 2D keypoint localization [56]. Unlike KITTI-3D where car images are captured on real roads and mostly in low resolution, PASCAL VOC contains car images with larger appearance variations and heavy occlusions. In Table 5, we compare our results with the state-of-the-art [38], [58] on various sub-

TABLE 4
Ablative Study of Different Training Data Sources

Training Data			Test Data		
Full	Trunc.	Multi-Car	Full	Trunc.	Occ.
✓			91.8	53.6	68.3
	✓		89.9	73.8	61.7
		✓	91.3	74.7	82.7
✓	✓		92.9	71.3	63.4
✓		✓	92.5	73.2	**84.1**
	✓	✓	90.5	70.4	81.2
✓	✓	✓	**93.1**	**78.5**	83.2

PCK[$\alpha = 0.1$] accuracies (%) of DISCO for 2D keypoint localization on KITTI-3D dataset.

[4] We cannot report Zia et al. [10] on occluded data because only a subset of images has valid result in those classes.

TABLE 5
PCK[$\alpha = 0.1$] Accuracies (%) of Different Methods for 2D Keypoint Localization on the Car Category of PASCAL VOC

PCK[$\alpha = 0.1$]	Full	Full[$\alpha = 0.2$]	Occluded	Big Image	Small Image	All[APK $\alpha = 0.1$]
Long[58]	55.7	NA				
VpsKps[38]	81.3	88.3	**62.8**	**90.0**	67.4	40.3
DSN-2D	75.4	87.8	54.5	85.5	63.3	NA
plain-2D	76.7	90.6	50.4	80.6	69.4	NA
plain-all	75.9	90.4	53.0	82.4	65.1	41.7
DISCO-reverse	64.5	84.5	41.2	55.5	67.0	24.9
DISCO-3D-2D	81.5	92.0	61.0	87.6	73.1	NA
DISCO	**81.8**	**93.4**	59.0	87.7	**74.3**	**45.4**

Bold numbers indicate the best results.

classes of the test set: fully visible cars (denoted as "Full"), occluded cars, high-resolution (average size 420 × 240) and low-resolution images (average size 55 × 30). Please refer to [38] for details of the test setup. Note that these methods [38], [58] are trained on real images, whereas DISCO training exclusively leverages synthetic training data.

We observe that DISCO outperforms [38] by 0.6 and 5.1 percent on PCK at $\alpha = 0.1$ and $\alpha = 0.2$, respectively. In addition, DISCO is robust to low-resolution images, improving 6.9 percent accuracy on low-resolution set compared with [38]. This is critical in real perception scenarios where distant objects are small in images of street scenes. However, DISCO is inferior on the occluded car class and high-resolution images, attributable to our use of small images (64 × 64) for training and the fact that our occlusion simulation does not capture the complex occlusions created by non-car objects such as walls and trees. Finally, we compute APK accuracy at $\alpha = 0.1$ for DISCO on the same detection candidates used in [38].[5] We can see that DISCO outperforms [38] by 5.1 percent on the entire car dataset (Full+Occluded). This suggests DISCO is more robust to noisy detection results and more accurate on keypoint visibility inference than [38]. We attribute this to global structure modeling of DISCO during training where the full set of 2D keypoints resolves the partial view ambiguity whereas traditional methods like [38] only are supervised with visible 2D keypoints.

Note that some definitions of our car keypoints [10] are slightly different from [56]. For example, we annotate the bottom corners of the front windshield whereas [56] labels the side mirrors. In our experiments, we ignore this annotation inconsistency and directly assess the prediction results. We reemphasize that unlike [38], [58], we do not use the PASCAL VOC train set. Thus, even better performance is expected when real images with consistent labels are used for training.

5.2.3 PASCAL 3D+

PASCAL3D+ [16] provides object viewpoint annotations for PASCAL VOC objects by manually aligning 3D object CAD models onto the visible 2D keypoints. Because only a few CAD models are used for each category, the 3D keypoint locations are only approximate. Thus, we use the evaluation metric proposed by [16] which measures 2D overlap (IoU) against projected model mask. With a 3D skeleton of an object, we are able to create a coarse object mesh based on the geometry and compute segmentation masks by projecting coarse mesh surfaces onto the 2D image based on the estimated 2D keypoint locations.

Table 6 reports object segmentation accuracies on two types of ground truth. The column "Manual GT" uses manual pixel-level annotation provided by PASCAL VOC 2012, whereas "CAD alignment GT" uses 2D projections of aligned CAD models as ground truth. Note that "CAD alignment GT" covers the entire object extent in the image including regions occluded by other objects. DISCO significantly outperforms a state-of-the-art method [33] by 4.6 and 6.6 percent despite using only synthetic data for training. Moreover, on "Manual GT" benchmark, we compare DISCO with "Random CAD" and "GT CAD" which stand for the projected segmentation of randomly selected and ground truth CAD models respectively, given ground truth object pose. We find that DISCO yields even superior performance to "GT CAD". This provides evidence that joint modeling of 3D geometry manifold and viewpoint is better than the pipeline of object retrieval plus alignment. Finally, we note that a forward pass of DISCO only takes less than 10ms during testing, which is far more efficient compared with sophisticated CAD alignment approaches [10] that usually needs more than 1s for one image input.

TABLE 6
Object Segmentation Accuracies (%) of Different Methods on PASCAL3D+

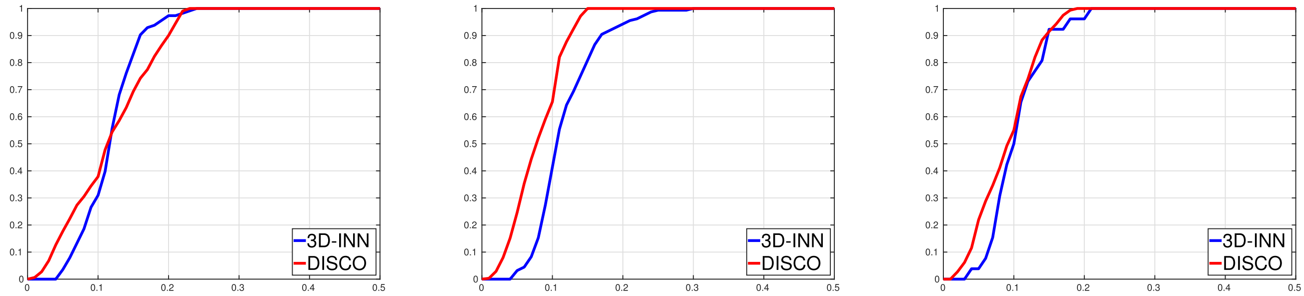
Method	CAD alignment GT	Manual GT
VDPM-16 [16]	NA	51.9
Xiang et al. [59]	64.4	64.3
Random CAD [16]	NA	61.8
GT CAD [16]	NA	67.3
DSN-2D	66.4	63.3
plain-2D	67.4	64.3
plain-all	66.8	64.2
DISCO-reverse	54.2	56.0
DISCO	**71.2**	**67.6**

Best results are shown in bold.

TABLE 7
Average Recall and PCK[$\alpha = 0.1$] Accuracy(%) for 3D Structure Prediction on the Sofa and Chair Classes on IKEA Dataset

Method	Sofa		Chair		Bed	
	Recall	PCK	Recall	PCK	Recall	PCK
3D-INN	**88.0**	31.0	87.8	41.4	**88.6**	42.3
DISCO	84.4	**37.9**	**90.0**	**65.5**	87.1	**55.0**

[5] We run the source code [38] to obtain the same object candidates.



(a) IKEA Sofa

(b) IKEA Chair

(c) IKEA Bed

Fig. 6. 3D PCK (RMSE[27]) curves of DISCO and 3D-INN on sofa (Fig. 6a), chair (Fig. 6b) and bed (Fig. 6c) classes of IKEA dataset. In each figure, X axis stands for α of PCK and Y axis represents the accuracy.

5.2.4 IKEA

In this section, we evaluate DISCO on the IKEA dataset [17] with 3D keypoint annotations provided by [27]. One question remaining for the DISCO network is whether it is capable of learning 3D object geometry for multiple object classes simultaneously. Therefore, we train a single DISCO network from scratch which jointly models three furniture classes: sofa, chair and bed. At test time, we compare DISCO with the state-of-the-art 3D-INN [27] on IKEA. Since 3D-INN evaluates the error of 3D structure prediction in the

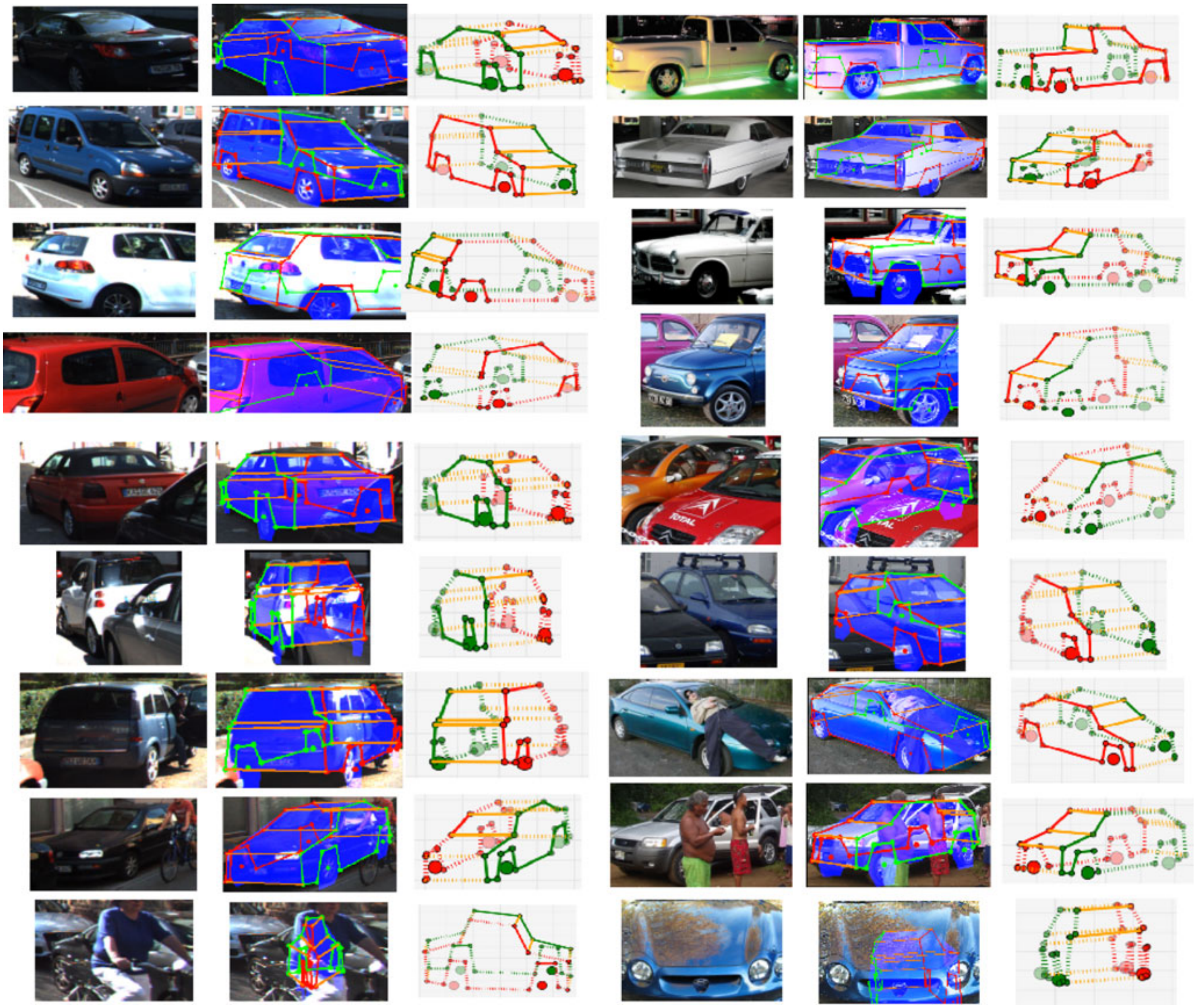


Fig. 7. Visualization of 2D/3D prediction, visibility inference and instance segmentation on KITTI-3D (left column) and PASCAL VOC (right column). Last row shows failure cases. Circles and lines represent keypoints and their connections. Red and green indicate the left and right sides of a car, orange lines connect two sides. Dashed lines connect keypoints if one of them is inferred to be occluded. Light blue masks present segmentation results.

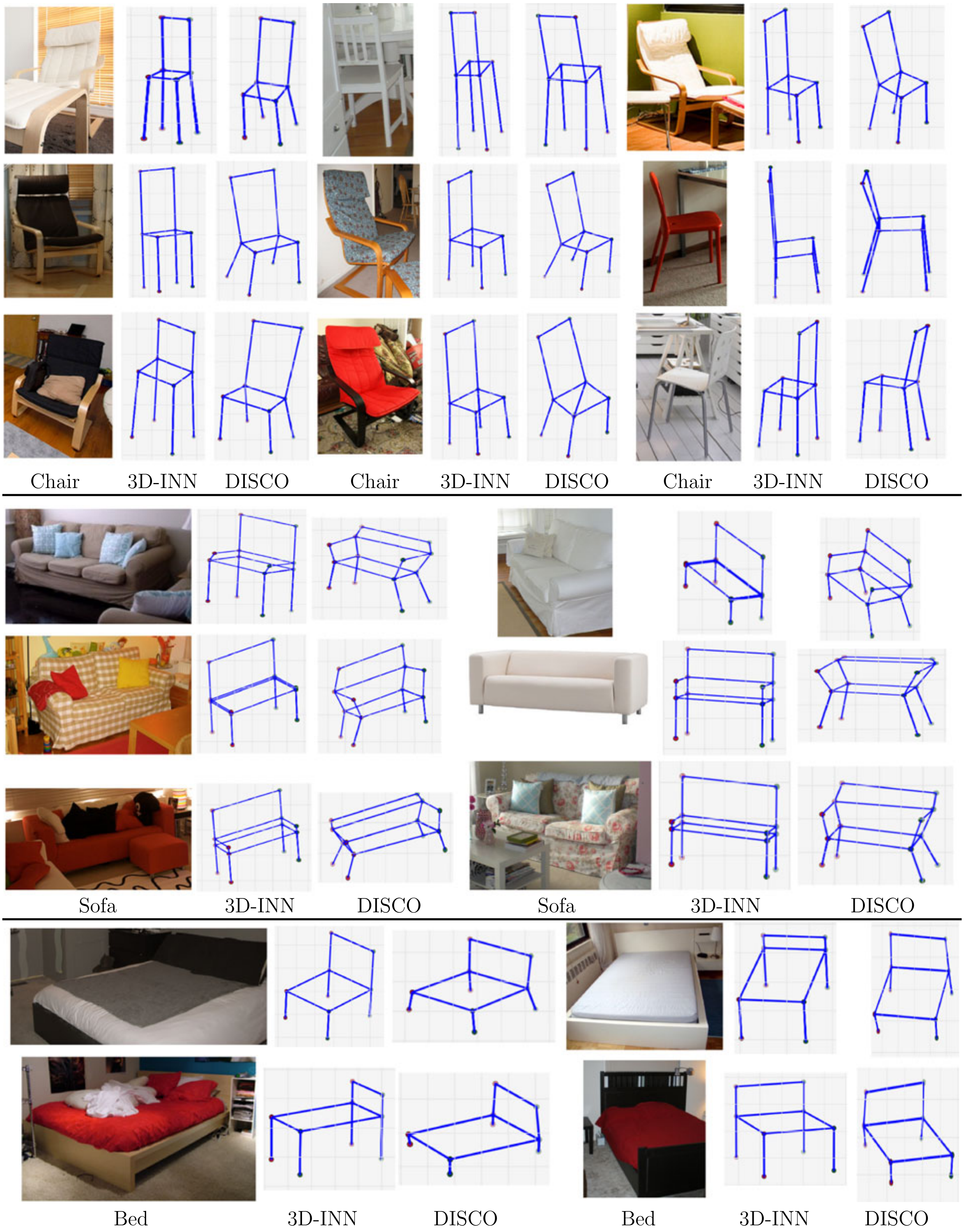


Fig. 8. Qualitative comparison between 3D-INN and DISCO for 3D stricture prediction on IKEA dataset.

object canonical pose, we align the PCA bases of both the estimated 3D keypoints and their groundtruth. Table 7 reports the PCK[$\alpha = 0.1$] and average recall [27] (mean PCK over densely sampled α within $[0, 1]$) of 3D-INN and DISCO on all furniture classes. The corresponding PCK curves are visualized in Fig. 6. We retrieve PCK accuracies of 3D-INN on the IKEA dataset from its publicly released results. DISCO significantly outperforms 3D-INN on PCK by 6.6, 24.1, 12.7 percent on sofa, chair and bed respectively, which means that DISCO obtains more correct predictions of keypoint locations than 3D-INN. This substantiates that direct exploitation of the rich visual details from images adopted by DISCO is critical to infer more accurate and fine-grained 3D structure than lifting sparse 2D keypoints to 3D shapes like 3D-INN. However, DISCO is inferior to 3D-INN in terms of average recall on the sofa and bed class. As shown in Fig. 6a, the incorrect predictions by DISCO deviate more from the groundtruth than 3D-INN. This is mainly because 3D predicted shapes from 3D-INN are constrained by shape bases so even incorrect estimates have realistic object shapes when recognition fails. Moreover, our 3D keypoint labeling for the sofa CAD models is slightly different from [27]. We annotate the corners of reachable seating areas of a sofa while IKEA labels the corners of the outer volume parallel to the seating area We conclude that DISCO is able to learn 3D patterns of object classes other than the car category and shows potential as a general-purpose approach to jointly model 3D geometric structure of multiple objects *in a single model*.

5.2.5 Qualitative Results

In Fig. 7, we visualize example predictions from DISCO on KITTI-3D (left column) and PASCAL VOC (right column). From left to right, each column shows the original object image, the predicted 2D object skeleton with instance segmentation and the predicted 3D object skeleton with visibility. From top to bottom, we show the results under no occlusion (row 1-2), truncation (row 3-4), multi-car occlusion (row 5-6), other occluders (row 7-8) and failure cases (row 9). We observe that DISCO is able to localize 2D and 3D keypoints on real images with complex occlusion scenarios and diverse car models such as sedan, SUV and pickup. Moreover, the visibility inference is mostly correct. These capabilities highlight the potential of DISCO as a building block for holistic scene understanding in cluttered scenes. In failure cases, the left car is mostly occluded by another object and the right one is severely truncated and distorted in projection. We may improve the performance of DISCO on these challenging cases by exploiting more sophisticated data simulated with complex occlusions [60] and finetuning DISCO on real data.

In addition, we qualitatively compare 3D-INN and DISCO on three categories in IKEA dataset in Fig. 8. For the chair, 3D-INN fails to delineate the inclined seatbacks in the example images while DISCO being able to capture this structural nuance. For the sofa, DISCO correctly infers the location of sofa armrest whereas 3D-INN merges armrests to the seating area or predicts an incorrect size of the seatback. Finally, DISCO yields better estimates of the scale of bed legs than 3D-INN. We attribute this relative success of DISCO to direct mapping from image evidence to 3D structure, as opposed to lifting 2D keypoint predictions to 3D.

6 Conclusion

Visual perception often involves sequential inference over a series of intermediate goals of growing complexity towards the final objective. In this paper, we have employed a probabilistic framework to formalize the notion of intermediate concepts which points to better generalization through deep supervision, compared to the standard end-to-end training. This inspires a CNN architecture where hidden layers are supervised with an intuitive sequence of intermediate concepts, in order to incrementally regularize the learning to follow the prescribed inference sequence. We practically leveraged this superior generalization capability to address the scarcity of 3D annotation: learning shape patterns from synthetic training images with complex multiple object configurations. Our experiments demonstrate that our approach outperforms current state-of-the-art methods on 2D and 3D landmark prediction on public datasets, even with occlusion and truncation. We applied deep supervision to fine-grained image classification and showed significant improvement over single-task as well as multi-task networks on CIFAR100. Finally, we have presented preliminary results on jointly learning 3D geometry of multiple object classes within a single CNN. Our future work will extend this direction by learning shared representations for diverse object classes. We also see wide applicability of deep supervision, even beyond computer vision, in domains such as robotic planning, scene physics inference and generally wherever deep neural networks are being applied. Another future direction is to extract label relationship graphs from the CNN supervised with intermediate concepts, as opposed to explicitly constructed Hierarchy and Exclusion graphs [23].

Acknowledgments

This work was part of C. Li's intern project at NEC Labs America, in Cupertino. We acknowledge the support by NSF under grants IIS-127228 and IIS-1637949. We also thank Rene Vidal, Alan Yuille, Austin Reiter and Chong You for helpful discussions.

References

[1] B. J. Smith, "Perception of organization in a random stimulus," *Comput. Vis., Graph., Image Process. (CVGIP)*, vol. 31, no. 2, pp. 242–247, Aug. 1985.

[2] D. Lowe, *Perceptual Organization and Visual Recognition*. Norwell, MA, USA: Kluwer, 1985.

[3] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. Roy. Soc. London. Series B*, vol. 200, pp. 269–294, 1978.

[4] R. Mohan and R. Nevatia, "Using perceptual organization to extract 3D structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 11, pp. 1121–1139, Nov. 1989.

[5] S. Sarkar and P. Soundararajan, "Supervised learning of large perceptual organization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 504–525, May 2000.

[6] Y. S. Abu-Mostafa, "Hints," *Neural Comput.*, vol. 7, pp. 639–671, 1995.

[7] C. Li, M. Z. Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with shape concepts for occlusion-aware 3D object parsing," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 388–397.

[8] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, Ontario, 2009.

[9] Y. Xiang and S. Savarese, "Object detection by 3D aspectlets and occlusion reasoning," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 530–537.

[10] M. Z. Zia, M. Stark, and K. Schindler, "Towards scene understanding with detailed 3D object representations," *Int. J. Comput. Vis.*, vol. 112, pp. 188–203, 2015.

[11] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Category-specific object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1966–1974.

[12] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. B. Tenenbaum, "Deep convolutional inverse graphics network," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2539–2547.

[13] T. Wu, B. Li, and S.-C. Zhu, "Learning and-or model to represent context and occlusion for car detection and viewpoint estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1829–1843, Sep. 2016.

[14] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros, "Learning dense correspondence via 3D-guided cycle consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 117–126.

[15] H.-J. Lee and Z. Chen, "Determination of 3D human body postures from a single view," *Comput. Vis. Graph. Image Process.*, vol. 30, pp. 148–168, 1985.

[16] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2014, pp. 75–82.

[17] J. J. Lim, H. Pirsiavash, and A. Torralba, "Parsing IKEA objects: Fine pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2992–2999.

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[19] C.-Y. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. 18th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2015, vol. 38, pp. 562–570.

[20] R. Caruana, "Multitask learning," *Mach. Learning*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

[21] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 94–108.

[22] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with Rendered 3D model views," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2686–2694.

[23] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 48–64.

[24] J. Baxter, "A model of inductive bias learning," *J. Artif. Intell. Res.*, vol. 12, pp. 149–198, 2000.

[25] A. Maurer, M. Pontil, and B. Romera-Paredes, "The benefit of multitask representation learning," *J. Mach. Learn. Res.*, vol. 17, pp. 2853–2884, 2016.

[26] Ç. Gülçehre and Y. Bengio, "Knowledge matters: Importance of prior information for optimization," *J. Mach. Learn. Res.*, vol. 17, pp. 226–257, 2016.

[27] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman, "Single image 3D interpreter network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 365–382.

[28] A. Dosovitskiy, J. Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1538–1546.

[29] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3D models from single images with a convolutional network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 322–337.

[30] P. Moreno, C. K. Williams, C. Nash, and P. Kohli, "Overcoming occlusion with inverse graphics," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 170–185.

[31] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 628–644.

[32] D. Rezende, S. Eslami, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised learning of 3D structure from images," in *Proc. 30th Conf. Neural Inf. Process. Syst.*, 2016, pp. 4996–5004.

[33] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D voxel patterns for object category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1903–1911.

[34] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic, "Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3762–3769.

[35] J. J. Lim, A. Khosla, and A. Torralba, "FPM: Fine pose parts-based model with 3D CAD models," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 478–493.

[36] A. Bansal, B. Russell, and A. Gupta, "Marr revisited: 2D-3D alignment via surface normal prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5965–5974.

[37] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik, "Inferring 3D object pose in RGB-D images," *arXiv:1502.04652*, Feb. 2015, https://arxiv.org/abs/1502.04652

[38] S. Tulsiani and J. Malik, "Viewpoints and keypoints," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1510–1519.

[39] X. Yu, F. Zhou, and M. Chandraker, "Deep deformation network for object landmark localization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 52–70.

[40] A. Kanazawa, D. W. Jacobs, and M. Chandraker, "WarpNet: Weakly supervised matching for single-view reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3253–3261.

[41] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 32–39.

[42] B. Pepikj, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3286–3293.

[43] H. Lebesgue, "Intégrale, longueur, aire," *Annali di Matematica Pura ed Applicata (1898–1922)*, vol. 7, no. 1, pp. 231–359, 1902.

[44] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, pp. 359–366, 1989.

[45] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. Int. Conf. Learning Represent. (ICLR)*, 2017.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[47] Y. Li, L. Gu, and T. Kanade, "Robustly aligning a shape model and its application to car alignment of unknown pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1860–1876, Sep. 2011.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, Nov. 2014, https://arxiv.org/pdf/1409.1556.pdf

[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *J. Mach. Learn. Res.*, vol. 37, pp. 448–456, 2015.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[51] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, et al., "ShapeNet: An information-rich 3D model repository," *arXiv:1512.03012*, 2015.

[52] D. Mishkin and J. Matas, "All you need is a good init," 2016.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[55] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

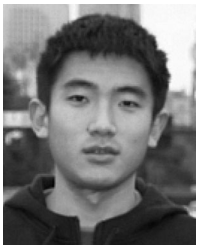
[56] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1385–1392.

[57] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[58] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1601–1609.

[59] R. Mottaghi, Y. Xiang, and S. Savarese, "A coarse-to-fine model for 3D pose estimation and sub-category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 418–426.

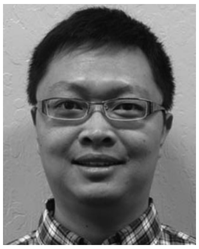
[60] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 102–118.



Chi Li received the BEng degree in cognitive science from Xiamen University, China, in 2012 and the PhD degree in computer science from Johns Hopkins University, in 2018. He is now a computer vision and machine learning engineer in Apple Inc. His main research interests are learning representation towards understanding scene semantics and geometry and deploy such techniques for sequential scene understanding and single/multi-view object recognition.

M. Zeeshan Zia is a senior scientist at Microsoft. He works on holistic 3D scene understanding, and is interested in exploring how data-driven techniques can contribute to robust 3D perception. He attended the Swiss Federal Institute of Technology (ETH), Zurich for graduate studies, and subsequently spent a few years at Imperial College London as a postdoctoral associate and NEC Laboratories America as a researcher.

Quoc-Huy Tran received the BEng degree in computer science and engineering from Vietnam National University, Vietnam, in 2010 and the PhD degree in mathematical and computer sciences from the University of Adelaide, Australia, in 2014. He is currently a researcher at NEC Laboratories America. His main research interests include structure from motion, 3D localization, and 3D reconstruction with applications in autonomous driving.



Xiang Yu received the BS and MS degrees from Tsinghua University, and the PhD degree from Rutgers, the State University of New Jersey. He is a researcher at NEC Labs America, Media Analytics Department. His research mainly focuses on object key feature localization, large scale face recognition, factors of variation disentanglement and generative models for data synthesis. He is also expertise in emotion analysis and scalable image retrieval.

Gregory D. Hager is the Mandell Bellmore professor of computer science at Johns Hopkins University and founding director of the Malone Center for Engineering in Healthcare. His research interests include collaborative and vision-based robotics, time-series analysis of image data, and medical applications of image analysis and robotics. He has served on the editorial boards of the *IEEE Transactions on Robotics*, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and the *International Journal of Computer Vision*. He is a fellow of the IEEE, the ACM, AIMBE, and the MICCAI Society.

Manmohan Chandraker heads the Media Analytics Department at NEC Laboratories America, and is on the faculty of University of California at San Diego (UCSD). His principal research interests are sparse and dense 3D reconstruction, with applications to autonomous driving and robotics. His work on provably optimal algorithms for structure and motion estimation received the Marr Prize Honorable Mention for Best Paper at ICCV 2007. His work on shape recovery from motion cues for complex material and illumination received the Best Paper Award at CVPR 2014.