# Comparative Analysis of Segmentation Models on the Human Visual Diet Dataset

Francesco Plastina
*Harvard College '28*

**Teaching an algorithm to identify and separate objects within an image (segmentation) is central to AI applications, from delineating tumors in medical scans to safe robotics, augmented reality, and assistive navigation. Yet most benchmarks use web-style photos, whereas people learn from everyday, human-centric views. In this study, I ask whether training modern segmentation models on human-like images (Human Visual Diet dataset), scenes that resemble what people routinely encounter, changes models' performance and what these models attend to. I compare two conventional convolutional networks (ResNet-18 and DenseNet-121 with custom upsampling decoders) and a transformer-based SegFormer (pre-trained ViT) on their segmentation performance. To interpret model focus, I compute Gradient-weighted Class Activation Mapping (Grad-CAM) for the convolutional neural networks (CNNs) and inspect self-attention maps for the transformer (ViT). My results show that the ViT achieved the highest evaluation accuracy of ~80%, outperforming DenseNet (~74% accuracy) and ResNet (~57% accuracy). All models struggled with fine object boundaries and smaller objects, indicating room for improvement in edge localization. Grad-CAM for ResNet and DenseNet often appeared in diffuse or highlighted background regions, while the ViT attention maps were sometimes inconsistent, suggesting that these models do not consistently provide precise object degrees. While prior studies (e.g., Madan et al., 2024) suggest superior generalization for transformers, our HVD results address segmentation performance only: the transformer leads in accuracy, but all models remain below human-level segmentation, especially at fine boundaries and small objects. These findings underscore the difficulty of achieving human-level generalization in segmentation, aligning with recent work emphasizing the importance of more human-like data 'diets' and architectures to close the gap between neural networks and human vision.**

## Introduction

Semantic segmentation, the process by which an algorithm labels every pixel in an image, supports applications such as tumor outlining in medical scans, robotics, and assistive navigation. A central challenge in segmentation is generalization: keeping high accuracy when the viewpoint, lighting, or context changes. Most benchmarks use web-style, object-centric photos that miss much of everyday visual variation. The Human Visual Diet (HVD) dataset addresses this by presenting objects in varied indoor scenes with controlled changes, more closely resembling what people routinely see (Madan et al., 2024).

This study tests how architecture affects segmentation on HVD and what image regions models rely on. I compare two convolutional neural networks (CNNs) (ResNet-18 and DenseNet-121 with custom upsampling decoders) against a transformer-based SegFormer with a Vision Transformer (ViT) backbone. CNNs are a type of neural net that look at small patches of an image with the same tiny filter sliding around, so it can spot edges, textures, and shapes and build up to full objects (LeCun et al., 2015), while a transformer is a model that looks at all parts of the input image at once and decide which parts matter to each other using "attention" (Vaswani et al., 2017). To see where models "look," I use Grad-CAM, a method to explain a CNN's prediction by coloring the image to show which regions most influenced the decision, using gradients from the last convolution layer (Selvaraju et al., 2017), and attention maps, visualizations of the transformer's attention weights that indicate which patches it focused on, for the ViT. Results are reported as segmentation performance on a specific subset of HVD, without evaluating out-of-distribution generalization. In this framework, we compare the three architectures on HVD, examine training behavior and error patterns with an emphasis on small objects and fine boundaries, and relate accuracy differences to spatial focus using Grad-CAM and attention maps.

Prior work links computer-vision progress to findings in biological vision (Cox & Dean, 2014; Yamins et al., 2014), yet also shows important divergences: CNNs can rely on cues humans do not, and performance can drop under mild changes to images (Bowers et al., 2022; Linsley & Serre, 2023; Nagaraj et al., 2023; Serre, 2019). Vision transformers may capture object–context relations differently, but gains are not universal. Training on more realistic, human-like data has been proposed to improve alignment with human behavior (Nagaraj et al., 2023), and models trained on such "visual diets" can perform better under real-world transformations (Madan et al., 2024). Building on these observations, we use HVD to assess how architectural choices shape segmentation performance and whether higher accuracy coincides with more appropriate spatial focus.

## Methods

### Dataset and Preprocessing

I used the Human Visual Diet (HVD) dataset introduced by Madan et al. (2024), focusing on the photorealistic renderings of indoor scenes contained within the *main_xml* folder. Each image depicts a target object in a fully furnished room with varied layouts, materials, and lighting. Ground-truth segmentation labels were provided as .npy masks in *labels_main_xml*. Each pixel is labeled with an integer class ID corresponding to an object category (e.g., bed, sofa, table, wall, floor, etc.). There are 27 distinct class IDs present in our subset of HVD (1, 4, 5, 10, 11, 13, 21, 23, 31, 32, 37, 41, 42, 43, 44, 45). The object each class corresponds to can be found in the dataset folder. For training,

I applied minimal preprocessing: images were resized to 64 × 64 pixels (to reduce memory, as a compromise given computational constraints) and masks were likewise downsampled to 64 × 64 using nearest-neighbor interpolation. The pixel intensity values were normalized to [0, 1]. While this downsampling sacrificed detail (i.e., edges become less crisp), it allowed training of multiple models within a reasonable time with low computational power. I randomly split the dataset into 80% training and 20% validation images. No test set of completely novel scenes was used; the evaluation focused on the validation performance for generalization to unseen images of similar distribution. Because our evaluation was in-dataset (no novel-scene test), I kept the training setup the same for every model. This ensured a fair basis for comparison: any differences stemmed from the architectures, not from training choices.

**Model Architectures**

Three segmentation models were implemented:

1. **ResNet-18 + Upsampling Decoder:** Using a ResNet-18 backbone pretrained on ImageNet (up to the final convolutional block, removing the classification head). This provided a 512-channel feature map (of size 2 × 2 after our 64 × 64 input), which was fed into a custom decoder, a series of 2 × 2 bilinear upsamples with intervening 3 × 3 convolutional layers reducing channels. The final layer was a 1 × 1 convolutional layer producing class logits for each of the 27 classes, and I upsampled this to the original image size.
2. **DenseNet-121 + Upsampling Decoder:** Similarly, I took DenseNet-121 pretrained (on ImageNet) and truncated at its convolutional feature map. A decoder with three upsample steps (and convolutional layers halving channels each step) was used, yielding final class logits at 128 × 128, which were then interpolated to 128 × 128 (my chosen output size for DenseNet model). The ResNet and DenseNet decoders were implemented from scratch, but following a typical fully convolutional network design for segmentation.
3. **SegFormer (ViT):** A vision transformer-based segmentation model. I used the HuggingFace Transformers library's SegformerForSemanticSegmentation implementation, starting from the pre-trained weights on ADE-20k. This model has an efficient MiT-B0 backbone and a lightweight MLP decoder. I modified the final classification head to produce 27 classes instead of ADE-20k's 150, and allowed the head weights to initialize randomly.

For fair comparison, I fine-tuned all models on the HVD training set.

**Training Procedure**

I trained ResNet and DenseNet for up to 100 epochs, while SegFormer was fine-tuned for 50 epochs as it converged faster. I employed cross-entropy loss on the pixel labels, ignoring unlabeled pixels (if any, since most pixels were labeled). During training, I tracked the pixel-wise accuracy (percentage of pixels correctly classified) across classes on both training and validation sets. Pixel accuracy is a coarse measure that can be high if large areas are correctly labeled, even if smaller objects are not correctly classified. No heavy data augmentation was applied, given that the dataset already contained substantial variation.

**Grad-CAM Implementation**

To interpret the CNN models used in this project, specifically for ResNet and DenseNet, I implemented Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al., 2017). This method highlights the regions of an input image that most strongly influence the model's decision for a particular class. The process begins by passing an image through the network to obtain the prediction scores for each class. For a chosen class, I computed the gradient of its score with respect to the feature maps of the last convolutional layer in the encoder. These gradients indicate how sensitive the class score is to changes in different spatial regions of the feature maps. By averaging the gradients across the spatial dimensions, I obtained a single weight for each channel in the feature map, reflecting its importance to the class prediction. Each channel was then multiplied by its corresponding weight, and the results were summed to create a raw heatmap. Applying a ReLU function ensures that only positive contributions are kept, emphasizing the regions that positively influence the prediction. Finally, the heatmap was upsampled to the size of the input image and overlaid using a color map for visualization. This technique visualized where the model was focusing when identifying objects, and assessed whether it was attending to relevant parts of the image or being distracted by background features. This method was developed by Selvaraju et al. (2017).

**Transformer Attention Maps**

To interpret the SegFormer model's predictions, I extracted its internal self-attention matrices to visualize where the model focuses when making decisions for a given class. SegFormer, unlike some transformers, doesn't use a special classification token. Instead, it processes the input image as a grid of patches. For the B0 variant I used, the image was downsampled by a factor of 32 in height and width so that the final attention maps operate on a much smaller patch grid. During evaluation, I enabled the model to return its attention weights, which are square matrices showing how much each patch in the image attends to all other patches.

To generate attention maps for a specific class, I first ran a forward pass to obtain the predicted segmentation mask and the attention weights from the final transformer layer. I then identified the set of patch positions that were labeled as the class of interest, such as "lamp" or "bed." For each of these patches, I extracted its corresponding row in the attention matrix, which represents how strongly that patch attends to every other patch. I averaged these attention vectors across all selected patches and across all attention heads to get a single, representative attention map for that class. Then, I reshaped this one-dimensional attention vector into a 2D grid matching the layout of the patch map, and upsampled it to the size of the original input image. Finally, I overlaid the resulting heatmap on the image to visualize where the model was attending when making its class prediction. This attention visualization doesn't directly explain what caused the prediction, like Grad-CAM, but it shows which regions the model relied on when assigning a class to specific parts of the image. I applied this technique to two different class predictions to study how the transformer uses spatial context. My implementation was

based on the HuggingFace SegFormer model's output, and while it was inspired by more advanced attention rollout methods, I kept the approach simple and focused on per-class attention accumulation.

**Comparison**

I directly compared the models on their pixel accuracy performance to evaluate their segmentation capabilities, estimating their IoUs (Intersection over Union) that describe how much two regions (the ground truth and the prediction of the model) overlap. Moreover, I compared how the different models decide on assigning a specific pixel class by directly showing their Grad-CAM (for ResNet and DenseNet) and attention (for ViT) maps.

## Results

**Segmentation Performance**

All three models successfully learned to segment the HVD scenes to a reasonable degree, but with notable differences in accuracy. The Vision Transformer (SegFormer-B0) achieved the best overall performance (Fig. 1), followed by DenseNet (Fig. 2), then ResNet (Fig. 3). On the validation set, the SegFormer attained pixel accuracy of approximately 81% and an estimated mean IoU of around 0.60. DenseNet's accuracy was about 72% at convergence (mean IoU ~ 0.50), and ResNet's was around 57% (mean IoU ~ 0.35–0.40). These quantitative gaps reflect the models' capacity differences and possibly the benefit of SegFormer's pretrained features. Training and validation loss curves showed that ResNet-18 underfitted the data: its validation accuracy peaked early and then stagnated, indicating it struggled with the complexity of the task (many classes, high variability), given its smaller capacity. DenseNet-121, with its deeper architecture, learned faster and attained higher accuracy, but eventually showed signs of overfitting, as validation loss began rising after about 90 epochs while training loss kept decreasing. The SegFormer converged rapidly, within about 10–15 epochs, to a low validation loss, suggesting that the pretrained transformer features were already well-aligned to the segmentation task, especially since ADE-20k pre-training provides a strong prior on indoor scenes.

**Training Dynamics**

Figures 1–3 illustrate the validation pixel accuracy curves for the models. ResNet's curve starts lowest and improves slowly, leveling off around 55–58%. DenseNet starts higher and climbs to approximately 74% before slightly dipping (because of overfitting). SegFormer starts around 62% (already high due to pretraining) and improves to about 81%. The relative ordering of the curves stayed consistent across training. Validation accuracy and training accuracy followed similar trends, though training accuracy could reach above 90% for DenseNet and SegFormer, indicating some overfitting (the gap between training and validation accuracy for DenseNet was about 20% by epoch 100). I attempted mild regularization (dropout in the decoder, weight decay) for DenseNet, which narrowed this gap but did not fundamentally change the final results. The transformer, despite its capacity, did not severely overfit, likely because the pretrained weights acted as a strong regularizer.

**Qualitative Segmentation Examples**

I inspected segmentation outputs on several validation images to understand the models' successes and failures. In general, SegFormer produced the most coherent and correct segmentations, although it showed some weird overlays with the input images. DenseNet also performed surprisingly well, as its prediction mask did not differ greatly from the ground truth. ResNet performed poorly, though it was still able to segment between walls, floor, and other objects. Each model had significant difficulty in recognizing edges and boundaries, and in general, the 3D structure of the images. Figures 4—6 show a sample of the same input image and the outputs of each model.

**Grad-CAM Visualization (CNNs)**

I applied Grad-CAM to interpret where ResNet-18 and DenseNet-121 focus their attention for predicting certain classes. For the target object class in each scene—the object the model was asked to identify in context—Grad-CAM maps tended to highlight broad regions that only partially overlapped the object. For example, in a living room scene where the target is "floor", the Grad-CAM for class "floor" on DenseNet highlighted a specific side of the floor (Fig. 7), and highlighted part of the wall and of an object nearby. This suggests the model's prediction of "wall" was influenced by features in adjacent regions or simply due to imprecise localization in the feature maps. ResNet's Grad-CAM for the same class in other images was even more spread out, essentially covering the entire image (Fig. 8). Overall, Grad-CAMs for DenseNet were slightly more localized than ResNet's, reflecting DenseNet's superior segmentation performance. Yet in comparison to humans, who would squarely focus on the object of interest, the Grad-CAMs appeared noisy and overinclusive. This aligns with known issues that CNNs don't always have interpretable internal representations, but they may rely on context textures.

**Transformer Attention Visualization**

For the SegFormer model, I examined its last-layer attention maps for selected classes. While I expected the transformer's attention mechanism to show a more structured pattern, I instead found largely unstructured attention maps. For example, two very different classes, "cabinet" (42) and "floor" (44) shared pretty much the same attention maps and did not consider the "floor" at all (Fig. 9). In summary, the ViT's attention maps clearly did not align with human-like rationale, which tends to focus the attention on the object for recognition. This mirrors the Grad-CAM issue: higher accuracy didn't necessarily translate into more pinpointed attention. It seems the SegFormer achieves higher accuracy through its ability to integrate features over a larger receptive field, but it still spreads its attention across the scene in ways that are hard to interpret.

## Discussion

**Practical Significance**

On the HVD indoor scenes, a compact transformer (SegFormer) achieved the highest in-dataset pixel accuracy (approximately 80%) relative to two CNN baselines (DenseNet: approximately 74%, ResNet: approximately 57%). For professionals working with HVD-like data and limited compute, this identifies a strong default baseline. At the same time, all models showed consistent errors on small objects and along fine boundaries. Therefore, in applications where edge precision matters (e.g.,

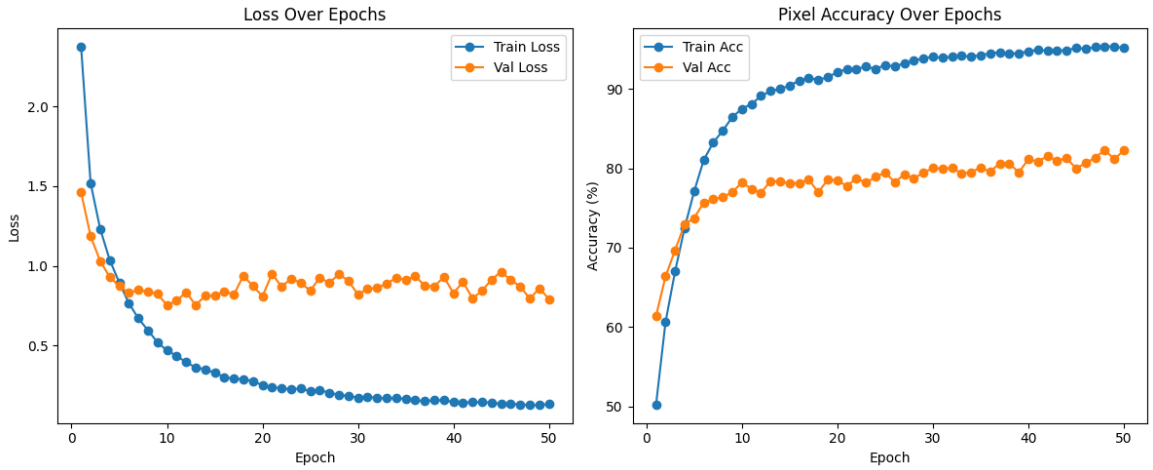


**Figure 1.** *ViT Segmentation Performance:* The graph on the left shows ViT's training and the validation loss. Since the training loss (blue) is going down while the validation loss (orange) is almost constant, also ViT is overfitting, but less than the previous two models. Indeed, despite the pixel accuracy of the model on the training data (blue) is higher than the accuracy of the model on the validation data (orange), the two graphs are closer than before.

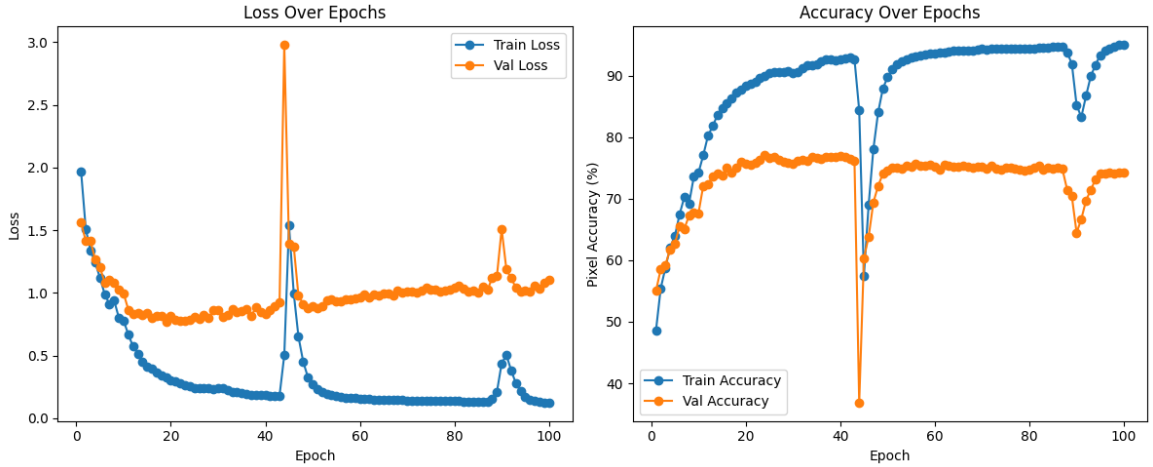


**Figure 2.** *DenseNet Segmentation Performance:* The graph on the left shows DenseNet's training and the validation loss as explained in figure 1. Since the training loss is going down and the validation loss is going slightly up, DenseNet is aslo overfitting, but less than ResNet. The overfitting is more clear in the figure on the left where the pixel accuracy of the model on the training data (blue) is higher than the accuracy of the model on the validation data (orange), while still better than ResNet performance.

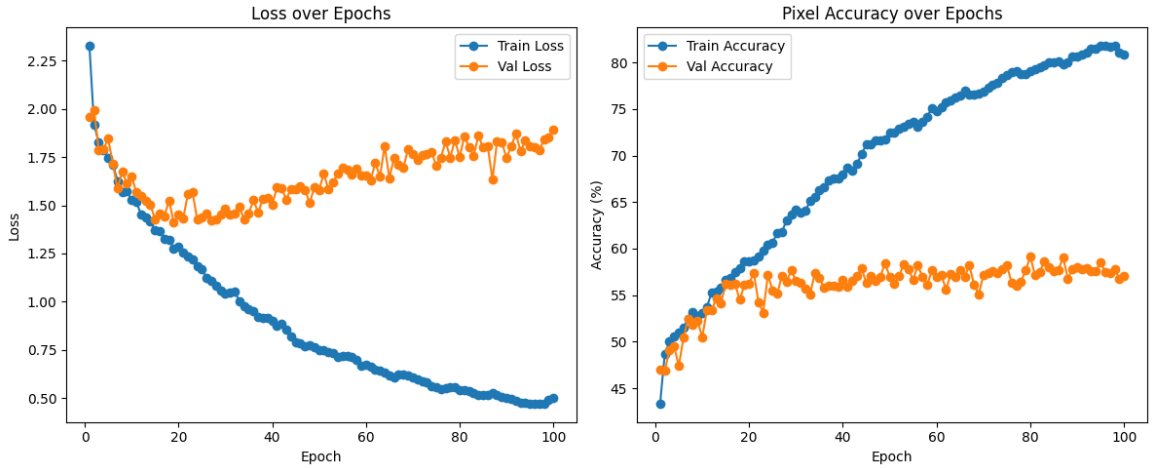


**Figure 3.** *ResNet Segmentation Performance:* The graph on the left shows ResNet's training and the validation loss. The training loss (blue) is measured on the training data while the model is learning, while the validation loss (orange) is measured on a separate hold-out set that the model never learns on and therefore estimates how well the model will do on new data. Since the training loss is going down while the validation loss is going up, ResNet is overfitting, which means that the model learns the training data so well that it doesn't generalize to new data. The overfitting is more clear in the figure on the left where the pixel accuracy of the model on the training data (blue) is way higher than the accuracy of the model on the validation data (orange).

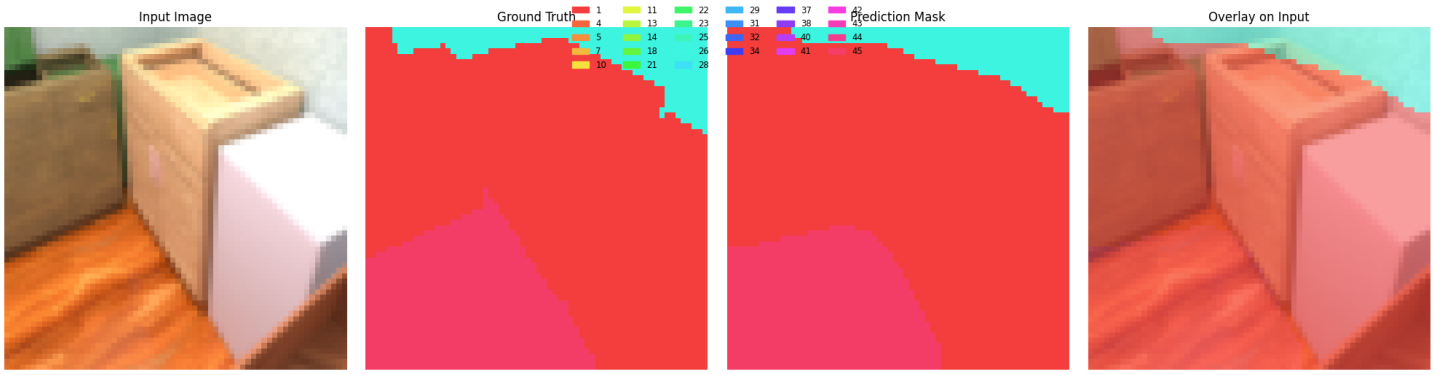


**Figure 4.** *ResNet output sample:* These images show a sample of ResNet's output prediction and compare it with the input image. Starting from the left, in order it is possible to see the input image, the ground truth, the prediction mask of the model (its output), and the overlay of the prediction on the input image. For each color corresponds one of the 45 different classes of the dataset.

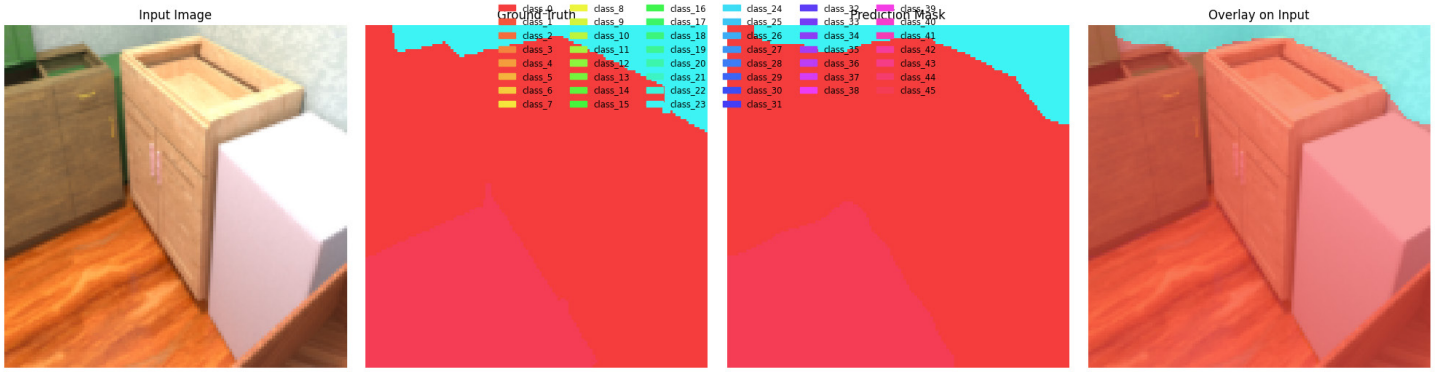


**Figure 5.** *DenseNet output sample:* These images show a sample of DenseNet's output prediction and compare it with the input image. Starting from the left, in order it is possible to see the input image, the ground truth, the prediction mask of the model (its output), and the overlay of the prediction on the input image. For each color corresponds one of the 45 different classes of the dataset.

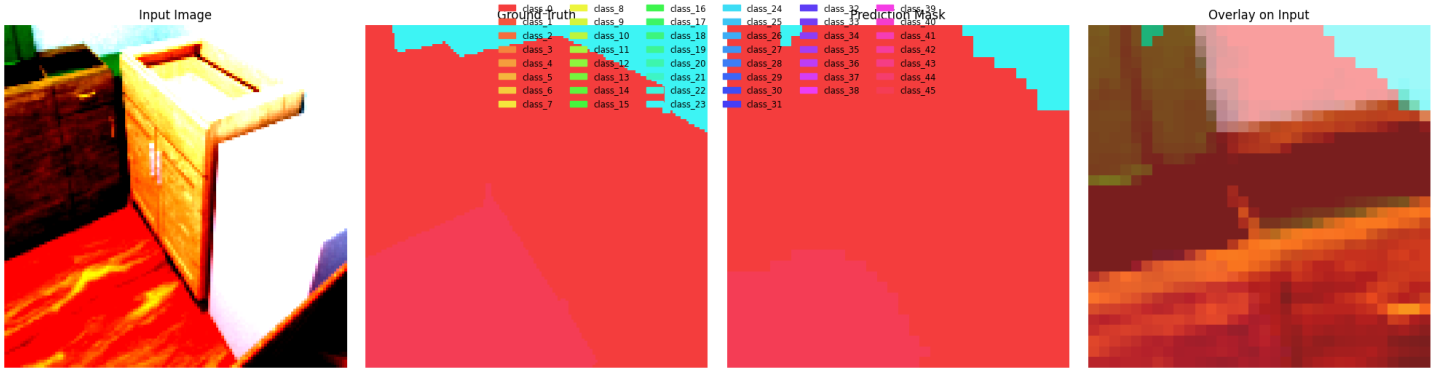


**Figure 6.** *ViT output sample:* These images show a sample of ViT's output prediction and compare it with the input image. Starting from the left, in order it is possible to see the input image, the ground truth, the prediction mask of the model (its output), and the overlay of the prediction on the input image. For each color corresponds one of the 45 different classes of the dataset.

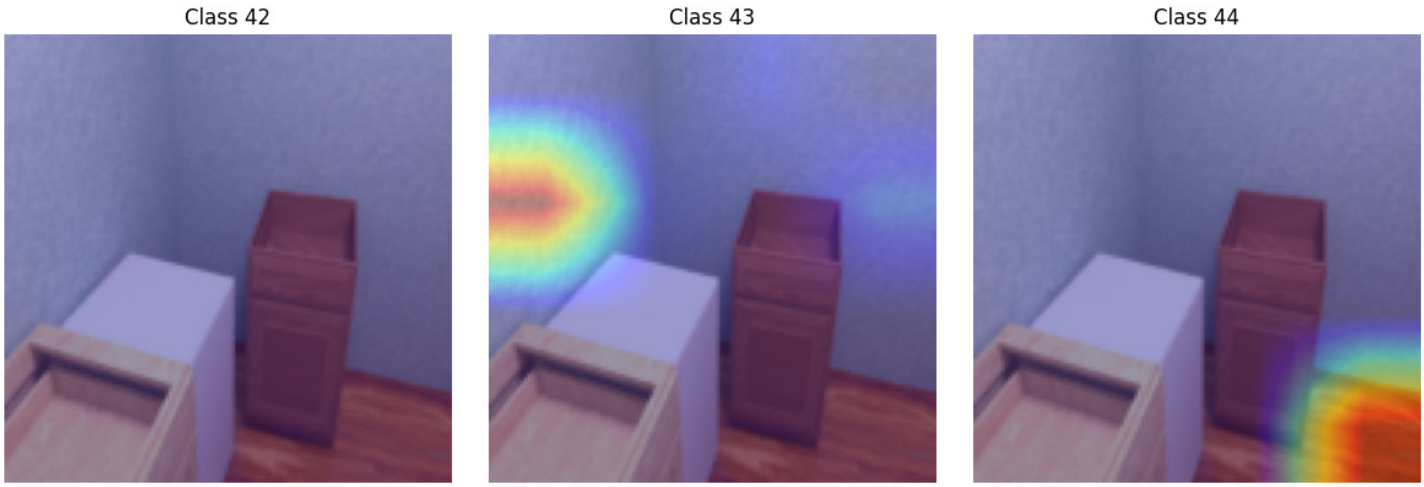


**Figure 7.** *DenseNet Grad-Cam visualization sample:* These images show which elements of the image DenseNet focused on to make the segmentation decision (assigning the pixels to their specific class) for three random classes (42,43,44). Warm colors correspond to higher "focus", while cool colors to less "focus".

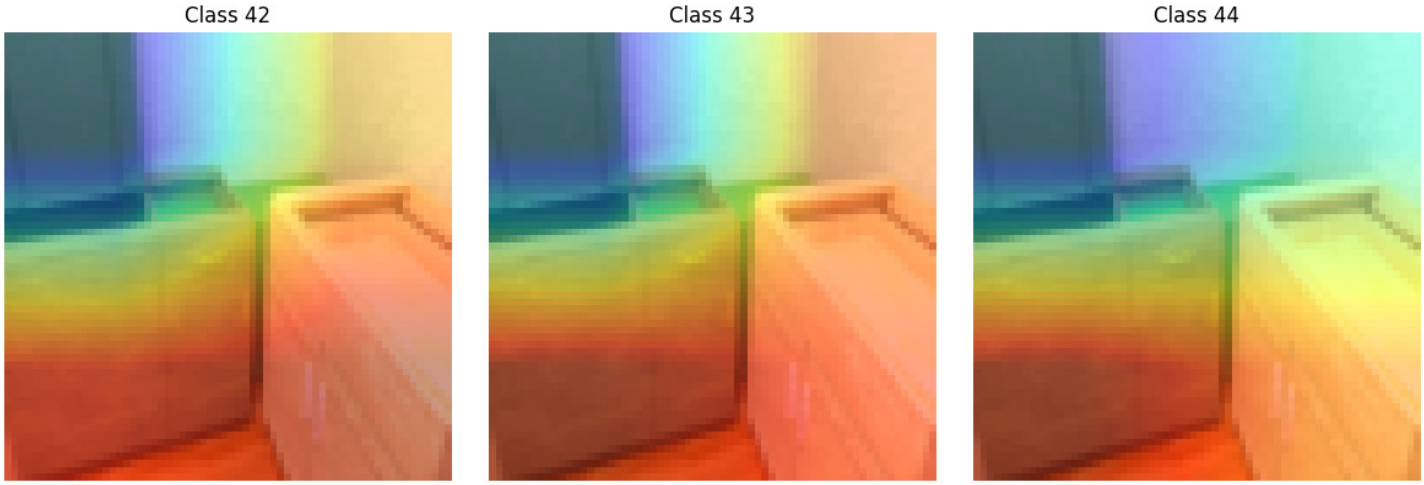


**Figure 8.** *ResNet Grad-Cam visualization sample:* These images show which elements of the image ResNet focused on to make the segmentation decision (assigning the pixels to their specific class) for three random classes (42,43,44). Warm colors correspond to higher "focus", while cool colors to less "focus".

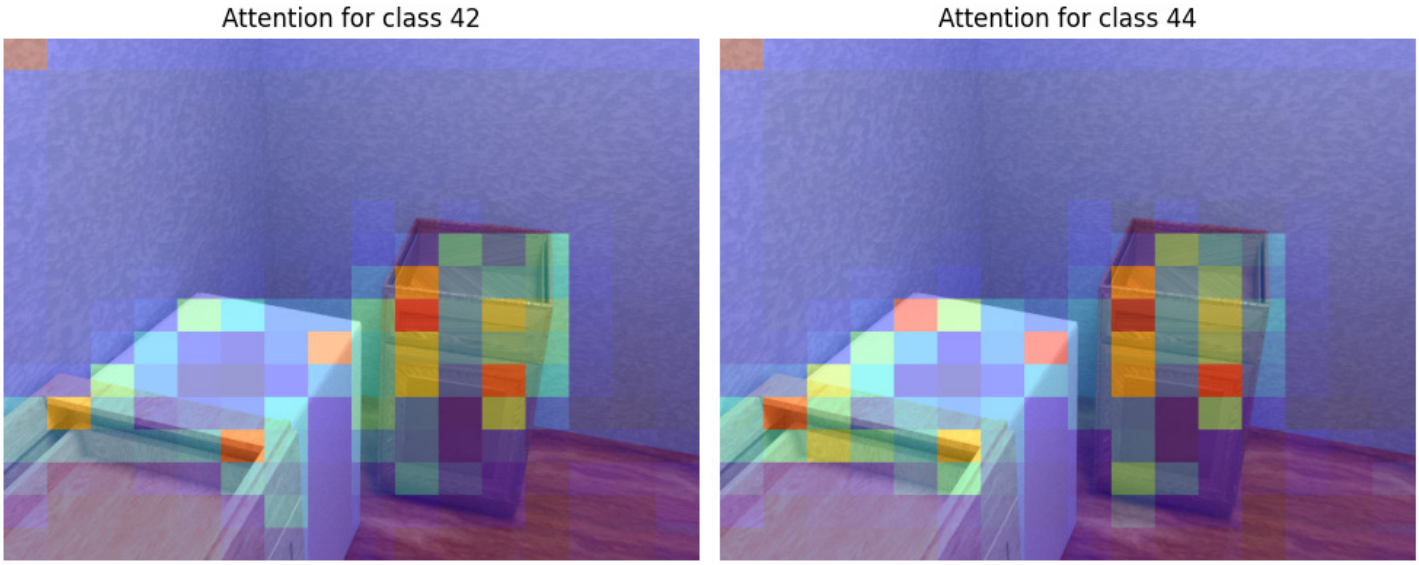


**Figure 9.** *ViT attention maps:* These images show which elements of the image ViT focused its "attention" on to make the segmentation decision (assigning the pixels to their specific class) for two random classes (42,44). Warm colors correspond to higher "focus", while cool colors correspond to less "focus".

medical delineation, manipulation in robotics), these failure modes will directly limit utility unless addressed.

**Accuracy vs Human-like Reasoning**

Qualitative analyses (Grad-CAM for CNNs, attention maps for the transformer) did not reveal reliable, object-tight focus. CNN heatmaps often spread into the background; transformer attention was instead broad or inconsistent. These analyses caution against interpreting higher accuracy as evidence of human-like spatial reasoning and suggest that current explanation maps are, at best, sanity checks rather than proof of mechanism.

**Context: Necessary but Insufficient**

HVD contains natural variation in viewpoint, lighting, and materials, and the models maintained in-dataset performance across these variations in the validation split. The results, therefore, support using human-like data for evaluation, but also indicate that data realism alone does not fix boundary or occlusion errors.

**Takeaway**

Within HVD, transformers offer a practical accuracy advantage under matched (or near-matched) conditions, but all tested models remain limited by boundary and small-object errors and do not show reliably human-like spatial focus. Closing this gap likely requires pairing human-like data with inductive biases and objectives that explicitly encode edges, scale, and occlusion, not just scaling existing architectures or relying on generic pretraining.

## Limitations

**Pre-training and Resolution Mismatches**

To keep the study reproducible and focused, I used standard pre-trained features for each network (ImageNet for CNNs; ADE20K for SegFormer with a reinitialized 27-class head) and matched evaluation on a common 128×128 grid despite differing training resolutions between ResNet and the other networks due to GPU limits. These choices favor practical, commonly used configurations over custom, model-specific tuning. As a result, I limit claims to in-dataset performance; I do not claim out-of-distribution generalization, and I note that resolution and pretraining differences may contribute to the observed gaps. Qualitative analyses (Grad-CAM and attention maps) and a discussion of small-object/boundary errors are included to make these trade-offs explicit.

**Downsampling**

The study was made with a limited computational power and memory using the free time-limited GPUs offered on Google Colab. The large size of the dataset and of the images was impossible to handle with such limited computational power. To tackle this challenge, I decided to downsample. While I recognize that downsampling may create a loss of some information, since these important results were shown on a smaller scale, it is reasonable to assume that they will be shown on the larger data.

**Grad-CAM and Attention Maps**

The use of Grad-CAM and attention maps in the study aims to be only a qualitative analysis of how these models are making their decisions while assigning each class to each pixel. Further studies may focus on follow-up analysis to quantify the attention/feature maps.

## Conclusions

On the Human Visual Diet dataset, a compact transformer (SegFormer-B0) outperformed two CNN baselines, yet none approached human-level segmentation, with persistent errors on fine boundaries and small objects. Interpretability analyses (using Grad-CAM for CNNs and attention maps for the transformer) revealed diffuse or inconsistent focus, cautioning against equating higher accuracy with human-like reasoning. These results suggest that human-like data are helpful but insufficient. Closing the gap will require objectives and architectures that explicitly encode edges, scale, and occlusion, alongside higher-resolution training and evaluation beyond in-dataset splits. Future work should probe out-of-distribution generalization to novel scenes and explore edge-aware losses or multi-scale decoders.

## Code Availability

I built the training pipeline and custom decoders from scratch in Google Colab for ResNet and Densenet models, while the SegFormer model utilization relied on the HuggingFace Transformers library implementation. The code for this project is available at https://github.com/fran0324/Neuro-240-Project.

## References


Bowers, J. S., Malhotra, G., Dujmovic, M., Montero, M. L., Tsvetkov, C., Biscione, V., Puebla, G., Adolfi, F., Hummel, J. E., Heaton, R. F., Evans, B. D., Mitchell, J., & Blything, R. (2022). *Deep Problems with Neural Network Models of Human Vision. The Behavioral and Brain Sciences*, *46*, e385.

Cox, D. D., & Dean, T. (2014). *Neural Networks and Neuroscience-Inspired Computer Vision. Current Biology*, *24*(18), 2073-2216.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep Learning. Nature*, *521*(7553), 436–444.

Linsley, D., & Serre, T. (2023). *Fixing the problems of deep neural networks will require better training data and learning algorithms. The Behavioral and Brain Sciences*, *46*, e400.

Madan, S., Li, Y., Zhang, M., Pfister, H., & Kreiman, G. (2024). *Improving out-of-distribution generalization by mimicking the human visual diet. "NeuroAI: Fusing Neuroscience and AI for Intelligent Solutions" NeurIPS workshop.*

Nagaraj, A., Ashok, A. K., Linsley, D., Lewis, F. E., Zhou, P., & Serre, T. (2023). *Ecological data and objectives align deep neural network representations with humans. "UniReps: Unifying Representations in Neural Models" NeurIPS workshop.*

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). *Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. IEEE International Conference on Computer Vision (ICCV)*, 618-626.

Serre, T. (2019). *Deep Learning: The Good, the Bad, and the Ugly. Annual Review of Vision Science*, *5*, 399–426.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need. Advances in Neural Information Processing Systems 30* (NIPS 2017).

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). *Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciences of the United States of America*, *111*(23), 8619–8624.