Object Recognition

Universitat de Barcelona

# Body and Clothes Depth Estimation

Academic year: 2021/2022

Author: Joaquim Marset Alsina

# 1 Introduction

The present report describes the results of the body and clothes depth estimation task applied to the *CLOTH3D++* dataset [1]. As the name suggests, we want to predict the depth of the body and clothes of persons appearing in images, predicting it for each pixel using only an RGB image as input to the network.

In section 2, we briefly explain the dataset and the pre-processing we perform on the data. In section 3, we explain the training schedule we have applied, the different experiments we have performed, and the numeric results of those experiments. In Appendix A, we show some qualitative results we obtain when applying data augmentation, as it is the most different experiment. In Appendix B we show qualitative results with our best model.

# 2 Dataset

*CLOTH3D++* is a dataset containing video segments of synthesized 3D human models in different poses wearing different kinds of clothes. The clothes are classified into seven classes (T-shirt, shirt, top, trousers, skirt, jumpsuit and dress), and they can have different fabrics, colours, tightness and topology. For our particular task, we need to extract the different frames of those videos together with the ground-truth depth map, which will serve as our training data. Given that we are only interested in the body and clothes estimation, we use a binary mask to filter the background and possible objects that could appear beside the person.

Thanks to the provided code, we can extract all the data we need, storing the frames as images and the depth maps and masks as `.npy` binary files. Then, we preprocess by extracting square crops centred on the human subject, leaving a border of 10 pixels in the four directions. However, we discard those frames where the person appears cut or do not even appears. We also discard those frames where the square crop (including the extra border) does not fit the image (i.e. the crop is outside the image). We ensure this way always having an image centred on the person.

# 3 Training

In this assignment we have followed a *Keras* post [2] performing depth estimation on a dataset of indoor and outdoor scenes. They train a *UNet* using a combination of three losses. The first one is the Structural Similarity Index (SSIM). The second one is the L1 distance (or point-wise depth). The third one is a depth smoothness loss. Each loss is computed between the ground truth and predicted depth maps, averaging the values within each batch and averaging again to obtain the final value at each epoch.

Given that we have the mask to filter the person as an extra input, we have modified the *Keras* code, as explained in the assignment description. However, we have decided to perform additional modifications to those proposed in the task, as they have been essential in training the model. In particular:

- We do not convert the ground truth depth maps to meters, as we later scale them to the range of [0, 1]. Therefore, we only subtract the minimum to move it to 0.
- We divide the depth maps by the corresponding value when feeding them to the network. We ensure this way a range of [0, 1].
- The last layer of the network has a linear activation instead of a `tanh`. We tried using a sigmoid activation, given the range of values, but we obtained better results with the linear.
- We compute additional metrics to assess the results of the predicted depth maps. We have used some of the metrics they use in most depth estimation papers. In particular, we compute the Root Mean Squared Error (RMSE) and two accuracy metrics $\delta_1$ and $\delta_2$ defined in [1]. They compute an accuracy in terms of the fraction of pixels in the predicted depth map such that the difference in depth values between the prediction and the ground truth is not higher than 25% and 56% of the predicted values, respectively. In the papers, they use additional metrics like the Absolute Relative Error or the Squared Relative Error, but we found them similar to the RMSE, and we decided not to include them. We compute the value at each epoch like we do with the loss.

[1] `https://chalearnlap.cvc.uab.cat/dataset/38/description/`

[2] `https://keras.io/examples/vision/depth_estimation/`

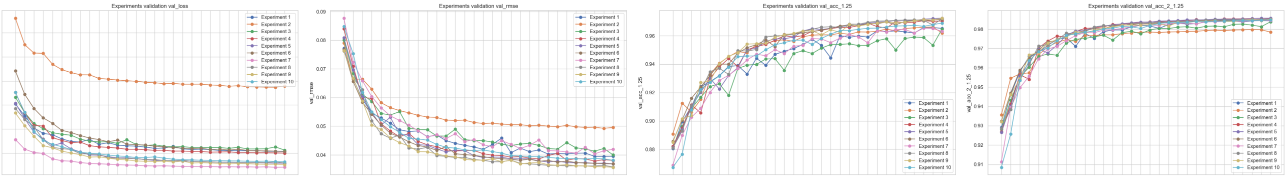
## 3.1 Experiments

We have performed different experiments as requested in the task description. Regarding the training hyperparameters, we have changed the image size (the same for the image, depth map and mask), the batch size, the learning rate schedule, and the loss function. We have changed the loss function by changing the weights of the three components and adding some of the additional metrics we compute as part of the loss computation (influencing the weights updates). We also perform data augmentation as we did in the first assignment. Given that we do not have the depth for those added objects, we assume they are placed in front of the person, occluding it and hindering the body and clothes depth prediction.

The experiments are listed in Table 1. In Figure 1 we show plots comparing the evolution of the validation loss, RMSE, and accuracies, for all the experiments:

| Experiment | Batch size | Image Size | Learning Rate | Loss Weights | | | Extra Loss | Augmentation |
|---|---|---|---|---|---|---|---|---|
| | | | | SSIM | L1 | Edge | | |
| 1 | 32 | 256 | 2e-4 | 0.85 | 0.1 | 0.9 | No | No |
| 2 | 32 | 128 | 2e-4 | 0.85 | 0.1 | 0.9 | No | No |
| 3 | 64 | 256 | 2e-4 | 0.85 | 0.1 | 0.9 | No | No |
| 4 | 32 | 256 | 1e-3 with exp decay | 0.85 | 0.1 | 0.9 | No | No |
| 5 | 32 | 256 | 1e-3 with exp decay | 0.9 | 0.1 | 0.4 | No | No |
| 6 | 32 | 256 | 1e-3 with exp decay | 0.9 | 0.7 | 0.4 | No | No |
| 7 | 32 | 256 | 1e-3 with exp decay | 0.4 | 0.1 | 0.9 | No | No |
| 8 | 32 | 256 | 1e-3 with exp decay | 0.9 | 0.1 | 0.4 | RMSE (equal) | No |
| 9 | 32 | 256 | 1e-3 with exp decay | 0.9 | 0.0 | 0.4 | RMSE (0.7 vs 0.3) | No |
| 10 | 32 | 256 | 1e-3 with exp decay | 0.9 | 0.0 | 0.4 | RMSE (0.7 vs 0.3) | Yes |

Table 1: Experiments configurations



(a) Loss

(b) RMSE

(c) $\delta_1$

(d) $\delta_2$

Figure 1: Experiments validation metrics for 30 epochs

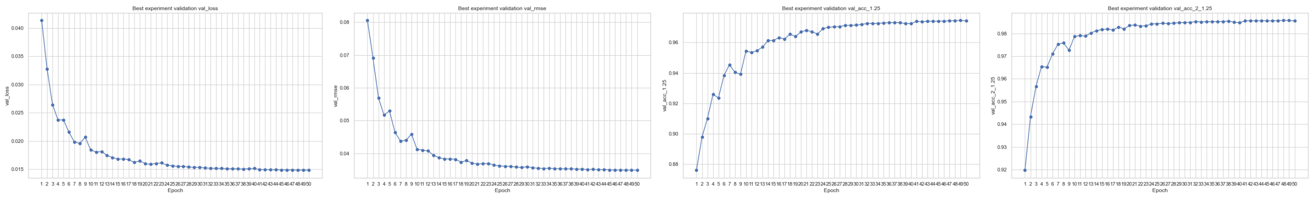
We have executed each experiment for 30 epochs, which is plenty for the model to more or less converge, given the saturation we observe during the last epochs. From the first two experiments, we can extract that using a larger image size is beneficial because we can better capture small details that we would lose with a smaller image size, as we are not modifying the network. From the third experiment, we can see that using a bigger batch size slightly worsens the results. In the fourth experiment, we wanted to try if we could improve the saturation in the last epochs with a higher learning rate with an exponential decay schedule. With the batch size of 32, we perform almost 1000 steps per epoch, so we decided to decay the learning rate every 1000 steps using a decay rate of 95%. We slightly improved the performance, achieving the best one so far, so we decided to keep it for the rest of the experiments.

We moved to modify the loss function, trying different configurations. We started changing the weights of the three losses, concluding the SSIM to be the most important, followed by the depth smoothness and the L1 loss. The best configuration is the one from the fifth experiment, reaching the best performance in both loss and metrics. However, given the difficulties of reducing the RMSE, we tried modifying the loss. We decided to compute the final loss as the weighted sum of the original loss and the RMSE. We tried giving equal weight to both (0.5) and more weight to the RMSE (0.7 vs 0.3). Also, given the low weight and importance the L1 loss has, we decided to try to remove it by giving a 0 weight in the ninth experiment. As we can see, we are slightly improving the RMSE with both experiments, and given the ninth seems to be able to reduce it even more than the eighth, we choose it as the best experiment so far.

Finally, we moved to perform data augmentation with the best configuration. As we did in the first assignment, we augment an image by adding objects with a probability of 50%. In this case, we add one or two objects, allowing them to overlap the person, performing rotations and rescalings. As mentioned, we want to see if the model can predict the depth of the body and the clothes with the potential overlap we are introducing. The loss and metrics get worse, as expected, given the increased difficulty. In Appendix A, we show some predictions compared to their ground truth. As we can see, they are very decent, as we obtain quite good depth predictions, even though sometimes we predict deeper depths in the heads or the legs. Given these results, we can see that the model can ignore the added objects to some extent and learn the desired objective.

## 3.2 Final Results

We can conclude that the ninth experiment is the best one, so we executed it again with 50 epochs instead of 30. We saw the performance saturation with only 30 epochs, but we want to run it as a final experiment to get the final results. In Figure 2 we show the evolution of the loss and the other three metrics.



(a) Loss (b) RMSE (c) $\delta_1$ (d) $\delta_2$

Figure 2: Best experiment validation metrics for 50 epochs

We are not getting improvement in any of the metrics from the thirtieth epoch onwards, as expected given the saturation we obtained during the experimentation phase. However, we wanted to execute it again during more epochs to get the final quantitative results. Given this almost nonexistent improvement, we can check the qualitative results in Appendix B, where we plot the same frames as in Appendix A for comparison reasons. We cannot perceive any difference, and we commit the same mistakes. We predict some heads or legs as the deepest element when they are not. However, we are getting pretty decent results considering that we are predicting depth from an RGB image and not having any other supervision. Nevertheless, further experimentation would be good to correct the mistakes and improve the metrics to escape that saturation zone.

# 4 Conclusions

In this assignment, we have trained a U-Net model from scratch to perform body and clothes depth estimation, getting very decent results. Given the fixed model, we had a small room for experimenting and trying to improve the results even more. Also, we have to mention the hurdles we had with the baseline code to make it run with our dataset. The extra modifications we introduced (besides those mentioned in the task description) have been essential to training the model. In fact, without scaling the depth maps, neither the loss nor the other metrics changed during all the epochs, which was very frustrating. For future work, we would like to change the model architecture and try other metrics and loss configurations. Finally, it would be interesting to perform data augmentation having depths for the objects we add, as we consider it more interesting that not having depth at all (i.e. being placed at the front).

# References

[1] Max Hermann et al. "Self-Supervised Learning for Monocular Depth Estimation from Aerial Imagery". In: *CoRR* abs/2008.07246 (2020). arXiv: 2008.07246. URL: https://arxiv.org/abs/2008.07246.

# A Qualitative Results with Data Augmentation

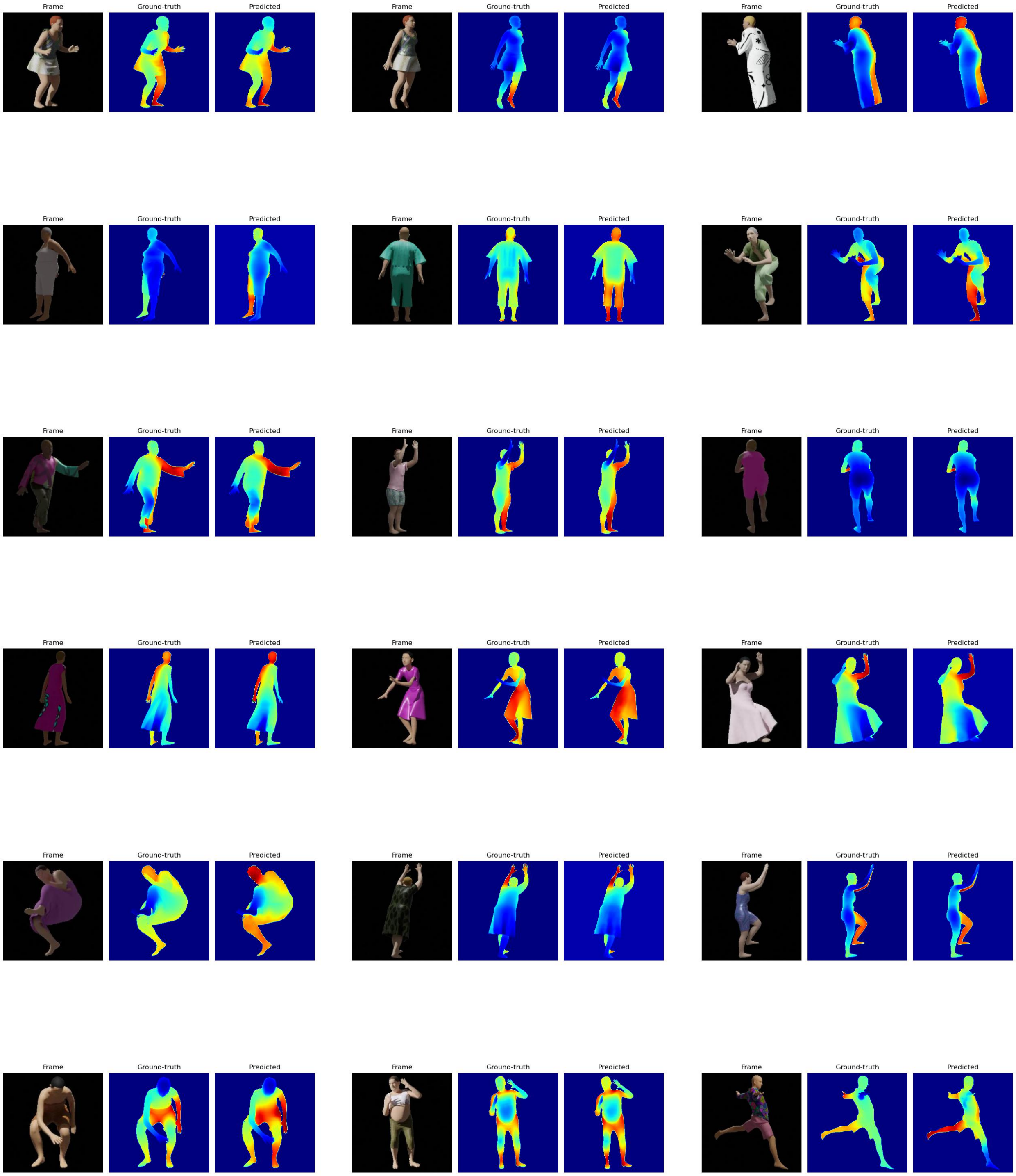


Figure 3: Results obtained with the model trained with data augmentation

# B Best Model Qualitative Results

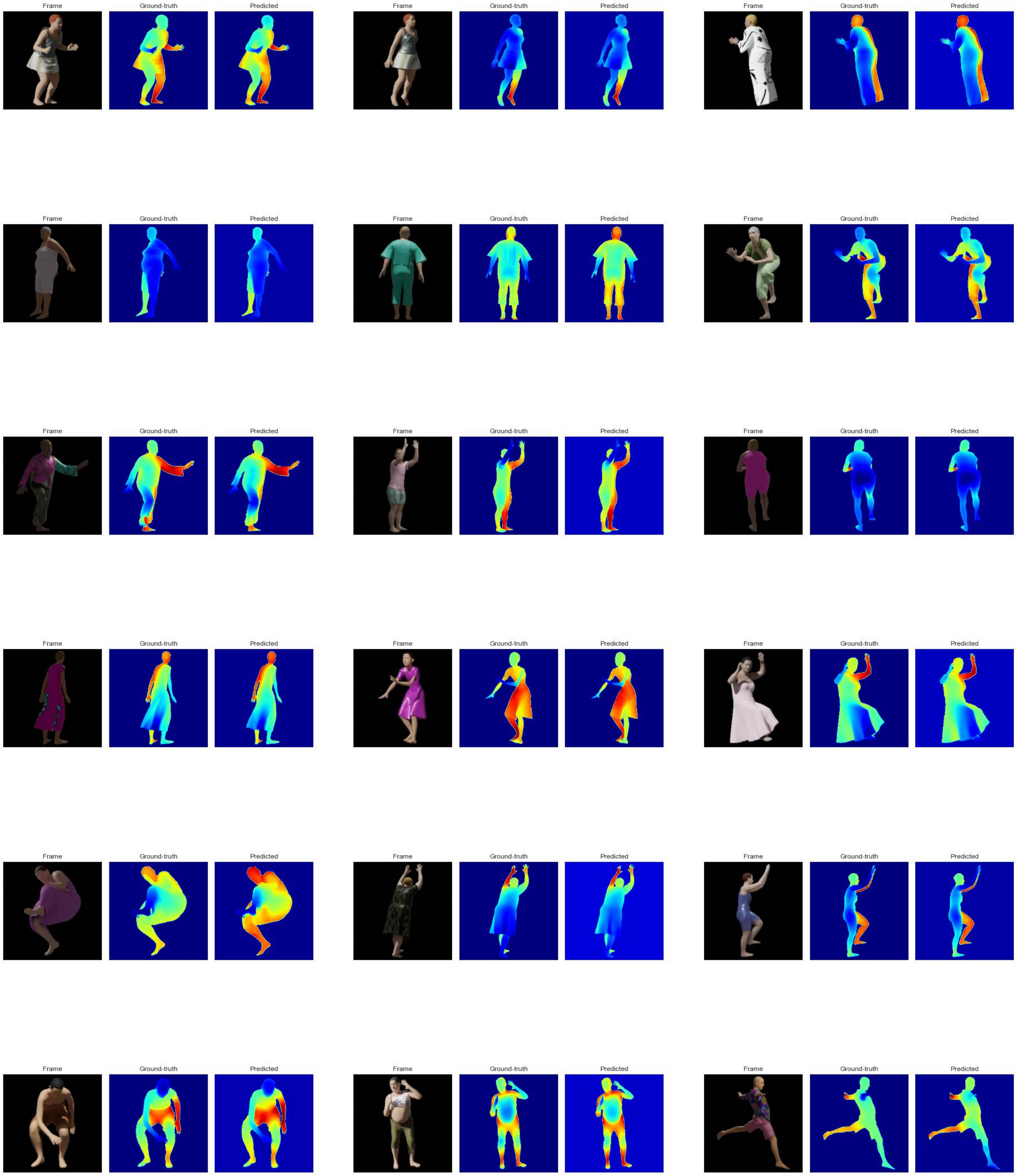


Figure 4: Results obtained with the best model