A Comparative Analysis and Study of Multiview CNN Models for Joint Object Categorization and Pose Estimation
Supplementary Materials

Mohamed Elhoseiny*, **Tarek El-Gaaly*** M.ELHOSEINY@CS.RUTGERS.EDU, TGAALY@GMAIL.COMU

Amr Bakry*, **Ahmed Elgammal** AMRBAKRY@CS.RUTGERS.EDU, ELGAMMAL@CS.RUTGERS.EDU

Rutgers University, Computer Science Department, Piscataway, NJ, USA, 08854, * Equal contribution

1. k-NN Results

1.1. Pascal3D

KNN figures for categorization on Pascal3D dataset.

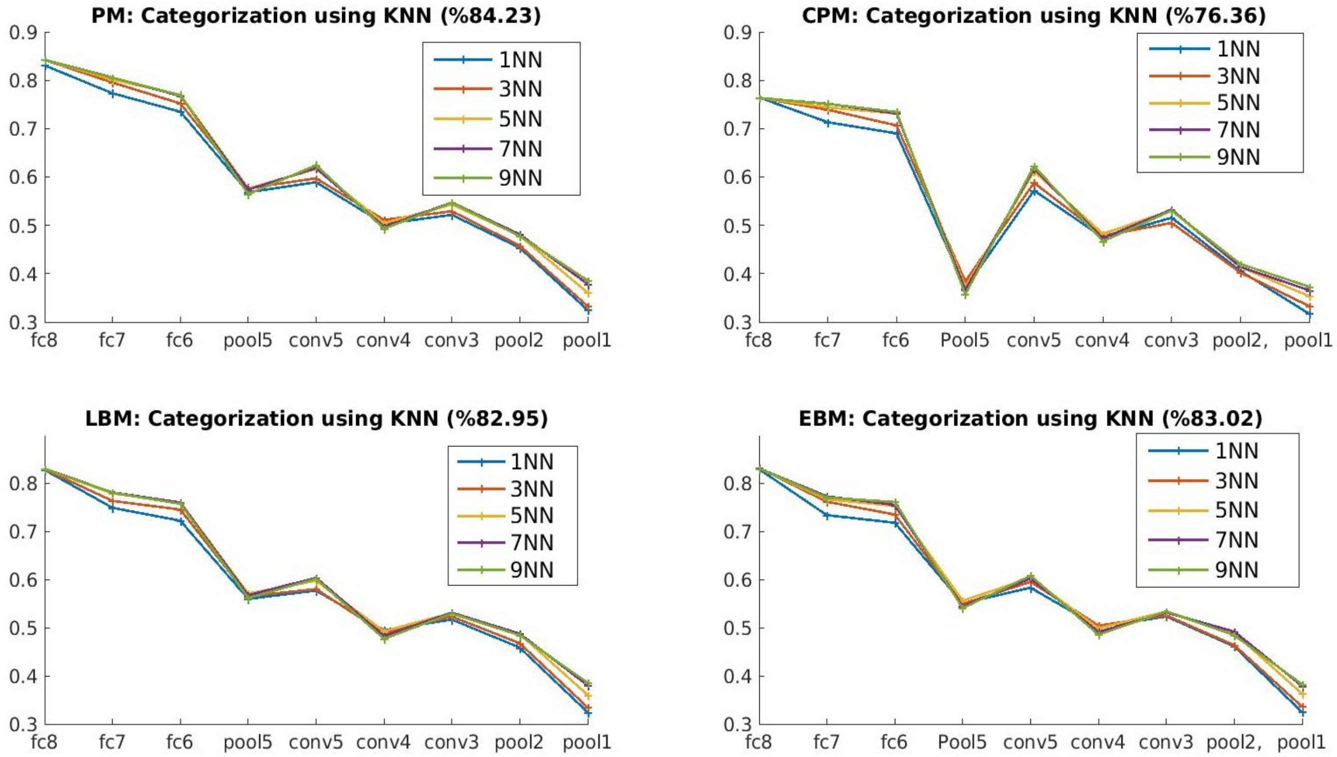

Figure 1. Comparison of the categorization at each layer of the CNN using k-NN with varying $k = \{1, 3, 5, 7, 9\}$ from top to bottom. This experiment was conducted on the PASCAL3D dataset categorization

1.2. RGBD Dataset

It is clear how CPM is very unstable for dense poses that exist in RGBD dataset.

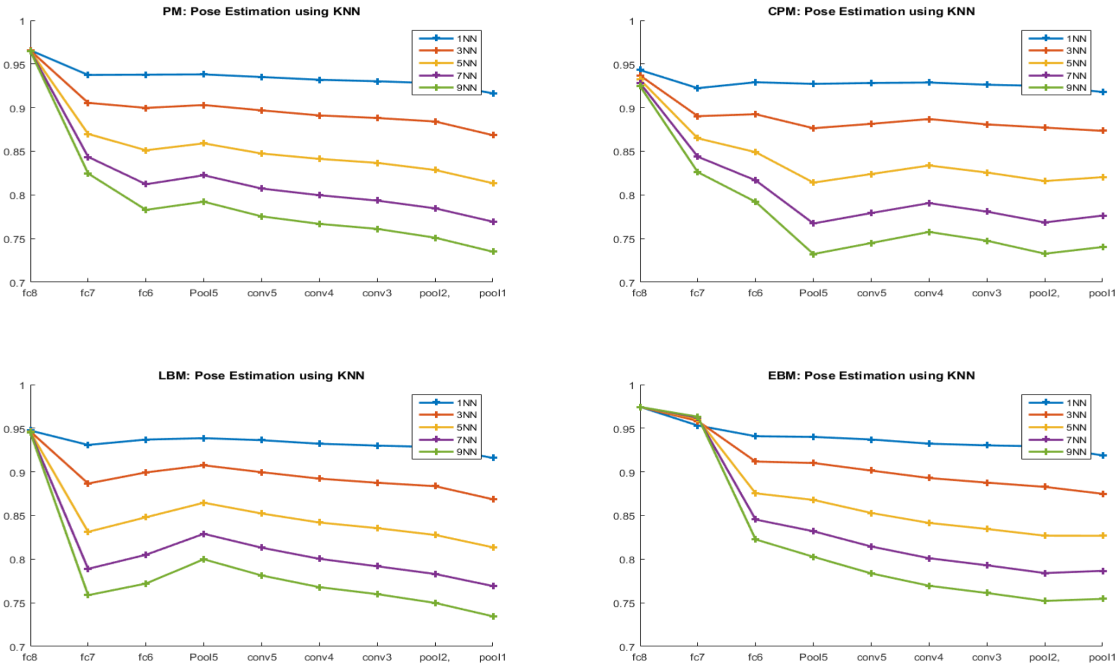

Figure 2. Comparison of the pose estimation at each layer of the CNN using k-NN with varying $k = \{1, 3, 5, 7, 9\}$ from top to bottom. This experiment was conducted on the RGBD dataset categorization (training points)

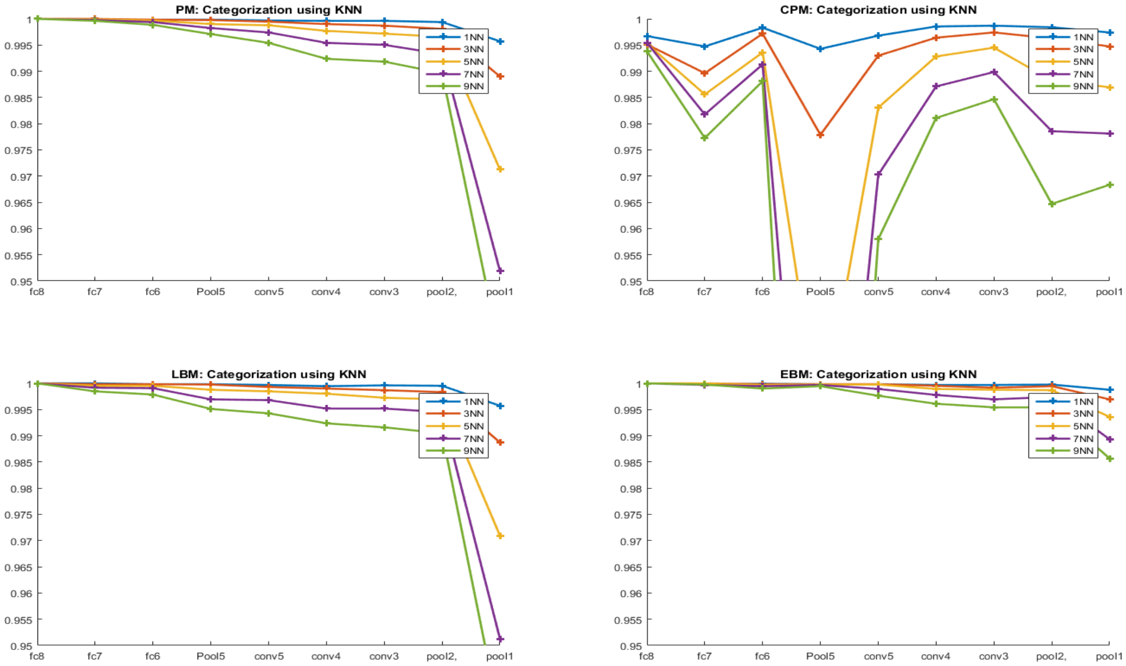

Figure 3. Comparison of the categorization at each layer of the CNN using k-NN with varying $k = \{1, 3, 5, 7, 9\}$ from top to bottom. This experiment was conducted on the RGBD dataset categorization (training points)

2. Local Pose Measurement Analysis on RGBD Dataset

We applied three local measurement analysis proposed in (Bakry et al., 2015) to analyze features against dense poses. For more details about the description of these measurements, please refer to (Bakry et al., 2015). The main property that these measurements quantified is how these representations align with the the circle manifold that represent the pose of the categories.

All the figures shows that EBM achieves the best behavior in untangling both the categorization and the pose branches. It is clear that CPM behaves the worst for pose estimation as we argued in the paper for several reasons.

1. **Z-EffectiveSV 90 (The lower the better)**
2. **TPS-RCond-CF-poly (The higher the better)**”
3. **Nuclear Norm (The higher the better)**
4. **KPLS-Kernel Regression Error (The lower the better)**

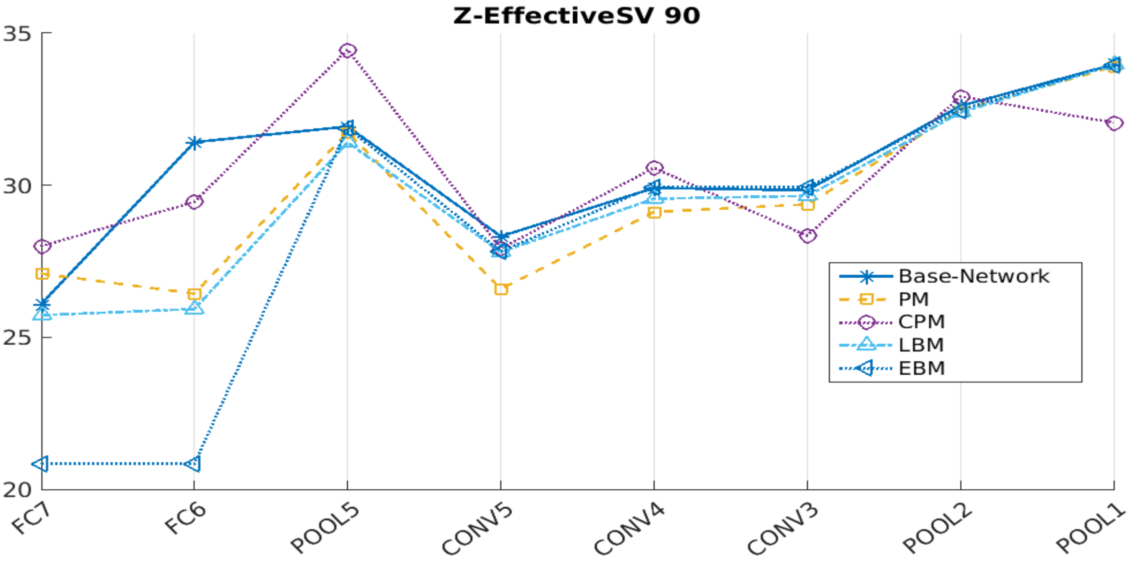

Figure 4. Comparison of the pose estimation at each layer of the CNN using “Effectibe SV 90%”

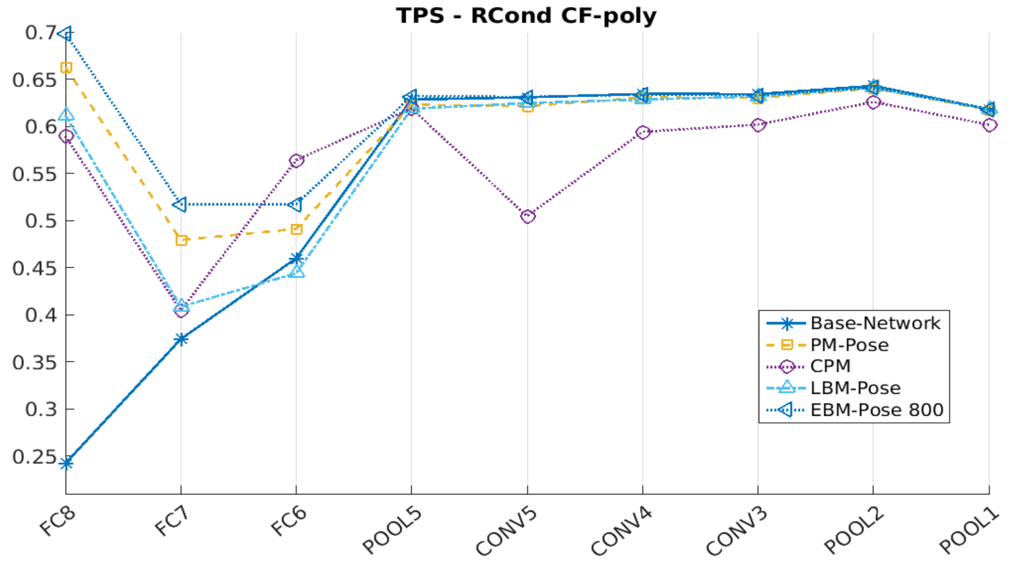

Figure 5. Comparison of the pose estimation at each layer of the CNN using “TPS-RCond (polynomal)[” measurement

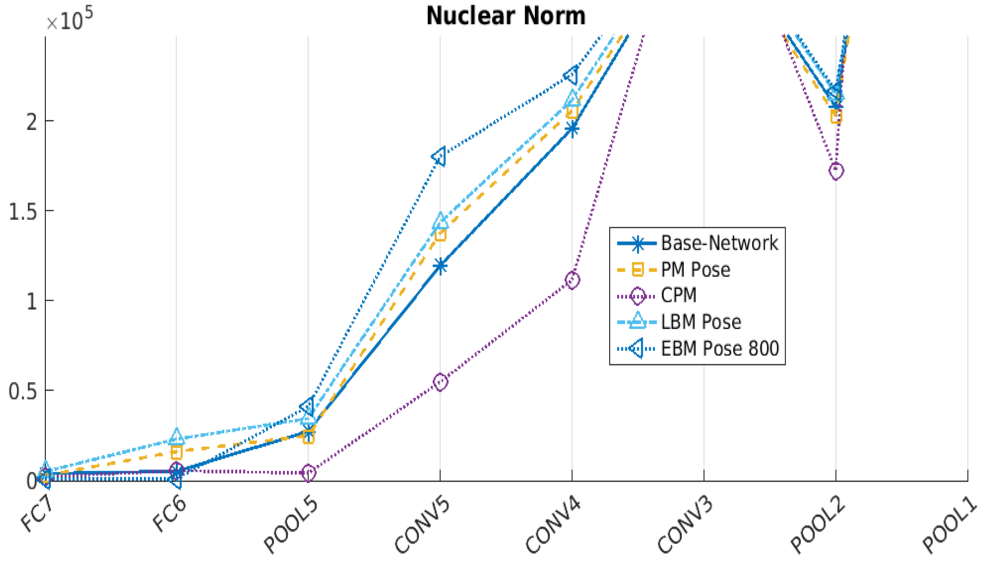

Figure 6. Comparison of the pose estimation at each layer of the CNN using “Nuclear Norm” measurement (FC8 to Pool5)

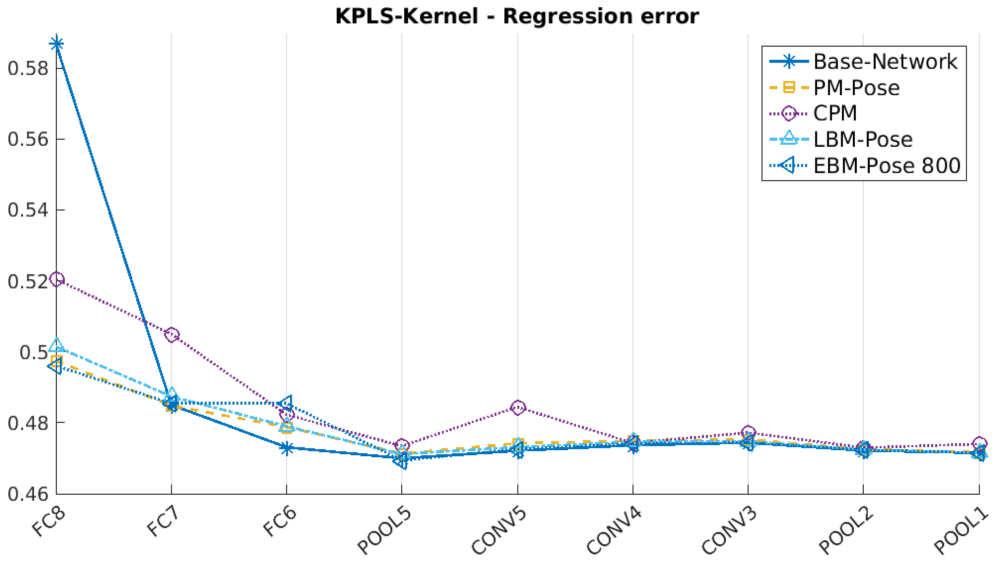

Figure 7. Comparison of the pose estimation at each layer of the CNN using KPLS Kernel Regression Error measurement

3. Computational Analysis and Convergence(More Details)

The following figures show the loss/validation curves for the trained CNNs. The loss is shown per the batch being processed at each iteration(one training batch/iteration). The interesting behavior we notice in all the networks is that the categorization part converges very quickly, while the pose part takes sometime to converge. This is since, in all networks, the layers were initialized with the ImageNet categorization CNN. For the pose part, initialization for a categorization network might not be helpful especially for the top layers (e.g. FC6,FC7, and FC8), since they were trained for a different purpose that might be conflicting. Furthermore, training on a joint loss as in EBM and LBM positively affects the convergence as can be seen in figures 11 and 12. It is not hard to see that the Early Branching reduced the validation pose error significantly faster compared to the remaining models despite having much more parameters than many of the other models. Refer to section 4 for the number of parameters.

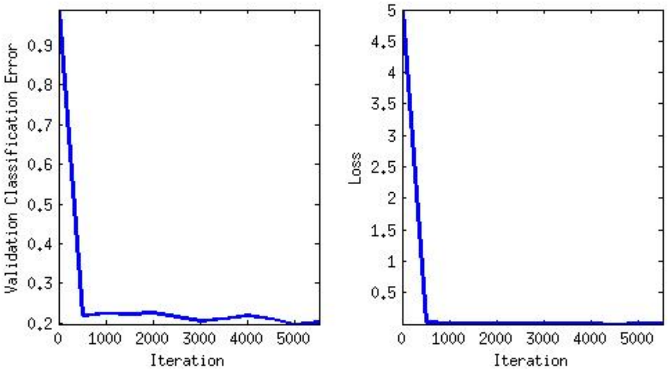

Figure 8. PM Category CNN Training

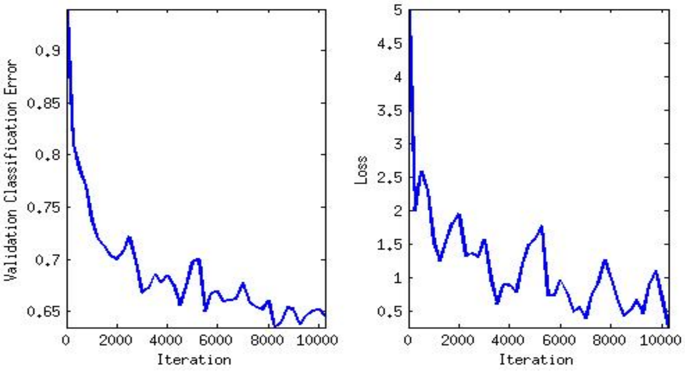

Figure 9. PM Pose CNN Training

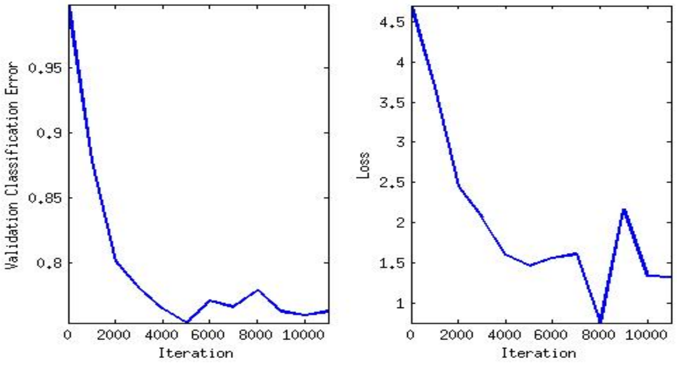

Figure 10. CPM Training Error (this is the error of both classifying the correct category in the correct pose bin)

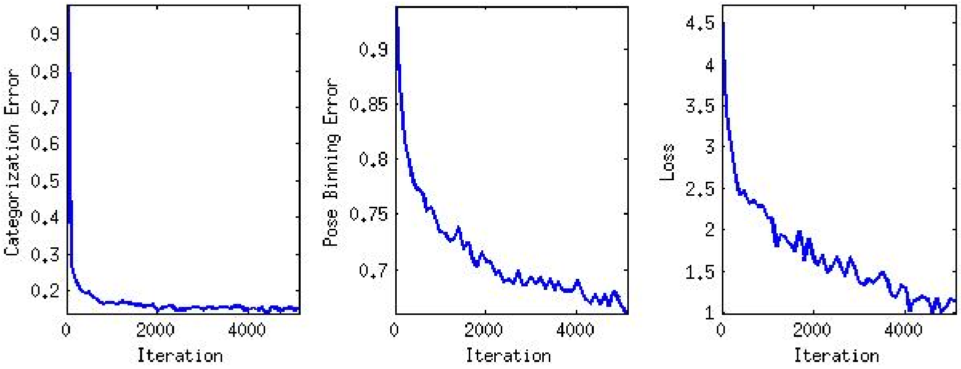

Figure 11. LBM Training (Categorization Error on the left, Pose Binning Error in the middle, Loss on the right)

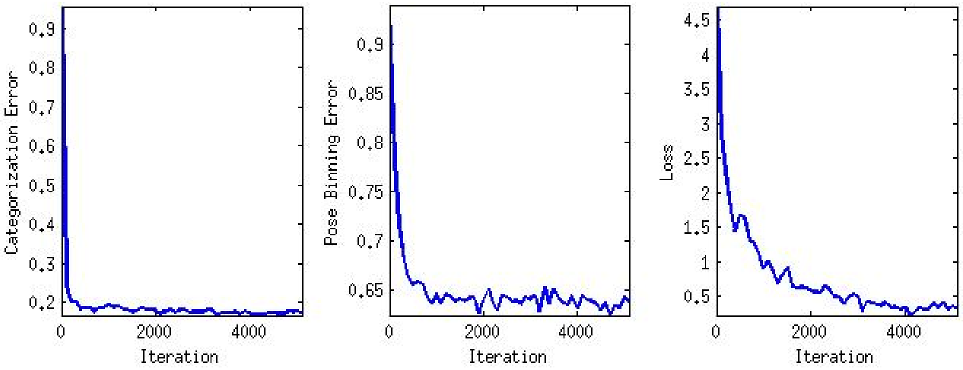

Figure 12. EBM Training (Categorization Error on the left, Pose Binning Error in the middle, Loss on the right)

4. Error Metrics

The two metrics < 22.5 and < 45 used to evaluate the performance of pose estimation are the percentages of test samples that satisfy $AE < 22.5°$ and $AE < 45°$, respectively, where the Absolute Error (AE) is $AE = |EstimatedAngle - GroundTruth|$). The AAAI pose accuracy (used extensively in the previous work we compare with) is equal to $1 - [min(|\theta_i - \theta_j|, 2\pi - |\theta_i - \theta_j|)/\pi]$.

5. Training Parameters

All models are trained by back propagation with Stochastic gradient descent. Refer to the supplementary (Sec 5) for parameter settings, *e.g.* learning rate, decay, *etc*. At training time, we randomly sample 227x227 patches from the downscaled 256x256 images. At test time the center 227x227 patches are taken.

The base learning rate is assigned 0.5×10^{-3}. For fine-tuning, the learning rate of the randomly initialized parameters (*e.g.* FC8 parameters in PM) are assigned to be ten times higher than the learning rate of the parameters initialized from the pretrained CNN (*e.g.* Conv1 to Pool5 in all the models). The decay of the learning rate γ is 0.1. While training our CNNs, we drop the learning rate by a factor of γ every 5000 iterations. The momentum and the weight decay were assigned to 0.9 and 0.0001 respectively. Training images are randomly shuffled before feeding the CNN for training. The training batch size was 100 images.

6. Initialization and Models' Parameters

The ImageNet CNN used in our paper (AlexNet) [16] has ∼60 million parameters. In this section, we present how all the models were initialized in our experiments. Then, we analyze the number of parameters in the model for each of RGBD and Pascal3D datasets.

6.1. Initialization

6.1.1. EBM Model

In EBM, we initialize all the convolutional layers by the convolution layer parameters of AlexNet. We initialize FC6 and FC7 of the category branch by the parameters of AlexNet model. The remaining layers were initialized randomly (*i.e.*FC6, FC7, and FC8 of the pose branch subnetwork, and FC8 of the category Branch subnetwork).

6.1.2. LBM Model

For LBM, we initialize all the layers by the pretrained AlexNet model for the convolution layers, FC6, and FC7. FC8 weights are initialized randomly.

6.1.3. CPM Model

We initialize all the layers by the pretrained AlexNet model for the convolution layers, FC6, and FC7. We initialized FC8 parameters randomly.

6.1.4. PM Model

Since there are two separate models, one for category and one for Pose. We initialize all the layers by the pretrained AlexNet model for the convolution layers, FC6, and FC7. We initialized FC8 parameters randomly.

6.2. Number of Model Parameters for RGBD Dataset

6.2.1. EBM Model

EBM Model has 111,654,944 parameters. These are as follows starting from the input layer (number of filters x filter width x filter height x number of channels): 96x11x11x3 + 256x5x5x48 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + Pool5-FC6 (pose) 9216x4096, FC6(pose)-FC7(pose) 4096x4096, FC7(pose)-FC8(pose) 4096x16 Pool5-FC6(category) 9216x4096, FC6(category)-FC7(category) 4096x4096, FC7(category)-FC8(category) 4096x51.

6.2.2. LBM Model

LBM has 57,133,088 params = convolution-layers' parameters (96x11x11x3 + 256x5x5x48 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + 9216x4096 + 4096x4096 + (fc8-cat) 4096x51 + (fc8-pose) 4096x16. These are organized as (number of filters x filter width x filter height x number of channels) and starting from the input layer.

6.2.3. CPM Model

CPM has 60,200,992 params = convolution-layers' parameters (96x11x11x3 + 256x5x5x48 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + 9216x4096 + 4096x4096 + (fc8-cat and pose) 4096x51x16. These are organized as (number of filters x filter width x filter height x number of channels) and starting from the input layer.

6.2.4. PM Model

PM has 113,991,744 parameters (56,924,192 for pose and 57,067,552 for category). These are shown below as (number of filters x filter width x filter height x number of channels) and starting from the input layer.

PM Model Pose Parameters: The 56,924,192 pose parameters comes from convolution-layers' parameters (96x11x11x3+ 256x5x5x48 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + fully connected layers' parameters (9216x4096 + 4096x4096 + 4096x16).

PM Model Category Parameters: The 57,067,552 category parameters comes from convolution-layers' parameters (96x11x11x3+ 256x5x548 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + fully connected layers' parameters (9216x4096, + 4096x4096 + 4096x51).

6.3. Number of Model Parameters for Pascal3D Dataset

6.3.1. EBM Model

EBM Model 111,495,200 params = convolution layers' parameters (96x11x11x3 + 256x5x5x48 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + Pool5-FC6 (pose) 9216x4096, FC6(pose)-FC7(pose) 4096x4096, FC7(pose)-FC8(pose) 4096x16 Pool5-FC6(category) 9216x4096, FC6(category)-FC7(category) 4096x4096, FC7(pose)-FC8(pos) 4096x16. These are organized as (number of filters x filter width x filter height x number of channels) and starting from the input layer.

6.3.2. LBM Model

LBM has 56,969,248 params = convolution layers' parameters (96x11x11x3 + 256x5x5x48 + 384x3x3x256 + 384x3x3x192+ 256x3x3x192)+ 9216x4096 + 4096x4096 + (fc8-cat) 4096x11 + (fc8-pose) 4096x16. These are organized as (number of filters x filter width x filter height x number of channels) and starting from the input layer.

6.3.3. CPM Model

CPM has 57,579,552 params = convolution layers' parameters (96x11x11x3 + 256x5x5x48 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + 9216x4096 + 4096x4096 + (fc8-cat and pose) 4096x11x16. These are organized as (number of filters x filter width x filter height x number of channels) and starting from the input layer.

6.3.4. PM Model

PM has 113,827,904 parameters (56,924,192 for pose and 56,903,712 for category. These are shown below as (number of filters x filter width x filter height x number of channels) and starting from the input layer.

PM Model Pose Parameters: The 56,924,192 pose parameters comes from convolution layers' parameters (96x11x11x3+ 256x5x5x48 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + fully connected layers' parameters (9216x4096, + 4096x4096 + 4096x16).

PM Model Category Parameters: The 56,903,712 category parameters comes from convolution layers' parameters (96x11x11x3 + 256x5x5x48 + 384x3x3x256 + 384x3x3x192 + 256x3x3x192) + fully connected layers' parameters (9216x4096, + 4096x4096 + 4096x11).

7. Effect of Loss Function Weights on EBM model

We found that changing the weights for EBM slightly affected the performance; see table 1. Our intuition behind this behavior is that EBM splits into separate parameters starting from Pool5, which makes each of the pose and the category have some independent parameters (FC6,FC7,FC8) in addition to the shared parameters (Conv1 to Pool5).

Table 1. Effect of λ_1 and λ_2 for EBM

Parameters	Categorization %	Pose %
$\lambda_1 = 1$ $\lambda_2 = 1$	**89.94**	**82.00**
$\lambda_1 = 1$ $\lambda_2 = 2$	89.39	81.80
$\lambda_1 = 2$ $\lambda_2 = 1$	89.25	81.89

Since $\lambda_1 = 1$, $\lambda_2 = 1$ is slightly better than others, we performed all of our Model 5 experiments in the paper with this setting.

References

Bakry, Amr, Elhoseiny, Mohamed, El-Gaaly, Tarek, and Elgammal, Ahmed. Digging deep into the layers of cnns: In search of how cnns achieve view invariance. *arXiv preprint arXiv:1508.01983*, 2015.