Learning from Synthetic Vehicles

Tae Soo Kim Bohoon Shim Michael Peven Weichao Qiu
Alan Yuille Gregory D. Hager
Johns Hopkins University
3400 N. Charles St, Baltimore, MD
{tkim60,jdjones,hager}@jhu.edu

Abstract

In this paper, we release the Simulated Articulated VEhicles Dataset (SAVED) which contains images of synthetic vehicles with moveable vehicle parts. SAVED consists of images that are much more relevant for vehicle-related pattern-recognition tasks than other popular pretraining datasets such as ImageNet. Compared to a model initialized with ImageNet weights, we show that a model pretrained using SAVED leads to much better performance when recognizing vehicle parts and orientation directly from an image. We also find that a multi-task pretraining approach using fine-grained geometric signals available in SAVED leads to significant improvements in performance. By pretraining on SAVED instead of ImageNet, we reduce the error rate of one of the state of the art vehicle orientation estimators by 51.2% *when tested on real images. We release SAVED and instructions on its usage here*[1].

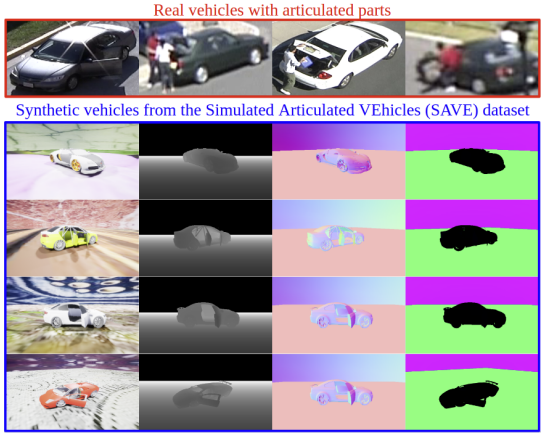

Figure 1: SAVED is the first large scale dataset of synthetic vehicles with articulated vehicle parts such as doors and trunks. Top: vehicles with articulated parts from the DIVA-Doors dataset, Bottom: simulated instances with domain randomized RGB examples, depth maps, surface normals and semantic segmentation labels (from left to right).

1. Introduction

Access to a large set of images paired with accurate annotations is often a prerequisite for successfully training a visual perception model. In the era where data-driven methods powered by highly parameterized deep neural networks dominate the field of visual recognition, the need to acquire sufficiently large and high quality training data is common across different applications and problem domains. However, annotating large scale datasets may be prohibitively expensive or practically impossible depending on the problem domain. For example, while annotating a presence of a common object in an image may be suitable for large scale data collection with a crowd sourced workforce, collecting detailed 3D information of object parts from real images at scale is far more challenging. When access to a large annotated dataset is limited, practitioners typically rely on pretraining a deep network model on a large scale, but unrelated, dataset and later finetune the poorly initialized model using a small set of annotated samples from the target domain. Our experiments show that this standard practice often leads to models with sub-optimal performance.

We explore the use of synthetic data to address this challenge. There is growing evidence that a dataset with both real and synthetic images can successfully train deep network models for various vision problems [19, 28, 14, 12]. Given the advancements in graphical renderers such as Unreal Engine 4 and Blender [4] coupled with software developments such as UnrealCV [20], researchers now have direct access to a simulation engine that can generate large scale datasets. Compared to real datasets, the cost of annotating a very large dataset is inconsequential. Moreover, most simulation engines provide various auxiliary information regarding the scene in addition to the generated task label. Often, graphical engines by default provide scene

[1] https://taesoo-kim.github.io/

level geometric information such as pixel-wise depth values and object surface normal information as these factors govern how the scene gets rendered. Many of such rendering parameters and other auxiliary information that can be extracted from the simulator provide a rich set of 'free' annotations that are difficult to obtain in natural images.

In this paper, we present a synthetic dataset, Simulated Articulated VEhicles Dataset (SAVED), which contains over 500K images of synthetic vehicles with various labels. Each image comes with a full set of scene-level information regarding the geometry of the vehicle in the form of surface normals and depth maps. Moreover, we provide per-pixel level annotation to localize the vehicle in the scene. Finally, the most noticeable novelty of SAVED is the granularity of vehicle part information that it provides. Vehicles in SAVED contain articulated parts with annotations specifying the exact encoder position of door joints. Using this information, we can accurately assess the state of a vehicle part (ie. door or trunk) and how much it is rotated about its axis. Though there are real [31, 17, 7] and simulated [6, 25] datasets with vehicle annotations, the granularity of annotations are insufficient for such fine-grained analysis of a vehicle's state.

In addition to the dataset, we present a simple yet an effective approach for training with synthetic images. We show that we can obtain a stronger model trained using synthetic images by requiring a model to predict scene level geometric information (ie. per-pixel depth and normal values) in addition to the main prediction task. In our experiments, we show that a model pretrained using our synthetic images with the presented multi-task training strategy outperforms other models pretrained using other popular large scale datasets such as [5, 31, 7]. Moreover, we show that by pretraining on a large scale synthetic training dataset, we can reduce the number of real training images with labels to finetune the model. Using SAVED, we present the first approach to recognize parts of vehicles and their states (i.e. opened-doors and closed-trunks) from natural images. We demonstrate that we can not only obtain a model to perform the novel task of articualted vehicle recognition but also improve existing state of the art methods on standard tasks such as vehicle pose estimation. In summary, the following are contributions of this paper:

1. The Simulated Articulated VEhicles Dataset (SAVED): A large scale dataset of rendered synthetic vehicle images with fine-grained vehicle part annotations, 3D geometry annotations and move-able vehicle parts.
2. The first model trained using simulated data to recognize vehicle parts and orientation from natural images.
3. Experimental evidence that multi-task training with geometric signals (i.e. surface normals and depth maps) is critical when pretraining a model using a simulated dataset.

2. Related Work

Learning from simulation. Researchers have successfully trained various visual perception models using simulated data for applications in stereo-vision [32], semantic segmentation [22, 26] and 3D pose estimation [14, 2, 23, 12]. For such tasks, groundtruth annotations on real images are insufficient to train deep neural networks. Using a simulation engine with a software such as UnrealCV [20], groundtruth data that is otherwise difficult to obtain can be generated in large amounts with significantly less effort. To the best of our knowledge, there is no dataset, synthetic or real, that has annotations at the vehicle part level which includes how much the part is rotated about its connection point to the vehicle frame.

Related simulated vehicle datasets. Compared to datasets with only natural images of vehicles such as KITTI [7], PASCAL 3D+ [31] and EPFL [17], the SAVED dataset leverages the power of simulation to generate significantly more samples with larger diversity. In our experiments, we show that by pretraining on SAVED then later finetuning to the target dataset using the available real samples, we achieve much better results on articulated vehicle recognition and vehicle orientation estimation tasks.

The most notable simulated datasets with vehicle annotations are SYNTHIA [25] and V-KITTI [6]. The biggest motivating application for these datasets is in the domain of autonomous vehicles. Hence, viewpoints are limited in these datasets because samples are captured from the point of view of a driver. Hence, the virtual vehicles found in the two datasets are not well suited for training fine-grained models for reasoning about vehicle parts. With more diverse camera viewpoints and articulated vehicle parts, SAVED is a better dataset to train models for problems such as 3D pose estimation and recognizing parts of vehicles.

Vehicle orientation estimation. Estimation of object orientation can be cast a camera viewpoint estimation problem and has been active fields of research, including methods for estimating the pose of human heads [15], pedestrians [21], vehicles [12, 13, 9] and common objects [27, 12, 9]. Following the observation that training with a classification loss consistently outperforms a regression loss setup for pose estimation in previous approaches, our method models vehicle orientation estimation as classification problem by converting continuous angular values to a one-hot vector label using evenly sized discrete bins. Our approach builds upon the state-of-the-art methods for orientation estimation using conservative labels [13] and intermediate supervision [12] to enable the first approach for articulated vehicle recognition using simulated data.

Simulation to real transfer. Several studies have shown

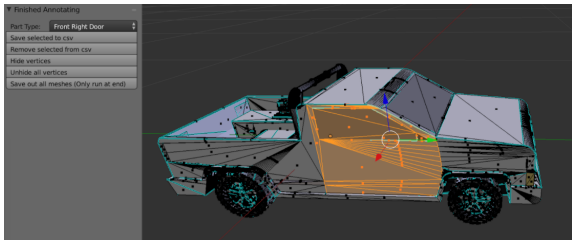

Figure 2: The custom Blender plugin for annotating vehicle parts such as doors and trunks.

that classifiers trained using simulated images often require methods for simulation-to-real transfer to perform well on real images [30]. We show in our experiments that the use of geometric signals during pretraining with simulation data helps mitigate the issue of domain shift. When there are small number of labeled real instances, we show that a simple approach of pretraining using simulated instances and then later finetuning with real examples leads to best results. We also show that domain randomization techniques [29] as well as intermediate supervision [12] are important when training with synthetic datasets.

3. The Simulated Articulated Vehicles Dataset (SAVED)

We describe the details of the Simulated Articulated VEhicles Dataset (SAVED). In contrast to existing real or synthetic datasets of vehicles, the simulated instances in SAVED have moveable parts such as doors, trunks and hoods with ground truth annotations on how much the vehicle part is rotated around its axis. As illustrated in Figure 1, SAVED provides per-pixel depth, surface normal and semantic part labels.

We use Unreal Engine 4 as our renderer of choice and use UnrealCV [20] to interact with the virtual environment to simulate and capture data. We simulate vehicles by rendering the 3D CAD models provided by the ShapeNet dataset [1]. The synthetic vehicles found in ShapeNet do not provide vehicle part annotations as standard. Thus, we manually annotated doors, trunks and hoods of vehicles in order to articulate them as needed using the simulator.

For this purpose, we built a custom Blender plugin (depicted in Fig. 2) to label the sections of the mesh as its corresponding part. To maximize diversity of simulated vehicle appearance in the dataset, we search for similar vehicles via hierarchical clustering over features extracted from rotation invariant 3D shape descriptors using spherical harmonics [11]. We annotated 103 vehicle meshes corresponding to the center of the largest clusters. Given the knowledge of corresponding vehicle parts, we implemented a mechanism through UnrealCV which manipulates each vehicle part individually. Figure 3 illustrates sample data points from SAVED generated using our approach.

Table 1 compares SAVED to other vehicle datasets. Our dataset contains the most number of images captured from a diverse set of camera viewpoints. SAVED is the first dataset with annotations on vehicle parts: we provide the extent to which each door is rotated in degrees. Next, we describe our approach for training with simulated data.

4. Learning from synthetic vehicles

In this section, we discuss our general strategy for training deep neural network based image classifiers using synthetic images. We use extra geometric information about the scene such as depth maps and surface normals as auxiliary tasks in addition to the main task for the model to optimize for during pretraining with synthetic data. We observe in our experiments that a model initialized using simulation data with this simple multi-task training approach leads to much better classification performance when tested on real images.

Multi-task approach with geometric signals. We describe our approach for a general classification scenario but our method can be generalized trivially to other problems such as detection and pose estimation. Let $X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ be a synthetic training dataset with pairs of a rendered image of a vehicle $x_n \in \mathbb{R}^{H \times W \times C}$ and some corresponding ground truth task label $y_n \in \{1, \ldots, M\}$. The objective is to learn a classifier $\hat{y} = F(x)$ such that the following classification loss $\mathcal{L}_{cls}$ is minimized:

$$\mathcal{L}_{cls} = -\sum_n \sum_m y_n^m log(\hat{y_n^m}) \quad (1)$$

We use a deep neural network for F. This is a standard formulation for optimizing a deep neural network using a cross-entropy loss.

One of the biggest benefits of synthetic data is that a simulation engine has a representation for the 3D geometry of the virtual scene that is readily available. Let $Z = \{z_1, z_2, \ldots, z_N\}$ be a set of some geometric representations such as surface normals or depth maps extracted from the simulator. The intuition behind this approach is that various tasks regarding a vehicle such as vehicle part detection and pose estimation are fundamentally related to its geometry. We use an encoder-decoder framework to jointly predict the task label $\hat{y_n}$ and the geometry $\hat{z_n}$ from x_n. We refactor the classifier F such that:

$$\hat{y} = F(x) = \text{softmax}(f_{cls}(f(x))) \quad (2)$$

where $f(x) \in \mathbb{R}^D$ is an output of an encoder that maps an image to a feature representation of some dimension D and f_{cls} is a linear classification layer that maps feature vectors with D dimension to the output space with M outputs. The

	SAVED (Ours)	KITTI [7]	V-KITTI [6]	SYNTHIA [25]	Pascal 3D+ [31]	EPFL [17]
Real/Simulated	Sim	Real	Sim	Sim	Real	Real
# annotated samples	586,340	80,000	80,000	200,000	6704	2137
Background	Random Texture	Outdoor	Sim. Outdoor	Sim. Outdoor	Indoor+Outdoor	Indoor
Orientation label	yes	no	yes	yes	yes	yes
Azimuth label	yes	no	yes	yes	yes	no
Depth and normal labels	D+N	D	D	D	no	no
Vehicle part Label	yes	no	no	no	no	no

Table 1: Compared to existing datasets with vehicle annotations, SAVED provides vehicle part information and the most comprehensive set of 3D geometry information.

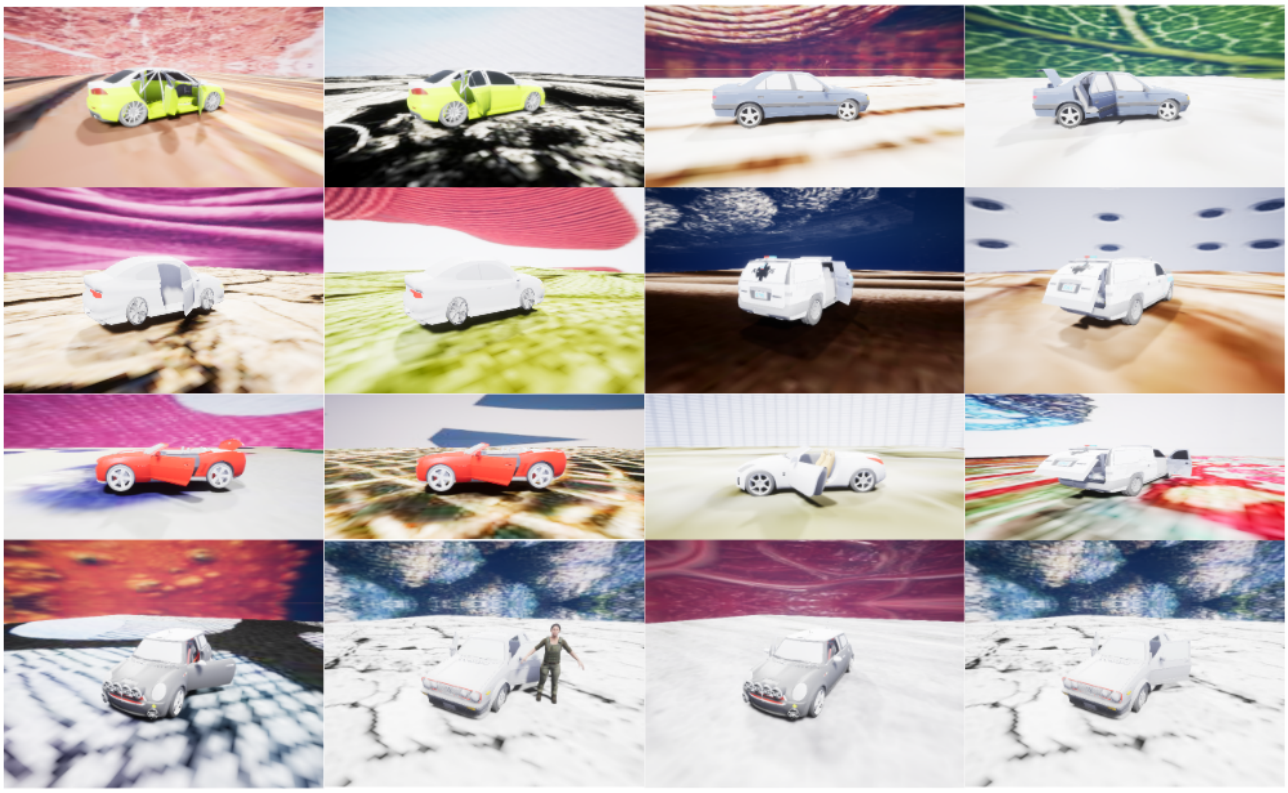

Figure 3: Sample images from SAVED. We use the original texture associated with the 3D CAD models provided by ShapeNet. SAVED captures a vehicle with randomized door/trunk states from various azimuth and elevation angles. We randomize the texture of the background.

output of the encoder $f(x)$ then becomes the input to the decoder g to predict the geometric signal $\hat{z}$:

$$\hat{z} = g(f(x)) \tag{3}$$

Then, we define the geometric loss $\mathcal{L}_g$ over all samples in the synthetic training set as:

$$\mathcal{L}_g = \sum_n d(z_n, \hat{z_n}) \tag{4}$$

where d is a distance function which produces a large scalar when differences between z_n and $\hat{z_n}$ are large. L1 or L2 norms are suitable functions for d and we use the L2 norm in our experiments. The final objective $\mathcal{L}_{final}$ to minimize is then:

$$\mathcal{L}_{final} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_g\mathcal{L}_g \tag{5}$$

where λ_{cls} and λ_g are weighting coefficients. During pre-training, we optimize the entire encoder-decoder to minimize $\mathcal{L}_{final}$. During finetuning with real images, we discard the decoder g and only use the encoder f initialized using synthetic training examples. We set λ_{cls}, λ_g to 0.5 in our experiments.

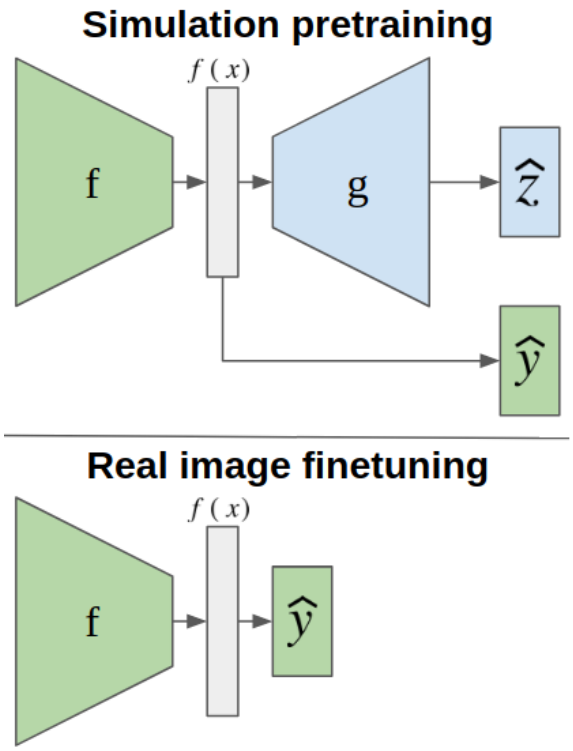

Figure 4: Top: A multi-task formulation to train a model with synthetic images where a decoder predicts geometric signals $\hat{z}$ in addition to the task labels $\hat{y}$. Bottom: The encoder trained using synthetic images is then finetuned using real images and the decoder is discarded.

5. Experiments

In Section 5.1, we show that the fine-grained vehicle part information contained in the SAVED dataset enables a model to recognize articulated vehicle parts from natural images. Then in Section 5.2, we show that we can improve existing state of the art vehicle pose estimation models by simply pretraining them on the SAVED dataset. For both tasks, we show that using synthetic data with our training strategy leads to much better results for both tasks.

5.1. Recognizing Vehicle Parts

Using the vehicle part ground truth from SAVED, we train a model for recognizing opened vehicle doors directly from an image. The ability to understand an image of a vehicle at such granularity benefits a wide array of domains ranging from autonomous vehicle applications to video surveillance.

Dataset with real vehicles with opened doors. There are no datasets with real images that have ground truth annotations on vehicle door states. To test the ability to train an 'opened-door' detector using simulated images, we manually annotate a small set of real images from the DIVA dataset[2], visualized in Figure 1.

The DIVA dataset is an activity detection benchmark adapted from the VIRAT [16] dataset with annotations regarding human actions in surveillance videos. We sample frames with action labels 'person-opens-vehicle' and 'person-closes-vehicle' and manually annotate frames with closed and opened vehicle doors. A vehicle door is 'open' when at least one door of the vehicle is opened and the deformation is clearly visible.

The DIVA dataset consists of videos captured from five independent sites with large differences in capture conditions such as viewpoint, scale of objects and levels of occlusion. The training set contains images from four scenes and the validation set consists of instances from the held-out fifth scene. There are 2144 training examples of vehicles with all doors closed and 1803 samples with at least one door opened. In the validation set, there are 309 images with 'closed' labels and 420 images with 'opened' labels.

implementation details: We adopt a U-Net [24] architecture to implement our encoder-decoder architecture. For the encoder, we use the ResNet-101 [10] model with a convolutional stem C followed by four residual blocks, R1,R2,R3,R4. Let a convolutional layer with 256 filters with spatial filter dimensions of 3 and a stride 1 be denoted as: (C-256-3-1). The decoder consists of three identical convolutional layers specified by (C-256-3-1). A geometry prediction module is appended to the decoder and it consists of two convolutional layers: (C-64-3-1)-(C-N-3-1) where N is the number of required output channels for the problem (N=3 for predicting normals, N=1 for predicting depth). There are lateral connections between the first decoder conv layer and R4, the second decoder conv layer and R3 and the third decoder conv layer with R2. Finally, the task prediction layer is a linear layer with 2048 nodes attached to the output of R4.

We implement the models using the PyTorch [18] framework. All models are initialized from ImageNet [5] pretrained weights. We optimize using stochastic gradient descent with 0.9 as the momentum term. For pretraining with simulation data, we use an initial learning rate of 0.001 with learning rate decrease with gamma of 0.9 every 10000 iterations. For finetunining, we use an initial learning rate of 0.0001 and a learning rate decrease with gamma of 0.9 every 2 epochs. For both settings, we use a small batch size of 8 due to hardware restrictions. We resize RGB inputs to square crops of 224 pixels. For the auxiliary prediction tasks, the geometric signals are resized to square crops of 56 pixels and the auxiliary predictions from the model are of same dimensions.

Results. Table 2a shows the performance of a ResNet-101 [10] model pretrained on ImageNet finetuned to our task

[2]https://actev.nist.gov/

Figure 5: We visualize the attention values of the model trained using only real iamges (left) and the model pretrained using SAVED.

	Pretrain	Train	Acc (%)
R101-E	ImgNet (R)	DIVA-Doors (R)	75.0
R101-E	ImgNet (R)	Joint (R+S)	75.3
R101-E	SAVED (S)	-	51.8
R101-E	SAVED (S)	DIVA-Doors (R)	80.5

(a) Results from an encoder only model (Res101-E) trained with only classification loss. R: real images. S: simulated images.

	Pretrain	Train	Acc (%)
R101-E	ImgNet (R)	DIVA-Doors (R)	75.0
R101-ED-N	SAVED (S)	-	52.8
R101-ED-N	SAVED (S)	DIVA-Doors (R)	85.6
R101-ED-D	SAVED (S)	-	52.3
R101-ED-D	SAVED (S)	DIVA-Doors (R)	83.7

(b) Results comparing encoder-decoder models that trained with auxiliary geometric signals using surface normals (Res101-ED-N) or depth maps (Res101-ED-D).

Table 2: Results on articulated vehicle recognition. We show that pretraining with synthetic auxiliary geometric signals greatly improves model performance on real images.

on DIVA-Doors. A naive approach for training with simulation data is to simply augment the existing real training set with additional synthetic data points. This naive joint training approach only leads to a minor improvements of 0.3 points over the train-on-real-test-on-real baseline. Instead, we observe a much more substantial performance gain of 5.5 points when we follow the pretrain-on-sim-then-finetune-on-real paradigm achieving an accuracy of 80.5%.

When the model uses geometric signals during pretraining, we observe significantly improved classification results in Table 2b. The model, which uses surface normals (R101-ED-N) to compute the geometric loss during pretraining, has an accuracy of 85.6%, a significant improvement (+10.6%) over the model trained without any simulation data. A model pretrained using surface normals as the multi-task signal outperforms the model that uses depth maps (R101-ED-D) for this application by a small margin. We suspect that the knowledge of surface normals of objects is more predictive than the knowledge of relative depth of objects in the scene. More in-depth study on the causal relation between the auxiliary geometric signal and the performance on the downstream task is an interesting topic for further investigation which we reserve for future work.

To provide qualitative assessment of the model, we additionally train models to compute simple attention masks [3]. In Figure 5, we visualize the attention mask values superimposed on vehicle images and compare the model trained using only real images to the model pretrained using synthetic images from SAVED. We observe that the attention mask of the model trained using only real images attends to regions of the image that contain noise. In comparison, the model pretrained using synthetic images produces much sharper attention masks where most of high attention values are placed on doors and trunks of the image.

In Figure 6, we visualize the auxiliary surface normal predictions, $\hat{z}$. The model has not been trained with a single instance of a real image with surface normal annotations. However, the model can still predict reasonable surface normal values for a vehicle in the natural image. Interestingly, we observe that the model learns to ignore the humans interacting with the vehicle. Combined with the attention values visualized in Figure 5, we believe the model's ability reason about the geometry of the vehicle leads to a model that can accurately reason about states of vehicle parts.

5.2. Vehicle Orientation Estimation

In the previous section, we demonstrated that the fine-grained vehicle part annotations provided by SAVED enables a model to recognize the state of vehicle parts which has not been possible before. In this section, we demonstrate that SAVED benefits existing state of the art models and improve their performance on the task of vehicle orientation estimation.

Dataset. We use the EPFL [17] dataset which is a small dataset with 20 real image sequences of 20 car types at

Figure 6: We visualize the surface normal predictions of the model. Though the model has never seen an example of a natural image paired with annotated surface normal values, it still outputs consistent and accurate surface normal predictions.

	Pretrain	Train	MeanAE (↓)
[8]	ImgNet	EPFL	23.8
Our Impl. of [8]	ImgNet	EPFL	24.4
Our Impl. of [8] + N	SAVED	-	23.4
Our Impl. of [8] + N	SAVED	EPFL	11.9
[9]	ImgNet	EPFL	9.86
Our Impl. of [9]	ImgNet	EPFL	10.1
Our Impl. of [9] + N	SAVED	-	12.3
Our Impl. of [9] + N	SAVED	EPFL	6.46
[13]*	ImgNet	EPFL	6.04

Table 3: Results on the EPFL dataset. We improve the existing state-of-the-art models using our approach and pretraining on SAVED. + N indicates that the model is pretrained with surface normals as the geometric auxiliary signal. Lower is better. * We were unable to replicate [13]

a show. We follow the settings in [9, 13] and report the mean-absolute-error (MeanAE) for evaluated models. We follow the settings presented in [9]: we use the ResNet-101 backbone pretrained on Imagenet and we take the output of the 22-nd residual layer as our visual feature for all experiments. All simulation pretraining settings are similar to the DIVA-Doors experiment. For finetuning, we follow all settings consistent with [9] and use the PyTorch framework to run experiments.

Results. We implement existing state-of-the-art methods reported for this dataset and report our replication results in Table 3. We then show that we can improve these models by attaching a decoder to predict geometric signals as the auxiliary output. We choose surface normals as our source for the geometric signal during pretraining.

In the DIVA-Door experiments shown in the previous section, a model trained only using synthetic images failed to transfer to real natural images as shown in Table 2. Interestingly, for the task of vehicle orientation estimation, we observe that a direct simulation-to-real transfer is possible meaning that a model trained without a single instance of a real natural image performs well when tested on real images. In Table 3, we show that models trained using only synthetic images from SAVED *without* finetuning on any real images from the target dataset perform on par with the model trained using real images from the target EPFL dataset.

When both existing state of the art models ([8] and [9]) are pretrained using SAVED and then finetuned using real images from the target dataset, we see significant relative improvements of 51.2% and 36.0% respectively. Just by changing the dataset used for pretraining the models without altering anything else about them, we improve the accuracy of the models by significant margins. This shows the importance of good model initialization for visual perception tasks using neural networks.

6. Conclusion and Discussion

The presented SAVED dataset is the first dataset of synthetic vehicles with articulated vehicles parts with 3D geometry annotations. Using SAVED to pretrain deep neural networks, we showed that we can recognize vehicle parts such as opened doors directly from real images. Using our multi-task formulation with geometric auxiliary signals, we obtained models that generalize to real images much more effectively. In the case of vehicle orientation estimation, a model trained using only synthetic images transferred directly to real images. Moreover, we showed that by simply pretraining existing state of the art models for vehicle orientation estimation on SAVED, we dramatically reduce the error rate.

We showed empirical evidence that pretraining with synthetic images using a multi-task learning formulation with auxiliary geometric signals improved model's performance on downstream tasks. We also showed that a model that

predicted surface normals as the auxiliary output performed better than the model that learned to predict depth values. We believe studying the effect of different geometric signals on various downstream tasks is an interesting and a promising direction for future work. We wish SAVED contributes to development of new methods for training with synthetic images and approaches for more fine-grained analysis of vehicles.

Acknowledgements. Omitted during the review process.

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. Technical report, Standford, Princeton, TTIC, 2015.

[2] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Synthesizing training images for boosting human 3d pose estimation. In *3DV*, pages 479–488, 2016.

[3] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, 2015.

[4] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[6] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[8] Kota Hara and Rama Chellappa. Growing regression tree forests by classification for continuous object pose estimation. *Int. J. Comput. Vis.*, 2017.

[9] Kota Hara, Raviteja Vemulapalli, and Rama Chellappa. Designing deep convolutional neural networks for continuous object orientation estimation. *CoRR*, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Symposium on Geometry Processing*, 2003.

[12] C. Li, M. Z. Zia, Q. Tran, X. Yu, G. D. Hager, and M. Chandraker. Deep supervision with intermediate concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[13] Xiaofeng Liu, Yang Zou, Tong Che, Peng Ding, Ping Jia, Jane You, and B.V.K. Vijaya Kumar. Conservative wasserstein training for pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[14] Jiteng Mu, Weichao Qiu, Gregory D. Hager, and Alan L. Yuille. Learning from synthetic animals. In *CVPR*, 2020.

[15] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4), Apr. 2009.

[16] Sangmin Oh, A. Hoogs, A. Perera, N. Cuntoor, Chia-Chih Chen, Jong Taek Lee, S. Mukherjee, J. K. Aggarwal, Hyungtae Lee, L. Davis, E. Swears, Xioyang Wang, Qiang Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, Bi Song, A. Fong, A. Roy-Chowdhury, and M. Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011.

[17] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[19] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *CVPR*, 2018.

[20] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.

[21] Mudassar Raza, Saeed-Ur Rehman, Peng Wang, and Bao Peng. Appearance based pedestrians' head pose and body orientation estimation using deep learning. *Neurocomputing*, 272, 01 2018.

[22] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

[23] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *NIPS*, Red Hook, NY, USA, 2016.

[24] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[25] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[26] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[27] Hao Su, Charles Ruizhongtai Qi, Yangyan Li, and Leonidas J. Guibas. Render for CNN: viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, 2015.

[28] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *CVPR Workshops*, pages 969–977, 2018.

[29] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, July 2017.

[30] David Vazquez, Javier Marin, Antonio Lopez, Daniel Ponsa, and David Geronimo. Virtual and real world adaptation for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):797–809, 2014.

[31] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014.

[32] Yi Zhang, Weichao Qiu, Qi Chen, Xiaolin Hu, and Alan L. Yuille. Unrealstereo: Controlling hazardous factors to analyze stereo vision. In *3DV*, 2018.