

# Designing Deep Convolutional Neural Networks for Continuous Object Orientation Estimation

Kota Hara, Raviteja Vemulapalli and Rama Chellappa

Center for Automation Research, UMIACS, University of Maryland, College Park

## Abstract

*Deep Convolutional Neural Networks (DCNN) have been proven to be effective for various computer vision problems. In this work, we demonstrate its effectiveness on a continuous object orientation estimation task, which requires prediction of  $0^\circ$  to  $360^\circ$  degrees orientation of the objects. We do so by proposing and comparing three continuous orientation prediction approaches designed for the DCNNs. The first two approaches work by representing an orientation as a point on a unit circle and minimizing either L2 loss or angular difference loss. The third method works by first converting the continuous orientation estimation task into a set of discrete orientation estimation tasks and then converting the discrete orientation outputs back to the continuous orientation using a mean-shift algorithm. By evaluating on a vehicle orientation estimation task and a pedestrian orientation estimation task, we demonstrate that the discretization-based approach not only works better than the other two approaches but also achieves state-of-the-art performance. We also demonstrate that finding an appropriate feature representation is critical to achieve a good performance when adapting a DCNN trained for an image recognition task.*

## 1. Introduction

The effectiveness of the Deep Convolutional Neural Networks (DCNNs) has been demonstrated for various computer vision tasks such as image classification [17, 19, 33, 31], object detection [12, 11, 27], semantic-segmentation [20, 4, 22, 25], human body joint localization [38, 36, 5], face recognition [34] and so on. Due to the large number of network parameters that need to be trained, DCNNs require a significant number of training samples. For tasks where sufficient number of training samples are not available, a DCNN trained on a large dataset for a different task is tuned to the current task by making necessary modifications to the network and retraining it with the available training data [42]. This transfer learning technique has been proven to be effective for tasks such as fine-grained recog-

nition [30, 13, 2], object detection [12, 29], object classification [43], attribute detection [30, 2] and so on.

One of the tasks for which a large number of training samples are not available is the continuous object orientation estimation problem where the goal is to predict the continuous orientation of objects in the range  $0^\circ$  to  $360^\circ$ . The orientations are important properties of objects such as pedestrians and cars and precise estimation of the orientations allows better understanding of the scenes essential for applications such as autonomous driving and surveillance. Since in general it is difficult to annotate the orientations of objects without a proper equipment, the number of training samples in existing datasets for orientation estimation tasks is limited. Thus, it would be interesting to see if and how it would be possible to achieve good performance by adapting a DCNN trained on a large object recognition dataset [28] to the orientation estimation task.

The first consideration is the representation used for prediction. When a DCNN is trained for an image classification task, the layers inside the network gradually transform the raw pixel information to more and more abstract representation suitable for the image classification task. Specifically, to achieve good classification ability, the representations at later layers have more invariance against shift/rotation/scale changes while maintaining a good discriminative power between different object classes. On the other hand, the orientation estimation task requires representation which can capture image differences caused by orientation changes of the objects in the same class. Thus, it is important to thoroughly evaluate the suitability of representations from different layers of a DCNN trained on the image classification task for the object orientation estimation task.

The second consideration is the design of the orientation prediction unit. For the continuous orientation estimation task, the network has to predict the angular value, which is in a non-Euclidean space, prohibiting the direct use of a typical L2 loss function. To handle this problem, we propose three different approaches. The first approach represents an orientation as a 2D point on a unit circle, then trains the network using the L2 loss. In test time, the net-

work’s output, a 2D point not necessarily on a unit circle, is converted back to the angular value by atan2 function. Our second approach also uses a-point-on-a-unit-circle representation, however, instead of the L2 loss, it minimizes a loss defined directly on the angular difference. Our third approach, which is significantly different from the first two approaches, is based on the idea of converting the continuous orientation estimation task into a set of discrete orientation estimation tasks and addressing each discrete orientation estimation task by a standard softmax function. In test time, the discrete orientation outputs are converted back to the continuous orientation using a mean-shift algorithm. The discretized orientations are determined such that all the discretized orientations are uniformly distributed in the output circular space. The mean-shift algorithm for the circular space is carried out to find the most plausible orientation while taking into account the softmax probability for each discrete orientation.

We conduct experiments on car orientation estimation and pedestrian orientation estimation tasks. We observe that the approach based on discretization and mean-shift algorithm outperforms the other two approaches with a large margin. We also find that the final performance significantly varies with the feature map used for orientation estimation. We believe that the findings from the experiments reported here can be beneficial for other object classes as well.

The paper is organized as follows. Section 2 discusses related works. Section 3 presents three proposed approaches. Section 4 shows experimental results and conclusions are given in section 5.

## 2. Related Work

Object orientation estimation problem has been gaining more and more attention due to its practical importance. Several works treat the continuous orientation estimation problem as a multi-class classification problem by discretizing the orientations. In [23], a three-step approach is proposed where a bounding box containing the object is first predicted, then orientation is estimated based on image features inside the predicted bounding box, and finally a classifier tuned for the predicted orientation is applied to check the existence of the object. [10] address the orientation estimation task using Fisher encoding and convolutional neural network-based features. [3] learns a visual manifold which captures large variations in object appearances and proposes a method which can untangle such a visual manifold into a view-invariant category representation and a category-invariant pose representation.

Some approaches address the task as continuous prediction in order to avoid undesirable approximation error caused by discretization. In [16], a joint object detection and orientation estimation algorithm based on structural SVM is proposed. In order to effectively optimize a non-

convex objective function, a cascaded discrete-continuous inference algorithm is introduced. In [14, 15], a regression forest trained with a multi-way node splitting algorithm is proposed. As an image descriptor, HOG features are used. [35] introduces a representation along with a similarity measure of 2D appearance based on distributions of low-level, fine-grained image features. For continuous prediction, an interpolation based approach is applied. A neural network-based model called Auto-masking Neural Network (ANN) for joint object detection and view-point estimation is introduced in [41]. The key component of ANN is a mask layer which produces a mask passing only the important part of the image region in order to allow only these regions to be used for the final prediction. Although both our method and ANN are neural network-based methods, the overall network architectures and the focus of the work are significantly different.

Several works consider learning a suitable representation for the orientation estimation task. In [37], an embedded representation that reflects the local features and their spatial arrangement as well as enforces supervised manifold constraints on the data is proposed. Then a regression model to estimate the orientation is learned using the proposed representation. Similarly to [37], [8, 9] learn a representation using spectral clustering and then train a single regression for each cluster while enforcing geometric constraints. [7] formulates the task as a MAP inference task, where the likelihood function is composed of a generative term based on the prediction error generated by the ensemble of Fisher regressors as well as a discriminative term based on SVM classifiers.

[40] introduces PASCAL3D+ dataset designed for joint object detection and pose estimation. Continuous annotations of azimuth and elevation for 12 object categories are provided. The average number of instances per category is approximately 3,000. The performance is evaluated based on Average Viewpoint Precision (AVP) which takes into account both the detection accuracy and view-point estimation accuracy. Since the focus of this work is the orientation estimation, we employ the EPFL Multi-view Car Dataset [23] and the TUD Multiview Pedestrian Dataset [1] specifically designed to evaluate the orientation prediction.

Despite the availability of continuous ground-truth view point information, majority of works [40, 39, 21, 24, 10, 32] using PASCA3D+ dataset predict discrete poses and evaluate the performance based on the discretized poses. [39] proposes a method for joint view-point estimation and key point prediction based on CNN. It works by converting the continuous pose estimation task into discrete view point classification task. [32] proposes to augment training data for their CNN model by synthetic images. The view point prediction is cast as a fine-grained (360 classes for each angle) discretized view point classification problem.

### 3. Method

Throughout this work we assume that a single object, viewed roughly from the side, is at the center of the image and the orientation of the object is represented by a value ranging from  $0^\circ$  to  $360^\circ$ . Before feeding to the network, we first resize the input image to a canonical size and then subtracted the dataset mean. The network then processes the input image by applying a series of transformations, followed by an orientation prediction unit producing the final estimates. In this section, we present each of the proposed orientation prediction units in details. All the prediction units are trained by back propagation.

#### 3.1. Orientation Estimation

##### 3.1.1 Approach 1

We first represent orientation angles as points on a unit circle by  $v = (\cos \theta, \sin \theta)$ . We then train a standard regression layer with a Huber loss function, also known as smooth L1 loss function. The Huber loss is used to improve the robustness against outliers, however, a standard L2 or L1 loss can also be used when appropriate. During testing, predicted 2D point coordinate  $v = (x, y)$  is converted to the orientation angle by  $\theta = \text{atan}2(y, x)$ . A potential issue in this approach is that the Huber loss function, as well as L2 or L1 loss functions, consider not only the angular differences but also the radial differences that are not directly related to the orientation.

##### 3.1.2 Approach 2

As in approach 1, we represent orientation angles as points on a unit circle and train a regression function, however, we use a loss function which focuses only on the angular differences:

$$L(v_g, v) = 1 - \cos(\theta) = 1 - \frac{v_g \cdot v}{|v_g||v|} = 1 - \frac{x_g x + y_g y}{\sqrt{x^2 + y^2}} \quad (1)$$

where  $v_g = (x_g, y_g)$  is the ground-truth. Note that  $|v_g| = 1$  by definition. The derivative of  $L$  with respect to  $x$  is computed as

$$\frac{\partial L}{\partial x} = \frac{(x_g x + y_g y) \frac{x}{\sqrt{x^2 + y^2}} - x_g \sqrt{x^2 + y^2}}{x^2 + y^2} \quad (2)$$

We compute  $\frac{\partial L}{\partial y}$  similarly. These derivatives allow us to train the network parameters by back propagation. As in approach 1, during testing, the predicted 2D point coordinates are converted to orientation angles by the atan2 function.

A potential issue in this approach is that the derivatives approaches 0 when angular difference becomes close to  $180^\circ$ , making the optimization more challenging.

##### 3.1.3 Approach 3

We propose an approach based on discretization. The network architecture illustrating this approach is presented in Fig. 1. We first discretize the  $0$ - $360$  range into  $N$  unique orientations which are  $G = 360/N^\circ$  degree apart and convert the continuous prediction task into an  $N$ -class classification task. Each training sample is assigned one of the  $N$  class labels based on its orientation's proximity to the discretized orientations. In order to alleviate the loss of information introduced by the discretization, we construct  $M$  classification tasks by having a different starting orientation for each discretization. The  $M$  starting orientations equally divide  $G$  degree. Formally, the discrete orientations for the  $m$ -th classification task are  $\{m \times G/M + k \times G\}_{k=0,\dots,N-1}$ , where  $m = 0, \dots, M-1$ . The example discretization with  $N = 4, M = 3$  is depicted in Fig. 2. As an orientation estimation unit, we thus have  $M$  independent  $N$ -way softmax classification layers, which are trained jointly.

During testing, we compute softmax probabilities for each of the  $M$  classification tasks separately. Consequently, we obtain probabilistic votes at all of the  $M \times N$  unique orientations. We then define a probability density function using weighted kernel density estimation adopted for the circular space. We use the von-Mises distribution as a kernel function. The von-Mises kernel is defined as

$$k_\nu(\theta) = \frac{1}{2\pi I_0(\nu)} \exp(\nu \cdot \cos(\theta)) \quad (3)$$

where  $\nu$  is the concentration parameter and  $I_0(\nu)$  is the modified Bessel function of order 0.

Formally, the density at the orientation  $\theta$  is given by

$$\hat{p}(\theta; \nu) \propto \sum_{i=1}^{M \times N} p_i k_\nu(\theta - \theta_i). \quad (4)$$

where  $\theta_i$  is the  $i$ -th discrete orientation and  $p_i$  is the corresponding softmax probability.

Then final prediction is made by finding the orientation with the highest density:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \hat{p}(\theta; \nu) \quad (5)$$

In order to solve the above maximization problem, we use a mean-shift mode seeking algorithm specialized for a circular space proposed in [15]

The same level of discretization can be achieved by different combinations of  $N$  and  $M$ . For instance, both  $(N, M) = (72, 1)$  and  $(N, M) = (8, 9)$  discretize the orientation into 72 unique orientations, however, we argue that larger  $N$  makes the classification task more difficult and confuses the training algorithm since there are smaller differences in appearances among neighboring orientations. Setting  $M$  larger than 1 and reducing  $N$  could alleviate this

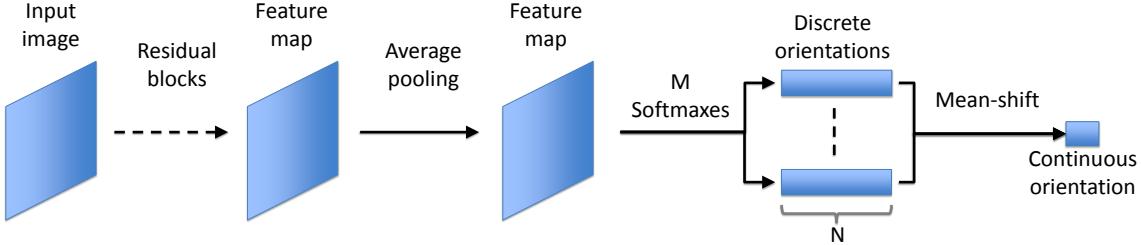


Figure 1. The network architecture for the discretization based approach.

problem while maintaining the same level of discretization. This claim is verified through the experiments.

A potential problem of the proposed approach is the loss of information introduced by the discretization step. However, as shown later in the experiment section, the mean-shift algorithm successfully recovers the continuous orientation without the need for further discretization.

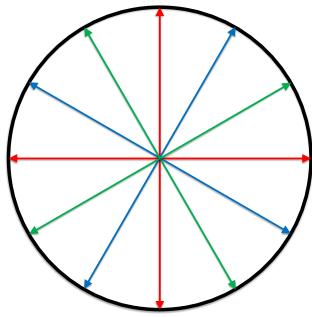


Figure 2. The example discretization constructed by  $N = 4$ ,  $M = 3$ .  $N$  Orientations depicted in the same color are used for one of the  $M$  classification task.

## 4. Experiments

We evaluate the effectiveness of the proposed approaches on the EPFL Multi-view Car Dataset [23] and TUD Multiview Pedestrian Dataset [1]. Both datasets have continuous orientation annotations available.

### 4.1. DCNN

As an underlying DCNN, we employ the Residual Network [17] with 101 layers (ResNet-101) pre-trained on ImageNet image classification challenge [28] with 1000 object categories. The ResNet-101 won the 1st place on various competitions such as ImageNet classification, ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. Although ResNet-152, which is deeper than ResNet-101, achieves better performance than the ResNet-101, we employ ResNet-101 due to its smaller memory footprint.

The key component of the ResNet is a residual block, which is designed to make the network training easier. The residual block is trained to output the residual with reference to the input to the block. The residual block can be easily constructed by adding the input to the output from the block. ResNet-101 consists of 33 residual blocks. Each residual block contains three convolution layers, each of which is followed by a Batch Normalization layer, a scale Layer and the ReLU layer.

### 4.2. Training details

Unless otherwise noted, the weights of the existing ResNet-101 layers are fixed to speed up the experiments. The parameters of the orientation prediction unit are trained by Stochastic Gradient Descent (SGD). All the experiments are conducted using the Caffe Deep Learning Framework [18] on a NVIDIA K40 GPU with 12GB memory. In order to include contextual regions, bounding box annotations are enlarged by a scale factor of 1.2 and 1.1 for EPFL and TUD datasets, respectively. We augment the training data by including the vertically mirrored versions of the samples.

For all experiments, we apply average pooling with size 3 and stride 1 after the last residual block chosen and then attach the orientation prediction unit. The batch size, momentum and weight decay are set to 32, 0.9 and 0.0005, respectively. Weights of the all the orientation prediction layers are initialized by random numbers generated from the zero-mean Gaussian distribution with  $\text{std} = 0.0001$ . All biases are initialized to 0.

For the approach 3, we set  $M$ , the number of starting orientations for discretization, to 9 and  $N$  to 8 for all the experiments unless otherwise stated.

### 4.3. EPFL dataset

The EPFL dataset contains 20 sequences of images of cars captured at a car show where each sequence contains images of the same instance of car captured with various orientations. Each image is given a ground-truth orientation. We use the first 10 sequences as training data and the remaining 10 sequences for testing. As a result, the number of training samples is 2,358 after data augmentation and

that of testing samples is 1,120. We use bounding box information which comes with the dataset to crop out the image region to be fed to the network. The performance of the algorithm is measured by Mean Absolute Error (MeanAE) and Median Absolute Error (MedianAE) in degree following the practice in the literature. Unless otherwise noted, the number of training iterations is 2000 with 0.000001 as a learning rate, followed by the additional 2,000 iterations with 10 times reduced learning rate.

First we conduct experiments to figure out the most suitable representation for the orientation estimation task by attaching the orientation prediction unit to different residual blocks. For these experiments, we use approach 3.

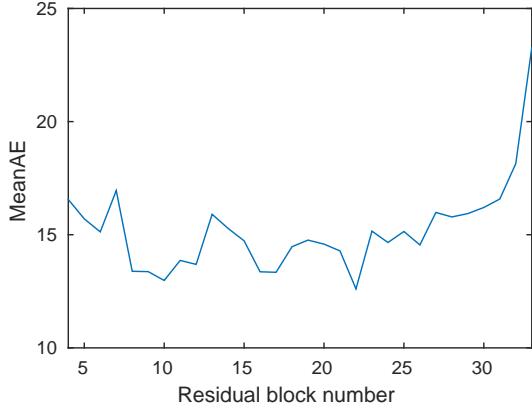


Figure 3. The performance on EPFL dataset obtained by using a different residual block to attach the proposed orientation prediction unit.

In Fig. 3, we show the MeanAE on the EPFL dataset obtained by using different residual blocks. As can be seen, both the earlier and later residual blocks do not provide a suitable representation for the orientation estimation task. The 22nd residual block produces the best representation among 33 residual blocks for our task. The following experiments are conducted using the 22nd residual block.

We analyze the effect of  $M$ , the number of starting orientations for discretization, and  $N$ , the number of unique orientations in each discretization, of the discretization-based approach. The results are summarized in Table. 1. It is observed that when  $N = 8$ , increasing  $M$  leads to better results, however, no significant improvement is observed after  $M = 9$ . When  $N = 72$ , increasing  $M$  from 1 to 5 does not lead to better performance. These results indicate that the larger number of the total orientations leads to better performance up to some point.

When the total number of unique orientations is same, e.g.,  $(N, M) = (72, 1)$  and  $(N, M) = (8, 9)$ , MeanAE is smaller with  $(N, M) = (8, 9)$  while MedianAE is smaller with  $(N, M) = (72, 1)$ . Since MedianAE is very small with both settings, both of them achieve high accuracy in most of

Table 2. Comparison among the proposed approaches on the EPFL dataset.

Approach	MeanAE	MedianAE
1	22.9	11.3
2	26.7	10.2
3	12.6	3.0

the cases.

Table 2 shows the performance of the other two approaches based on a-point-on-a-unit-circle representation. For the approach 1, we train the model with the same setting used for the approach 3. For the approach 2, we train the model for 40,000 iterations with a learning rate of 0.0000001 as it appears necessary for convergence. As can be seen, the discretization-based approach significantly outperforms the other two approaches.

In Table 3, we present results from the literature and the result of our final model (approach 3). For this comparison, instead of fixing the existing ResNet weights, we fine-tune all the network parameters end-to-end, which reduce the MeanAE by 21.7%. As can be seen, our final model advances the state of the art performance.

The information on whether or not the ground-truth bounding box annotations are used in test time is also included in the table. In methods which do not utilize the ground-truth bounding boxes, an off-the-shelf object detector such as DPM [6] is used to obtain the bounding boxes [7] or the localization and orientation estimation are addressed jointly [16, 41, 35, 26, 23].

Finally, we show representative results in Fig. 4 with ground-truth bounding boxes overlaid on the images and a ground-truth orientation and predicted orientation indicated in a circle. Note that many of the failure cases are due to the flipping errors ( $\approx 180^\circ$ ) and tend to occur at a specific instance whose front and rear look similar (See the last two examples in the row 4.)

#### 4.4. TUD dataset

The TUD dataset consists of 5,228 images of pedestrians with bounding box annotations. Since the original annotations are discrete orientations, we use continuous annotations provided by [15]. In total, there are 4,732 images for training, 290 for validation and 309 for testing. Note that the size of the dataset is more than two times larger than that of EPFL Multi-view Car Dataset and unlike the EPFL dataset, images are captured in the wild. Since most of the training images are gray scale images and thus not adequate to feed into the DCNN, we convert all the grey scale images into color images by a recently proposed colorization technique [45]. The performance of the algorithm is measured by Mean Absolute Error (MeanAE), Accuracy-22.5 and Accuracy-45 as in [15]. Accuracy-22.5 and Accuracy-

Table 1. MeanAE and MedianAE with different values for  $N$  and  $M$  on the EPFL dataset.

$N$	72			8			
$M$	1	5	1	3	5	9	15
MeanAE	13.6	13.7	19.0	13.3	13.2	12.6	12.6
MedianAE	2.7	2.8	10.8	3.0	3.0	3.0	3.1

Table 3. Comparison with the existing works on the EPFL dataset. The performance is measured in Mean Absolute Error (MeanAE) and Median Absolute Error (MedianAE).

Methods	MeanAE	MedianAE	Ground-truth Bounding box?
<b>Ours</b>	<b>9.86</b>	<b>3.14</b>	Yes
Fenzi et al. [7]	13.6	3.3	No
He et al. [16]	15.8	6.2	No
Fenzi and Ostermann [9]	23.28	N/A	Yes
Hara and Chellappa [15]	23.81	N/A	Yes
Zhang et al. [44]	24.00	N/A	Yes
Yang et al. [41]	24.1	3.3	No
Hara and Chellappa [14]	24.24	N/A	Yes
Fenzi et al. [8]	31.27	N/A	Yes
Torki and Elgammal [37]	33.98	11.3	Yes
Teney and Piater [35]	34.7	5.2	No
Redondo-Cabrera et al. [26]	39.8	7	No
Ozuyosal et al. [23]	46.5	N/A	No

45 are defined as the ratio of samples whose predicted orientation is within  $22.5^\circ$  and  $45^\circ$  from the ground truth, respectively. For this dataset, the number of training iterations is 10,000 with 0.00001 as a learning rate.

In Fig. 5, we show the MeanAE obtained by attaching the orientation estimation unit of the approach 3 to different residual blocks. As is the case with the EPFL dataset, the performance varies significantly depending on the residual block used. Furthermore, the use of a proper representation is more critical on this dataset. Interestingly though, as in the EPFL dataset, the 22nd residual block performs well. Following experiments are thus conducted by using the 22nd residual block.

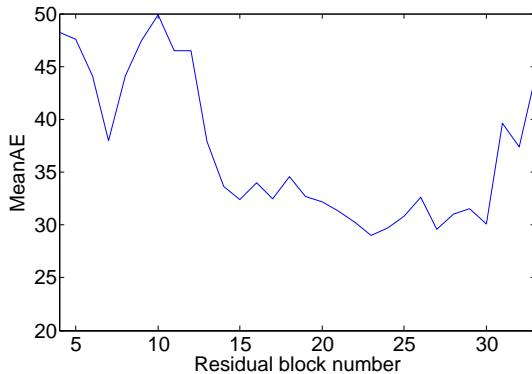


Figure 5. The performance on TUD dataset obtained by using a different residual block to attach the proposed orientation prediction unit.

Table 5. Comparison among the proposed approaches on the TUD dataset.

Approach	MeanAE	Accuracy-22.5	Accuracy-45
1	33.7	46.9	75.7
2	34.6	44.0	70.9
3	30.2	63.1	82.8

In Table 4, we show the effect of  $M$  while keeping  $N = 8$ . As can be seen, in general larger  $M$  produces better results, however, no significant improvement is observed after  $M = 9$ . In order to evaluate the effect of having multiple non-overlapping discretization, we compare  $N = 72, M = 1$  setting, whose number of discrete angles is same as  $N = 8, M = 9$  setting. As can be seen in the table, the effect of having multiple discretization is prominent.

Table 5 shows the performance of all the proposed approaches. For the approach 2, we increase the training iterations to 70,000 as it appears to take more iterations to converge. It is observed again that the approach 3 performs best.

Finally, we train our model by fine-tuning all the layer parameters end-to-end. The result is shown in Table 6 along with the result of prior art. The end-to-end training reduces the MeanAE by 11.9 %. Our final model outperforms the state-of-the-art with 23.3 % reduction in MeanAE. The table contains the performance of human which is significantly better than the algorithms, necessitating further al-

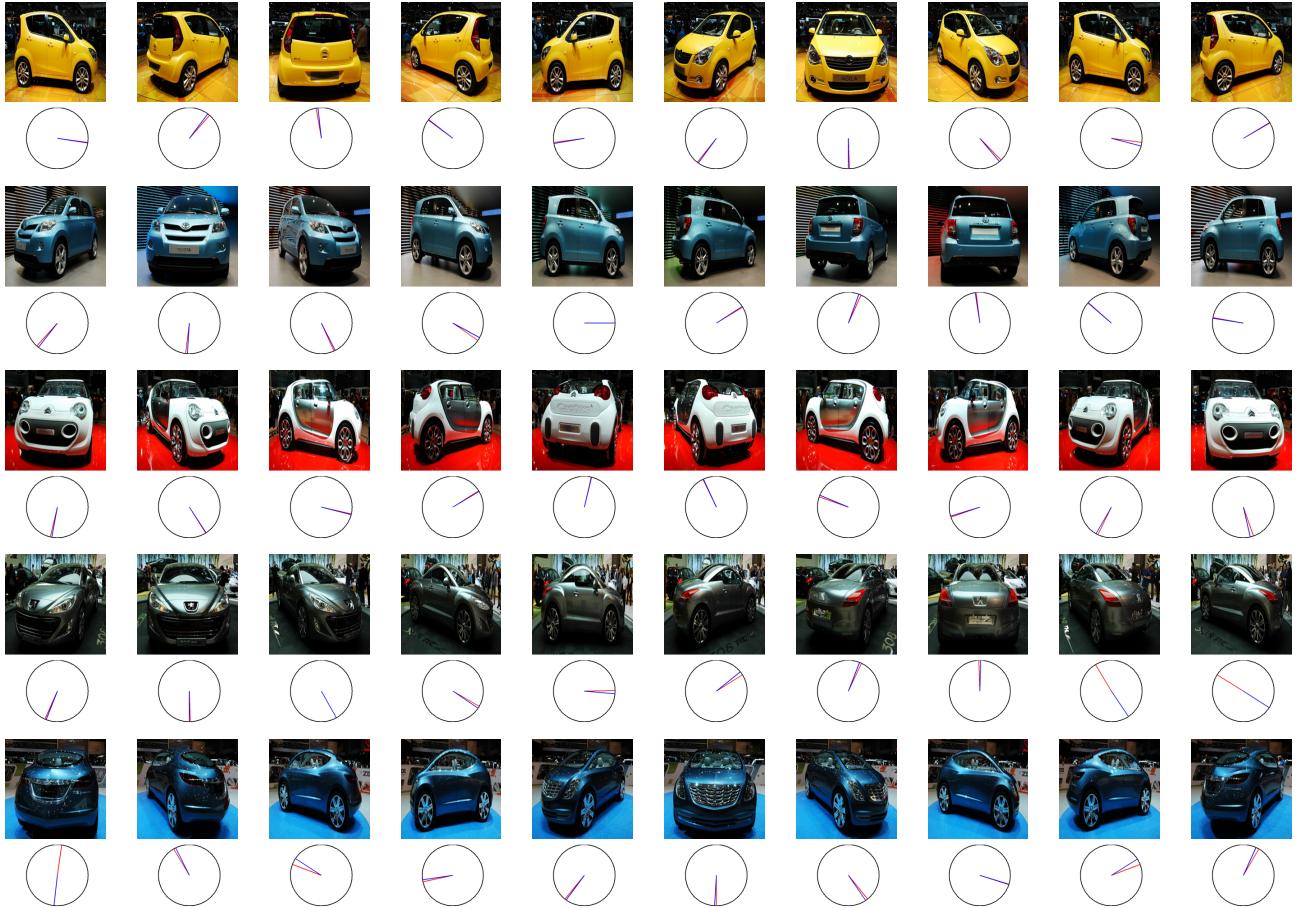


Figure 4. Representative results obtained by the proposed method ( approach 3,  $N = 8, M = 9$ ). A ground-truth orientation (red) and predicted orientation (blue) are indicated in a circle. Each row contains 10 example results from a testing sequence. From left to right, images are selected with 10 frames apart, starting from the first frame.

Table 4. MeanAE, Accuracy-22.5 and Accuracy-45 with different  $M$  on the TUD dataset.

$N$	72		8					
$M$	1	5	1	3	5	9	15	
MeanAE	35.4	33.5	40.0	32.7	31.1	30.2	30.9	
Accuracy-22.5	63.1	62.5	55.0	61.5	61.5	63.1	62.5	
Accuracy-45	79.6	80.6	75.4	79.6	82.2	82.8	82.5	

gorithm development.

Finally, Fig. 6 shows some representative results. The last row includes failure cases.

## 5. Conclusion

This work proposed a new approach for a continuous object orientation estimation task based on the DCNNs. Our best working approach works by first converting the continuous orientation estimation task into a set of non-overlapping discrete orientation estimation tasks and converting the discrete prediction to a continuous orientation by a mean-shift algorithm. Through experiments on a car

orientation estimation task and a pedestrian orientation estimation task, we demonstrate that the DCNN equipped with the proposed orientation prediction unit works significantly better than the state of the approaches, providing another successful DCNN application. Our experiments also indicate that selecting a suitable representation is critical in transferring DCNNs trained on an image classification task to an orientation prediction task.

Table 6. Continuous pedestrian orientation estimation: Mean Absolute Error in degree, Accuracy- $22.5^\circ$  and Accuracy- $45^\circ$  are shown.

Method	MeanAE ( $^\circ$ )	Accuracy- $22.5^\circ$	Accuracy- $45^\circ$
<b>Ours</b>	26.6	70.6	86.1
Hara and Chellappa [15]	34.7	68.6	78.0
Human	9.1	90.7	99.3 [15]

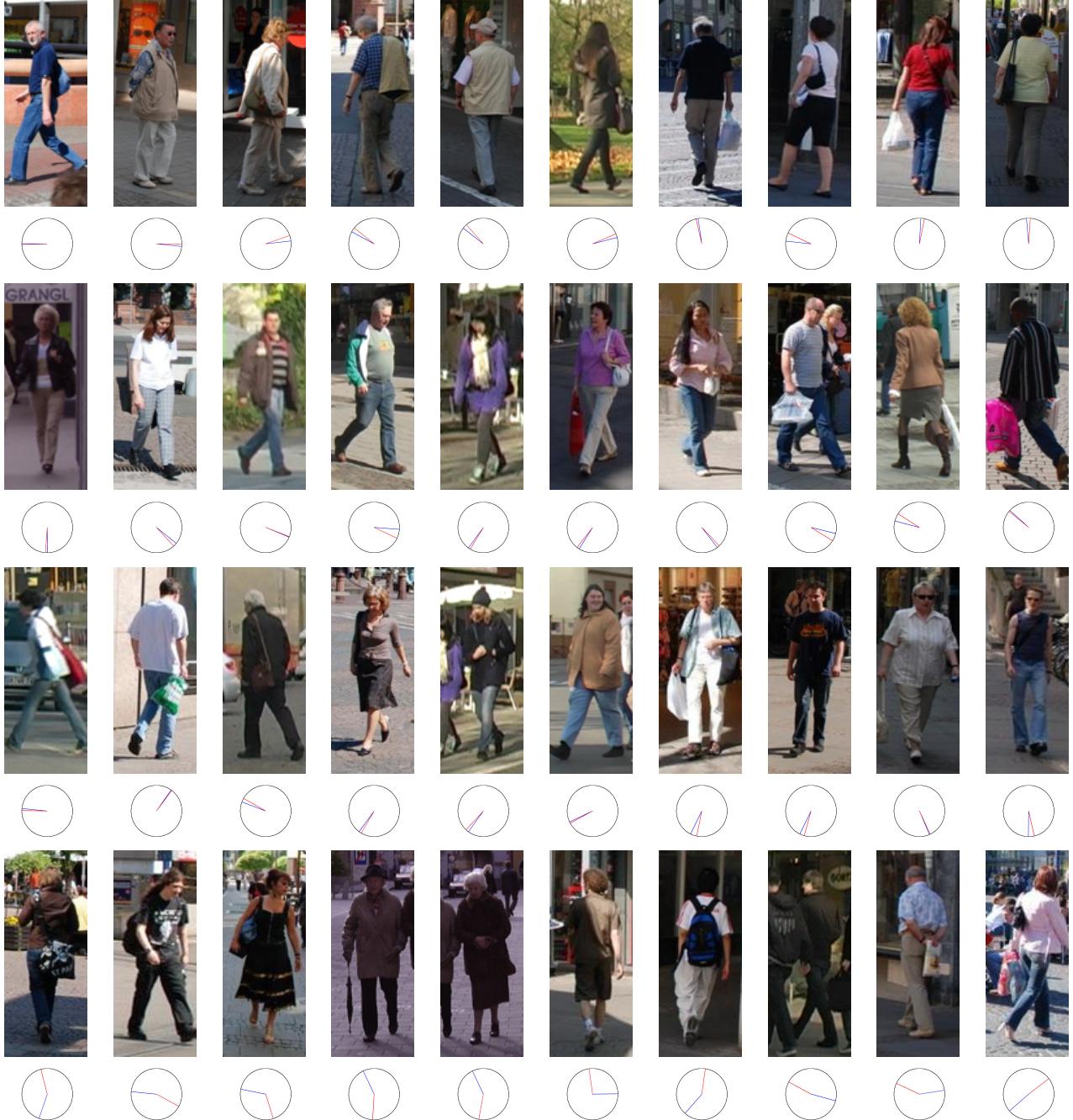


Figure 6. Representative results obtained by the proposed method ( approach 3,  $N = 8, M = 9$ ). A ground-truth orientation (red) and predicted orientation (blue) are indicated in a circle. The first three rows show successful cases while the last row shows failure cases.

## References

- [1] M. Andriluka, S. Roth, and B. Schiele. Monocular 3D Pose Estimation and Tracking by Detection. In *CVPR*, 2010. [2](#), [4](#)
- [2] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2015-Octob:36–45, 2015. [1](#)
- [3] A. Bakry and A. Elgammal. Untangling object-view manifold for multiview recognition and pose estimation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8692 LNCS(PART 4):434–449, 2014. [2](#)
- [4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *Iclr*, pages 1–14, 2014. [1](#)
- [5] X. Chen and A. Yuille. Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations. *NIPS*, pages 1–10, 2014. [1](#)
- [6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, sep 2010. [5](#)
- [7] M. Fenzi, L. Leal-taixé, J. Ostermann, and T. Tuytelaars. Continuous Pose Estimation with a Spatial Ensemble of Fisher Regressors. In *ICCV*, 2015. [2](#), [5](#), [6](#)
- [8] M. Fenzi, L. Leal-taixé, B. Rosenhahn, and J. Ostermann. Class Generative Models based on Feature Regression for Pose Estimation of Object Categories. In *CVPR*, 2013. [2](#), [6](#)
- [9] M. Fenzi and J. Ostermann. Embedding Geometry in Generative Models for Pose Estimation of Object Categories. *BMVC*, pages 1–11, 2014. [2](#), [6](#)
- [10] A. Ghodrati, M. Pedersoli, and T. Tuytelaars. Is 2D Information Enough For Viewpoint Estimation? *BMVC*, 2014. [2](#)
- [11] R. Girshick. Fast R-CNN. *Arxiv*, pages 1440–1448, 2015. [1](#)
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014. [1](#)
- [13] L. Gui and L.-p. Morency. Learning and Transferring Deep ConvNet Representations with Group-Sparse Factorization. *ICCVW*, pages 1–6, 2015. [1](#)
- [14] K. Hara and R. Chellappa. Growing Regression Forests by Classification: Applications to Object Pose Estimation. In *ECCV*, 2014. [2](#), [6](#)
- [15] K. Hara and R. Chellappa. Growing Regression Tree Forests by Classification for Continuous Object Pose Estimation. *IJCV*, 2016. [2](#), [3](#), [5](#), [6](#), [8](#)
- [16] K. He, L. Sigal, and S. Sclaroff. Parameterizing Object Detectors in the Continuous Pose Space. In *ECCV*, 2014. [2](#), [5](#), [6](#)
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *Arxiv.Org*, 7(3):171–180, 2015. [1](#), [4](#)
- [18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv*, jun 2014. [4](#)
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012. [1](#)
- [20] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2014. [1](#)
- [21] F. Massa, R. Marlet, and M. Aubry. Crafting a multi-task CNN for viewpoint estimation. pages 1–12, 2016. [2](#)
- [22] H. Noh, S. Hong, and B. Han. Learning Deconvolution Network for Semantic Segmentation. *ICCV*, 1:1520–1528, 2015. [1](#)
- [23] M. Ozuysal, V. Lepetit, and P. Fua. Pose Estimation for Category Specific Multiview Object Localization. In *CVPR*, 2009. [2](#), [4](#), [5](#), [6](#)
- [24] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D geometry to deformable part models. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (June):3362–3369, 2012. [2](#)
- [25] P. O. Pinheiro and R. C. Com. Recurrent Convolutional Neural Networks for Scene Labeling. *ICML*, 32, 2014. [1](#)
- [26] C. Redondo-cabrera, R. López-Sastre, and T. Tuytelaars. All together now : Simultaneous Object Detection and Continuous Pose Estimation using a Hough Forest with Probabilistic Locally Enhanced Voting. In *BMVC*, 2014. [5](#), [6](#)
- [27] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *ArXiv 2015*, pages 1–10, 2015. [1](#)
- [28] O. Russakovsky, J. Deng, and H. Su. ImageNet Large Scale Visual Recognition Challenge. *arXiv preprint arXiv: ...*, 2014. [1](#), [4](#)

- [29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat : Integrated Recognition , Localization and Detection using Convolutional Networks. *ICLR*, 2014. 1
- [30] A. Sharif, R. Hosseini, A. Josephine, S. Stefan, and K. T. H. Royal. CNN Features off-the-shelf: an Astounding Baseline for Recognition. *Cvprw*, 2014. 1
- [31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Iclr*, pages 1–14, 2015. 1
- [32] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. *Proceedings of the IEEE International Conference on Computer Vision*, 11-18-Dece:2686–2694, 2015. 2
- [33] C. Szegedy, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 1
- [34] Y. Taigman, M. A. Ranzato, T. Aviv, and M. Park. DeepFace : Closing the Gap to Human-Level Performance in Face Verification. *CVPR*, 2014. 1
- [35] D. Teney and J. Piater. Multiview feature distributions for object detection and continuous pose estimation. *Computer Vision and Image Understanding*, 125:265–282, 2014. 2, 5, 6
- [36] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. *Nips 2014*, pages 1–9, 2014. 1
- [37] M. Torki and A. Elgammal. Regression from Local Features for Viewpoint and Pose Estimation. In *ICCV*. Ieee, 2011. 2, 6
- [38] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *Computer Vision and Pattern . . .*, 2014. 1
- [39] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 1510–1519, 2015. 2
- [40] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond PASCAL: A benchmark for 3D object detection in the wild. *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, mar 2014. 2
- [41] L. Yang, J. Liu, and X. Tang. Object Detection and Viewpoint Estimation with Auto-masking Neural Network. *ECCV*, 8691:441–455, 2014. 2, 5, 6
- [42] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *NIPS*, pages 1–9, 2014. 1
- [43] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS(PART 1):818–833, 2014. 1
- [44] H. Zhang, T. El-gaaly, A. Elgammal, and Z. Jiang. Joint Object and Pose Recognition using Homeomorphic Manifold Analysis. In *AAAI*, 2013. 6
- [45] R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. pages 1–25, 2016. 5