Proceedings of the Joint Austrian Computer Vision and Robotics Workshop 2020 DOI: 10.3217/978-3-85125-752-6-25

Image Synthesis in $SO(3)$ by Learning Equivariant Feature Spaces

Marco Peer, Stefan Thalhammer, and Markus Vincze
Faculty of Electrical Engineering and Information Technology, TU Wien, 1040 Vienna, Austria
marco.peer@tuwien.ac.at, {thalhammer,vincze}@acin.tuwien.ac.at

Abstract. *Equivariance is a desired property for feature spaces designed to make transformations between samples, such as object views, predictable. Encoding this property in two dimensional feature spaces for 3D transformations is beneficial for tasks such as image synthesis and object pose refinement. We propose the Trilinear Interpolation Layer that applies $SO(3)$ transformations to the bottleneck feature map of an encoder-decoder network. By employing a 3D grid to trilinearly interpolate in the feature map we create models suited for view synthesis with three degrees of rotational freedom. We quantitatively and qualitatively evaluate on image synthesis in $SO(3)$ providing evidence of the suitability of our approach.*

1. Introduction

Invariant feature spaces are agnostic to input transformations in order to help models overcome variations in the data capturing process. Equivariant feature spaces are exploitable with respect to image space transformations, thus more suited for reasoning about changes in image space [9]. As a consequence the property of equivariance is desired for feature spaces that are used for predicting transformations of or in the image space. More precisely, equivariant feature spaces can be exploited to predict unseen views based on known transformations or to estimate relative transformations between two inputs. Feature spaces that correlate an input with a transformed output via observable transformation parameters are desired for applications such as image synthesis or object pose refinement.

In this work we study the equivariance of features spaces of Convolutional Neural Networks (CNN) as motivated by the task of object pose refinement. This motivation arises from recent RGB-based object pose refinement methods that use pairs of images [10, 21]:

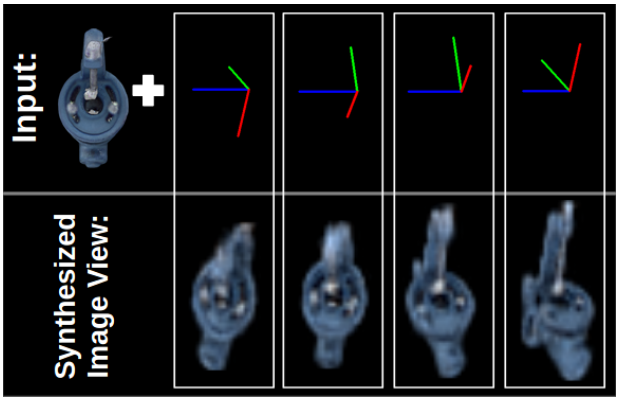


Figure 1: Given an object view and a relative 3D rotation, unseen views are synthesized.

One image represents the observation of the desired object and the other image usually a rendering of the object in a hypothesized pose. A network is trained to predict the relative transformation between the input pair. We study how to correlate such an image pair, in feature space, in order to achieve predictability of the relative object transformations.

The Spatial Transform Network (STN) [6] provides a mean to learn image space transformations conditioned on the input to produce a transformed output feature map. Studies such as [2, 13] apply a sub part of the STN, known as the Spatial Transformer Layer (STL), to properly align the network's output with its input by applying image space translations. The authors of [19] wrap a projection function around the in- and outputs of the STL in order to make image properties such as lighting and $SO(3)$ transformation in a limited range predictable. Alternatively to their approach, we directly modify the structure of the STL. We extend the STL to enable trilinear interpolation of a feature map in order to interpret transformations in all of $SO(3)$. In the remainder of the paper it is referred to as the Trilinear Interpolation Layer (TIL).

Our contributions are:

- We propose a Trilinear Interpolation Layer suited for creating equivariant feature spaces in $SO(3)$.
- We provide quantitative and qualitative evidence for the advantage of equivariant feature spaces by predicting unseen views in $SO(3)$ of objects from the LineMOD dataset [4].

The remainder of the paper is structured as follows. Section 2 reviews related work. In Section 3 we describe our approach. Section 4 presents our experimental results. Finally, Section 5 concludes the paper.

2. Related Work

Object pose refiners rely on the availability of prior stages to produce pose hypotheses [7, 10, 12, 16, 18, 20, 21]. When depth data is available, the Iterative-Closest-Point algorithm (ICP) can be used to refine initial pose estimates [18, 7, 20]. Recent RGB-based approaches do not rely on the availability of depth data for pose refinement [7, 10, 12, 16, 21]. CNN-based object pose refinement architectures such as [10, 12, 21] pass two input images to the network in order to estimate the relative rotation between these. These images are an observation of the object in the desired pose and a rendering of the prediction. In [10] the authors base their network architecture on an approach for optical flow estimation [1] and predict optical flow, mask and relative pose deviation in $SE(3)$. The authors of [21] use a similar approach with two encoders, one per input image. The encoders' outputs are subtracted and further encoded to predict the refined pose in $SE(3)$. We present a concept suitable for enhancing such methods by guiding the network to learn an equivariant feature space.

The STN introduced by [6] is widely used for feature and image space transformation [2, 13, 14, 15, 19]. It consists of the combination of a localization network, a grid generator and a sampler.. The authors of [2] apply STL to properly align the features to their inputs. In [13] the authors predict deep heatmaps from randomly sampled object patches to predict poses under occlusion. They apply the STL to upsample their predictions. In [14, 15] an analog of the localization network is used to produce feature maps invariant to input transformations. The authors of [11] leverage on the methodology of STN to generate realistic looking images from the intersection of the natural image and geometric manifold, using an adapted Generative Adversarial Network. Conversely to these approaches we modify the STL component of STN to enable $SO(3)$ transformations of input feature maps with spatial dimension.

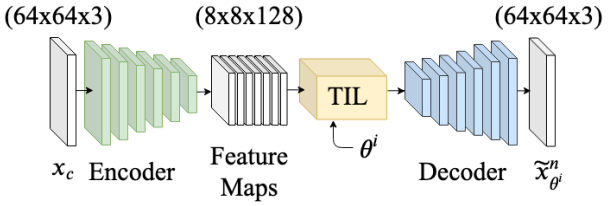


Figure 2: Encoder-decoder architecture for image synthesis.

3. Approach

This section presents our approach for learning equivarient features in $SO(3)$ in order to synthesize images from unseen viewpoints. We first give a problem definition, then describe the Trilinear Interpolation Layer. Finally, we outline how the TIL is used in an encoder-decoder architecture for image synthesis.

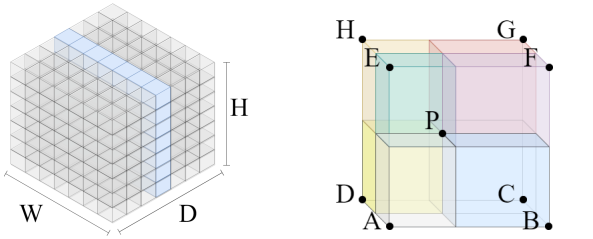
3.1. Problem Statement

Let $X = \left\{x_c, \left(\tilde{x}^0_{\theta^0}, ..., \tilde{x}^n_{\theta^i}\right)\right\}$ be a set of training examples where x_c refers to the projection $\prod$ of object o_c, in its canonical pose, to the image space I. The set of $\tilde{x}^n_{\theta^i}$ are the projections of transformed objects o_{θ^i} where θ^i represent the transformation in $SO(3)$ for the projection into I. Our goal is to learn the inverse of the mapping function $\prod^{-1}$ in order to produce transformed images. In other words, to learn $\tilde{x}^n_{\theta^i} = \prod\left[\prod^{-1}(x_c), \theta^i\right]$ given an image of the object in its canonical pose and transformation parameters.

In order to model the inversion of the mapping function $\prod$, we utilize a CNN due to their power to encode statistical relationships from visual data into feature spaces [8]. To provide information regarding relative transformations θ^i in $SO(3)$ between pairs of images to our model, we modify the STL of [6]. An overview of the encoder-decoder architecture for image synthesis using the modified STL is presented in Figure 2.

3.2. Trilinear Interpolation

The STL [6] allows $SE(2)$ transformations to be applied to feature maps. This works well in image



(a) Three dimensional grid. The feature map are centered in D on initialization.

(b) Interpolation scheme of P using adjacent grid cell values.

Figure 3: Trilinear Interpolation Layer (TIL) components.

space, however, requires adaptation for the $SO(3)$ domain [19]. The STL is composed of a grid generator and a sampler.

The grid generator is modified by adding a depth dimension D. The input feature map of space $\mathbb{R}^{H\times W\times C}$ thus becomes $\mathbb{R}^{H\times W\times D\times C}$, where H, W and C are the height, width and number of channels, respectively. The feature map is centered along D as shown in Figure 3a. The sampler of the STL bilinearly interpolates between corner points using the corresponding areas. For volumes, this scheme is unsuitable. Therefore, trilinear interpolation is used instead, as shown in Figure 3b. Feature maps are interpolated channel-wise and projected back to 2D by averaging along the depth dimension. In order to guarantee proper interpolation in 3D, H and W must be greater than 1. The proposed modification enables transformations in $SO(3)$ and only affects non-trainable layers. Thus, the additional computational overhead compared to STL is negligible.

Since averaging over D is used for projecting the grid back to 2D no feature map scaling can be applied while sampling. Thus, modifying the trilinear interpolation by allowing scaling along depth would enable transformations in $SE(3)$, thus yielding full 6DoF. However, this is out of the scope in this paper.

3.3. Network Architecture

The network in Figure 2 is an encoder-decoder architecture. The encoder consists of a truncated ResNet18 [3], pretrained on ImageNet [17], for feature encoding. ResNet18 consists of five stages. In order to preserve a larger spatial image dimension we remove the fourth and the fifth stage and take the outputs of the last Rectified Linear Unit (ReLU) of stage three. The final output is a tensor of size 8×8 with 128 feature maps.

The encoded image $\prod^{-1}(x_c)$ as well as the transformation parameters θ^i are passed to the TIL. Feature maps are trilinearly interpolated to produce the mapping of the encoded transformed image $\tilde{x}^n_{\theta^i}$. The transformed encoding is forwarded to the decoder stage of the network.

The design of the decoder is rather ad-hoc to show that the TIL is not restricted to a certain architecture. A transposed convolution with ReLU activation is followed by stacks of deconvolution layers with ReLU activation and upsampling layers. These stacks are repeated two times and a final transposed convolution layer with linear activation is added. Feature channels are reduced gradually. Kernel sizes of the transposed convolutional layers are $5-3-3-5$ and upsampling kernel sizes of 3×3 are used. All strides are set to 1. The output of the decoder is an image of size 64×64.

In each training iteration, the deviation of $\tilde{x}_\theta$ to x_θ is minimized. The loss function to be optimized is $l2$. The network is trained to correlate objects views with its corresponding transformation in $SO(3)$ in the camera frame. Consequently, a feature space is created that enables to synthesize views not included in the training set.

4. Experiments

This section presents experiments for image synthesis of unseen views of household objects with little texture. These experiments show that the extension of an encoder-decoder network with the proposed TIL reconstructs objects views in $SO(3)$. In addition, the method can also reconstruct views in regions of $SO(3)$ where no data was provided to the network during training.

4.1. Dataset

Our experiments are conducted on a subset of the LineMOD dataset [4]. We use the object models of *Benchvise*, *Cat*, *Glue*, *Camera* and *Lamp*. These objects represent elongated and asymmetric shapes as well as complex shapes with self occlusion. With this subset we cover the representative challenges when synthesizing views for objects.

4.2. Dataset Creation

Dataset images are rendered using the renderer provided by [5]. For our purposes, the RGB images are scaled to 64×64 pixels. To each object's canonical pose, $45°$ are added to elevation in order to only train on views of the upper hemisphere of the object.

metric	latent space	loss			
		l1	*l2*	DSSIM	DSSIM + *l1* ($\delta = 0.85$)
l1	2x2x512	0.03 ± 4.9e-04	0.03 ± 4.2e-04	0.028 ± 4.0e-04	0.031 ± 4.3e-04
l2		0.096 ± 2.1e-05	0.093 ± 1.6e-02	0.093 ± 1.8e-03	0.1 ± 1.2e-03
DSSIM		0.102 ± 3.5e-03	0.105 ± 2.6e-03	0.09 ± 3.0e-03	0.1 ± 3.4e-03
l1	4x4x256	0.018 ± 3.5e-04	0.02 ± 3.6e-04	0.0243 ± 4.0e-04	0.018 ± 3.2e-04
l2		0.065 ± 1.8e-05	0.0064 ± 1.7e-05	0.086 ± 2.5e-06	0.063 ± 1.7e-05
DSSIM		0.061 ± 3.0e-05	0.067 ± 3.6e-05	0.07 ± 2.8e-05	0.059 ± 3.2e-05
l1	8x8x128	**0.016 ± 2.6e-04**	0.017 ± 2.6e-04	0.017 ± 25e-05	0.017 ± 2.4e-04
l2		0.06 ± 1.6e-03	**0.057 ± 1.7e-03**	0.06 ± 1.7e-03	0.066 ± 1.6e-03
DSSIM		0.055 ± 3.0e-03	0.06 ± 3.0e-03	**0.053 ± 2.9e-03**	0.055 ± 2.6e-03

Table 1: Performance study for latent spatial dimension and loss function. We present the error and variance, averaged over all objects, using *l1*, *l2* and DSSIM respectively.

Based on the newly defined canonical pose, images are rendered in a range of -43° to +43° azimuth and elevation. This is similar to [19] but with approximately three times the range in azimuth angle.

For training, only views in a range of -37° to +37° azimuth and elevation are used. Of these 950 images, 43 images are exclusively used for testing. The selected samples are distributed uniformly in the viewing cone. An additional 59 images are included in the test set. These are in an angle range of negative and positive 37° to 43° azimuth and elevation. Thus, views in a range that are not shown to the network during training.

4.3. Training Protocol

For training we use the Adam optimizer with the learning rate set to 10^{-3}. A batch size of 1 is used. We train 40 epochs per object for quantitative ablation studies. After 30 epochs, the learning rate is decreased by one magnitude. Qualitative evaluation is presented after 40 epochs of training. During training, Gaussian blur with uniformly sampled $\sigma = [0.0, 1.5]$ is used as online augmentation.

4.4. Hyperparameter Studies

We study the choice of loss function used for optimization and the optimal size of the bottleneck feature maps. Table 1 presents results averaging over the test sets of all five objects. Presented are the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Structural Similarity Index (SSIM) as well as their corresponding variances.

The loss functions compared are $l1$, $l2$, Structural Disimilarity (DSSIM) and a combination of $l1$ and DSSIM as used by [19], where δ is the weighting parameter. The bottleneck tensor size is adjusted by truncating ResNet18. For a dimension of $4 \times 4 \times 256$ we use the outputs of the fourth and upsample using three stacks of transposed convolutions plus upsampling layers. For $2 \times 2 \times 512$ we use four stacks starting with a 5×5 transposed convolution.

Tensor size	2x2x512	4x4x256	8x8x128
parameters	13,330,508	3,753,804	1,881,932

Table 2: Network parameters per bottleneck tensor size.

Quantitative evaluation shows that the metric used for evaluating the reconstruction quality correlates with the loss function used, which is to be expected. Using $l2$ is reasonable. However, when synthesizing views for a specific application more carefully choosing the loss function will be obligatory. Surprisingly, a bottleneck tensor size of $8 \times 8 \times 128$ leads to image synthesis with the lowest error even though this network has far fewer parameters than the other spatial dimensions (see Table 2). This leads to the conclusion that bigger spatial dimensions are more important for synthesizing views than network depth. Based on the chosen hyperparameters we further present experiments for synthesizing views.

4.5. Studies on View Synthesis

Studies are presented to illustrate that the proposed formulation generates feature spaces suited for view synthesis in $SO(3)$. Figure 4 shows views synthesized from unseen transformations during training time. Additionally, we present view predictions outside of the training range. Views inside the training range are reconstructed with sufficient quality to visually verify the expected object orientations. Despite the reconstruction quality being poor for

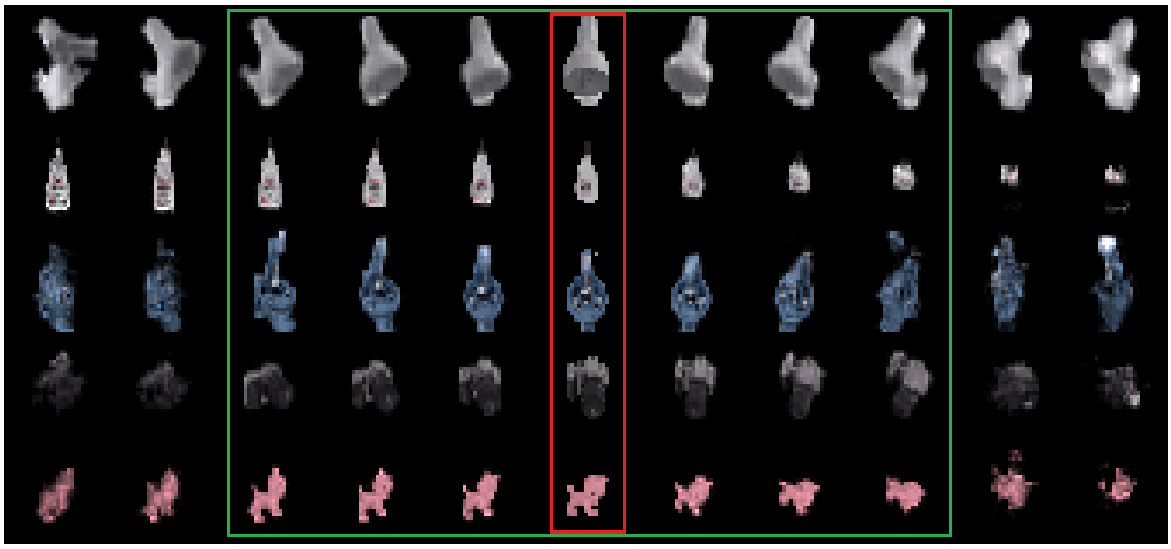


Figure 4: View synthesis from $SO(3)$ transformations unseen during training time. First row: reconstructed *Lamp* with varying azimuth from -43° to 43°. Second row: reconstructed *Glue* with elevation variation from -43° to 43°. Row three to five: objects *Benchvise, Camera, Cat* reconstructed with azimuth/elevation range from (-43°,-43°) to (43°,43°). Object poses outside the green box are samples out of training distribution. Centered images, in the red box, mark the canonical poses.

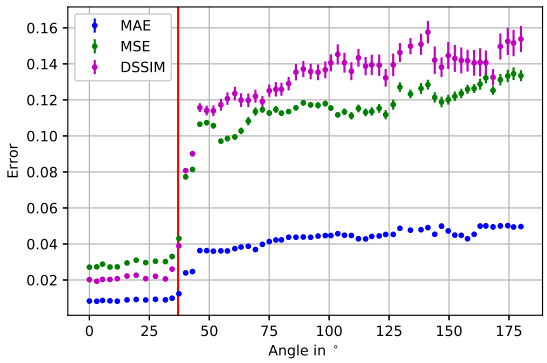


Figure 5: Error values and its variance over azimuth angle. The network was trained on its corresponding loss function with a spatial bottleneck dimension of $8 \times 8 \times 128$. The vertical line shows the training set range.

some of the synthesized views outside of the training range, it is visible that views can be predicted properly based on $SO(3)$ transformations.

Figure 5 provides reconstruction error and variance over an extended azimuth and elevation angle range of [0, 180°]. The results in the figure are averaged over all objects. The training dataset contains images with azimuth angles up to 37°. A sharp rise in error and variance is observed at azimuth angle of approximately 45°. For angles above this value, error and variance increase rapidly. As such, the network cannot properly reconstruct these views.

These results show that our formulation for creating equivariant feature spaces has the desired property to correlate spatial transformations with 2D views of the transformed object. Thus, the proposed Trilinear interpolation layer guides the network towards learning an equivariant feature space in $SO(3)$.

5. Conclusion

We extend recent work for learning equivariant feature spaces for synthesizing object views in $SO(3)$. The proposed extension of the Spatial Transform Network [6], that we call the Trilinear interpolation Layer, applies $SO(3)$ transformations to feature maps from 2D data. Validity of the approach is provided by training a simple encoder-decoder network architecture. Our experiments show that our formulation not only enables the prediction of views unseen during training time but also in a small range outside.

The current formulation enables control for 5DoF, $SO(3)$ and translations in image space. Future work will tackle adapting the proposed layer to create object view synthesis in all of $SE(3)$. We then plan to integrate this in a pose refinement strategy to improve object pose estimation.

ACKNOWLEDGMENT

This work has been supported by the Austrian Research Promotion Agency in the program Production of the Future funded project MMAssist_II (FFG No. 858623), the Austrian Ministry for Transport, Innovation and Technology (bmvit) and the Austrian Science Foundation (FWF) under grant agreement No. I3969-N30 (InDex).

References

[1] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[4] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Proceedings of Asian conference on computer vision*, pages 548–562, 2012.

[5] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[6] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[7] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. *CoRR*, abs/1711.10006, 2017.

[8] 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.

[9] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

[10] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[11] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9455–9464, 2018.

[12] F. Manhardt, W. Kehl, N. Navab, and F. Tombari. Deep model-based 6d pose refinement in rgb. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 800–815, 2018.

[13] M. Oberweger, M. Rad, and V. Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[16] M. Rad and V. Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3836, 2017.

[17] 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. *International Journal of Computer Vision*, 115:211–252, 2015.

[18] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018.

[19] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Interpretable transformations with encoder-decoder networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5726–5735, 2017.

[20] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018.

[21] S. Zakharov, I. Shugurov, and S. Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1941–1950, 2019.