Deep Synthetic Viewpoint Prediction

by

Andy Thomas Hess

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

Determining the viewpoint (pose) of rigid objects in images is a classic vision problem with applications to augmented reality, semantic SLAM, robotic grasping, autonomous navigation and scene understanding in general. While most existing work is characterized by phrases such as "coarse pose estimation", alluding to their low accuracy and reliance on discrete classification approaches, modern applications increasingly demand full 3D continuous viewpoint at much higher accuracy and at real-time speeds. To this end, we here decouple localization and viewpoint prediction, often considered jointly, and focus on answering the question: how accurately can we predict, at real-time speeds, full 3D continuous viewpoint for rigid objects given that objects have been localized? Using vehicles as a case study, we train our model using only black and white, synthetic renders of fifteen cars and demonstrate its ability to accurately generalize the concept of 'vehicle viewpoint' to color, real-world images of not just cars, but vehicles in general even in the presence of clutter and occlusion. We report detailed results on numerous datasets, some of which we have painstakingly annotated and one of which is new, in the hope of providing the community with new baselines for continuous 3D viewpoint prediction. We show that deep representations (from convolutional networks) can bridge the large divide between purely synthetic training data and real-world test data to achieve near state-of-the-art results in viewpoint prediction but at real-time speeds.

To my wife Leah,
thanks for all your patience,
and,
to my son Anthony,
keep on smiling :)
and always remember...
...you can do anything you set your mind to.

Acknowledgements

Firstly, I would like to thank professor Martin Jagersand for sparking my interest in computer vision and professor Nilanjan Ray for his continued support and for inspiring me further in the areas of image processing and medical segmentation especially while taking courses and working on summer projects together.

I will always be indebted to professor Hong Zhang for his continued support, approach-ability and positivity throughout my master's degree (thank you). Speaking of positivity, I would also like to thank Ilya Sutskever for unapologetically pronouncing "It just works!" during a deep learning talk at NIPS 2014.

Finally, many thanks to the Highlevel Diner (and staff), where bowl lattes fueled many long paper reading sessions.

Contents

List of Figures

Chapter 1

Introduction

Modern applications, such as augmented reality, semantic SLAM, robotic grasping, autonomous navigation and scene understanding in general, are placing an increasing demand on vision systems to provide object viewpoint (pose) predictions at much higher accuracy than is common in the literature today. Further, these applications call out for 3D continuous viewpoint (elevation, azimuth) and at real-time speeds. For example, augmented reality or vehicle collision avoidance systems have little use for azimuth estimates within 30 or 45° but rather to the nearest degree if possible. Recent work [19] acknowledges this by reporting continuous viewpoint accuracy results.

We proceed to decouple localization and viewpoint prediction into a two-stage process and focus on the latter. We ask the question: How accurately can we predict viewpoint for rigid objects given that objects have been localized in the image? Of note is the remarkable efficacy of current deep learning systems such as R-CNN [14], which can localize objects in images with greater accuracy than once thought possible (in a viewpoint independent manner), makes this assumption even more prudent. Our hope is that by focusing on this second stage and assuming the first, as is done in fine-grained recognition [4] and text spotting [26] for example, we can improve performance of viewpoint systems in general.

In this work, we propose an accurate, real-time, 3D continuous viewpoint prediction system for RGB images of rigid objects, trained solely on deep representations of synthetic data, amenable to real-time scene understanding. Our approach is applicable to rigid objects *in general* but we restrict ourselves here to the *vehicle* class. Our contributions are as follows:

i) We show that deep representations allow us to bridge the divide between synthetic and real-world images allowing for high levels of viewpoint accuracy even in the presence of occlusion and clutter.

ii) Continuous 3D viewpoint prediction as a single, matrix-vector multiply amenable to real-time GPU computation.

iii) Datasets. A new synthetic and real-world dataset well-suited for studying how synthetically-trained systems perform in the real-world. Furthermore, we annotate 3D viewpoint for existing datasets, one of them ideal for the study

of viewpoint under occlusion.

To the best of our knowledge, we are the first to focus exclusively on real-time, 3D continuous viewpoint prediction given localization by training on purely synthetic images using deep representations. We refer to our approach as Synthetic Viewpoint Prediction (SVP).

This thesis begins with a discussion of deep representations, their exponential efficiency as compared to (shallow) engineered features, and finally, generic convolutional neural network representations and how they are used in our system. This is followed by an introduction to the viewpoint prediction problem in the context of related work.

We then discuss the numerous contributions of this work followed by experimental results including dataset details and applications and conclude with a discussion of future work.

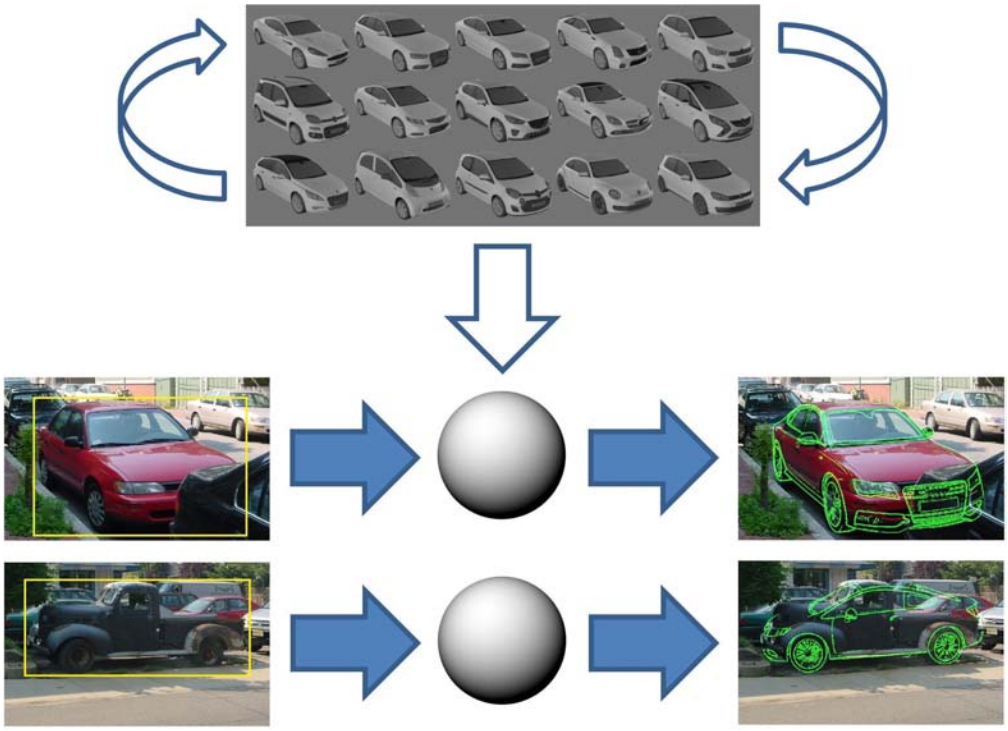

Figure 1.1: Synthetic Viewpoint Prediction (SVP) overview: Continuous, real-time, 3D viewpoint prediction even under occlusion. Large amounts of synthetic, viewpoint-dense training data is generated from 15 CAD models of cars (top) and used to train our model (down arrow symbolizes training). At test time (solid arrows), viewpoint is predicted using contents of bounding box. Representations used are deep. Red car (middle) from image SSDB01312 (SSV+ dataset) has ground truth GT=(elevation, azimuth)=(ϕ, θ)=$(13, 248)$ with predicted viewpoint P=$(15, 247)$ visualized by superimposing associated synthetic view of 2012 Audi S7 Sportback. Black truck (bottom) from image SSDB01575 (SSV+ dataset) has GT=$(14, 350)$ with P=$(18, 351)$ showing viewpoint generalization outside vehicle types seen in training set. Best viewed under pdf magnification.

Figure 1.2: Accurate viewpoint in the presence of occlusion, illumination change and clutter. Top: Image SSDB00271 from SSV+ dataset showing remarkable robustness given the extreme specular highlights. Here, ground truth GT=(elevation, azimuth)=(ϕ, θ)=$(11, 247)$ with predicted viewpoint P=$(16, 240)$. Bottom: High-clutter image SSDB00271 from SSV+ dataset with ground truth GT=(ϕ, θ)=$(11, 60)$ with predicted viewpoint P=$(19, 60)$. Best viewed under pdf magnification.

Part I

Background and Literature Review

Chapter 2

Deep Representations

2.1 Overview

Since the turn of the century, computer vision has witnessed tremendous progress in the area of generating good representations (features) for visual data. At the core of this research push is the growing realization that good representations are not only an intermediate stage of a multi-stage path toward the solution, but rather, the *elemental foundation* upon which any solution rests. In fact, solutions themselves are just representations. For example, whether an image contains a dog or not can be represented with a single bit.

The rising prominence of the role of representation in computer vision was predominantly expressed in the previous decade in the context of hand-engineered representations [7, 38, 37] with a great deal of research directed toward hand-crafting increasingly better features for specific tasks. However, progress using this approach slowed in the early years of the current decade (for example in object detection [10, 47]).

Alongside these developments in hand-crafted visual representations, others were looking into learning representations from the data itself. In 1986, in the seminal work [54], Rumelhart, Hinton and Williams wrote: "We demonstrate that a general purpose and relatively simply procedure is powerful enough to construct appropriate internal representations." This drive toward learned representation was fueled by the increasing interest in end-to-end learning systems which was thought to naturally embody, at least in spirit, a move toward artificial intelligence. In later years, up to about 2012, progress in learning internal representations came in the form of systems such as Autoencoders [23] and Restricted Bolzmann Machines [22] being two examples.

During these investigations into systems that learned their own internal representations, one thing became increasingly clear; multiple layers of representations often performed much better than one or few layers [2]. This insight was also supported by knowledge of the human visual cortex which has long been known to consist of many layers exhibiting increasingly complex function (V1, V2, etc.).

Regarding terminology, a *deep* system is simply one with multiple layers of (learned) representations whereas a *flat/shallow* system is one with a single (or few) layers. Hand-crafted features are also generally referred to as *flat* or *shallow* as they generally do not build representations in a hierarchical manner based on lower level representations. Paramount to this concept of *deep system* is the notion of abstraction; the idea that higher level representations implicitly encode lower level representations into higher levels of abstraction. This distinction is an important one as hierarchy by itself is not sufficient when referring to deep systems.

In 2012, the work of Krizhevsky, Sutskever and Hinton [32] forever changed the landscape of modern computer vision. In this work, the authors showed that a Convolutional Neural Network (CNN) [11] with large enough capacity, using Rectified Linear Unit (ReLU) activations [40], GPU-trained with Dropout [24] and large amounts of training data surpassed all state-of-the-art recognition systems at that time by substantial margins. Since the publication of this work, academia and industry have rapidly embraced CNNs and end-to-end learning systems in general. For example, all ImageNet challenge winners since 2012 have used CNNs [47].

2.2 The Exponential Efficiency of Deep Models

The tremendous success of deep systems have led many to ask the most natural question: "Why do deep systems perform so much better than shallow ones?"

It has been known since 1989 that neural networks with a single hidden layer (using sigmoid activations) are universal function approximators of continuous functions on compact subsets of $\mathbf{R}^n$ [6]. [25] later generalized this result by showing that the type of activation function is not important.

However, although any continuous function can be approximated by a single layer, these results say nothing about *how many nodes* are needed in this layer. It was shown in 2011 that there exist polynomial functions that can be computed using a deep sum-product network that require exponentially more units using a shallow sum-product network [8]. This work, showing the exponential efficiency gain of using deep networks pointed the way to more recent investigations into the representational power of modern networks that use ReLU activations [40] for example.

In 2013, [43] showed that given a fixed number of nodes, a deep ReLU network, in the limit of the number of hidden layers, can separate the input space into exponentially more regions than a shallow ReLU network can. More recently, [39] investigate the representational power of deep networks in terms of number of linear regions *in general* and find that as the number of layers increases, the number of linear regions increases exponentially. In this work they write: "The layer-wise composition of the functions computed in this

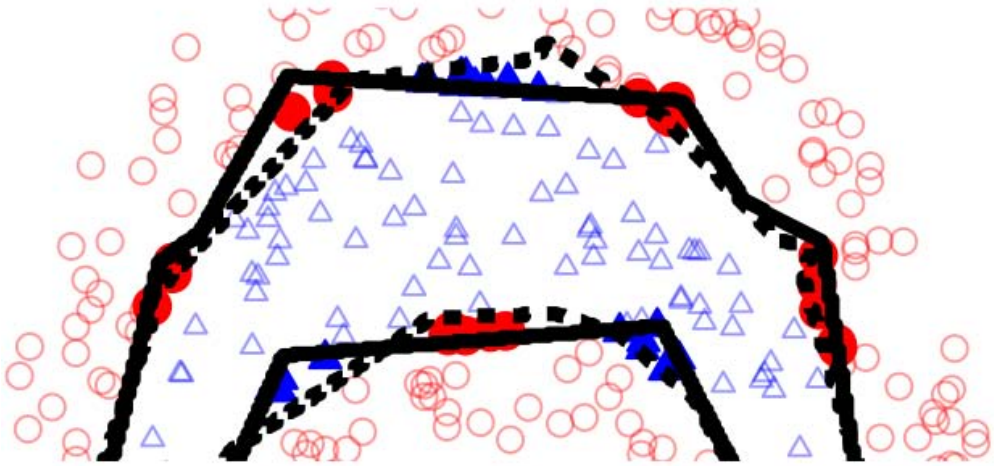

Figure 2.1: Example showing the advantage of depth for two-dimensional binary classification (circles/triangles). Solid line represents decision boundary for an MLP with a single hidden layer containing 20 units. Dashed line represents decision boundary for an MLP model with two hidden layers having 10 units each. Solid markers are incorrectly classified by shallow model but correctly classified by deeper model. Figure taken directly from [39].

way re-uses low-level computations exponentially often as the number of layers increases."

Indeed, it is the *compositional* structure of deep networks that leads to their exponential efficiency. A feature of an intermediate layer for example, can assimilate and re-use the computations of possibly an exponential number of lower level features and herein lies the power of deep networks.

As a more concrete example, consider a deep network trained for face recognition (although hypothetical, current trained CNNs exhibit hierarchical structure of exactly this nature). First layer features may represent edge-like features while second layer features may represent curves or other general shapes. At deeper layers, features may evolve during training to represent face parts like eyes, mouths or noses for example. Yet deeper layers would combine these parts to form portions of the face (left eye and nose together) until reaching a layer which would represent an entire face.

In this scenario, learned representations for eyes at a deeper layer would rely on multiple shape features learned at the second layer and each of these shapes would in turn rely on multiple edge features from the first layer. This effect is compounded for even deeper layers (for example, a layer containing representations of entire faces). One can see here that one specific edge feature in the first layer could be *re-used exponentially often* as network depth increases. This exponential depth advantage, absent from hand-engineered flat models, provides in large part, justification for the tremendous recent success of deep models.

Figure 2.2: Correlations across channels tend to be local. This is true for a color input image with three channels (as above) but also true at deeper layers where correlations tend to occur across previous layers' output channels each of which being expressions of more abstract features. Photo: Flickr (original image followed by red, green & blue channels).

2.3 Convolutional Neural Networks & Generic Feature Representations

Using a fully-connected neural network, where each pixel is connected to each unit is computationally prohibitive for all but the smallest of images. Moreover, local image statistics are often the same in different parts of an image so there is generally no need to model every spatial location separately. For example, given a collection of images of cats, a cat head could generally appear anywhere in the image; cat head features are generally translation invariant.

An early neural model incorporating this translation invariance was the Neocognitron [11]. This work, the first designed with *local connectivity* as an integral part of it's architecture, can be considered the birthplace of convolutional neural networks (CNNs).

Local connectivity in CNNs is implemented as convolution in the forward pass by considering the filter weights to be the same across all local spatial regions (*weight tying*). In this way, learned filter weights represent translation invariant features and at a fraction of the memory required compared to a fully-connected network. Most importantly, this huge reduction in network capacity (number of learnable parameters) leads to networks far easier to train with less risk of overfitting.

Also of note is that different channels of a single image tend to exhibit high local correlation (Figure 2.2). It is precisely this *spatial relationship* across channels which forms the motivation behind CNN filters being 3-dimensional. In this way, learned filters, at any depth, seek to summarize local correlations across all channels of the previous output layer. In the first layer, this convolution could be across color channels say; at deeper levels, across more abstract features.

After training [33, 32], these learned filters have been shown to represent increasingly abstract concepts at increasing depth [9, 14]. This hierarchical,

end-to-end feature learning approach contrasts sharply with shallow, engineered approaches as learned features are *emergent* from the training data and task at hand. In this context, these learned internal representations [54], customized to minimize the specified task's loss function for the training data presented to the network, have been shown to outperform almost all hand-engineered features for most vision tasks [5, 46].

Further, it is remarkable that these internal feature representations have been shown to be highly *transferable* [42] to varying tasks and to datasets different from those presented to the network at training time [32, 9, 59, 14, 20].

Following this *transfer learning* approach, we make use of *generic feature representations* for our task of viewpoint prediction using the pre-trained network [32] with implementation [9] (Figure 2.3). The use of the term *generic* refers to the fact that a pre-existing trained model, under the assumption of it being useful in the context of transfer learning, can be used to represent *any* input image. Learned (internal) representations that were useful for this model to perform well on the specified 1000-way classification task, given the 1.2 million images in the training set, have been shown to be useful for other tasks and datasets [9]. In this work, we use generic representations from multiple layers to represent both synthetic training images as well as real-world images.

Specifically, we utilize representations taken from three separate layers of this network, namely the 5^{th} pooling layer ($pool_5$) as well as the two fully connected layers (fc_6 & fc_7). After an input image is re-sized to 227x227 (a requirement for this architecture), the image if forward-passed through the network, resulting in activation values at all layers from which these representations are read.

After the 5^{th} convolutional layer (of dimension 13x13x128x2 = 43,264), overlapped max-pooling with 3x3 kernel and stride 2 is applied. This results in a layer of size 6x6x128x2 = 9,216 called layer $pool_5$ (after ReLU is applied). Note that this pooling layer is not shown separately in Figure 2.3. The reason we did not use earlier convolutional or pooling layers is because they were too large for our purposes and took up too much disk space. The fully-connected layer representations fc_6 & fc_7, each of size 2,048x2 = 4,096, were also read directly from the network after the forward pass.

In this work, we utilize these three generic deep representations and seek to bridge the divide between synthetic training data and real-world data for the purpose of continuous viewpoint prediction. Before doing this however, we introduce in the next section the viewpoint prediction problem in the context of related work.

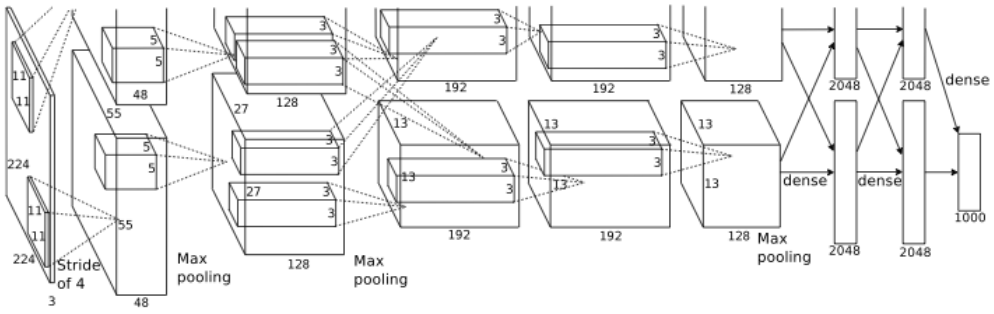

Figure 2.3: The canonical Krizhevsky, Sutskever & Hinton architecture [32], trained on ImageNet data, from which we extract generic representations from the last three layers for the purpose of viewpoint prediction (image from [32]). The $pool_5$ layer representation proceeds the 5^{th} convolutional layer (4^{th} layer from right above) and is the concatenation of all activation values (after stride-2, 3x3 max-pooling & ReLU [40]) giving a vector of dimension 6x6x128x2 = 9,216. The following two fully-connected layer representations used in this work (after ReLU is applied), fc_6 & fc_7, are each of dimension 2048x2 = 4,096.

Chapter 3

Viewpoint Prediction

3.1 Related Work

Previous viewpoint work, in the face of the multi-view recognition challenge, has striven to appeal to the inherent 3D nature of the problem in a number of ways. One approach has been to learn implicit 3D representations using keypoint-labeled, real-world images [1, 50]. Other works appeal directly to 3D CAD models which serve to connect 2D images with their 3D identities [35, 44, 60, 56, 45, 21, 57]. Recently, [19] utilizes RGB-D data along with 3D CAD models.

One can also consider previous work from the perspective of viewpoint resolution. Often characterized as "coarse viewpoint/pose estimation" or "viewpoint/pose classification" alluding to the discrete approach of training a separate model for each of a number of large azimuth ranges [48, 50, 1, 34, 60, 16, 36, 17, 56, 45, 13] and reporting on the 8-view dataset [48]. [41, 12, 44, 57] strive for more resolution by increasing the number of azimuth bins. This approach, of training a separate model for every view, becomes increasingly difficult, if not intractable, as we move toward 1° resolution of 3D viewpoint (elevation, azimuth).

Recent work acknowledges that a far greater level of accuracy and full 3D viewpoint is required to be useful for modern applications [19]. We prefer "viewpoint" over "pose" (as in [52]) since the former pays respect to the center role of the camera and distinguishes it from deformable work such as [15]. Further, with increasingly higher levels of accuracy, we prefer "prediction" over "estimation".

Inextricably entwined, object recognition and viewpoint prediction have naturally matured together. One prominent tack has been to approach the subject from a local perspective; with models that consider parts or keypoints of objects rather than the object as a whole [48, 50, 34, 16, 60, 36, 17, 45, 44, 21, 52]. However, recent work in deep learning has pointed to another way; where intra-class variation and viewpoint variation can be *implicitly* expressed in the language of learned (deep) representations rather than *explicitly* with

parts or keypoints.

The startling effectiveness of recent deep learning systems has taken many by surprise. Human-level performance in face recognition [49]; reCAPTCHA solved to 99.8% accuracy [18]; highly accurate classification on over 90K classes for natural scene text recognition [26]; not to mention recent image-captioning work [29, 53]. The now-seminal work [32] inspired success in large scale image recognition to the point where all recent ImageNet winners utilize convolutional neural networks (convnets) [47]. Large portions of industry and academia have already switched, or are in the process of transitioning, from feature engineering to neural network engineering. Custom, deep learned features that emerge from the context of data and task are proving consistently richer and more effective than shallow hand-crafted ones [5, 46].

Another surprise has been how transferable these rich features are to tasks outside those experienced during training (ie. transfer learning) [58, 46, 42]. Therefore, in this work, we use generic, pre-learned deep representations taken from various layers of the convnet architecture of [32], as implemented by [9], acknowledging that [28] gives equivalent representations but in as little as 2ms per image on appropriate GPU hardware.

For our purposes, it is of utmost importance to note that modern deep learning methods *have made tremendous gains* on the multi-view and intra-class variation problems that inspired part and keypoint related work of the past. Current work, including ours, is following suit, leveraging the power of learned representations [52, 13, 19].

However, current (supervised) deep learning methods often require large quantities of labeled data to be effective; their large capacities often struggle with overfitting. This issue is amplified for viewpoint prediction as it's difficult for humans to accurately annotate images with 3D viewpoints.

On the other hand, synthetic training data can be generated to any scale required with 100% accurate labels whether it be class or viewpoint [60, 18, 19, 27, 26]. Moreover, synthetically generated data is perfectly suited to satisfy the hunger of modern deep learning systems [32, 14, 51].

An existing challenge remains, however; the gap between synthetic training and real-world testing is often too large to be effective as synth-trained models have not experienced the wide range of background noise, clutter, shadows and illumination found in the real world. A main point of this work is to show that, at least for viewpoint prediction, the advantage conferred by being able to generate any amount of perfectly-labeled training data *can bridge this gap* between synthetic and real-world image distributions (Figure 3.2).

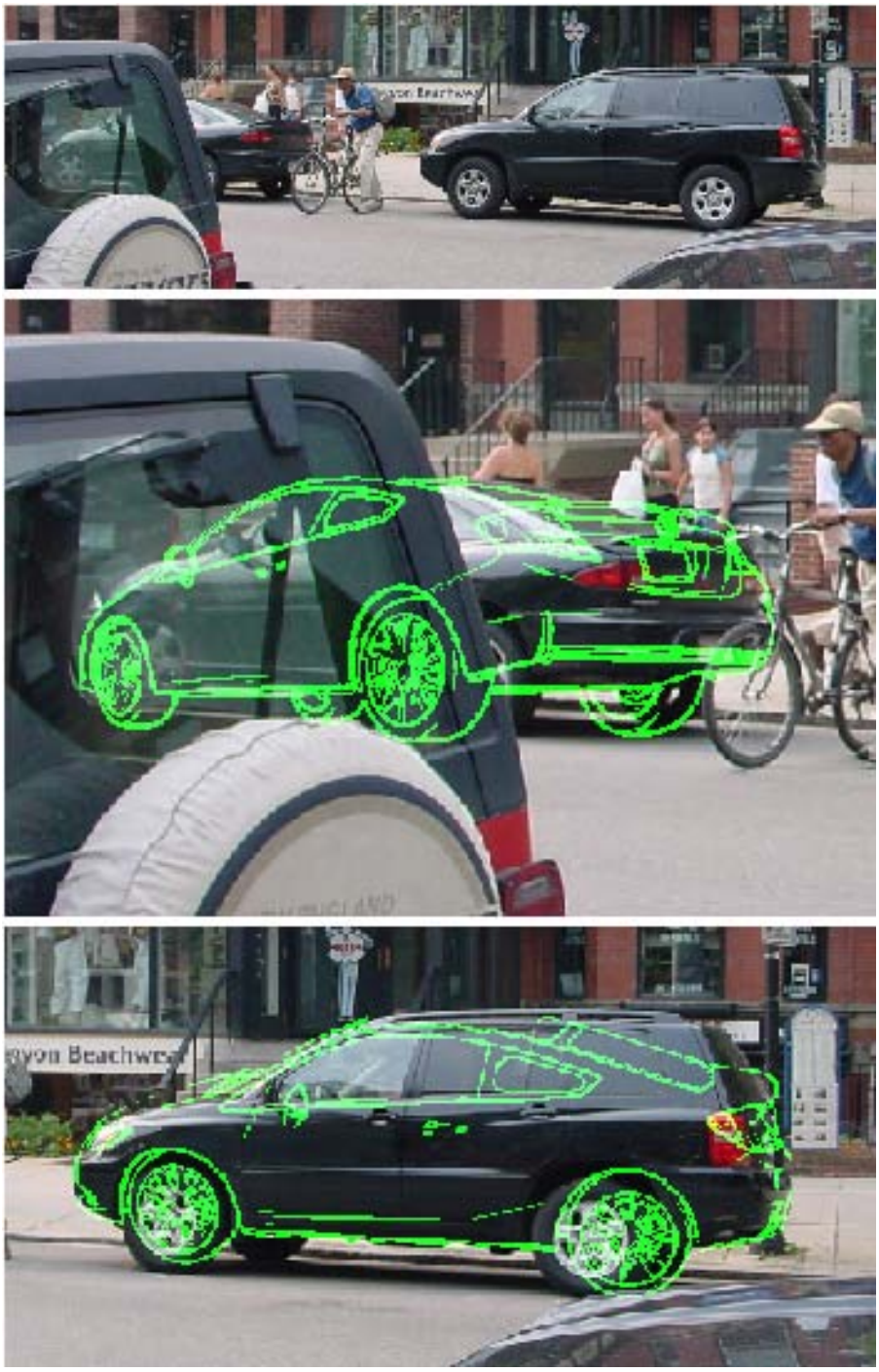

Figure 3.1: Viewpoint predictions for image SSDB02133 in SSV+ with and without occlusion. Middle: Left vehicle in top image (under heavy occlusion) has ground truth GT=(ϕ, θ)=$(3, 41)$ with prediction P=$(2, 52)$. Bottom: Right vehicle in top image has GT=$(5, 25)$ with P=$(17, 13)$. For visualization, viewpoint predictions superimposed using corresponding synthetic views of 2012 Honda Civic Coupe (green).

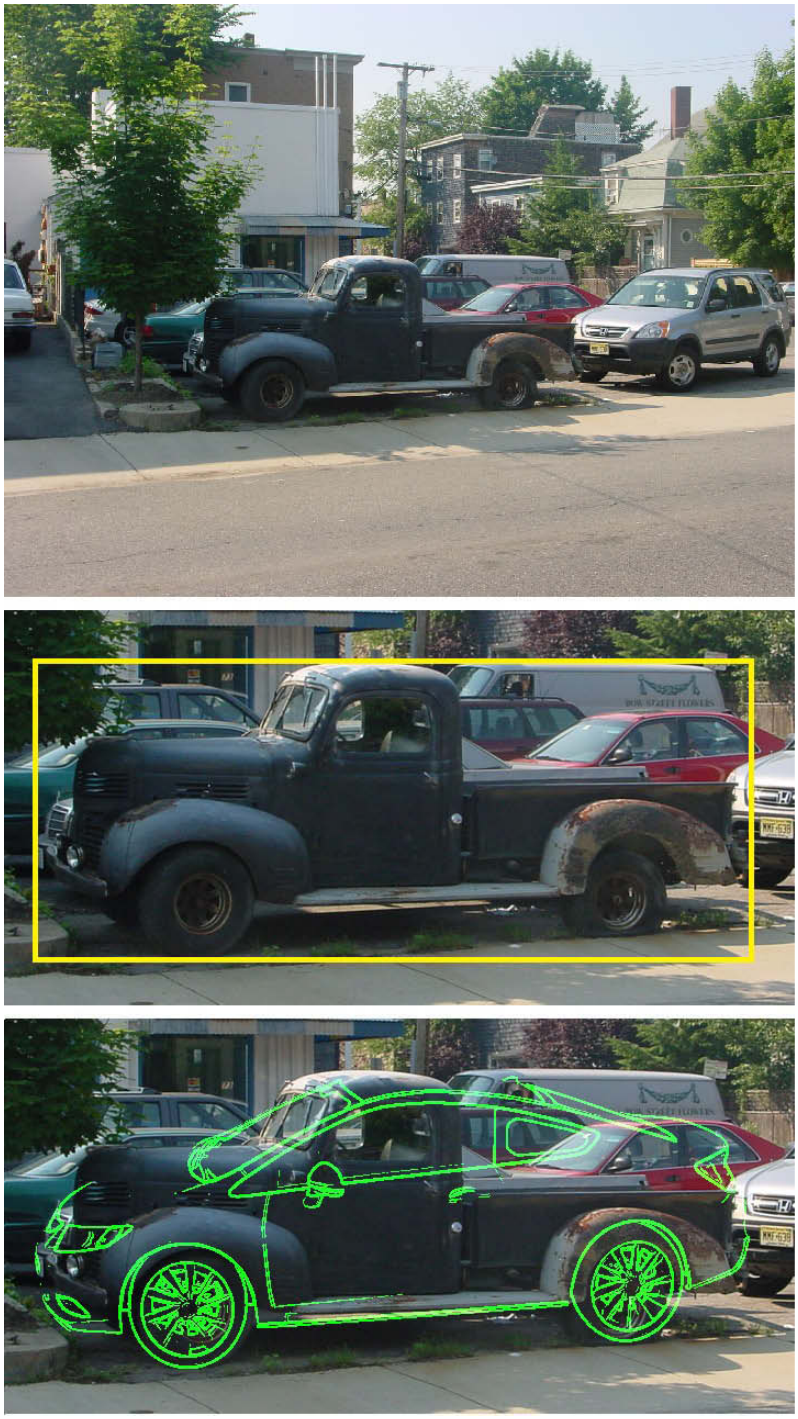

Figure 3.2: Remarkable viewpoint generalization to real-world unseen vehicle types using synthetic training data from 15 cars only. Top: original image 01575 from SSV+ dataset. Middle: zoomed-in view of truck with bounding box (annotated ground truth viewpoint is 14° elevation & 350° azimuth). Bottom: predicted viewpoint of 18° elevation & 351° azimuth shown as green edge overlay from a synthetic render from this viewpoint.

Part II

Contributions

Chapter 4

Deep Synthetic Viewpoint Prediction

4.1 Overview

The machine learning approach to solving problems often requires appropriate amounts of training data to be effective; certainly the training data must contain information representative of what one expects to find at test time. Viewpoint prediction, at any level of accuracy greater than simple binning approaches mentioned in the previous chapter, would require training data at least somewhat dense in the prediction space. However, existing datasets are often biased toward only a few viewpoints (Figure 6.7) and do not contain enough data to be effective for continuous viewpoint prediction.

In this work we turn to generating large amounts of dense, synthetic training data by rendering images of the object class of interest. A major contribution of this work is showing that *training on synthetic data and testing on real-world data* can be effective for a computer vision task such as viewpoint prediction. Somewhat surprisingly, this is true even though the training data (here in b&w with no backgrounds) has a seemingly very different distribution than the color, real-world images used at test time.

Emphasis should be placed on the fact that the proposed approach is not limited to the vehicle object class but to *any object class* where CAD models are available.

Another contribution of this work is showing the effectiveness of deep representations for helping to bridge this gap between synthetic and real-world image distributions. We compare pre-trained CNN representations from various layers with traditional HOG [7] features and show a large increase in viewpoint prediction accuracy.

This improvement in accuracy is especially pronounced with regard to viewpoint prediction in the presence of occlusion. This forms another point of this work; showing that our proposed system can provide accurate viewpoint even when objects are occluded and/or images contain large amounts of background

clutter.

Another contribution is showing that an increase in training data leads to an increase in viewpoint prediction accuracy. This scale-ability is a highly desirable attribute for our system as generating increasing amounts of training data is as simple a loading in more models into our rendering system. Of note is that the results in this work are obtained by using *only fifteen CAD models* of cars.

Further, the proposed system produces continuous 3D viewpoint predictions as a single, matrix-vector multiply amenable to real-time GPU computation. This statement contains multiple contributions: 1) continuous viewpoint rather than binned viewpoint 2) viewpoint in our case refers to elevation and azimuth whereas most previous work is restricted to azimuth only 3) a system amenable to real-time viewpoint prediction.

Dataset contributions: A new synthetic and real-world dataset well-suited for studying how synthetically-trained systems perform in the real-world. Furthermore, we annotate 3D viewpoint for existing datasets, one of them ideal for the study of viewpoint under occlusion.

To the best of our knowledge, this is the first work to focus exclusively on real-time, 3D continuous viewpoint prediction given localization by training on purely synthetic images using deep representations.

4.2 Approach

Driven to highlight the effectiveness of modern learned representations (convnets) for viewpoint prediction under real-time constraints, we purposefully sought out the simplest machinery that still achieved high levels of accuracy.

Synthetic b&w images from SRV15 (see 5.1) are cropped and re-sized to 227x227 using each image's corresponding foreground mask. We perform bounding box dilation (context padding) as in [14] with p=3 ensuring that the entire object, including its edges, are fully contained in the image. Lastly, we stack three copies of this b&w image resulting in an RGB image (a requirement for input to the architecture of [32]). Real-world (bounding box) images are processed the same way.

We leverage learned representations from layers $pool_5$, fc_6 and fc_7 from the generic pre-trained network [9, 32] resulting in features of dimensions 9,216, 4,096 and 4,096 respectively. Any image, synthetic or real, is forward-passed through the network and the appropriate layer extracted. We further normalize every vector in anticipation for comparison using cosine distance.

Our three synthetic models, corresponding to each network layer ℓ above, consist solely of representations of all synthetic images from SRV15 (see 4.1 below), normalized to the unit ball, each one forming a single row of matrix M_ℓ. Each row i of M_ℓ has an associated 3D viewpoint label $v[i] = (\phi, \theta)_i$. At test time, we pass image I through the network, extract the corresponding representation r_ℓ, normalize, and compare with the synthetic model using

cosine distance, 1-nearest neighbor (1-NN):

$$V_\ell(I) = (\phi, \theta)_\ell(I) = v[argmin(\underline{1} - M_\ell r_\ell^\intercal)] \tag{4.1}$$

since cosine distance is $1 - a \cdot b$ when $\|a\| = \|b\| = 1$.

Suited for GPU computation, viewpoint prediction for a single test image is effectively reduced to a single matrix-vector multiply. Here, for layer $pool_5$, an NVIDIA TESLA K40 GPU, capable of 4.29TFLOPS (single precision) can easily handle the approximately 1GFLOP of multiply-add operations per prediction. On the same GPU, convnet implementations such as [28] can forward-pass in as little 2ms. Although we used a CPU implementation [9] for the results in this work, we submit as self-evident that these facts imply that SVP is clearly amenable to real-time GPU computation.

Part III

Experimental Results

Chapter 5

Datasets

We concern ourselves here with datasets containing vehicles annotated with bounding boxes and continuous 3D viewpoint. Contributions of this paper include a new synthetic-real vehicle dataset as well as precise 3D continuous viewpoint annotations for existing datasets. Viewpoint annotations were carefully produced by manually aligning synthetic models within the extents of object bounding boxes with new software written expressly for this purpose. We intend to release our new dataset as well as new viewpoint labels for other datasets upon publication.

SRV15: We introduce here our new Synthetic and Real Vehicle (SRV15) dataset containing 15 fine-grained classes of cars (Figure 5.1). Each class is represented by 7,560 high-res, b&w renders (with texture) using `blender.org` on CAD models from `doschdesign.com` rendered at 1° resolution between elevations 2-22° at all azimuths. Each class is also represented with corresponding real-world color images, each annotated with accurate bounding box and continuous viewpoint (elevation, azimuth). In total, 113,400 synthetic images and 2,726 real-world images that are of much higher resolution than commonly found in other recognition datasets. This dataset is ideal for the study of vision systems utilizing 3D synthetic training data for the purposes of fine-grained recognition and/or viewpoint prediction. We refer to the synthetic and real-world portions of SRV15 as SRV15-S and SRV15-R respectively.

One of the main themes of this work is to study the effectiveness of deep representations for the task of bridging the divide between synthetic and real images, in our case for viewpoint prediction (synthetic viewpoint prediction). One aspect is to study this effectiveness when the real-world test images are of the *same object class* as the synthetic test data; this dataset (SRV15) provides the means to study this question.

As mentioned above, this dataset could also be used to study other vision problems such as fine-grained classification for example, but we restrict this work to viewpoint prediction. In our case, even though vehicles are labeled with vehicle class, we are only concerned with viewpoint prediction across object classes.

A more general aspect of this work is to study generalization to real-world

images *of different fine-grained classes* of vehicles. In this case, we use only SRV15-S for training data but test on images of any type of vehicle often far different than what is found in the training set. For example, SRV-15 contains only 15 sedan-type cars where other datasets (below) contain images of trucks, vans, jeeps etc.

In summary, SRV-15 is used to study two aspects of viewpoint prediction from synthetic to real-world images. The first being viewpoint prediction when the real-world images look like, and are indeed, the same type of vehicles as found in the training set (from a fine-grained perspective). The second aspect, is to train on only SRV-S and study viewpoint prediction for real-world images that are often quite different from a fine-grained perspective. One would expect that the first aspect leads to more accurate results and our experiments show this.

SSV+: The "StreetScenes" dataset (SS) [3] contains many quality images with high levels of occlusion and clutter indicative of urban scenes. Originally constructed to study scene understanding in general, images were labeled with nine object classes commonly found in urban scenes such as vehicles, people, trees, pavement etc. In our work, we are interested in studying viewpoint prediction of only the vehicles found in these images as our synthetic training data covers only this class.

We leverage these excellent images by extensively re-annotating vehicles with accurate bounding boxes, 3D viewpoints as well as whether each vehicle is occluded or not. Of note is that the original bounding box labels included in the original SS dataset were so noisy as to be of little use to us; many labels did contain a vehicle at all or vehicles taking up only a small percentage of the bounding box for example. In any case, labeling each vehicle with fine-grained pose proved to be far more time-consuming compared to bounding boxes anyway.

In this work, we refer to the existing SS dataset along with our new vehicle-only labels as SSV+ (Street Scenes Vehicles). Our contribution here is limited to labels only with full image credit and acknowledgment to [3]. We believe SSV+ to be the largest and most accurate labels for 3D viewpoint prediction of vehicles under occlusion currently available; 1,804 occluded (SSV+O) and 2,837 unoccluded images (SSV+U) for a total of 4,641 images.

PASCAL3D+ [55]: PASCAL VOC 2012 [10] is augmented with ImageNet [47] images and annotated with continuous viewpoint for 12 different rigid classes. This dataset contains an average of 3000 instances per category, contains non-centered images as well as images exhibiting occlusion and clutter.

Viewpoint annotation was facilitated using their custom annotation tool. After the object in question is labeled with a tight bounding box from the real-world image, the user selects one from a selection of 3D models corresponding to the object class in question where each 3D model has been previously annotated with a fixed set of 3D landmarks (keypoints). For example, for the vehicle class, each 3D model would be labeled with the same set of 3D land-

marks such as front left light, front left tire etc. Different objects of course have a different fixed set, and different number of 3D labeled keypoints. The 3D model class is manually selected as to best match the real-world image in question.

Next, the user specifies as many visible landmarks on the real-world image as possible and labels each one with a corresponding 3D landmark label (left front light for example). The viewpoint is then estimated via optimization, assuming canonical camera intrinsics, by minimizing the re-projection error over the set of labeled landmarks.

We report in this work viewpoint results for the "car" class of the PASACAL validation dataset.

MVT [57]: 1,623 frames from 9 YouTube and 11 KITTI [12] videos with cars exhibiting significant viewpoint change for the purpose of multi-view tracking (MVT). Cars are annotated with elevation & azimuth.

KITTI [12]: An extensive autonomous driving dataset, one small part of which contains azimuth-only annotated cars. We use training images with cars marked as fully visible and not truncated (11,017 cars from 7,481 images) and report azimuth-only results.

EPFL [41]: Twenty different rotating cars photographed during a car show from static camera positions (2,299 images total). We report azimuth results on the full dataset.

3DObject [48]: A classic, carefully crafted, 8-viewpoint dataset containing 10 different object categories. We re-annotate all 480 car images with continuous elevation & azimuth. Note that [60] also provide continuous 3D annotations by aligning 3D CAD models but only for a single car.

Cars196 [30]: This fine-grained vehicle dataset, originally with 197 classes, was later condensed to 196 as two vehicles were found to be the same. This dataset does not come with viewpoint labels so we present only qualitative results in figure 6.17.

Figure 5.1: Top: Synthetic black and white renders from a single viewpoint of the 15 car classes found in SRV15 dataset (SRV15-S). First Row: 2010 Aston Martin Rapide, 2013 Audi A4 Avant, 2012 Audi S7 Sportback, 2011 Cadillac CTS-V, 2011 Citroen C4. Second Row: 2013 Fiat Panda, 2012 Honda Civic Coupe, 2013 Mazda CX-5, 2012 Mercedes SLK350, 2012 Opel Zafira Tourer. Third row: 2011 Peugeot 508, 2011 Peugeot iOn, 2012 Renault Twingo, 2012 VW Beetle, 2010 VW Golf GTD. Bottom: Examples of corresponding real-world images of each car class found in SRV15 (SRV15-R). Note: all images, synthetic and real, are annotated with bounding boxes and viewpoint (elevation, azimuth).

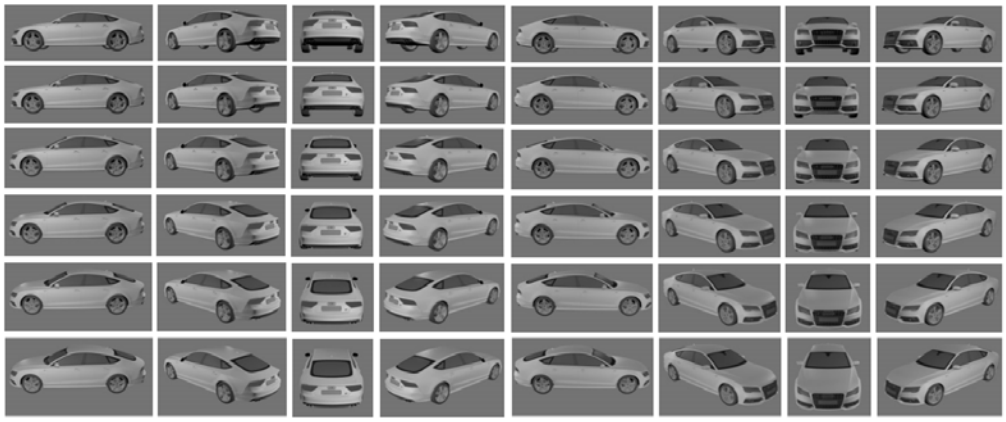

Figure 5.2: Some examples from the 7,560 high-res, b&w renders used as training data for the 2012 Audi S7 Sportback class in SRV15-S. Each column represents a change of 45 degrees azimuth and each row a 4 degree change in elevation.

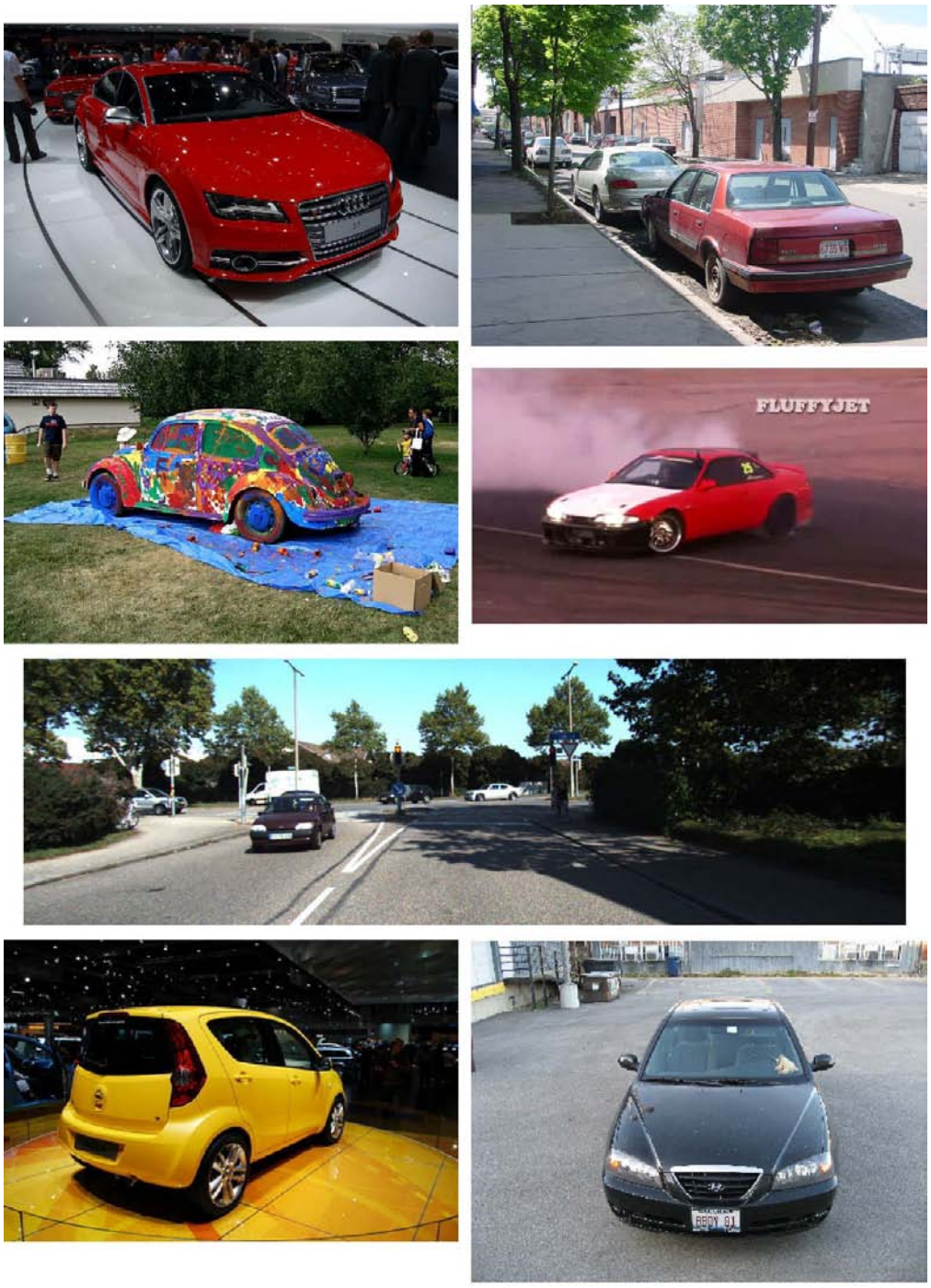

Figure 5.3: Example images from various datasets used in this work. First row: SRV15-R, SSV+. Second row: PASCAL3D+, MVT. Third row: KITTI. Fourth row: EPFL, 3DObject.

Chapter 6

Viewpoint Prediction

6.0.1 Convnet Representation Layer

We first compare viewpoint accuracy on SRV15-R based on from which layer ℓ we extract convnet representations ($pool_5$, fc_6 or fc_7). Figure 6.1 confirms that layer $pool_5$ performs best which we expected since known bounding boxes imply a stronger image alignment model [4]. Therefore, from this point forward, we use only $pool_5$ features.

6.0.2 Viewpoint Prediction

We concern ourselves with 3D viewpoint prediction (elevation, azimuth) and report results in Table 6.1. Following [19], we report continuous *accuracy at* θ rather than at a single discrete angle [52]. *Accuracy at* θ is defined as the percentage of correct predictions within θ of ground truth. As well, we report *median azimuth error*. Continuous results are shown in Figures 6.5 & 6.8, the latter focusing on SSV+ occlusion results. Qualitative results on the unlabeled Cars196 dataset can be found in Figure 6.17.

Not surprisingly, we achieve the highest level of accuracy on dataset SRV15-R which corresponds to the same car classes found in our model's training set. Surprising though is that we achieve almost 100% accuracy here (even 96.2% correct within 5° azimuth) which forms the basis of our claim that our deep (convnet) representations have successfully bridged the gap between synthetic and real images. Results on other datasets are also surprising accurate (Figure 6.16).

Although PASCAL3D+ contains some images with occlusion, we focus on occlusion results separately for dataset SSV+ in Figure 6.8. Of note here is the relatively small accuracy degradation between unoccluded to occluded images. See Figures 3.1 & 6.16 for occlusion examples.

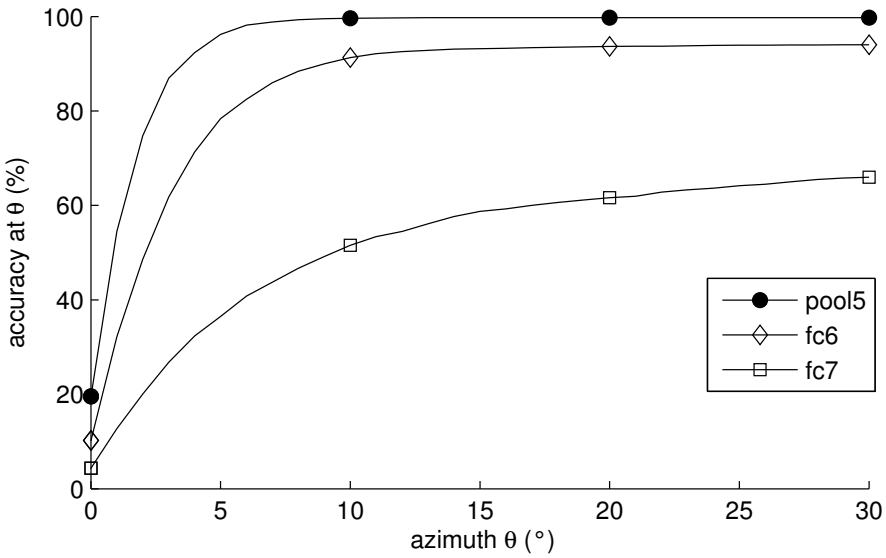

Figure 6.1: The effect of convnet representation layer on viewpoint azimuth accuracy for SRV15-R. Accuracy at θ is defined as the percentage of predictions within θ° of ground truth.

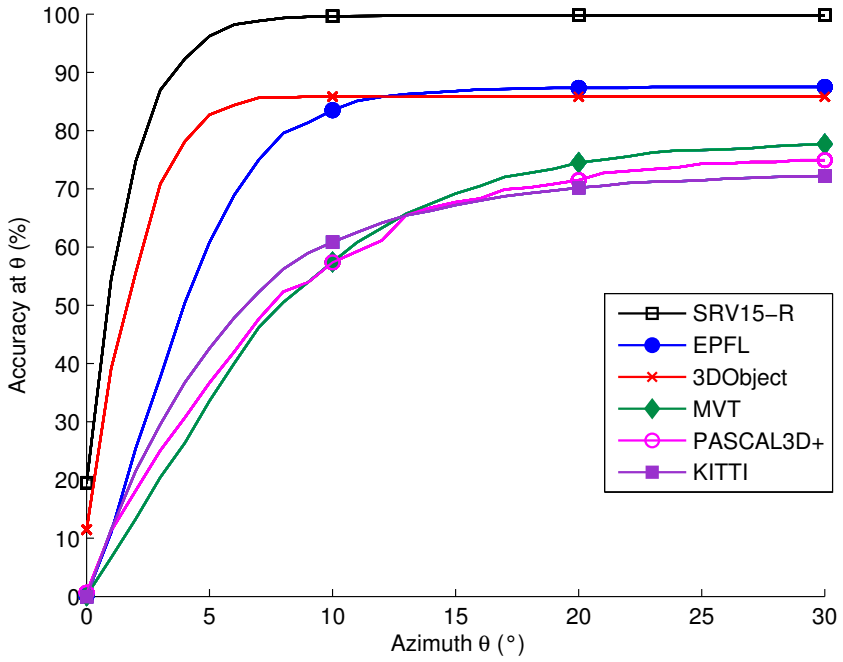

Figure 6.5: Continuous viewpoint prediction results. SRV15-R exhibits the highest level of accuracy since our SVP model is trained on SRV15-S. For other datasets, our model must not only make the jump from synthetic to real but also generalize to vehicle types different from the (only) 15 cars in our training set.

		Accuracy at θ						Accuracy at ϕ		
Dataset	Med (θ)	5°	10°	15°	20°	25°	30°	5°	10°	15°
SRV15-R	1.0	96.2	99.6	99.8	99.8	99.8	99.8	92.2	99.6	100
SSV+	3.0	65.4	80.2	84.2	85.7	86.3	86.4	72.9	94.3	99.2
SSV+U	3.0	69.0	84.2	87.7	89.0	89.4	89.4	75.4	95.5	99.4
SSV+O	4.0	59.6	73.9	78.7	80.6	81.4	81.8	69.0	92.5	98.9
PASCAL3D+	7.8	36.7	57.4	67.7	71.5	74.3	74.9	54.5	82.8	93.7
MVT	7.8	33.6	57.4	69.2	74.5	76.6	77.7	44.9	77.1	92.3
KITTI	6.4	42.6	60.8	67.2	70.2	71.5	72.2	-	-	-
EPFL	4.0	60.8	83.5	86.8	87.3	87.5	87.5	-	-	-
3DObject	2.0	82.7	85.8	85.8	85.8	85.8	85.8	95.0	99.6	100

Table 6.1: Azimuth (θ) and elevation (ϕ) viewpoint prediction results. Med (θ) is azimuth median error in degrees. Accuracy at θ is defined as the percentage of predictions within $\theta°$ of ground truth. Note: KITTI and EPFL do not have ground truth elevations.

We now consider comparison of our approach with two other works which report results using known bounding boxes. We do not use train data other than SRV15-S. In [13], results on the EPFL test dataset (last 10 sequences) are reported as the average of the diagonal of the confusion matrix. Here, we achieve 79.9% compared to their result of 82.8% by binning our continuous results as they do. In [52], for the validation portion of PASCAL VOC 2012, we achieve a median geodesic distance error of 11.5 compared to their result of 10.0.

Our results are close despite the fact that our approach amounts to not much more than a matrix-vector multiply. Further, consider that we have developed a *general* system that does not benefit from biasing to the training set of each dataset. An example of this bias can be found in Figures 6.6 & 6.7 where we do not benefit from bias to image size and azimuth respectively.

We have shown that SVP performs very well on a wide range of datasets, performing close to the state-of-the-art results found in [13] & [52].

6.0.3 Model Generalization

It is important to emphasize that we train on only 15 CAD models of cars. We were surprised, therefore, to find that our viewpoint model generalized so well to vehicle types far outside those found in the training set; pickup trucks, convertibles, vans, SUVs and jeeps for example (Figures 6.9, 6.16 & 6.17).

6.0.4 Comparison with HOG Features

Traditional hand-engineered image representations, one example being Histogram of Oriented Gradients (HOG) originating in 2005 [7], have been suc-

cessfully applied in many areas of computer vision. In the case of HOG, the distribution of gradients within local non-overlapping, uniformly-spaced cells, concatenated into a single descriptor has proven to be effective especially in cases where similar gradient patterns occur over the entire object class (such as pedestrian detection). HOG also incorporates local contrast normalization across larger blocks of cells in order to increase descriptor invariance in the presence of illumination changes. The cell size and number of orientations within which gradients are binned can be varied depending on the application.

Over the years, however, each successful application of traditional feature representations such as HOG has slowly shaped the direction of datasets toward increasingly difficult problems. By 2012, performance in object detection had already reached a point of diminishing returns [47] before the seminal work of Krizhevsky, Sutskever and Hinton showed that learned feature representations led to a dramatic jump in performance [32].

We confirm here that CNN features outperform HOG features for viewpoint prediction especially in the presence of occlusion. Using dataset SSV+, we compare the performance of CNN vs. HOG in the presence of occlusion and without.

We use the default, UoCTTI variant of the HOG implementation from VLFeat.org with a cell size of 8 pixels and 9 orientations on 227x227 resized vehicle localizations from dataset SSV+ to generate HOG descriptors of dimension 28 x 28 x 31 = 24,304. We compare the performance of this HOG feature with $pool_5$ CNN representations separating unoccluded and occluded images (SSV+U and SSV+O).

As can be seen in Figure 6.10, the CNN representations of dimension 9,216 outperform HOG features of dimension 24,304. Further, Figure 6.10 confirms that in the presence of occlusion, this difference is even more pronounced.

For any given image, HOG features are based entirely on flat, local gradient distributions whereas features derived from deeper layers of a CNN (here layer $pool_5$) have passed through a hierarchy of layers, each layer encoding a distributed expression of a wide variety of learned features emergent during training [14] (deep features can encode expressions of abstract concepts such as color, 'metal', 'glass', 'wheel' or 'fur' for example). Further, deep features at any given depth leverage and encode correlations occurring across features maps at the previous layer; in other words, abstractions encoded at any layer include the notion of *composition* of the previous layer's abstractions. In the case of a vehicle for example, this may encode the notion of 'front of car' at some depth by utilizing the notions of 'wheel' and 'front light' in the previous layer (including their relative positions in the image).

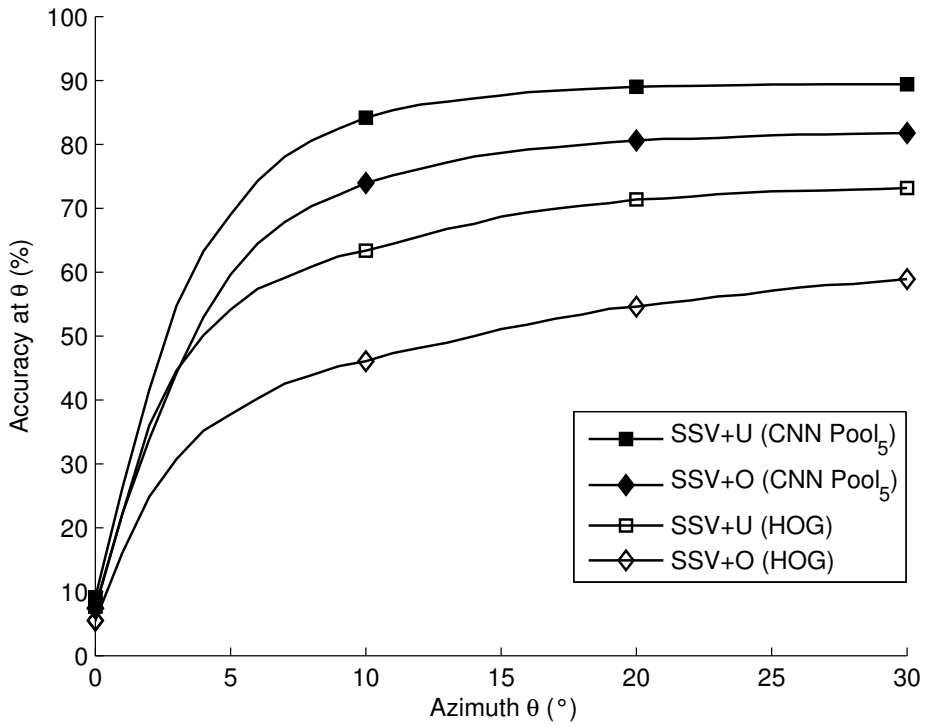

Figure 6.10: Learned $pool_5$ CNN features of dimension 9,216 outperform hand-crafted HOG features of dimension 24,304 for viewpoint prediction especially for occluded objects (Street Scenes Vehicles Unoccluded and Occluded datasets (SSV+U and SSV+O)).

The *flat* nature of HOG descriptors, based only on local image gradients, lead to even greater challenges in cases of occlusion. Figure 6.11 shows a single image from dataset SSV+O showing a vehicle occluded by a chain-link fence. HOG views the world exclusively through a single, local 'gradient' lens, and in the case of occlusion, will encode each occlusion as yet another feature of the image. However, even though learned CNN features at the first layer often encode gradient information (albeit color as well) [59], as discussed above, deeper layers encode more abstract concepts such as 'wheel' (for example) and therefore are less *solely* influenced by gradient alone. Figure 6.11 shows an example of this deep-feature advantage, apparently unaffected by the occlusion posed by the chain-link fence, with a predicted viewpoint that is only 2° azimuth off of ground truth. Similar affects can be seen in general as is shown in 6.10

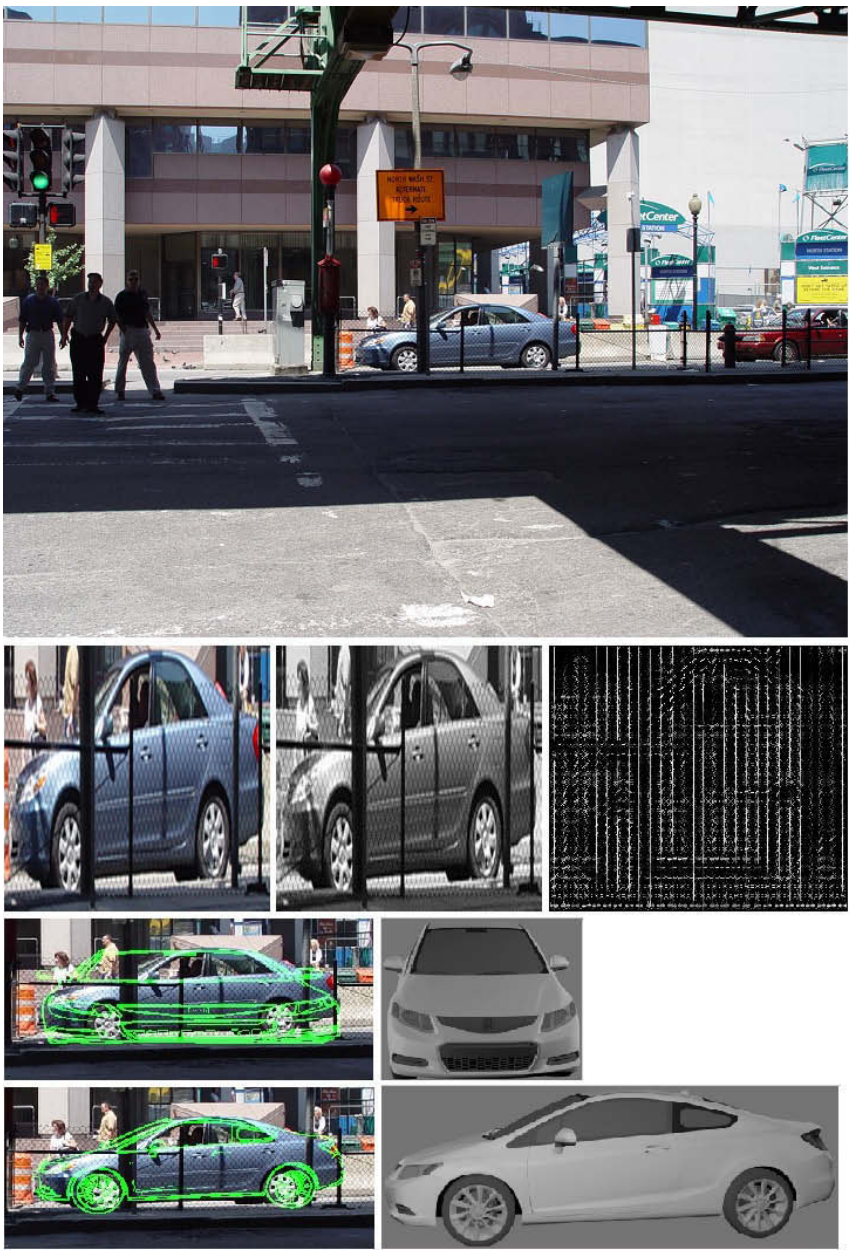

Figure 6.11: CNN features are more effective for viewpoint prediction than HOG features especially in cases of occlusion. Top: Original image SSDB03100 from dataset SSV+O showing car behind chain link fence with ground truth orientation of 5° elevation & 354° azimuth. 2nd Row: Car re-sized to 227x227 pixels in color and b&w followed by visualization of HOG feature histograms (cell size of 8 at 9 orientations). This results in a HOG feature of dimension 28 x 28 x 31 = 24,304 (HOG UoCTTI variant projects down to 31 dimensions for each cell). 3rd Row: Viewpoint prediction of 13° elevation & 267° azimuth using HOG features with overlay (left) of corresponding synthetic best match (right). Bottom Row: Viewpoint prediction of 12° elevation & 352° azimuth using CNN $pool_5$ features with overlay (left) of corresponding synthetic best match (right). Synthetic images: 2012 Honda Civic Coupe.

6.0.5 The Effect of Image Size

Fewer pixels, containing less information, lead to greater errors in viewpoint accuracy (Figure 6.12). Nevertheless, regardless of size, the hot spot in 6.12 (for dataset SSV+) shows that the majority of predicted azimuth values are closer to ground truth. We confirmed that this trend occurs across all datasets we tested leading us to conclude that, in general, viewpoint accuracy improves with higher input resolution.

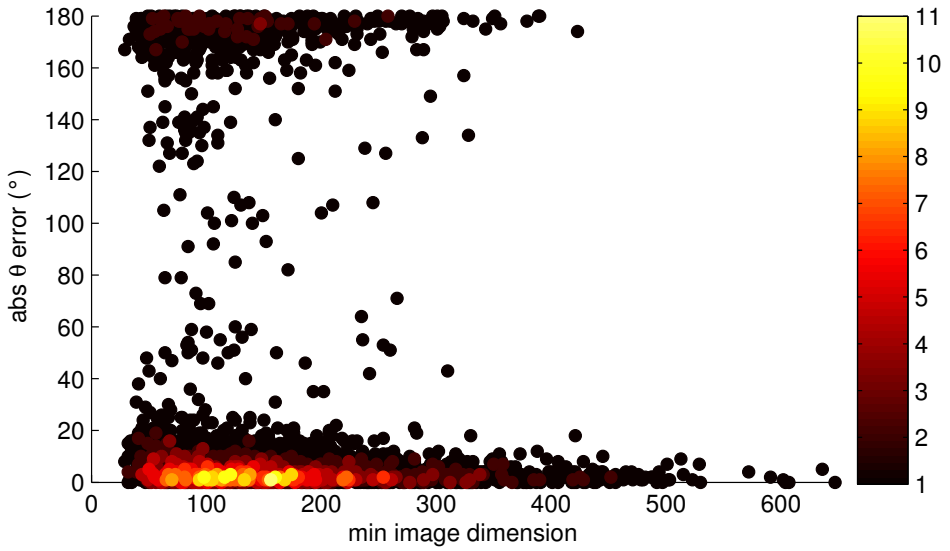

Figure 6.12: Azimuth viewpoint error depends on bounding box image size. Here we display results for dataset SSV+ as a color histogram with maximum occurring bin count of 11. Horizontal axis is min(width,height) of image. This shows the predominance of 180°-off errors with error increasing as image size decreases (see also Figure 6.4 and red-highlighted errors in Figure 6.17 showing predominance of 180°-off errors). The hot spot shows that many more are closer to correct than not. Best viewed in color.

6.0.6 Scaling Up Synthetic Training Data

SRV15-S contains 7,560 b&w renderings from each of 15 different CAD models of cars. Figure 6.15 shows that viewpoint accuracy increases dramatically as we increase the synthetic training set from 1 CAD model to 8 CAD models and finally the entire SRV15-S dataset of 15 CAD models (in no particular order). This forms an important contribution of this work; that viewpoint prediction for localized objects in real-world images can be made increasing accurate simply by adding more synthetic training data.

6.0.7 Applications

[52, 19] first compute coarse viewpoint followed by refinement steps and our fast SVP method could be applied here (see Figure 6.13). Similarly in [31], where coarse viewpoint is needed for fine-grained categorization, SVP should prove useful. Figure 6.17 shows a qualitative example of how effective SVP is on the unlabeled Cars196 dataset.

Figure 6.13: Coarse viewpoint prediction for the Cars196 dataset showing applicability to fine-grained classification as in [31] (classification can be simplified if images are first viewpoint-normalized). Images above (moving across, then down) show examples of correctly viewpoint-classified images into azimuth bins each spanning 45° centered on values 0, 45, 90, 135, 180, 225, 270, 315° (for example: upper left corner image represents image classified as being within azimuth values of -22.5° and 22.5°). Notice effective generalization of our model to vehicle classes far outside our synthetic training data like trucks and SUVs for example. Images here taken from just one row of 6.17.

SVP can also be applied to help solve the ultra-wide baseline matching problem where baselines approaching 180° become increasingly difficult (see Figure 6.14). This involves the calculation of the fundamental matrix between two views; a problem which is greatly simplified if one knows the 3D camera viewpoint angles (ϕ, θ) for each view. Using the 134 image pairs in [60] for the 3DObject dataset [48], we report the percentage of correctly predicted relative azimuth offsets for each pair. At baselines of 45, 90, 135 & 180°, we correctly predict 83.0, 82.9, 79.3 & 76.5% respectively. Our numerical values can not be directly compared to [60] (since they compare fundamental matrices) but nevertheless give a good indication of the accuracy and usefulness of our system to baseline matching.

Figure 6.14: Application to ultra-wide baseline matching here demonstrated for car #1 class from dataset 3DObject. When the angle between two views of the same object is large (above), traditional matching approaches fail as there is nothing to match. Using our synthetically trained model (SVP) for the object class in question (here car), we can predict the viewpoint of each image directly. Above (image car_A1_H2_S2): predicted viewpoint of 18° elevation & 87° azimuth. Below (image car_A6_H1_S1): predicted viewpoint of 12° elevation & 240° azimuth.

In this work, we trained a single synthetic model which sought to solve the vehicle viewpoint problem *in general* as opposed to fitting to any particular dataset. Since our approach is general to *any rigid object class for which CAD models exist* and not just vehicles, we emphasize that SVP could prove useful to any application where viewpoint prediction plays a role. Potential applications include augmented reality, semantic SLAM, robotic grasping, autonomous navigation and scene understanding in general.

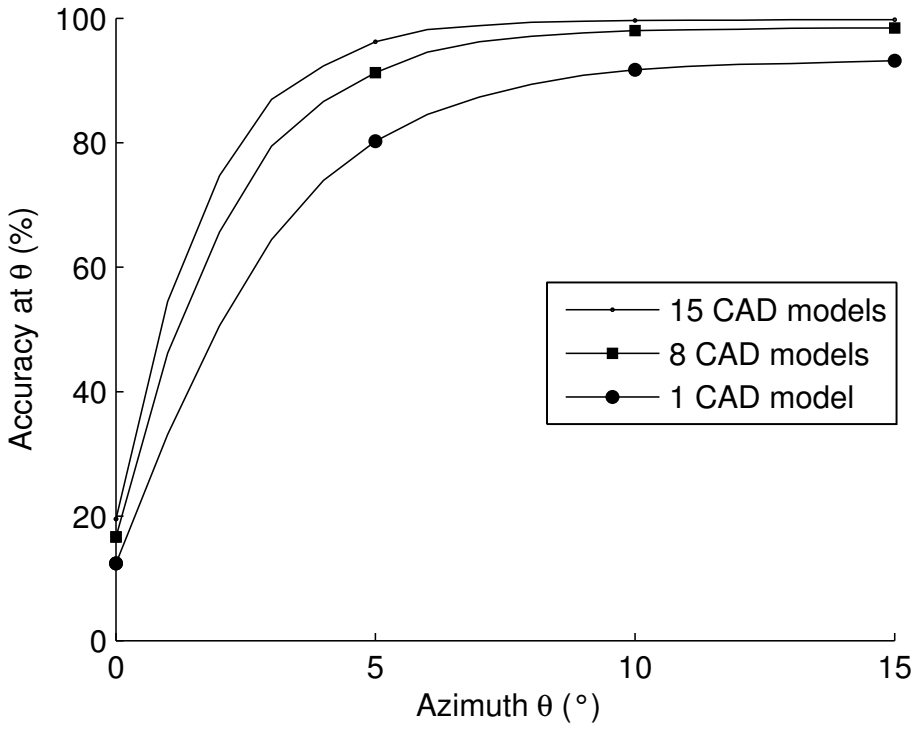

Figure 6.15: Azimuth viewpoint prediction accuracy on SRV15-R improves dramatically with expansion of synthetic training set. Each CAD model represents 7,560 b&w renders (one class of SRV15-S).

Figure 6.2: Transfer learning from synthetic to real images. Example viewpoint prediction of single real-world image from SRV15-R (2012 Renault Twingo). Green overlay represents edges of synthetic render (of 2012 Renault Twingo) corresponding to ground truth/predicted viewpoint and scaled within bounding box extents. Synthetic images to right are viewpoint renders associated with viewpoints. Top: ground truth viewpoint annotation of 10° elevation & 246° azimuth. Bottom: viewpoint prediction by model only 2°-off in elevation.

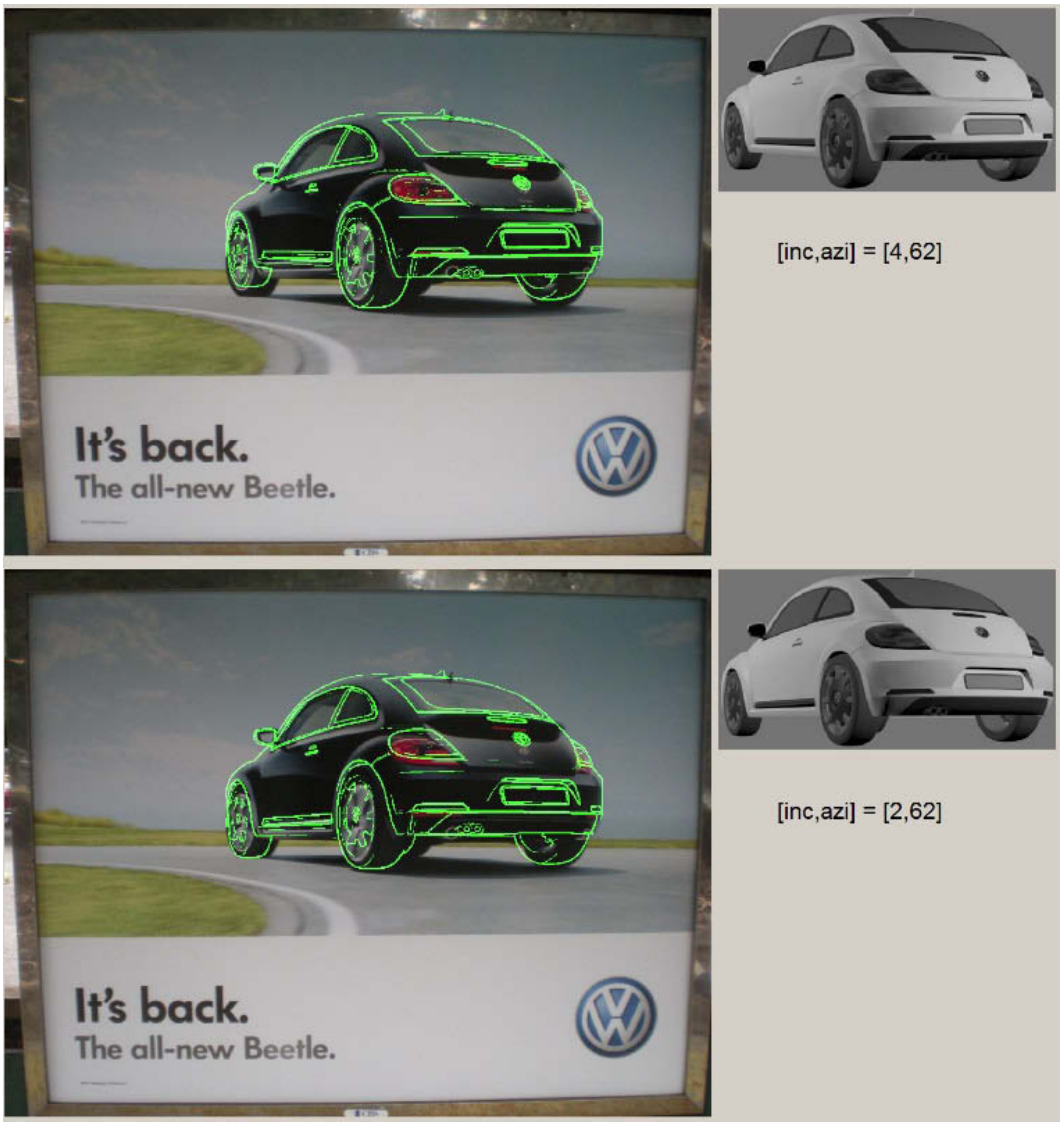

Figure 6.3: Another example of transfer learning from synthetic to real images. Example viewpoint prediction of single real-world image from SRV15-R (advertisement for 2012 VW Beetle). Green overlay represents edges of synthetic render (of 2012 VW Beetle) corresponding to ground truth/predicted viewpoint and scaled within bounding box extents. Synthetic images to right are viewpoint renders associated with viewpoints. Top: ground truth viewpoint annotation of 4° elevation & 62° azimuth. Bottom: viewpoint predicted by our synthetically trained model returns 2° elevation & 62° azimuth.

Figure 6.4: Viewpoint prediction errors, when they occur, are most often off by 180° azimuth. Top (2011 Peugeot 508): ground truth viewpoint annotation of 14° elevation & 68° azimuth with predicted viewpoint of 12° elevation & 248° azimuth (exactly 180° azimuth error). Bottom (2011 Peugeot iOn): ground truth viewpoint annotation of 9° elevation & 323° azimuth with predicted viewpoint of 21° elevation & 145° azimuth (178° azimuth error).

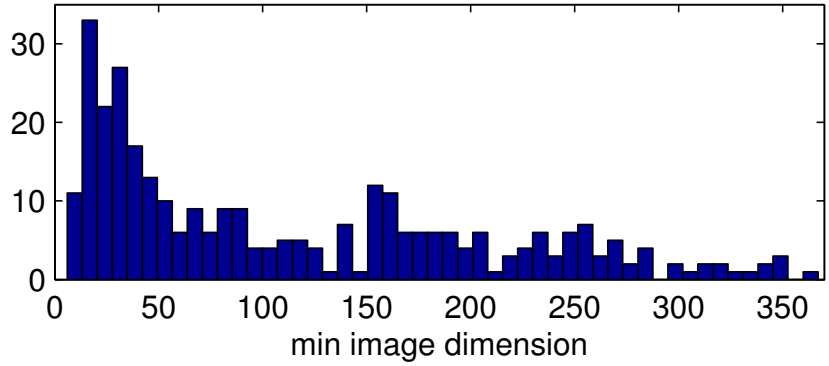

Figure 6.6: Distribution of minimum image dimension in PASCAL validation set.

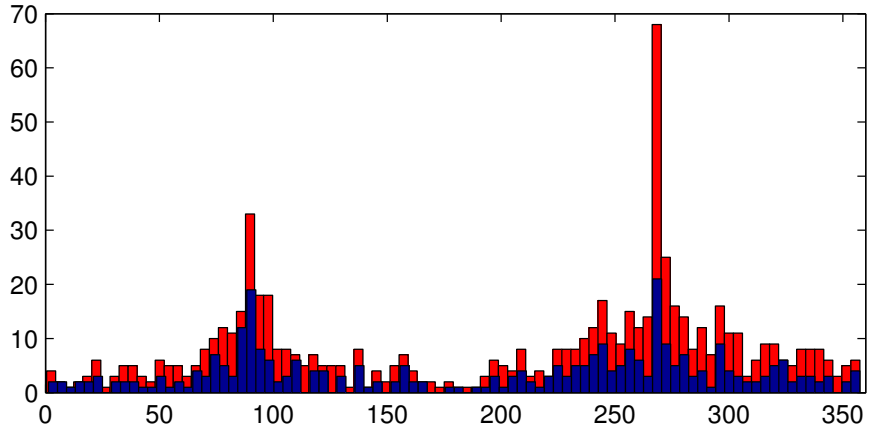

Figure 6.7: Distribution of ground truth azimuth viewpoint found in the PASCAL 2012 train (red) and validation (blue). Note validation set is used as test. Our SVP model does not benefit from learning dataset distributions such as this yet still performs well.

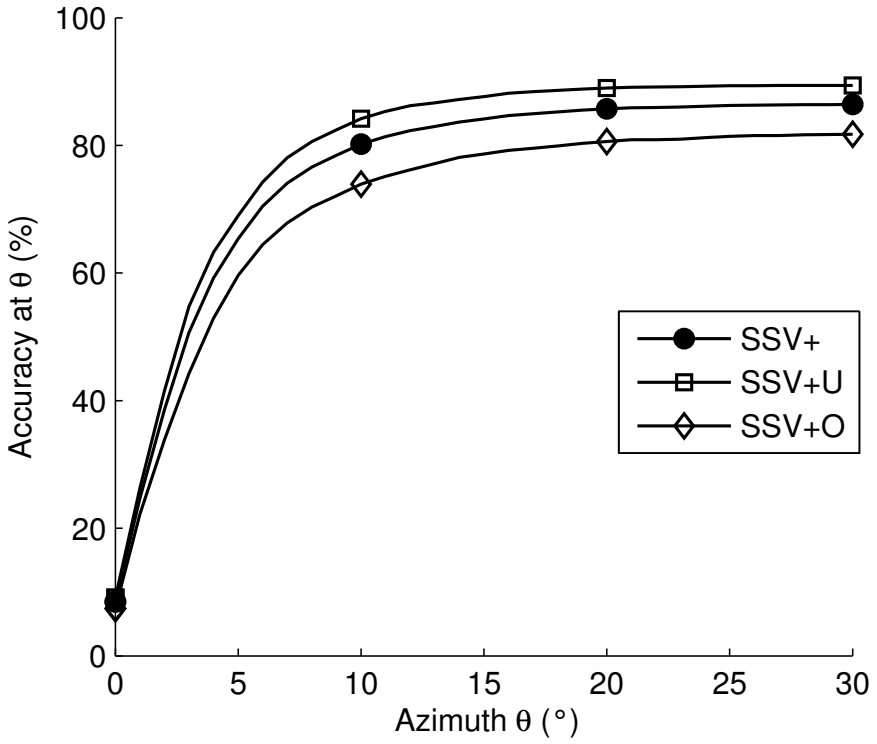

Figure 6.8: Viewpoint results on occluded (O) and unoccluded (U) images in dataset in SSV+ (4,641 images).

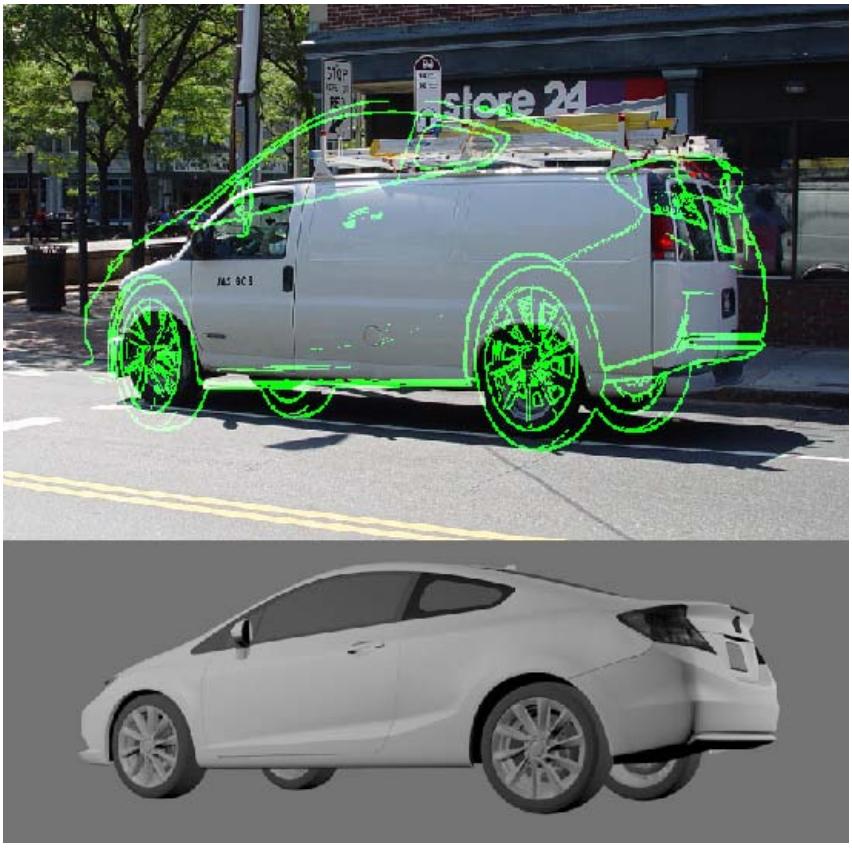

Figure 6.9: Viewpoint prediction generalization to vehicles outside synthetic training set. Top: portion of image SSDB0074 from SS dataset. Green lines are edges of corresponding synthetic view prediction (below) rendered within extents of bounding box.

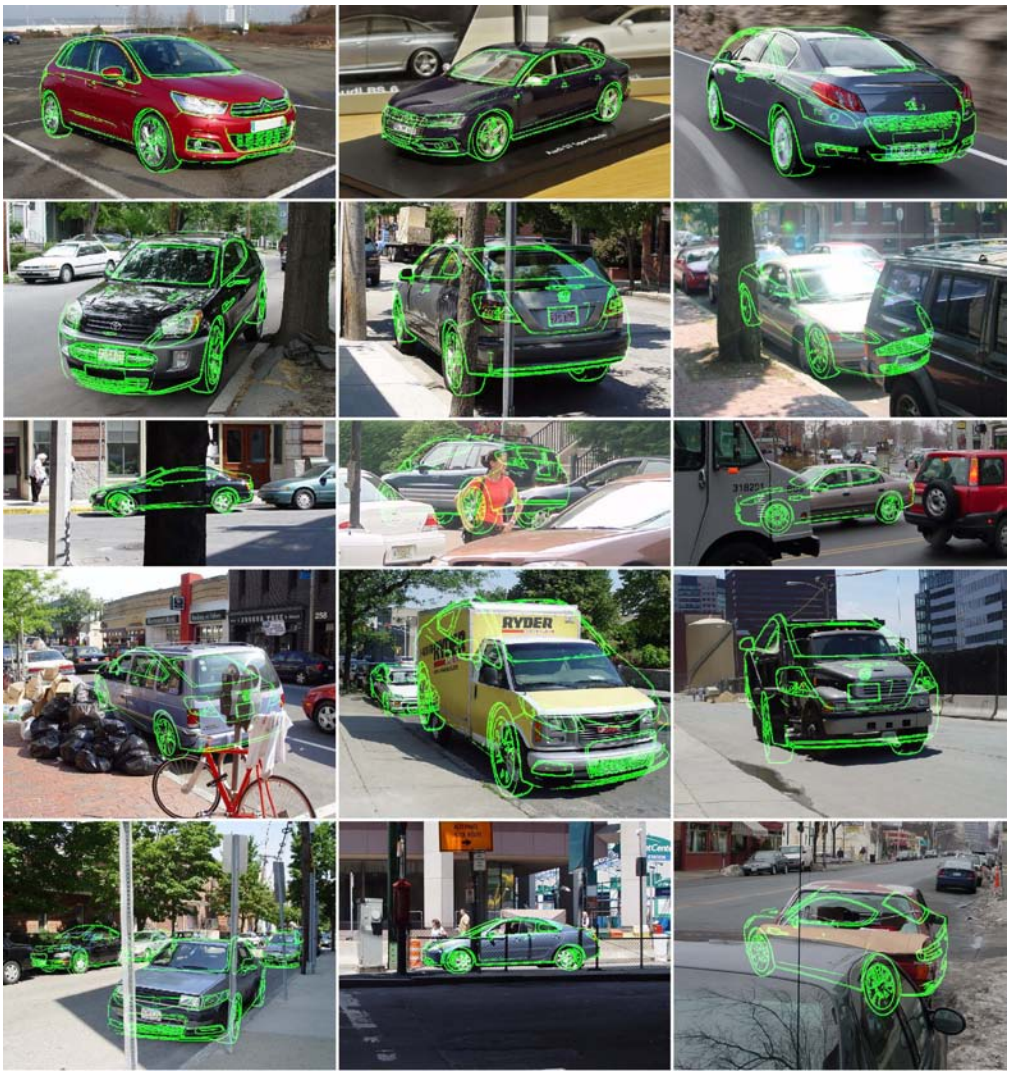

Figure 6.16: Viewpoint prediction results (GT=ground truth, P=prediction): 1st row: (dataset SRV15-R) (L) Exact prediction GT=P=(11,234) (M) Model of Audi S7 Sportback with GT=(15,311), P=(16,315) (R) Example of 180°-off θ error. Rows 2-5 from dataset SSV+. Row 2: (L) Typical unoccluded result (M) Shadow & occlusion (R) Success in presence of high specularity. Row 3: Heavy occlusion. Row 4: (L) Occlusion & clutter (M) Effective generalization to unseen vehicle type (R) Failed attempt to generalize to dump truck. Row 5: (L) Scene understanding for image SSDB00075 (5 occluded cars) (M) Chain-link fence occlusion (R) Occlusion failure. Best viewed under pdf magnification.

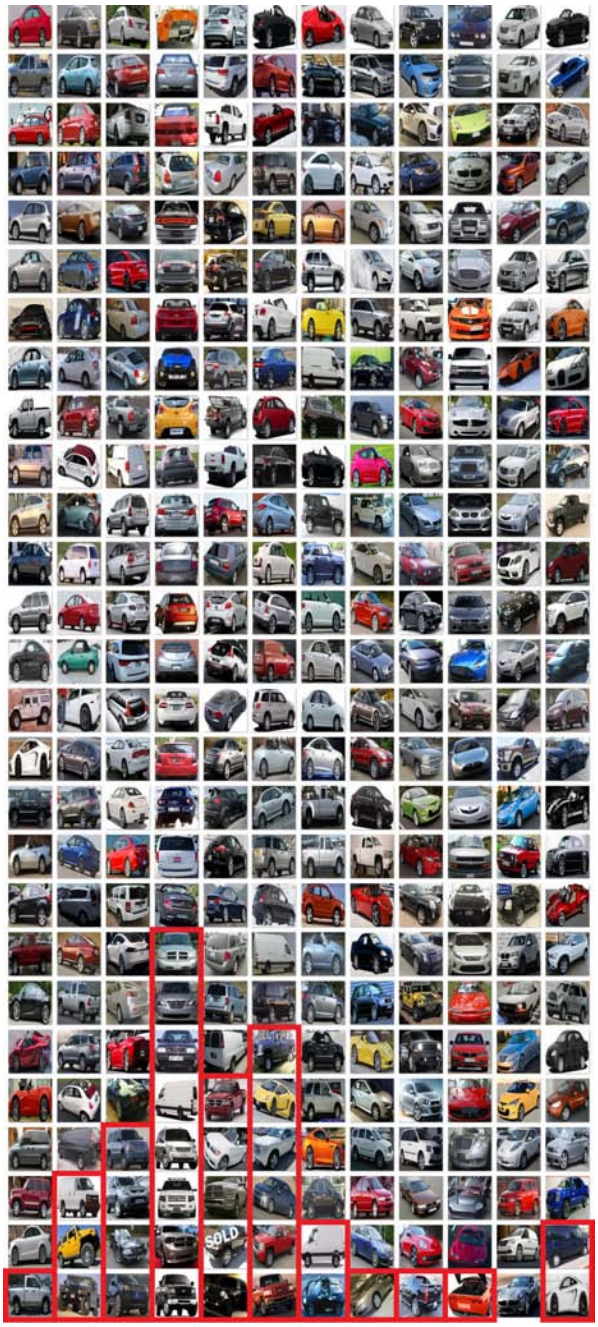

Figure 6.17: Viewpoint generalization to unseen vehicle types: SVP applied to the test portion of the Cars196 dataset [30]. Note, this dataset does not have viewpoint labels. We first predict viewpoint for all 8041 images and separate into 12, 30° azimuth bins. We then take the first 27 images in each bin (sorted alphabetically) and manually inspected whether it was correct (each column represents one bin). Erroneous predictions (those in the wrong bin) are moved to the bottom of each column and highlighted in red. Note: most are correctly predicted and those that are wrong are almost always off by 180°. Of these 324 images, 34 are incorrect (90% accuracy). If one considers 180°-off to be correct, only 7 are incorrect (98% accuracy). Best viewed under pdf magnification.

Chapter 7

Conclusion and Future Work

We synthetically trained a model using just 15 CAD models of cars and demonstrated generalization capable of accurate, continuous 3D viewpoint prediction to vehicles *in general* even under high levels of clutter and occlusion. We created a new synth-real dataset and new accurate labels for existing datasets, one of them ideal for studying the viewpoint-under-occlusion problem, as well as numerous results we hope will provide the community with new viewpoint baselines. Our annotation-free approach is not specific to cars but applicable to *any* object class given CAD models.

To our knowledge, we are the first to focus exclusively on viewpoint prediction decoupled from localization in the hopes that this specialization will lead to progress similar to that made in fine-grained categorization for example. Further, we believe we are the first to train a model using deep representations of purely synthetic images for the purpose of continuous 3D viewpoint prediction of real-world images amenable to real-time speeds. Most importantly, we have shown that deep representations can bridge the large divide between synthetic and real image distributions, overcoming clutter and occlusion.

We have shown that increasing the number of CAD models increases accuracy so we would like to scale up the number of models and expand their diversity to include other vehicle types (as well as other object classes entirely). This increase in training data might necessitate inquiry into representation dimensionality reduction. Experiments relating bounding box accuracy to viewpoint viewpoint accuracy would be of interest as would ablation studies simulating occlusion. We expect that fine-tuning existing convnets or training new architectures would lead to better results. Most interesting is how SVP could prove effective as part a larger system for scene understanding in general.

Especially for applications such as viewpoint prediction, where large amounts of dense, labeled training data is required, we believe that purely synthetically trained systems such as SVP will become increasingly prevalent.

Bibliography

[1] M. Arie-Nachimson and R. Basri. Constructing implicit 3d shape models for pose estimation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1341–1348. IEEE, 2009. 12

[2] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007. 6

[3] S. M. Bileschi. *StreetScenes: Towards scene understanding in still images*. PhD thesis, Citeseer, 2006. 22

[4] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014. 1, 27

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014. 10, 13

[6] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. 7

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 6, 17, 29

[8] O. Delalleau and Y. Bengio. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems*, pages 666–674, 2011. 7

[9] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 9, 10, 13, 18, 19

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 6, 22

[11] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 7, 9

[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012. 12, 23

[13] A. Ghodrati, M. Pedersoli, and T. Tuytelaars. Is 2d information enough for viewpoint estimation. In *Proceedings of the British Machine Vision Conference. BMVA Press*, volume 2, page 6, 2014. 12, 13, 29

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 1, 9, 10, 13, 18, 30

[15] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. R-cnns for pose estimation and action detection. *arXiv preprint arXiv:1406.5212*, 2014. 12

[16] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1275–1282. IEEE, 2011. 12

[17] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and continuous pose estimation. *Image and Vision Computing*, 30(12):923–933, 2012. 12

[18] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013. 13

[19] Gupta, Arbelez, Girshick, and Malik. Inferring 3d object pose in rgb-d images. *arXiv preprint arXiv:1502.04652*, 2015. 1, 12, 13, 27, 34

[20] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision–ECCV 2014*, pages 346–361. Springer, 2014. 10

[21] M. Hejrati and D. Ramanan. Analyzing 3d objects in cluttered images. In *Advances in Neural Information Processing Systems*, pages 593–601, 2012. 12

[22] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 6

[23] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 6

[24] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 7

[25] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991. 7

[26] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *arXiv preprint arXiv:1412.1842*, 2014. 1, 13

[27] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 13

[28] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 13, 19

[29] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014. 13

[30] J. Krause, J. Deng, M. Stark, and L. Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013. x, 23, 44

[31] J. Krause, T. Gebru, J. Deng, L.-J. Li, and L. Fei-Fei. Learning features and parts for fine-grained recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 26–33. IEEE, 2014. ix, 34

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. vii, 7, 9, 10, 11, 13, 18, 30

[33] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 9

[34] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1688–1695. IEEE, 2010. 12

[35] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3d feature maps. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 12

[36] R. J. Lopez-Sastre, T. Tuytelaars, and S. Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1052–1059. IEEE, 2011. 12

[37] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 6

[38] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 6

[39] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2924–2932, 2014. vi, 7, 8

[40] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010. vii, 7, 11

[41] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 778–785. IEEE, 2009. 12, 23

[42] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010. 10, 13

[43] R. Pascanu, G. Montufar, and Y. Bengio. On the number of inference regions of deep feed forward networks with piece-wise linear activations. corr. *arXiv preprint arXiv:1312.6098*, 2014. 7

[44] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3d2pm–3d deformable part models. In *Computer Vision–ECCV 2012*, pages 356–370. Springer, 2012. 12

[45] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3362–3369. IEEE, 2012. 12

[46] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014. 10, 13

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014. 6, 7, 13, 22, 30

[48] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 12, 23, 34

[49] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *arXiv preprint arXiv:1503.03832*, 2015. 13

[50] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 213–220. IEEE, 2009. 12

[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 13

[52] S. Tulsiani and J. Malik. Viewpoints and keypoints. *arXiv preprint arXiv:1411.6067*, 2014. 12, 13, 27, 29, 34

[53] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014. 13

[54] D. R. G. H. R. Williams and G. Hinton. Learning representations by back-propagating errors. *Nature*, pages 323–533, 1986. 6, 10

[55] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014. 22

[56] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3410–3417. IEEE, 2012. 12

[57] Y. Xiang, C. Song, R. Mottaghi, and S. Savarese. Monocular multiview object tracking with 3d aspect parts. In *Computer Vision–ECCV 2014*, pages 220–235. Springer, 2014. 12, 23

[58] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014. 13

[59] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014. 10, 31

[60] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Revisiting 3d geometric models for accurate object shape and pose. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 569–576. IEEE, 2011. 12, 13, 23, 34