Overcoming the reality gap: Studying synthetic image modalities for convolutional neural network training

Shu Zhang

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Artificial Intelligence and Robotics Commons

Recommended Citation

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Overcoming the Reality Gap - Studying Synthetic Image Modalities for Convolutional Neural Network Training

Shu Zhang

Abstract

Nowadays, convolutional neural networks (CNNs) are the first choice for a variety of computer vision tasks such as object recognition, 6-degree-of-freedom (DoF) pose estimation, navigation, and others. Despite the success, they come with a challenge: CNNs require training data, and acquiring this data is a laborious and time-intensive task. A solution is training with synthetic images: synthetic images mean one uses rendered content stacked on top of a random background photo. Unfortunately, research results following this approach show that CNNs trained with synthetic images underperform when exposed to recorded images, a discrepancy which is known as the reality gap. However, the results are limited to training using solely synthetic RGB images. The objective of this research is to study whether additional image modalities (normal vectors, depth maps) bridge the reality gap for 6-DoF pose estimation. The approach is experimental. A CNN architecture learns to estimate the pose using synthetic domain randomized datasets only; each dataset implements a different sensory modality (RGB, RGB-D[epth], RGB-N[ormal], RGB-DN). Once trained, the network is exposed to one synthetic and one recorded evaluation dataset, and the pose accuracy difference shows whether a different modality overcomes the reality gap. Experimental results on a synthetic test set shows that adding normal and depth modalities to the RGB modality increases the 5-degree accuracy by 4% and 6% respectively for the POLY dataset. The combination of all RGB, depth and normal modalities yielded promising results on both, the POLY and POSE datasets. More innovative fusion strategies can be explored to fully utilize the unique characteristics of each data modality. Multiple explorations on real recorded dataset revealed the necessity to generate synthetic training datasets which scene context matches real life data.

Key Words: CNN, object detection, 6D Pose estimation, reality gap, domain adaptation

Introduction

6D pose (3D translation and 3D rotation) estimation of a rigid object has import applications in areas such as augmented reality or robot manipulation. Traditional methods for 6D pose estimation include learning based methods, such as Brachmann’s regression forest [1] where a RANSAC-based optimization schema is used to create a pool of pose hypotheses; Template Matching methods [2], which vote for candidate templates to identify the nearest one; methods based on 3D local features [3], which find correspondences between an object model and the scene by matching local feature descriptors to generate a 6D pose candidate [4]. These fixed correspondence-groupings and hypotheses-verification methods can achieve good accuracy, however, the hand-crafted features limit their performances when there are heavy occlusion and lighting variation.
Convolutional neural networks (CNN) have yielded a remarkable performance on many image-related tasks, such as classification, object detection, and registration. The application of CNNs is gaining ground in different domains thanks to its advantages in automatic feature extraction and computational efficiency. This recent success has inspired many researchers to apply CNN for pose estimation problems, yielding new tools such as PoseCNN [5], BB8 [6], and PVnet [7]. However, these methods require pose-refinement using methods such as Perspective-n-Point (PnP) [8], Iterative Closest Point (ICP) [9], or hypothesis verification.

CNNs for 6-DoF pose estimation show promising results, however, current approaches are data-driven. The network requires sufficient data to learn the millions of weight values in the network. To annotate real life data is a very expansive and laborious process. One upcoming solution is to use synthetic data for training. By rendering 3D objects from different viewpoints and augmenting a background image with the rendering at different locations allows one to create a theoretically infinite dataset. However, even though the training data size is substantial, one experiences a large performance drop when evaluating a trained network on real life data. This performance drop between testing with synthetic and real images is called the reality gap. Today, the reality gap needs to be reduced by applying domain adaptation techniques. They fine-tune a trained network by learning on domain invariant representations or through transfer learning.

To explore new ways for domain adaptation, we investigate whether multiple image modalities have the potential reducing the reality gap.

**Related Work**

Using synthetic images to train the neural network for 6D pose estimation is now a popular topic. By augmenting a rendering of the object of interest on another images from different views and in different poses allows one to generate a substantial dataset with labels and ground-truth poses.

There are different ways to create synthetic images. First, one can render the object in a photorealistic way, where each object superimposes the background showing several application-common scenes. Technically, one can detect planes in those images so that the object can virtually fall onto the table, shelf, chair, or ground. A second approach is called domain randomization. Here, all objects of interest, including the scene content, are rendered. All objects can be placed at any position in the virtual scene. Hodan et.al [10] proposed to use photorealistic image for training. The authors generate realistic images by modeling a high-fidelity scene in terms of geometry, textures, materials, and illumination, including soft shadows, reflections, refractions, and indirect light. Their experimental results show that using photorealistic training images yields a significantly higher accuracy on the LineMOD dataset [11] comparing to non-photorealistic images. Tremblay et al. [12] also concurs that a combination of photorealistic and domain randomized images for training results in a higher accuracy in comparison to solely using domain randomized images. They also explored the size of the synthetic training set necessary to achieve the desired accuracy.
Generating photorealistic training images is one of the ways to approximate the color distribution of real images. Since objects shown as part of real evaluation data for domains such as robotics are mostly small objects on a table, thus, recognizing the scene ‘small objects on the table’ is an important context for this network. However, photorealistic images are not easy to generate comparing to domain randomized images. If we can use only domain randomized imaged for training, then there would have more flexibilities on the training data generation. Considering the cases where the real recorded image of an object does not show it on the table, for example, if the object is in a robot gripper, then maybe domain randomized images are sufficient for training. Therefore, we will explore the potential power of using sorely domain randomized images for training objects that are not all on the table.

**Proposed Approach**

In this section, we present an end-to-end fully CNN-based pipeline for 6-DoF pose estimation. The network is split into two segments: the first segment is a segmentation network. It is followed by two branches of regression networks as the illustration in **Figure 1** shows.

![Model pipeline for 6D pose estimation](image)

**Figure 1. Model pipeline for 6D pose estimation**

1. **Segmentation**
   Considering our object of interest is rendered using random background images and in a random pose, the image context does not help much recognizing the object. Thus, our first step is to do instance segmentation to pixel-wise segment the object of interest in the training images.
Many CNNs were already suggested performing pixel-wise segmentation. The typical segmentation architecture is composed of a regular convolutional path, down sampling the image and extracting coarse semantic features. Subsequently, a transpose convolutional path recovers the input image resolution at the output of the CNN and, optionally, a post-processing module such as Conditional Random Field refines the model predictions. Based on a new CNN architecture, Densely Connected Convolutional Networks (DenseNets), which has shown excellent results on image classification tasks, Jégou et al. [13] proposed a fully convolutional DenseNet for semantic segmentation. This 100-layer network achieved good performance on several benchmark datasets with far fewer parameters comparing to other state-of-the-art networks. For this research, we adopted a simplified version of fully convolutional DenseNet, which has 56-layer in total. There are 5 transition down, 5 transition up, 4 layers in each dense block and 12 new feature maps created by each layer in a dense block. A basic architecture from [13] is shown in the Figure 2.

![Figure 2. Basic diagram of Fully Convolutional DenseNet](image-url)
2. 6D Pose Estimation
We represent the pose of an object by its position \( t = (tx, ty, tz) \) and orientation \( q = (qx, qy, qz, qw) \), which are translations and rotations relative to the camera coordinate frame. Here, the translation coordinates are unconstrained, while the rotation is normalized and in the form of a unit quaternion. Since there is no constraining relation between translation and rotation, we process the two sets as separate regressions by using two branches of multi-layer-perceptron to estimate the 6-DoF pose.

3. Multimodality
To obtain real life annotated data, one can obtain RGB images and depth images by using an RGB-D camera. We can also generate a normal image from an RGB image. Different modalities of an image focus on different aspects of input data. RGB images are a visual representation of the content and represented by the color components of all the instances. However, the hue and saturation of the content is encoded as RGB colors, thus, subject to illumination changes, resulting in poor recognition under extreme light condition. Normal images represent the surface normal vectors of its content. They are less susceptible to various light conditions if captured with an RGB-D camera. Depth images represent information pertaining to the objects distance to the camera. Research results on several benchmark datasets show that depth images can contribute to the estimation of 6-DoF pose. Nevertheless, few explored the fusion of normal images with RGB images and/or depth images. Also, research pertaining to the fusion strategy of different modalities is also underrepresented, thus, worthy being studied to better understand how different data modalities can complement and contribute to each stage of pose estimation.

Here, we explored different combinations of data modalities, such as single branch, only RGB or depth or normal images; two branches, where any of the two kinds from the above-mentioned modalities are combined; or three branches, where all modalities are used. As for the fusion strategy, we explored the early fusion and late fusion. Early fusion concatenates different modalities after the down-sampling path, while late fusion concatenates just before the last segmentation layer by adding up all values of the feature map pixel-wise in different modalities.

4. Generating Training Images
Training images for this research were generated synthetically. The Stanford bunny model was used as the object of interest. The training image generation process requires multiple steps: In the first step, one changes the position and orientation of the Stanford bunny object randomly. For this research, 10000 bunny images with different poses were generated. In the second step, 10000 domain randomized images were generated using these renderings. Each bunny model rendering was combined with a color image by superimposing its content. ImageNet database was used to provide background content. Since the second step is a random process, only some of the 10000 unique bunny model renderings were used and present in the final dataset.
5. **Data Augmentation**
To introduce more variations into the training dataset, data augmentation techniques were employed. For each training image, randomly picked zero to two augmentations out of Crop, GaussianBlur, ContrastNormalization, AdditiveGaussianNoise and Multiply were applied. These augmentations introduce more data variance by varying light conditions and image quality. This type of data augmentation does not affect the pose annotations.

**Experiments**

1. **Evaluation Metrics**
To evaluate the segmentation results, we used precision recall score as well as the intersection-over-union (IOU) ratio. Both represent the accuracy that the instance segmentation yields. To assess the translation accuracy result, the L2-norm between the predicted translation and the ground-truth translation was calculated for each sample as well as for the whole dataset. To assess the rotation results, we calculated the angle distance between the predicted and the ground-truth labels. Additionally, all rotations with an angle-delta less than 5 degree were considered as acceptable, and its ratio in comparison to the entire dataset was calculated. Also, the mean angle was calculated.

2. **Dataset**
The data are generated using the software DNN Helpers [14]. For each rendering of the bunny object, an RGB image, depth image, normal image, a mask image, and annotations describing the region of interest were created, as well as the pose ground truth data as quaternion and translation coordinates. Experiments are conducted on several datasets, and the details of the datasets are summarized in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Datasets specifics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POLY data</strong></td>
</tr>
<tr>
<td><strong>POSE data</strong></td>
</tr>
<tr>
<td><strong>Combined</strong></td>
</tr>
<tr>
<td><strong>Evaluation</strong></td>
</tr>
</tbody>
</table>

3. **Loss function**
The CNN incorporates three tasks: segmentation, regression for translation, and regression for rotation. Each task employs a different loss function: sigmoid cross-entropy loss was used for the segmentation network and mean-squared-error for both, the translation and the rotation branches. Also, to avoid overfitting, an L2-regularization was added to the regression part. The total loss is a weighted average of the four loss values.
4. Implementations
The experiments were conducted using the TensorFlow V 1.12 framework. Details of the parameter settings are in Table 2.

Table 2: Experimental Settings

<table>
<thead>
<tr>
<th>Input size</th>
<th>(256x256)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input transform</td>
<td>Standardized to (0,1) range</td>
</tr>
<tr>
<td>Optimizer</td>
<td>RMSPropOptimizer</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.0001</td>
</tr>
<tr>
<td>Batch size</td>
<td>1</td>
</tr>
<tr>
<td>Decay</td>
<td>0.995</td>
</tr>
<tr>
<td>Epochs</td>
<td>100 or till convergence</td>
</tr>
<tr>
<td>GPU type</td>
<td>TITAN Xp</td>
</tr>
<tr>
<td># of GPU</td>
<td>1</td>
</tr>
</tbody>
</table>

5. Results
Table 3 shows the segmentation and pose result for the POLY dataset. The precision, recall and IOU are all 1.0 for all combinations of data modalities, which tells us that the segmentation task for big centered bunny is relatively easy. Using any single modality of data, we can obtain a highly accurate segmentation mask of the object of interest. For the rotation accuracy, RGB images only yielded the lowest accuracy comparing to other single modalities. Adding depth modality to RGB increased the 5-degree accuracy from 0.86 to 0.92, which is higher than adding normal vector to RGB data (0.90). Among all combinations of data modalities, a single normal modality obtained the highest 5-degree accuracy of 0.95, which means 95% of the test set predictions have less than 5-degree angle difference with the ground truth. A single depth modality achieved the highest 1-degree accuracy among all combinations, with 23% of the test set predictions less than 1-degree angle-delta. The combination of the depth with the normal modality achieved the highest 3-degree accuracy. The performance gaps among different data modalities are getting closer for higher thresholds like 7-degree or more.

Table 3: Results on POLY data

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>IOU</th>
<th>1 degree</th>
<th>3 degree</th>
<th>5 degree</th>
<th>7 degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.12</td>
<td>0.64</td>
<td>0.86</td>
<td>0.93</td>
</tr>
<tr>
<td>Depth</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.23</td>
<td>0.80</td>
<td>0.93</td>
<td>0.98</td>
</tr>
<tr>
<td>Normal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.18</td>
<td>0.78</td>
<td>0.95</td>
<td>0.97</td>
</tr>
<tr>
<td>RGB+Depth</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.11</td>
<td>0.68</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>RGB+Normal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.13</td>
<td>0.67</td>
<td>0.90</td>
<td>0.96</td>
</tr>
<tr>
<td>Depth+Normal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.22</td>
<td>0.81</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td>RGB+Depth+Normal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.19</td>
<td>0.81</td>
<td>0.94</td>
<td>0.97</td>
</tr>
</tbody>
</table>
The validation loss, IOU, translation and rotation accuracy for experiment on sorely RGB images are shown in Figure 3.

![Figure 3: loss, translation MSE, 5-degree, and IOU plots for POLY RGB modality](image)

From Figure 3, it becomes clear that the total loss converged after 35 epochs. The average IOU plot almost plateaued (0.999) after only 5 epochs, which means the segmentation task is relatively easy to train. The average translation Mean-Squared-Error plot converged to 0.05 after 20 epochs. The 5-degree plot takes sufficient longer time to converge, which means that rotation is a more challenging task.

For the POSE dataset, the segmentation, rotation, and translation results are shown in Table 4. From which we can tell the segmentation task is also easy to accomplish. The precision, recall, and IOU results showed that all combinations of data modalities achieved almost perfect segmentation outcomes. Among all data modalities, the normal modality again achieved the highest rotation accuracy in the threshold range form 1- to 7-degree. While the RGB modality reached the second highest 5-degree accuracy, the combination of all modalities achieved the second highest 7-degree accuracy. Adding the normal modality also yields the lowest translation error.
Table 4: Results on POSE data

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>IOU</th>
<th>1 degree</th>
<th>3 degree</th>
<th>5 degree</th>
<th>7 degree</th>
<th>translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.05</td>
<td>0.31</td>
<td>0.50</td>
<td>0.60</td>
<td>0.38</td>
</tr>
<tr>
<td>Depth</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.02</td>
<td>0.26</td>
<td>0.46</td>
<td>0.60</td>
<td>0.41</td>
</tr>
<tr>
<td>Normal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.08</td>
<td>0.46</td>
<td>0.67</td>
<td>0.78</td>
<td>0.25</td>
</tr>
<tr>
<td>RGB+Depth</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.01</td>
<td>0.07</td>
<td>0.20</td>
<td>0.36</td>
<td>0.51</td>
</tr>
<tr>
<td>RGB+Normal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.01</td>
<td>0.17</td>
<td>0.38</td>
<td>0.55</td>
<td>0.29</td>
</tr>
<tr>
<td>Depth+Normal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.08</td>
<td>0.26</td>
<td>0.45</td>
<td>0.39</td>
</tr>
<tr>
<td>RGB+Depth+Normal</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.02</td>
<td>0.23</td>
<td>0.48</td>
<td>0.64</td>
<td>0.48</td>
</tr>
</tbody>
</table>

The validation loss, IOU, translation and rotation accuracy for Normal data are shown in Figure 4.

Figure 4: loss, translation MSE, 5-degree, and IOU plots for POSE Normal modality

Figure 4 shows that total loss, translation MSE, and IOU almost converged after 50 epochs, while the 5-degree accuracy converged very slowly when training on the POSE dataset.

Segmentation results on synthetic test sets are shown in Figure 5. It’s clear to see that the bunny objects are well segmented. The predicted mask match the ground truths very well for both POLY and POSE data.
Figure 5. Segmentation results on synthetic set. 1st row: POLY, 2nd row: POSE

Discussion

1. **Comparison between the POLY data and POSE data**
   Discussing the segmentation task, the CNN can detect the bunny and can segment the object with almost perfectly for both datasets. However, for the pose estimation task, the model performed much better on POLY dataset than on the POSE dataset for both rotation and translation regression. The discrepancy could be attributed to the fact that the object average size of the bunny object in the POLY dataset is much larger (around 50%) relative to the whole image when comparing to the bunny object in the POSE dataset (around 1%). Larger objects are much easier to recognize due to the higher aspect ratio and clear surface. Besides, the bunny object instances in POLY dataset are all centered in the images, while they are at random locations in the POSE dataset. Some bunny images even get truncated in POSE dataset. These conditions caused the POSE dataset to be more challenging than the POLY dataset when estimating the 6D pose.

2. **Comparison among different data modalities**
   A single normal modality achieved the highest 5-degree accuracy for both the POLY dataset and the POSE dataset. For the POLY dataset, the RGB modality yielded the lowest accuracy comparing to other single modalities. Adding depth data to RGB data produces a higher accuracy in comparison to normal data and RGB. This result could be attributed to the fact that all normal images are generated from RGB images, thus, the two data modalities are highly correlated, adding one to the other does not help adding more meaningful information comparing adding Depth images. For both datasets, the combination of RGB, normal and depth data achieved the second highest 5-degree accuracy, and it even yielded the highest 3-degree accuracy for POLY dataset based on a
simple concatenation strategy. This result is promising. We believe further improvements are possible when more innovative fusion strategies are explored to better combine different unique features.

3. Evaluation on real recorded data

The results on the synthetic test sets are very close to the one yielded with the synthetic validation set, which means that the synthetic training set, validation set, and test set are from the same distribution, and the validation set is very representative, thus could generalize to the test set. However, when evaluating the trained models using a real camera recorded dataset, the tested network architectures failed to find the bunny object. We conducted the following experiments to find out the underlying problems.

a. CNN for very small objects

A typical CNN has alternating layers of convolutions and pooling which progressively results in smaller-resolution feature maps high up the processing pipeline. That can be disadvantageous for detection of fine-grained spatial information because, at each pooling layer, spatial information is lost. Consequently, the detection of a small object is a challenging task.

One way to detect a small object is to up-scale the source images, increase the object size using a pyramid approach, which processes images at multiple scales. Another approach is to avoid pooling layers and to use dilated convolution instead. Dilated convolutional layers can preserve a lot of spatial information but still include a large context.

The original proposed network failed to detect the bunny asset in the evaluation set, although it achieved relatively good performance on the synthetic test set. After applying certain filters to improve detection such as multi-scaled source images, reduce pooling layers, and dilated convolution in the up-sampling path, the network is able to find the general shape of the bunny. Some visual sample results are shown in Figure 6.

![Figure 6. Segmentation results after re-designed the network for small objects](image_url)
b. Examine the quality of evaluation set
Considering the fact that most of the networks achieved good segmentation accuracies on the synthetic dataset, however, failed on the evaluation dataset, we suspect that the synthetic dataset and the real evaluation dataset may expose the CNN to different data distributions. Consequently, the network trained on synthetic data encounters problems when confronted with the real dataset.

Even though both datasets contain the bunny asset, there is an essential difference between the two datasets. The synthetic datasets were rendered by augmenting the bunny onto a random background image, so that the bunny appears in the foreground. No other objects occlude the bunny in any way. However, this is not the case for the real dataset. Here, a human is holding the bunny, partially occluding its shape and boundary. Additionally, on the real data, the human dominates the image area, which might cause the network to ignore the much smaller bunny and take the bunny as part of the human hand.

This observation was further verified when a region proposal network was employed to detect the bunny. The result is shown in Figure 7. Here, we can clearly observe that the network cannot recognize the bunny when the human is holding it. On the other side, it can find the bunny with a high confidence score when the human is absent.

![Figure 7. Bounding box detection results using region proposal net](image)

**Conclusion**

In this study, we explored the potential of minimizing the reality gap by utilizing different data modalities. The results on synthetically generated test data show that adding depth and normal modalities do increase the 5-degree accuracy for the POLY dataset, and the combination of all three data modalities show promising results on both, the POLY and the POSE dataset. Also, we successfully debugged the network for the issues encountered when exposing it to the real dataset.
and identified challenges in the underlying data generation process. Thus, we conclude that not all models trained on synthetic data are transferable to real life data. Domain randomized images are limited when evaluated on real life data which are in certain obvious scenes. Scene context is very important information crossing different domains. More understandings on the differences among domains are needed to maximize our expectations when transferring. We believe we can have more control on the reality gap if we create training data which has scene context matches the evaluation set. Also, the data distributions must be close: incorporating a broad range of synthetic training instances in terms of data variations will largely increase the detection probability when evaluate on real life data. Consequently, we expect that these measures can increase the object detection and pose estimation accuracy.

References


[4] Hodan, Tomas; Michel, Frank; Brachmann, Eric; Kehl, Wadim; Glent Buch, Anders; Kraft, Dirk; Drost, Bertram; Vidal, Joel; Ihrke, Stephan; Zabulis, Xenophon; Sahin, Caner; Manhardt, Fabian; Tombari, Federico; Kim, Tae-Kyun; Matas, Jiri; Rother, Carsten. BOP: Benchmark for 6D Object Pose Estimation. (2018) eprint arXiv:1808.08319


[12] Tremblay, Jonathan; Prakash, Aayush; Acuna, David; Brophy, Mark; Jampani, Varun; Anil, Cem; To, Thang; Cameracci, Eric; Boochoon, Shaad; Birchfield, Stan. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. (2018) eprint arXiv:1804.06516
