Active Visual Exploration Based on Attention-Map Entropy – Supplementary Materials

Adam Pardyl[1,2,3], **Grzegorz Rypeść**[1,4], **Grzegorz Kurzejamski**[1], **Bartosz Zieliński**[1,2,6], **Tomasz Trzciński**[1,4,5]
[1]IDEAS NCBR
[2]Jagiellonian University, Faculty of Mathematics and Computer Science
[3]Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4]Warsaw University of Technology
[5]Tooploox
[6]Ardigen
{fistname.lastname}@ideas-ncbr.pl

In this Supplementary Materials, we present additional information regarding performed experiments, results, and visualizations for our Attention-Map Entropy active visual exploration method.

1 Details on models used for experiments

We implemented the Attention-Map Entropy algorithm using Python (version 3.9) and PyTorch (version 1.13) library. In this section we explain the structure, hyperparameters, and training regime of AME which we use in Section 4 of the paper.

1.1 Structure

For each task (reconstruction, segmentation, and classification), we use the same structure of the encoder. It consists of 24 standard transformer blocks, which are initialized with weights of the MAE ViT-Large model that was pre-trained on the ImageNet 1K dataset. Each encoder block consists of 16 self-attention heads. The input to the encoder are patches of size 16x16 which are later transformed into embedding vectors of size 1024. The feed forward network in each block consists of 4096 hidden neurons. As the number of input patches may vary between images in batch, we use padding tokens to even out the sequence length. We explicitly assign zero attention scores to padding tokens to mask them from computation, which is a standard approach used in NLP transformers. The difference in the architecture between tasks lies in the decoder part of the architecture.

For reconstruction, we use a decoder consisting of 8 transformer blocks followed by a linear head, initialized with MAE weights. The size of embedding in the decoder is 512, each block uses 16 self-attention heads. The feed forward network in each block has a size of 2048.

Similarly, for segmentation, we utilize 8 transformer blocks followed by the linear head as the decoder; however, we randomly initialize all parameters.

For classification, we attach an auxiliary classification head to the encoder output. The head takes as input the *cls* token embedding of size 1024, as returned by the encoder. It consists of linear layers, one for the *train-all* scenario and two for the *head-only* regime, as described in the Experimental Setup section of the main paper. For two linear layer classification head we set the latent dimension size to 256, and use a GELU activation between linear layer.

1.2 Training

During training we utilize AdamW optimizer and set the initial learning rate to 0.0001. We use the half-cycle cosine learning rate scheduling and set the number of epochs to 100. We set the weight decay factor to 0.0001. We choose the best model based on values of metrics on validation dataset.

2 Additional visualizations

In this section, we provide additional visualizations for both the reconstruction and segmentation tasks.

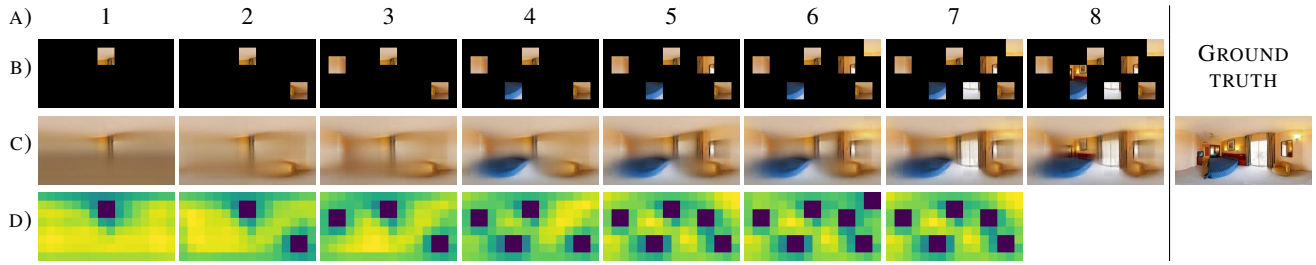

Figure 1: **Glimpse-based reconstruction step-by-step on SUN360:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

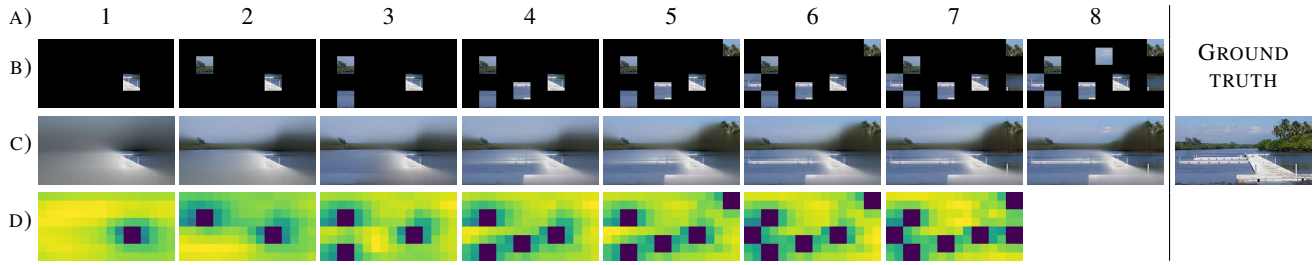

Figure 2: **Glimpse-based reconstruction step-by-step on ADE20K:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

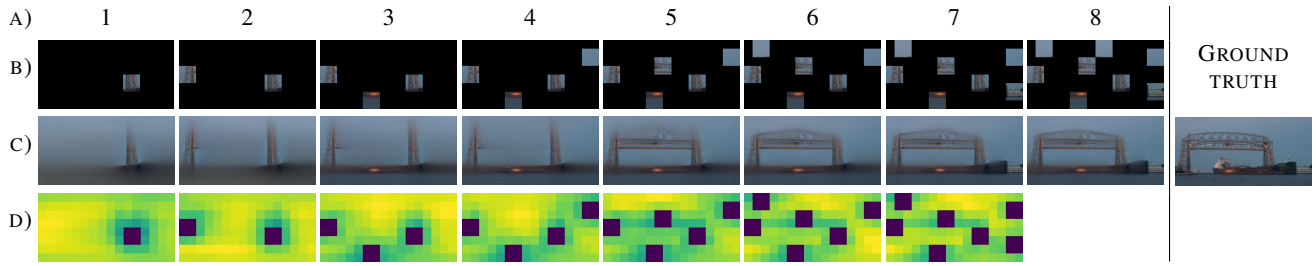

Figure 3: **Glimpse-based reconstruction step-by-step on ADE20K:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

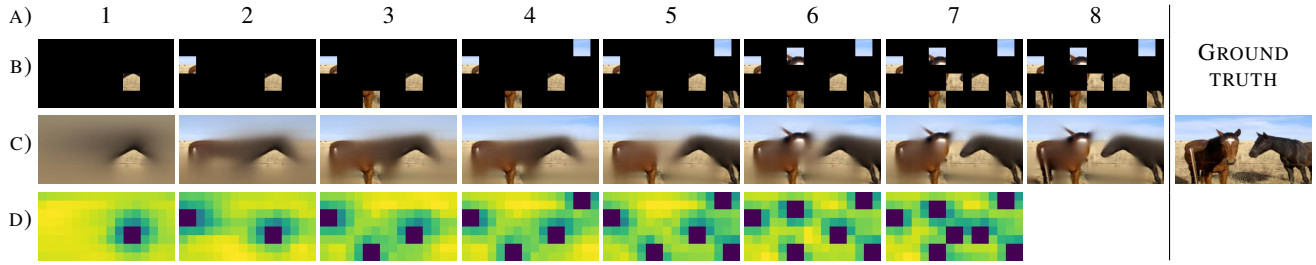

Figure 4: **Glimpse-based reconstruction step-by-step on MS COCO:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

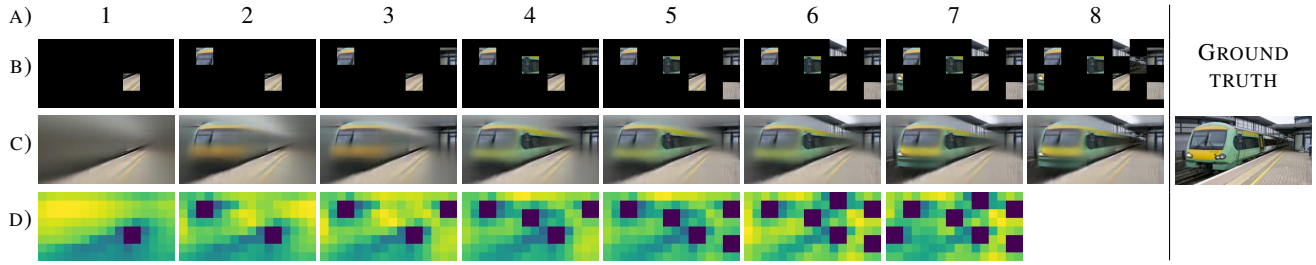

Figure 5: **Glimpse-based reconstruction step-by-step on MS COCO:** The figure shows a glimpse selection process based on AME for 8×32^2 glimpses for a sample 256×128 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

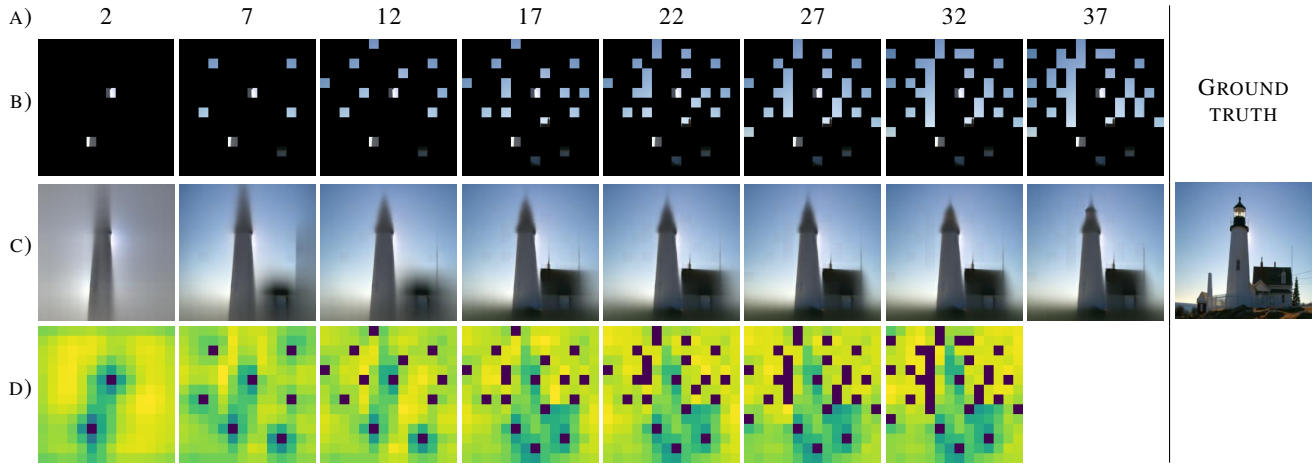

Figure 6: **Glimpse-based reconstruction step-by-step on ADE20K:** The figure shows a glimpse selection process based on AME for 37×16^2 glimpses for a sample 224×224 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

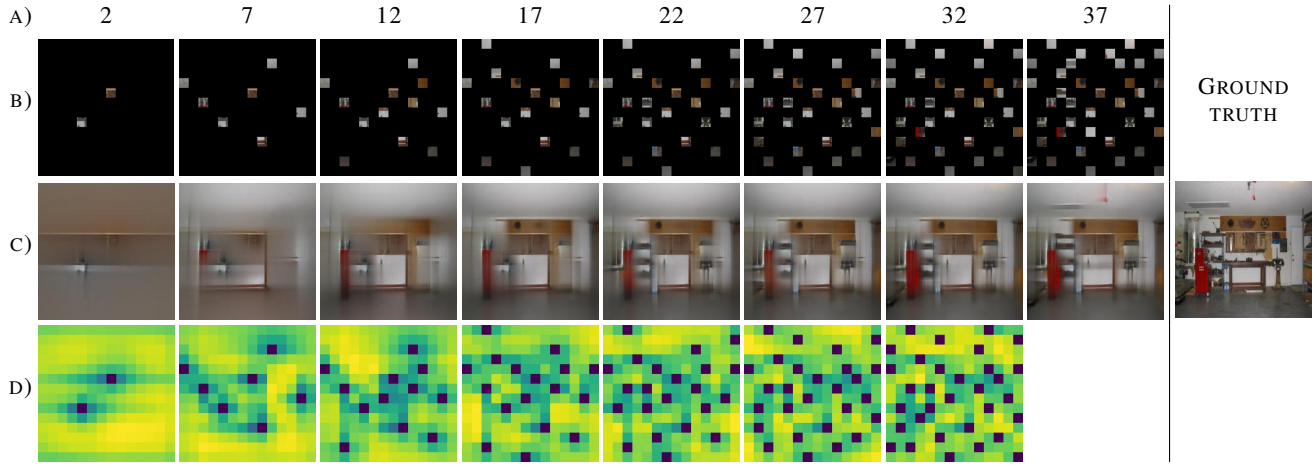

Figure 7: **Glimpse-based reconstruction step-by-step on ADE20K:** The figure shows a glimpse selection process based on AME for 37×16^2 glimpses for a sample 224×224 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

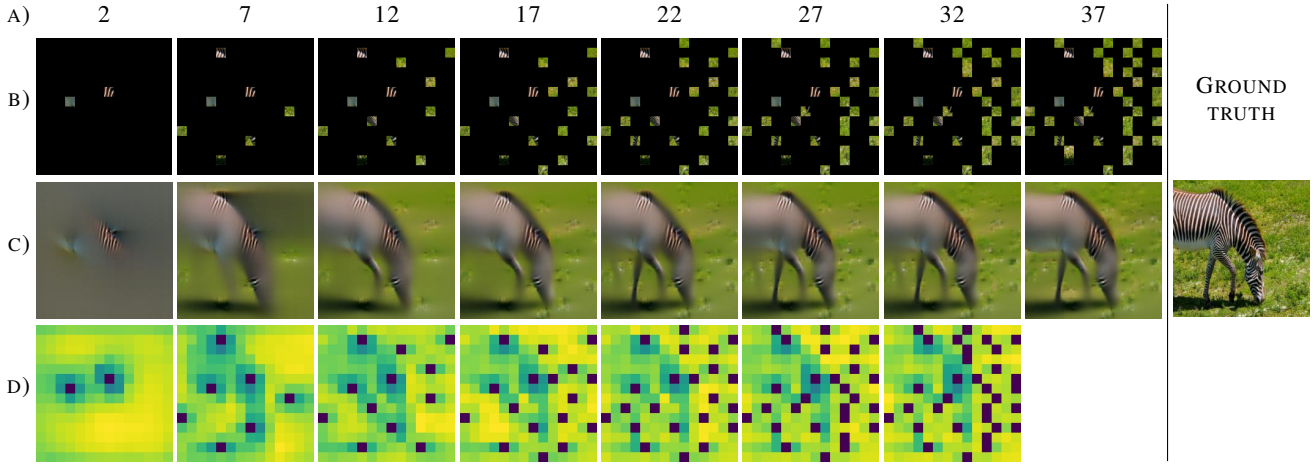

Figure 8: **Glimpse-based reconstruction step-by-step on MS COCO:** The figure shows a glimpse selection process based on AME for 37×16^2 glimpses for a sample 224×224 image. The rows correspond to A) step number, B) model input (glimpses), C) model prediction given, D) decoder attention entropy (known areas are explicitly set to zero).

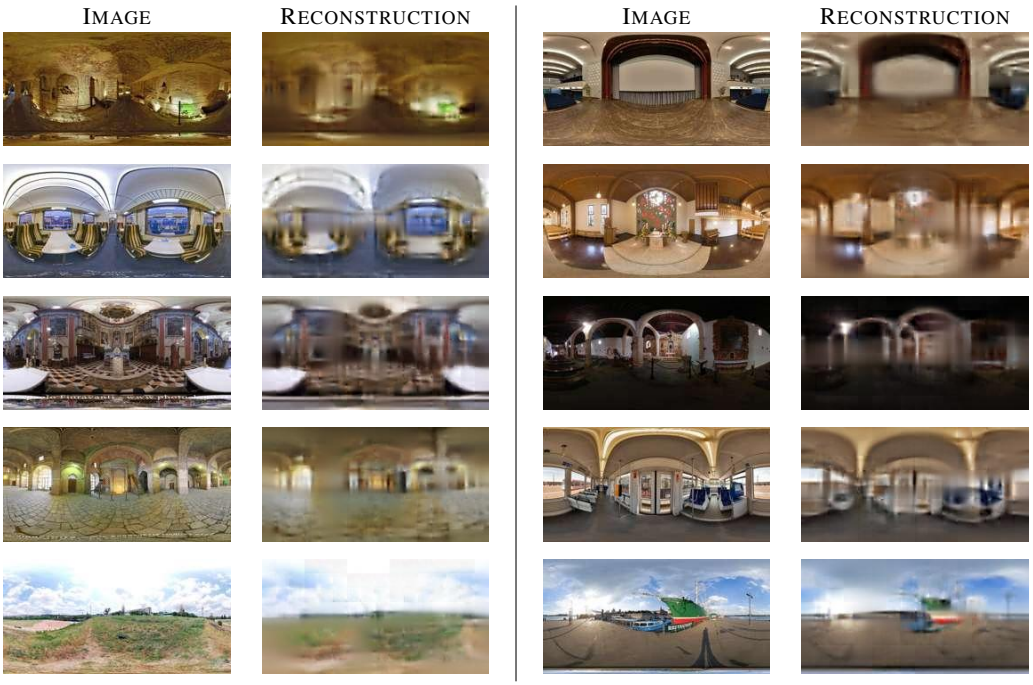

Figure 9: **Image reconstruction on SUN360:** Figure shows the qualitative results of the image reconstruction task. Image size is 256×128 and 8×48^2 retinal glimpses are used.

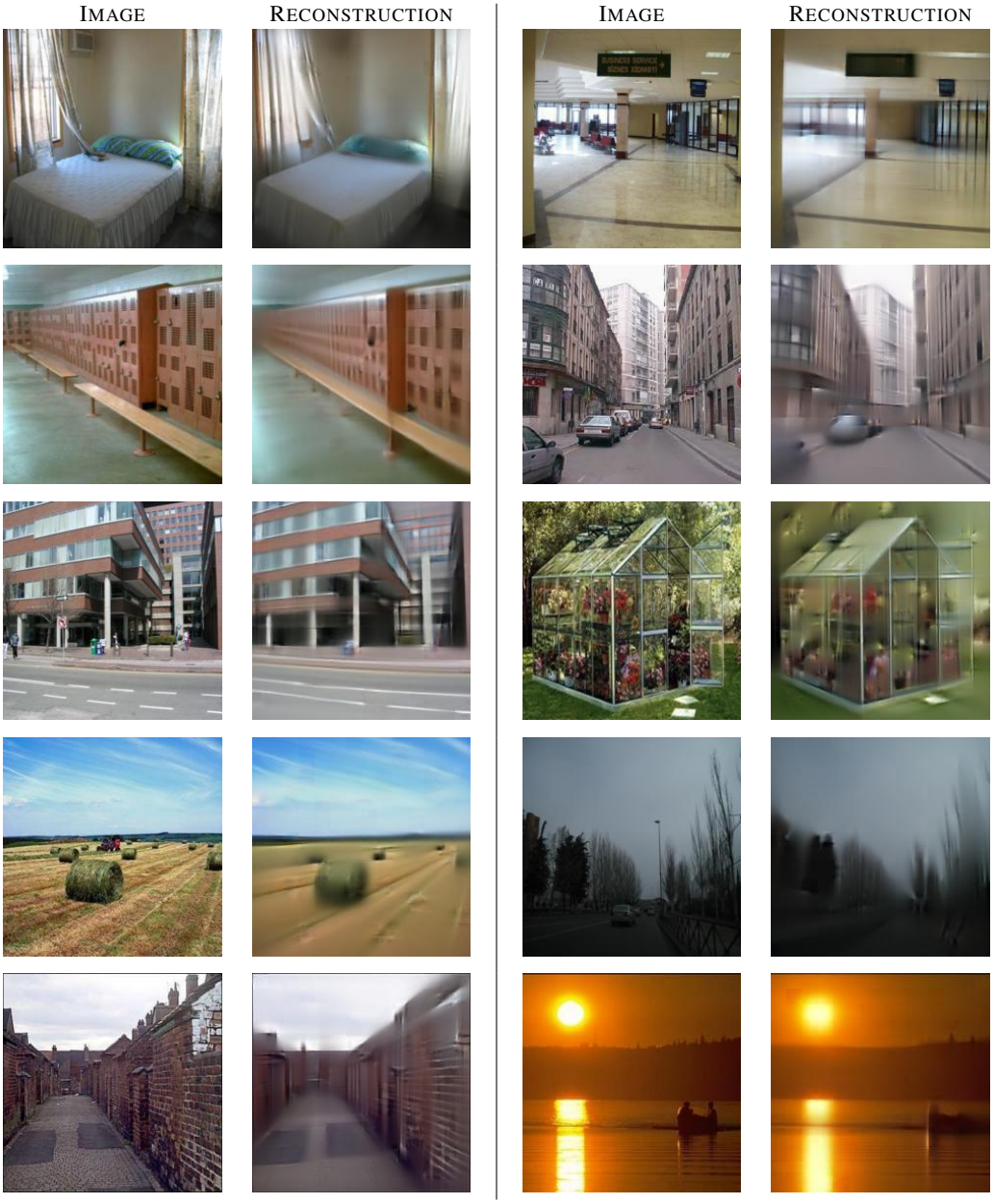

Figure 10: **Image reconstruction on ADE20K:** Figure shows the qualitative results of the image reconstruction task. Image size is 224×224 and 37×16^2 glimpses are used.

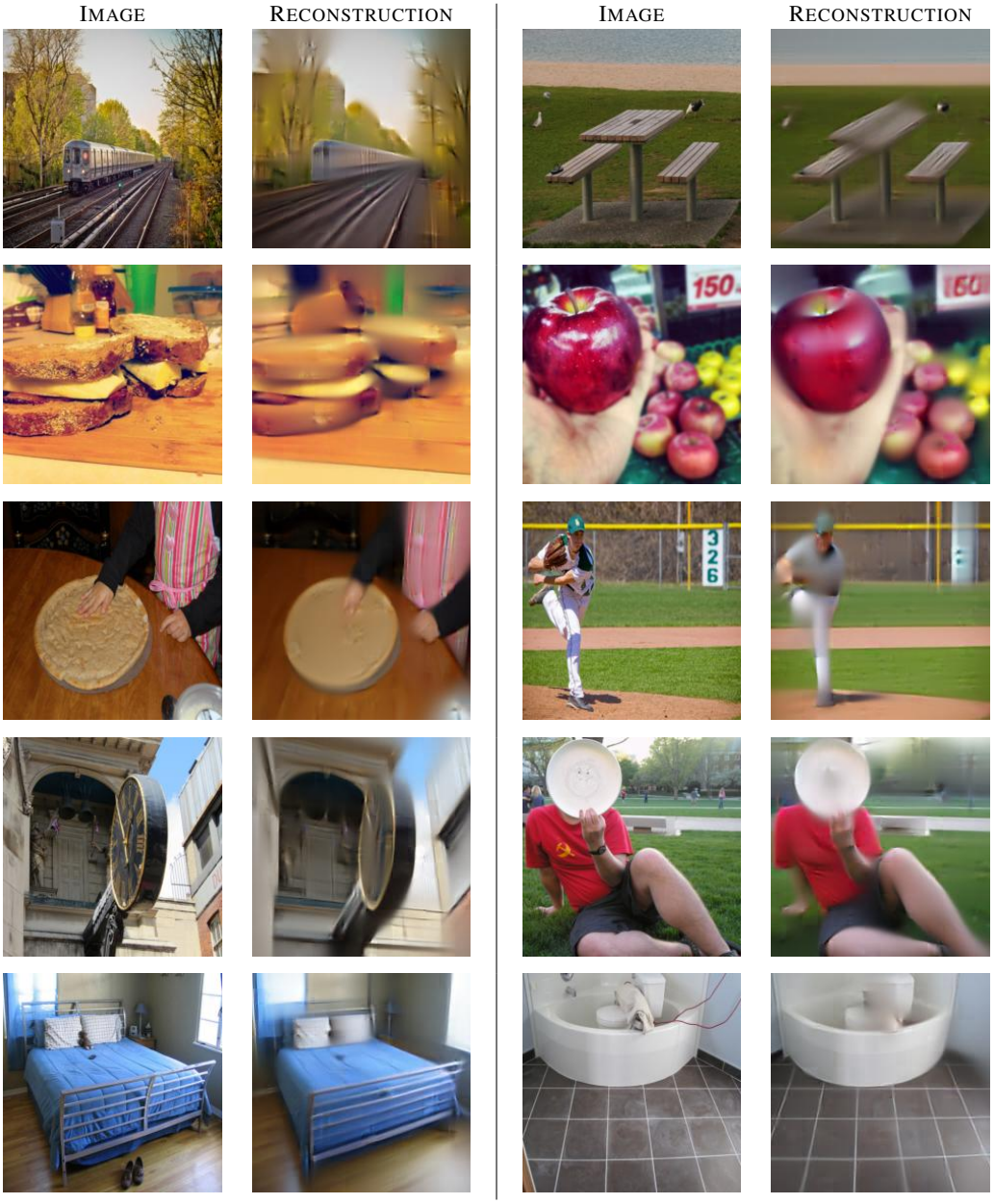

Figure 11: **Image reconstruction on MS COCO:** Figure shows the qualitative results of the image reconstruction task. Image size is 224×224 and 37×16^2 glimpses are used.

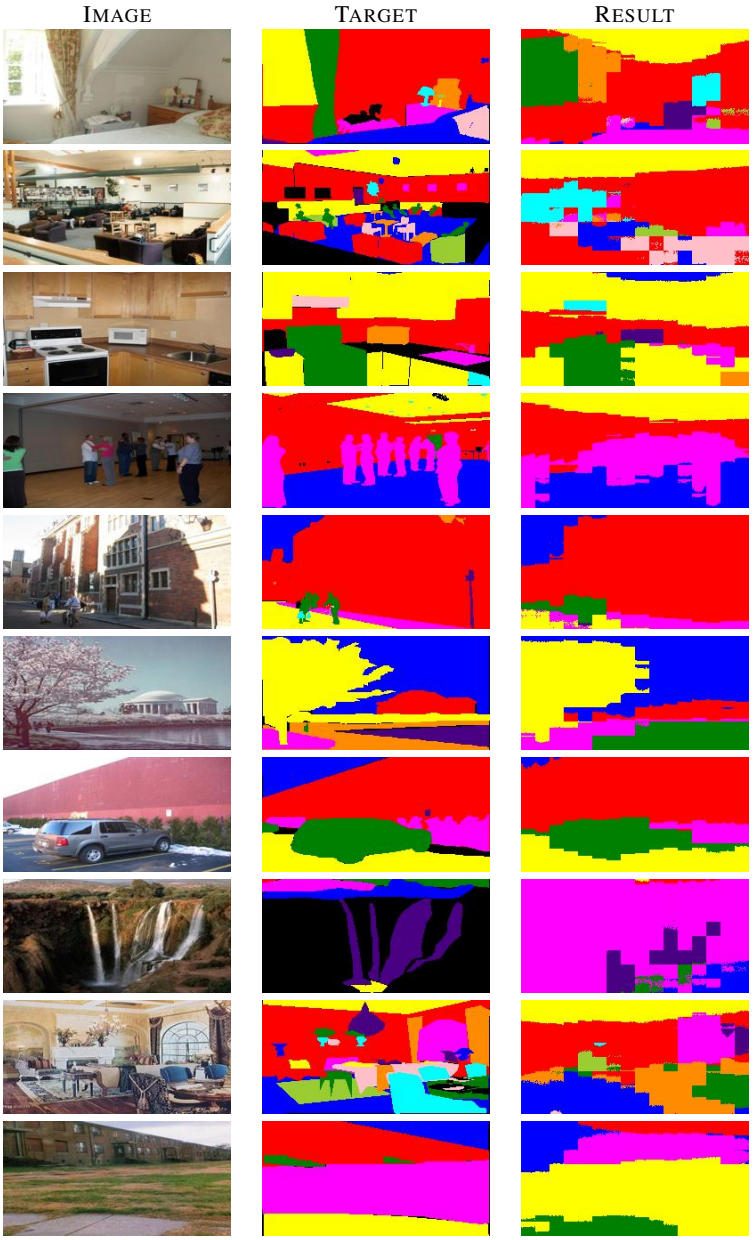

Figure 12: **Image segmentation on ADE20K:** Figure shows the qualitative results of the image segmentation task. Image size is 256×128 and 8×48^2 retinal glimpses are used.