Object Recognition Deliverable 3: Body and Cloth Depth Estimation

Agúndez, Pedro
pedro.agundez@estudiantat.upc.edu

Lang, Sheena
sheena.maria.lang@estudiantat.upc.edu

Parent, Zachary
zachary.parent@estudiantat.upc.edu

Recalde, Martí
marti.recalde@estudiantat.upc.edu

Sánchez, Bruno
bruno.sanchez@estudiantat.upc.edu

May 26, 2025

1 Introduction

Estimating depth from monocular RGB images of clothed human subjects is a challenging and highly relevant task with applications in virtual try-on systems, realistic 3D avatar reconstruction, and human motion analysis. This project addresses this problem by predicting dense depth maps from cropped RGB images using a subset of the CLOTH3D dataset [1], which contains synthetic video sequences of animated human figures dressed in a wide range of garments and poses. Preprocessing plays a crucial role in ensuring data consistency and involves center-aligned square cropping with margin control, as well as the removal of frames in which the subject is partially out of view. These steps help align the RGB and depth data for supervised learning and are discussed in detail in Section 2.

To establish a baseline, we first implemented a UNet-based architecture [2] and applied basic data augmentation along with hyperparameter tuning to improve its generalization. As part of our experimental design described in Section 3, we further investigated the potential of more advanced neural architectures, including *Attention U-Net* [3], *ResUNet-a* [4], and *TransUNET* [5]. These models incorporate attention mechanisms or transformer-based components to better capture long-range spatial dependencies and semantic structures in the input images.

In addition to architectural changes, we explored the impact of introducing a perceptual loss based on surface normals, designed to enforce geometric fidelity by penalizing differences between estimated and ground truth normal maps. This approach aims to capture finer details of human body shape and garment structure. Furthermore, we incorporated SMPL pose information by generating color-coded pose maps and concatenating them with the RGB input. This auxiliary input provides explicit structural priors about human joint locations and orientations, and we evaluate its effectiveness in enhancing depth prediction performance.

The outcomes of these experiments are presented in Section 4, with both quantitative metrics and qualitative visualizations. Finally, we summarize our key findings and outline directions for future work in Section 5.

2 Dataset and Preprocessing

For this project, we worked with the CLOTH3D dataset [1]. CLOTH3D is a large-scale dataset featuring 3D clothed human sequences. It provides data such as SMPL [6] body parameters, various garment geometries with textures, and animations across numerous sequences, encompassing a diverse range of subjects and motions.

To prepare this dataset, we utilized and adapted the provided starter-kit. This kit offered foundational modules for essential tasks such as reading the dataset's specific file formats, handling 3D mesh input/output, and basic depth rendering capabilities. Our main effort in this preprocessing phase was the development of the `preprocessing.py` script, which orchestrates the pipeline. Through this script, we implement the following steps to transform the raw CLOTH3D data into a structured and normalized format suitable for our subsequent modeling work.

2.1 Initial Frame Handling

For each sample sequence in CLOTH3D, we begin by extracting individual frames from the provided RGB video files and their corresponding segmentation mask videos. We use `ffmpeg` for this extraction, with the process managed within our `preprocessing.py` script. The extracted RGB frames and segmentation masks (treated as an alpha channel) are then merged into RGBA PNG images. Following this, we employ the `DataReader` class (part of the adapted starter-kit) to access frame-specific information, including SMPL parameters, 3D mesh data for the body and garments, and camera metadata.

2.2 Depth Map Generation

We generate depth maps for each frame to capture the 3D geometry of the scene:

1. **Mesh Aggregation:** We gather the 3D vertices and faces for the human body (derived from SMPL parameters) and any associated garments. Garment face data, which can comprise both triangles and quadrilaterals, is systematically converted into a triangle-only format.

2. **Rendering:** We use the `Render` class (which leverages DEODR, from the starter-kit), along with camera intrinsic and extrinsic parameters derived from the dataset's metadata, to render full-resolution (640×480 pixels) depth maps. A maximum rendering depth of 10.0 units is set, which also serves as the background depth value.

3. **Transformation and Storage:** From the rendered depth map, we identify a foreground mask and compute its bounding box. We then use these parameters to crop the region of interest and resize it to fit a 236×236 pixel content area using nearest-neighbor interpolation. This resized content is placed onto a 256×256 canvas, maintaining a 10-pixel margin and filling background areas with the maximum depth value. The final float32 depth map is saved as a `.npy` file. For inspection, we also save an 8-bit visualizable PNG version.

2.3 SMPL Pose Visualization

We also generate 2D visualizations of the SMPL pose for each frame:

1. **Joint Projection:** We obtain the 3D SMPL joint locations for the current frame and project them into the 640×480 2D image space using the dataset's camera parameters.

2. **Drawing and Transformation:** On a transparent 640×480 RGBA canvas, we draw the projected limbs and joints using a predefined color scheme. This full-resolution pose image is then transformed—cropped and resized to fit the 236×236 content area on a 256×256 canvas—using the same geometric parameters derived from the depth mask's bounding box. Linear interpolation is used for resizing.

3. **Storage:** The final 256×256 RGBA pose image is saved as a PNG file.

2.4 RGB Image Processing

The RGBA frames extracted in the initial step are also processed for consistency:

1. We ensure these frames are in RGBA format.
2. Using the same crop, scale, and margin parameters derived from the depth mask, we transform and resize these images to a final 256×256 resolution, with the main content occupying the central 236×236 area. Linear interpolation is used for resizing.
3. These processed 256×256 RGBA images are saved as PNG files.

These preprocessing steps create a dataset where depth maps, SMPL pose visualizations, and RGB images are co-registered and normalized in size, facilitating their use in our project. All processed outputs for each sample are organized into dedicated subdirectories.

Note: We saved the images with the alpha channels to ensure full consistency with the original given data but, in practice, the fourth channel is dropped when the images are fed into the models, since these only accept RGB data.

3 Experiment Design

This section outlines the experimental approach used to evaluate depth estimation performance. We begin by tuning a UNet-based baseline model. We then test three architectures: Attention U-Net [3], ResUNet-a [4], and TransUNET [5]. Additionally, we incorporate a perceptual loss based on surface normals and examine the effect of adding SMPL pose information as auxiliary input. Each component is detailed in the subsections below.

3.1 Baseline Model Tuning

To establish a strong baseline for the depth estimation task, we used a 2D UNet architecture [2], implemented as a configurable and modular network with separate downsampling and upsampling paths. The UNet consists of stacked convolutional blocks with optional batch normalization and ReLU or GELU activations, allowing flexible experimentation with network depth, filter sizes, and pooling strategies. The model takes cropped RGB images as input and outputs a single-channel depth map.

We performed an extensive hyperparameter search to explore the impact of various configurations on model performance. Each configuration was trained for 12 epochs across multiple random seeds to ensure robustness. The baseline configuration included five downsampling stages with filters of size $[64, 128, 256, 512, 1024]$, batch normalization enabled, GELU activation, and a learning rate of 1×10^{-4}.

To evaluate different design choices, we tested five additional variations:

- **Reduced Depth**: The deepest layer was removed, reducing the filter configuration to $[64, 128, 256, 1024]$.
- **No Batch Norm**: Batch normalization was disabled to test its effect on training stability and generalization.
- **Higher Learning Rate**: The learning rate was increased to 3×10^{-4} to assess convergence speed.
- **Lower Learning Rate**: The learning rate was decreased to 3×10^{-5} to test stability and fine-grained learning.

- **With Augmentation**: Data augmentation was applied using a composed transform pipeline consisting of random rotation, horizontal flip, color jitter, random erasing, and normalization using dataset-specific mean and standard deviation values.

This systematic tuning process helped identify which architectural components and training settings contributed most significantly to performance improvements, and it provided a solid baseline for comparing more advanced models.

3.2 Vision Transformer Architectures

Following the baseline model tuning, we extended our investigation to include more advanced architectures that incorporate attention and transformer mechanisms. We selected three representative models: *Attention U-Net* [3], *ResUnet-a* [4], and *TransUNET* [5]. These architectures were chosen for their diversity in design, introducing components such as spatial attention gates, residual connections, and transformer-based encoders, respectively. To ensure full compatibility with our PyTorch training pipeline and to allow for detailed architectural control, we re-implemented all models from scratch rather than using existing TensorFlow implementations.

3.2.1 Attention U-net

Attention U-Net extends the standard U-Net architecture by incorporating attention gates into the skip connections. These gates learn to emphasize relevant spatial regions and suppress less informative features, helping the network focus on important structures such as body contours and clothing details.

In our implementation, each upsampling block includes an attention gate that modulates the encoder features using a gating signal from the decoder. This attention-modulated output is then concatenated and passed through convolutional layers. The overall architecture mirrors the baseline UNet in depth and filter sizes, using GELU activations, batch normalization, and bilinear upsampling.

This mechanism enhances the model's ability to selectively recover fine-grained spatial features in the predicted depth maps.

3.2.2 ResUnet-a

ResUnet-a is an advanced encoder-decoder architecture that enhances the traditional U-Net by integrating residual connections, atrous (dilated) convolutions, and Pyramid Scene Parsing (PSP) pooling. These components improve the model's ability to capture multi-scale context and spatial dependencies, which are especially important for detailed depth estimation of complex human shapes and clothing.

In our implementation, we adapted the original architecture for single-channel depth prediction. The model features a deep residual encoder using multiple dilation rates within each block to expand the receptive field without losing resolution. The decoder combines these multi-scale features using skip connections and PSP pooling to refine spatial details at each level. The final output passes through a dedicated depth branch with normalization and activation layers, concluding with a sigmoid output.

ResUnet-a's deep, multi-resolution design and strong spatial reasoning capabilities make it particularly well-suited for recovering detailed geometry in challenging depth estimation tasks.

3.2.3 TransUNET

TransUNet is a hybrid architecture that integrates three main components: a ResNet-based encoder (ResNet-50), a Vision Transformer (ViT) bottleneck, and a U-Net-style decoder. The ResNet backbone extracts hierarchical spatial features, which are passed to the transformer block

for global context modeling. The final decoder gradually upsamples the features to recover spatial resolution and produce dense pixel-wise depth predictions.

In the original work, both the ResNet encoder and the ViT bottleneck were initialized with pre-trained weights, while the decoder was trained from scratch. Since the transformer block contains the majority of the model's parameters and requires extensive data and compute to train effectively, we consistently used pre-trained weights for the ViT in all experiments.

To further explore the role of pretraining, we conducted an ablation study comparing two versions of the model: one with pre-trained weights for both the ResNet and the Transformer, and another using pre-trained weights only for the Transformer. This allowed us to isolate the contribution of the ResNet encoder's pretraining and evaluate its impact on depth estimation performance.

This setup provides insight into how different components of the architecture benefit from transfer learning, especially in scenarios with limited computational resources and training data.

3.3 Perceptual Loss

To improve the geometric quality of predicted depth maps, we introduced a perceptual loss based on surface normals. Specifically, we computed normal maps from both the predicted and ground truth depth images using Sobel filters and normalized gradient vectors. The perceptual loss was then defined as the L1 or L2 discrepancy between these normal maps. This encourages the network to better capture fine-grained structural details, such as edges and contours, that are often lost with standard pixel-wise losses alone.

We applied this combined loss—comprising both standard mean squared error (MSE) and perceptual loss—to two models: the baseline UNet and our best-performing architecture, *TransUNet (pre-trained)*. We tested different discrepancy types (L1 and L2) and relative weights for the perceptual loss term in the total loss. This setup allowed us to evaluate the benefit of integrating geometric supervision alongside traditional loss functions and observe how it affects both shallow and transformer-based models.

3.4 SMPL

To evaluate whether providing explicit body pose information could enhance depth estimation, we conducted an additional experiment incorporating SMPL data. Using the SMPL model, we extracted joint locations and pose parameters, which were rendered as color-coded pose maps. These pose maps were concatenated with the RGB input, resulting in a six-channel input image instead of the standard three-channel RGB.

This augmented input was used with our best-performing model *TransUNet (pre-trained)* to assess whether the added structural priors improved prediction quality. Since the original ResNet backbone of TransUNet expects three-channel input (due to pretraining on ImageNet), we replaced its first convolutional layer with a new one configured to accept six channels. As a result, this initial layer was randomly initialized, while the rest of the backbone and transformer remained pre-trained.

This modification allowed us to isolate the effect of SMPL-based pose information without disrupting the benefit of pretraining in the rest of the model.

4 Results and Analysis

4.1 Quantitative Results

Through each of the experiments described in Section 3, we gathered the per-epoch losses (MSE and, when appropriate, Perceptual Loss) on the training and validation sets. Then, at the end of the training process, we evaluated the resulting model in the test set. For each of the experiments, we ran 3 seeds to validate the robustness and statistical significance of our results.

4.1.1 Baseline Model Tuning

The first experiment set corresponds to the baseline model tuning. In Figure 1, we can see the validation plots for each of these experiments. There are some interesting conclusions we can extract from this graph:

- The *data augmentation* techniques did not help with better generalization and, in fact, made the worst-performing models out of this experiment set. We expect that the many frames per video of the dataset allow the dataset to be sufficiently diverse to learn effectively without augmentation.
- The *lower learning rate* shows a stable but under-performing learning curve, and would probably have benefited further training or a decreasing learning rate schedule.
- *Batch normalization* was indeed useful, as the models without it performed worse than the ones with the *base* configuration.
- The models with *reduced depth* performed slightly worse than the *base* ones, but not by a great margin. A future study on the trade-off between computing time and performance between these two configurations could yield interesting results.
- Overall, the best performing configurations were the *base* and the one with *higher learning rate*, although we observed higher variation with the latter.

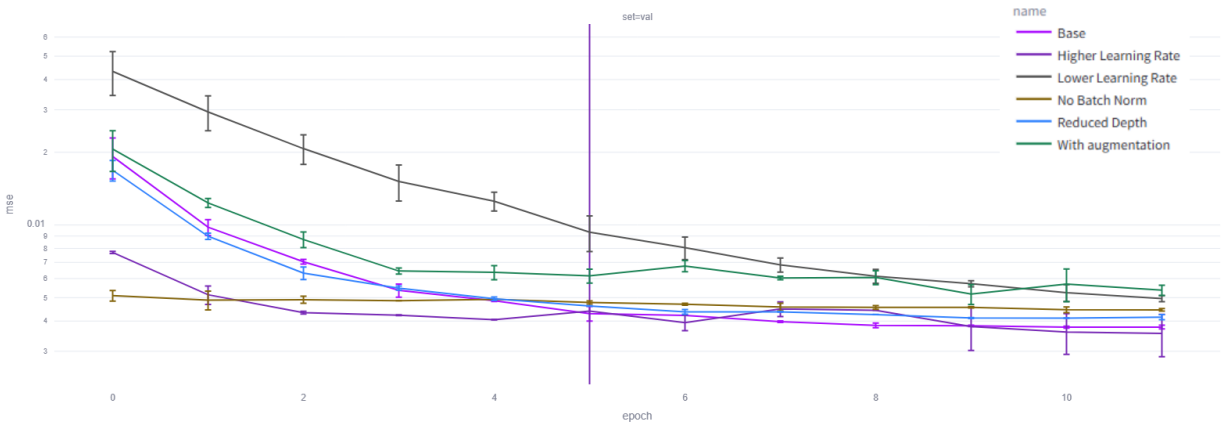


Figure 1: Validation plots for the baseline model tuning experiment set

On the test set, the *base* configurations had MSE scores in the range $[2.96 \cdot 10^{-3}, 3.04 \cdot 10^{-3}]$, while the *higher learning rate* ones scored in the range $[2.62 \cdot 10^{-3}, 3.38 \cdot 10^{-3}]$. Although the *higher learning rate* configuration was the one to obtain the best score out of all the experiments in this set, it also obtained one of the worst. For this reason, we chose to proceed with the *base* configuration for the next set of experiments, due to the instability of performance when using a *higher learning rate*.

4.1.2 Vision Transformer Architectures

In the second experiment set, we test 3 alternative architectures to compare with the base U-Net2D. Figure 2 shows the validation plots for these experiments.

For *ResUNet-a*, we can see that it does not seem to learn as well as the others (it has one of the worst performances), and its validation score seems to have plateaued by the end of the training process. This likely indicates that we do not have enough data to effectively train such a complex network with our current depth estimation dataset.

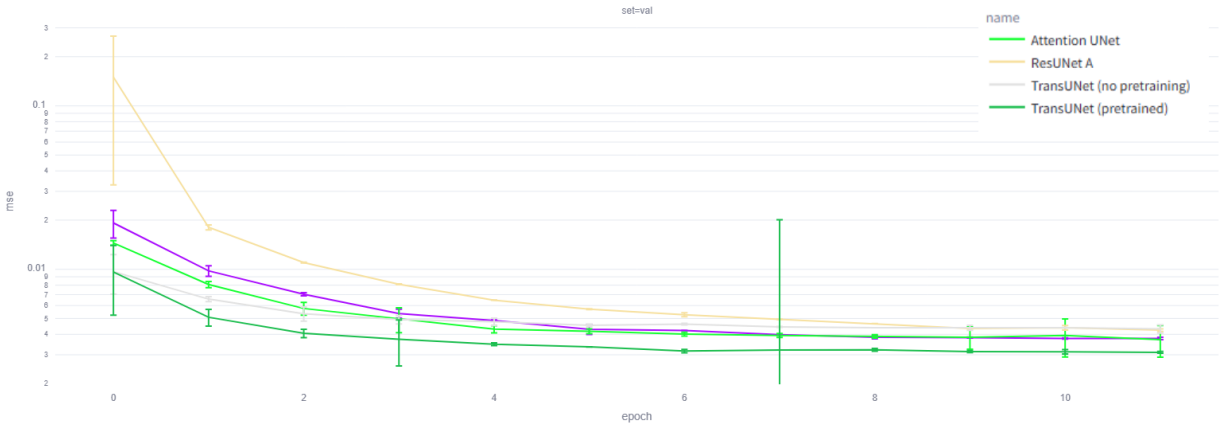


Figure 2: Validation plots for the vision transformer architectures experiment set. The purple curve corresponds to the *base* configuration from the previous experiment set.

As for *Attention UNet*, it achieves the single best validation score by the end, but its performance is somewhat unstable, compared with the *pretrained TransUNet*, which consistently achieves top scores.

The results for *TransUNet* are especially interesting, since we can see the great impact of using pretrained weights on its ResNet encoder:

- When not using them, *TransUNet* has the **worst** median MSE of this experiment set.
- When we do utilize the pretrained weights, this architecture has the **best** median score of the set.

On the test set, *TransUNet (pretrained)* achieved a median MSE of $2.44 \cdot 10^{-3}$, while *Attention UNet* scored $3.01 \cdot 10^{-3}$. Therefore, we proceeded to the next set of experiments with *TransUNet (pretrained)*.

4.1.3 Perceptual Loss

As a preliminary step for this experiment set, we performed an initial study using the baseline 2D UNet configuration to test different values for the weight, w, of the perceptual loss:

$$\text{Loss} = w \cdot \text{Perceptual} + (1 - w) \cdot \text{MSE}$$

We tested $w = \{0.25, 0.5, 0.75\}$ and detected no significant difference, with $w = 0.5$ yielding very slightly better performance than the other values. For this reason, we selected said value for the perceptual loss experiment set performed on our best model, *TransUNet (pretrained)*, where we studied the effect of using the L1 and the L2 discrepancy errors on the normal maps for the computation of the perceptual loss. Figure 3 shows the validation plots for these experiments.

The validation results show that there is no significant difference in performance when including the perceptual loss term, nor is there a difference between using the L1 or L2 discrepancies for its computation. In the test set, the median MSE scores were:

No perceptual loss	$2.44 \cdot 10^{-3}$
L1 perceptual loss	$2.50 \cdot 10^{-3}$
L2 perceptual loss	$2.42 \cdot 10^{-3}$

Even though the L2 perceptual loss seems to have very slightly improved performance on the test set, this conclusion does not agree with the one extracted from the validation results, and we

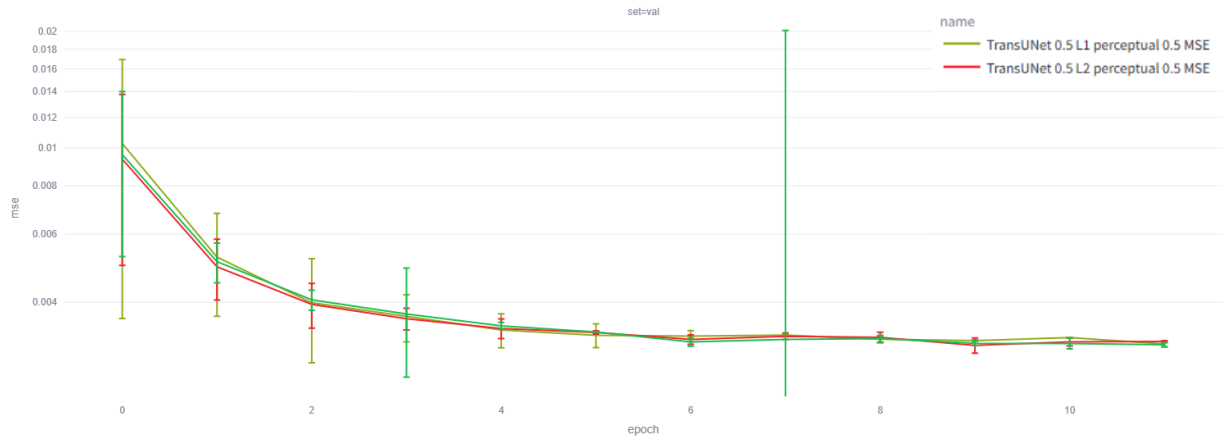


Figure 3: Validation plots for the perceptual loss experiment set. The bright green curve corresponds to the *TransUNet (pretrained)* configuration from the previous experiment set.

believe the difference is not significant enough to justify changing the loss term. Therefore, we still consider *TransUNet (pretrained)* our best-performing model. Nevertheless, for completeness, we performed our final set of experiments both with and without the L2 perceptual loss.

4.1.4 SMPL

For our final set of experiments, we studied the effect of including SMPL pose information along with the RGB image input. Figure 4 shows the validation plots for these experiments.

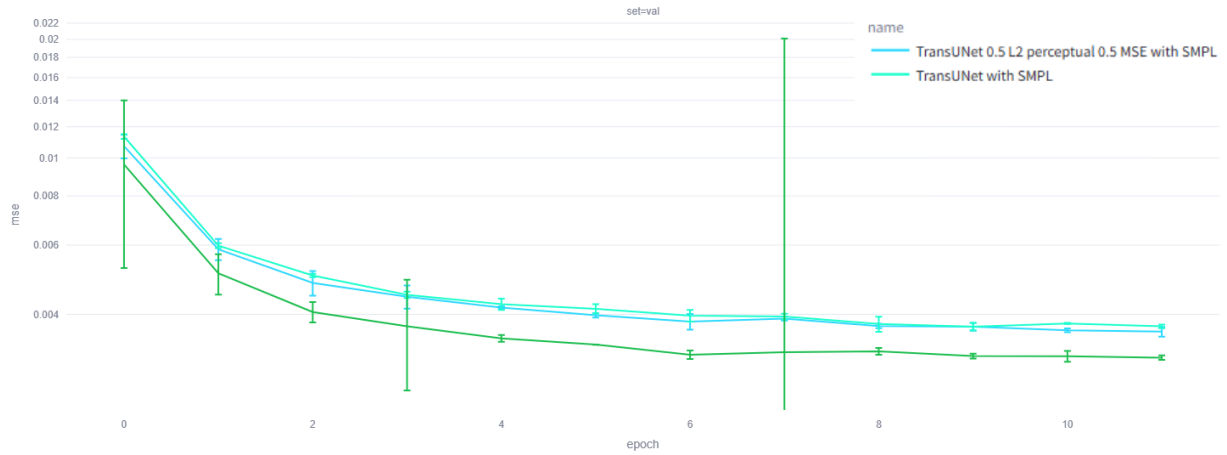


Figure 4: Validation plots for the SMPL experiment set. The bright green curve corresponds to the *TransUNet (pretrained)* configuration.

The validation results clearly show that the addition of SMPL information did not help the model with depth estimation, neither with nor without the L2 perceptual loss term (between which there seemed to be no significant difference). The test results agree with this conclusion: both of the experiments with pose information achieved a median MSE of $2.83 \cdot 10^{-3}$, which is worse than the $2.44 \cdot 10^{-3}$ achieved by *TransUNet (pretrained)*.

It is possible that the modification of the first convolutional layer of the ResNet encoder of TransUNet (with the corresponding removal of its pre-trained weights) negatively impacted the model's performance for the experiments where pose information is included. However, this

change is necessary to adapt the network to accept the three additional input channels. Different approaches to this modification could be studied in future experiments. For instance:

- Load the weights corresponding to the first three (RGB) channels, and only randomly initialize the weights corresponding to the three additional (SMPL) channels.
- Alternatively, do not load any pretrained weights at all, thereby isolating the true effect of including SMPL pose information.
- Finally, we could overlay the pose information directly over the renders, although this overloading of information may have unintended consequences.

4.2 Qualitative Results

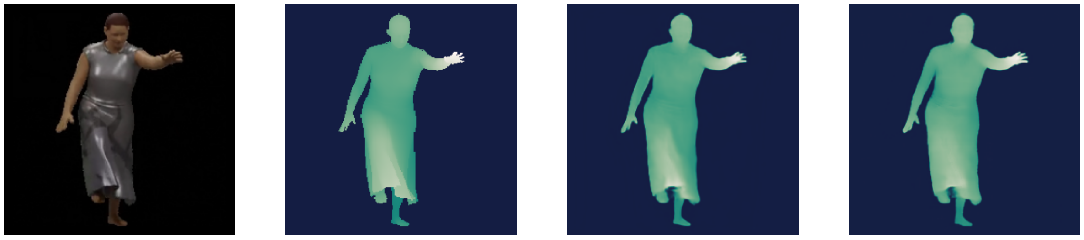
To further validate that the models were well-trained and estimating depth reasonably, we also gathered qualitative results, by inspecting and comparing the predicted depths on various frames. In Figure 5 we show the inference process, in which a raw frame is normalized, then passed through the network, and finally compared with the original ground truth depth for the baseline model UNet2D.

(a) Input image (b) Normalized input image (c) Ground truth (d) Depth estimation from *UNet2D base*

Figure 5: The inference process on a validation frame (video 158, frame 233)

.

In Figure 5d, we see evidence of one of the difficulties for the model: correctly separating finger depths. These small areas do not contribute much to the MSE loss, but do distract significantly from the faithfulness of the estimate. We hoped that perceptual loss might help prioritize better learning of these fine edges, but did not see significant improvement.



(a) Input image (b) Ground truth (c) *TransUNet (pretrained)* (d) *UNet2D with data augmentation*

Figure 6: Comparison of inference on a difficult test frame (video 173, frame 245) between the best model *TransUNet (pretrained)* and the worst model *UNet2D with data augmentation*

.

We were especially interested in understanding on which parts of the image the model failed for frames with lower MSE performance. Figure 6 shows a comparison of the output depth estimate on a frame which for which all models showed low MSE. We compared our best model, *TransUNet (pretrained)*, with one of the worst models, *UNet2D with data augmentation*.

We can see that one of the parts of the image where the model struggles is estimating the depth of the figure's right leg underneath the skirt. Figure 6d shows the model estimating both legs at roughly the same depth, whereas Figure 6c produces a better, though not perfect, estimate of the leg closer to the camera. Both models demonstrated decent estimation of the edges of the forward hand in this case. Perhaps this is an easier case, because the hand is over the background instead of over another part of the figure.

5 Conclusions

In this project, we addressed the problem of monocular depth estimation for clothed human subjects using the CLOTH3D dataset [1], a large-scale synthetic dataset providing rich annotations such as RGB frames, 3D meshes, and SMPL pose parameters [6] across diverse human motions and garments. By building a complete preprocessing and modeling pipeline, we explored several techniques to enhance depth prediction accuracy from single RGB images. Our key findings were:

- **Baseline 2D UNet Tuning**
  We constructed a baseline model based on the 2D UNet architecture and systematically tuned its hyperparameters. This baseline served as a foundation for comparing more advanced models.

- **Vision Transformer Architectures**
  Among the alternatives, we implemented and assessed Attention U-Net, ResUNet-a, and TransUNet. Our results showed that ResUNet-a was likely too complex for our dataset, leading to suboptimal learning. Attention U-Net showed promise, with strong validation performance, but lacked consistency across training runs.

  TransUNet, particularly the configuration with pretrained weights for both the ResNet encoder and the Vision Transformer bottleneck, consistently outperformed all other models in both validation and test metrics. This highlights the significant benefit of transfer learning and the effectiveness of combining convolutional and transformer-based approaches in capturing spatial and contextual information critical for accurate depth estimation.

- **Perceptual Loss (Surface Normals)**
  We further investigated the addition of a perceptual loss term based on surface normals to enforce geometric consistency. However, its impact on performance was minimal, with slight variations that were not statistically significant across configurations.

- **SMPL Pose Information**
  Similarly, while incorporating SMPL pose information via color-coded pose maps was conceptually promising, it did not improve depth prediction performance. We attribute this, at least in part, to the required changes in the model architecture—specifically the modification of the first convolutional layer—which disrupted the benefit of pretrained weights.

Despite these limitations, the pipeline we developed proved robust and modular, and our experiments provided valuable insights into the architectural and training design decisions that most impact depth prediction performance. Moving forward, we see several promising directions for future research. These include more sophisticated methods for integrating pose information without disrupting pretrained components, and training with larger or more diverse datasets to improve generalization.

In summary, pretrained *TransUNet* emerged as the most effective model for our task, achieving the lowest test MSE and producing the most visually accurate depth predictions. Our work demonstrates the viability of learning accurate depth maps from RGB inputs in realistic scenarios involving clothing and complex poses, and it lays the groundwork for future improvements in model design and data utilization.

References

[1] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. "CLOTH3D: clothed 3d humans". In: *European Conference on Computer Vision*. Springer. 2020, pp. 344–359.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

[3] Ozan Oktay et al. *Attention U-Net: Learning Where to Look for the Pancreas*. 2018. arXiv: 1804.03999 [cs.CV]. URL: https://arxiv.org/abs/1804.03999.

[4] Foivos I. Diakogiannis et al. "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162 (2020), pp. 94–114. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2020.01.013. URL: https://www.sciencedirect.com/science/article/pii/S0924271620300149.

[5] Jieneng Chen et al. *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*. 2021. arXiv: 2102.04306 [cs.CV]. URL: https://arxiv.org/abs/2102.04306.

[6] Matthew Loper et al. "SMPL: A Skinned Multi-Person Linear Model". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16.