ETHand - Semantic Amodal Segmentation for Hands

Daniele Chiappalupi
dchiappal@student.ethz.ch

Pierre Motard
pmotard@student.ethz.ch

Andrea Ziani
aziani@student.ethz.ch

ABSTRACT

With the spread of Convolutional Neural Networks (CNNs), researchers reached new state-of-the-art of in previously challenging problems. One popular task is image segmentation, which tackles the problem of detecting and marking objects inside an image. More recently, the tackle of amodal segmentation tasks added a level of complexity given by the fact of having to classify hidden parts of the images and, while well-known architectures perform great in modal segmentation tasks, in amodal tasks they tend to be overconfident about their predictions. In this work, we propose a supervised approach that addresses the problem of amodal segmentation for *hands*. Our proposed method, based on the U-Net architecture, has been able to obtain a mean-intersection-over-union score of 0.55804 on the public test split of the provided InterHand2.6M dataset.

1 INTRODUCTION

Semantic image segmentation is the process of giving a label to each pixel of an image such that it is possible to identify sub-parts, each of them representing a different object. The extraction of the location of objects inside an image is fundamental for many critical applications, such as autonomous driving, robotic navigation, localization, and scene understanding. *Hands* are an entity that has been subject of several studies on image segmentation [1, 2, 7]. Their pose and shape reveal what we plan to do or where we are directing our attention to. For these reasons, extracting hand regions is a critical step for understanding fine motor skills such as hand-object manipulation and hand-eye coordination.

Hand segmentation is often used as first step of 3D hand pose estimation: in [19] Zimmermann and Brox leverage a segmentation neural network to first localize the hand within an image, and reduce the research space for the subsequent pose estimation. In addition, for many applications in this domain, it is necessary to detect not only one single hand, but also two hands interacting with each other in their visible and invisible parts. In literature, the task of predicting occluded parts is called *Semantic Amodal Image Segmentation*. While this task has been approached in the past focusing on several objects (e.g. in [12] Qi et al. propose a new architecture to detect invisible parts using KITTI dataset [4]), to the best of our knowledge, there is no research focused just on amodal segmentation for hands.

The goal of our project is to estimate sixteen (plus 1 for background) class labels for both visible and occluded left-hand and right-hand parts (see Fig. 1). To solve this problem, we propose a semantic segmentation algorithm leveraging two U-Net [15] based segmentation networks trained in sequence, and an adversarial discriminator network. In our architecture we apply a similar methodology to the one proposed by Hung et al. [6] and we treat the second segmentation network as the generator in a Generative Adversarial Network (GAN) [5] framework. Usually a GAN consist of two parts: generator and discriminator. These two play a min-max game in the training process. The generator takes a sample vector and outputs a sample of the target data distribution, while the discriminator aims to distinguish generated samples from real ones. The generator is trained to fool the discriminator and generates samples that are more and more similar to those from the target distribution.

In our architecture the segmentation network plays the role of the generator. It outputs the probability maps of the semantic labels given an input image and passes them to the discriminator network which tells whether the mask looks real or fake. Under this adversarial setting, we enforce outputs to be as close as possible to the ground-truth label maps spatially.

In the beginning, the first segmentation network is trained on the entire training set, and the discriminator network is trained to distinguish between target masks and predicted masks. Once these networks have been trained we use the segmentation network to produce masks for a set of images lacking of target labels. These masks are used as "ground-truth" in the first few epochs of training of the second segmentation network. Afterwards, the training phase of the second segmentation network continues using a weighted combination of Cross-Entropy loss, Dice loss, and Adversarial loss.

This approach is able to obtain a mean-intersection-over-union score of 0.55804 on the public test split of the provided InterHand2.6M dataset This representing an improvement of 12.05% with respect to the baseline model provided by the skeleton code.
Organization: In Section 2, we extensively describe the idea behind our solution and we present some of the implementation details. In Section 3, we show the results achieved by running our best approach as well as different baselines we introduced while implementing it. In Section 4, we introduce an interesting idea that could have given us the ability to leverage both unpaired images and unpaired labels, but that unfortunately did not work as expected. In Section 5, we conclude the research summarizing the contributions of our work.

2 METHOD

2.1 Dataset

From InterHand2.6M dataset [11], roughly 68 thousand 128x128 RGB images with custom segmentation masks of left hand and right hand parts, are provided (we will refer to this as *paired training set*). Each segmentation mask is composed by pixels with value $0 < v \leq 16$ if the pixel is belonging to part of a hand, or with $v = 0$ if belonging to background. Also, to enable our unpaired training approach, an additional set of 519 thousand 128x128 RGB images is used (we will refer to this as *unpaired images*). A supplementary set of *unpaired labels* is available. However, in our best approach we do not make use of it.

2.2 General Network Architecture

In this project, we use a sequence of two U-Net based model variants and a convolutional discriminator network with a structure similar to the one proposed by Radford et al. [13].

Our first segmentation network, initialized with pre-trained weights on the ImageNet dataset, is a variation of the U-Net architecture which uses the Xception encoder proposed in [3]. We will refer to this model as *G1*.

The second segmentation network is a U-Net++, first proposed by Zhou et al. [17]. For this architecture, as the limited time did not give us the chance to apply a more complex encoder, we decided to make use of a RegNetY_032 [14], a good trade-off between complexity and training time. This encoder has been initialized with weights pre-trained on the ImageNet dataset. We will refer to this model as *G2*.

Lastly, the discriminator model is a simple convolutional network consisting of 5 convolution layers with 4×4 kernel and {64, 128, 256, 512, 1} channels with stride of 2. Each convolution layer, except for the last, is followed by a batch normalization layer and a Leaky-ReLU [10] layer with a negative slope of 0.2. We will refer to this model as *D*.

2.3 Data augmentation

Training a deep neural network is highly data-demanding, and the training dataset we have is rather small for the task we want to address. A first step to solve this problem is to use extensive data augmentation to increase the available samples significantly. We applied this technique by performing randomly parameterized transformations on every image of the training set (both paired and unpaired) and the corresponding ground-truth mask if present. By augmenting the data, the network will have a lower probability of incurring the same image more than once, thus reducing the risk of overfitting. The transformations we used are the following: scaling, rotation, and pixel intensity variation. The scaling parameter is sampled from a Gaussian distribution with mean 0 and standard deviation 0.01^2, clipped between -1 and 1. The rotation parameter is sampled from a Gaussian distribution with mean 0 and standard deviation 5^2, clipped between -2 and 2. Note that the rotation is applied with probability 0.6. The pixel intensity variation parameters are three random numbers sampled from a uniform distribution between 0.99 and 1.01; each channel of the input image is multiplied pixelwise with the corresponding parameter, and the values are then clipped between 0 and 255. Moreover, with probability 0.5, we exchange the right and left hands in the input image, performing a horizontal flip.

2.4 Training and loss functions

The key idea behind our solution is to make use of the unpaired training set to better generalize over unseen data. In order to achieve this, we split the training in two different phases (visually represented in Fig. 2).

Phase 1. During the first phase, *G1* is trained on the *paired training set* to produce segmentation masks. In this case, we apply the Cross-Entropy loss function and, as InterHand images are characterized by a heavy class imbalance (i.e. very few pixels represent the fingers and many more pixels depict the background and the areas of the palm and back of the hands), classes are manually weighted to give more importance to those with fewer pixels.

$$\mathcal{L}_{G1} = \mathcal{L}_{ce} = \frac{\sum_{i=0}^{16} w_i \dot{(}-x_i + log(\sum_j e^{x_j}))}{\sum_{i=0}^{16} w_i}$$

Furthermore, when a hand is heavily occluded, the segmentation network may produce a degenerate mask. To address this problem, we train *D* to classify whether the predicted labels are real or fake. Given y the ground-truth labels and $\hat{y}$ the labels predicted by *G1*, the loss for *D* can be summarized as:

$$\mathcal{L}_D = \mathcal{L}_{bce}(\hat{y}, 0) + \mathcal{L}_{bce}(y, 1)$$

where $\mathcal{L}_{bce}$ corresponds to the Binary Cross-Entropy loss function. We train the model in this phase, and we use it in the next phase to provide an adversarial loss for the second segmentation model. In this way, *G2* is discouraged to generate masks that do not look like valid hands.

Phase 2. After having successfully trained the two aforementioned neural networks, we train *G2*. This phase is itself split into two steps:

(1) *Weights Initialization.* We use *G1* trained during *phase 1* to generate imperfect labels from the *unpaired images*. In order to initialize the weights of *G2*, we train it considering these imperfect labels as ground-truth. This part of training on imperfect masks yields a good weights initialization which helps the network generalizing well. To avoid *G2* being influenced by very poorly generated labels, we drop at each epoch half of the images of the batch on which *G1* is less confident about its predictions. From tuning the losses, we reached the conclusion that the best performances result from a linear combination of the same Cross-Entropy Loss used in the first phase for *G*1 and the Dice Loss [9] which is well suited for class imbalanced problems. The loss for this step can be summarized with the subsequent formula:

$$\mathcal{L}_{G2}^{(init)} = \mathcal{L}_{segm} = 0.35 \times \mathcal{L}_{ce} + 0.65 \times \mathcal{L}_{dice}$$

(2) *Paired training.* In this step, we train *G2* on the *paired training set*. Also, as previously mentioned, we use *D* to help the segmentation network creating predictions as close as possible to the ground-truth distribution. The loss function for this step, summarized in the formula below, is a linear combination of Cross-Entropy Loss, Dice Loss and Adversarial Loss:

$$\mathcal{L}_{G2}^{(ptrain)} = 0.9 \times \mathcal{L}_{segm} + 0.1 \times \mathcal{L}_{adv}$$

Where $\mathcal{L}_{adv} = \mathcal{L}_{bce}(\hat{y}, 1)$. Indeed, while training *G2* we want the outputs from *D* to be 1, therefore classified as real masks, and thus increase the loss if the discriminator classifies them as 0.

These steps are repeated twice, for a different amount of epochs. Firstly, the *weights initialization* step for more epochs than the *paired training* step. Secondly, the same sequence but with a reverse balance of epochs for each step.

3 EVALUATION

To have a fair comparison among the different approaches, we run every experiment under the same setting in terms of number of epochs (140), learning rate (0.00015 when training with the

paired training set, and 0.00025 when training with the *unpaired images*), batch size (16), and optimizer (Adam [8]). We present each model along with its public test score, which measures the mean-intersection-over-union (mIoU) of the predicted segmentation labels against the ground truth masks. All the results commented below, are reported in Table 1.

Baseline model. As baseline model we adopted the plain U-Net given in the skeleton code. This model reached a baseline score of 0.49801. The loss we used for our baseline model is a simple Cross-Entropy Loss.

Architecture improvement. As first improvement we decided to change the segmentation network architecture, adopting the U-Net++ with a RegNetY_032 pre-trained encoder. By changing the segmentation network architecture we reached a score of 0.53739, that is already an improvement of 7.91% when compared to the aforementioned baseline model.

Loss function improvement. As second improvement, we experimented different loss functions. First, we added the class weights to the Cross-Entropy Loss, in order to address the imbalanced nature of the task. Adding such class weights, we reached a score of 0.53916. Furthermore, by applying the linear combination of losses described in Section 2.4, the mIoU on the public test set increased to 0.54449. This proves that in class unbalanced tasks, to give more importance to the pixels causing the major source of error increases the quality of the predicted masks.

Generated Labels. As third improvement, to the previous approach we introduced the sequence of two segmentation networks (*G1* and *G2*) to exploit *unpaired images*. Training the UNet++ on the imperfect masks generated by a pre-trained UXceptionNet while maintaining the linear combination of loss functions lead to a mIoU score of 0.55772. In terms of training phases splitting, as described in Section 2.4, the first *weights initialization* step is performed for 15 epochs and the *paired training* step is performed for 55 epochs. Instead, the second *weights initialization* step is performed for 5 epochs and the *paired training* step is performed for 65 epochs.

Adversarial learning. As final improvement, we introduced the adversarial learning to help *G2* not to predict degenerate segmentation masks as explained in Section 2.4. Even tough the introduction of the adversarial learning does not help as much as expected, with this final architecture we reached our highest mIoU score of 0.55804. This represents an improvement of 12.05% with respect to our baseline.

4 DISCUSSION

Along with our best approach described in the previous sections, we looked for a way to take advantage of both the *unpaired images* and the *unpaired labels*. One idea that came to our mind was to try a variant of the CycleGAN [18] suited to our task. Let the images and segmentation labels belong to two distinct domains that we denote as X and Y, respectively. The model, similar to what proposed in [16], consists of two generation networks $G_{XY} : X \rightarrow Y$, $G_{YX} : Y \rightarrow X$, as well as two discriminator networks D_X and D_Y. The idea is that G_{XY} takes images as input and generates segmentation masks, while G_{YX} does the inverse. D_X classifies whether an image is real or fake and D_Y does the same on the labels.

TABLE 1. Final results and improvements with respect to the baseline score of our models. For each model, we report its test score, improvement over the baseline, and improvement over the previous step.

Model	Test Score	Δ Baseline	Δ Previous
Baseline model	0.49801	-	-
U-Net++			
- w/ CE Loss	0.53739	7.91%	-
- w/ Weighted CE Loss$^{(1)}$	0.53916	8.26%	0.33%
- w/ $^{(1)}$, Dice Loss$^{(2)}$	0.54449	9.33%	0.99%
- w/ $^{(1)}$, $^{(2)}$, Gen. Labels$^{(3)}$	0.55771	11.99%	2.43%
- w/ $^{(1)}$, $^{(2)}$, $^{(3)}$, Adv.	**0.55804**	**12.05%**	0.06%

Using these models, the architecture allows to cycle on the data images (respectively on labels), generating the intermediary labels (images) using G_{XY} (G_{YX}). When training on *paired training set*, we can compare the generated label (image) to the real one by computing a loss that we call reconstruction loss. Additionally, this fake label (image) will be discriminated using D_Y (D_X), computing another loss called adversarial loss. Finally, G_{YX} (G_{XY}) is used to recover the initial image (label) from the generated label (image), we aim for reproducing the same image (label) as the one we start from. It yields another loss that compares the initial and recovered images (labels) called cycle loss. The more accurate the generated label (image), the better the recovered image (label) may be (and vice-versa). This method allows to leverage the unpaired datasets, simply skipping the reconstruction loss which needs pairs. Eventually, the loss is a combination of the reconstruction loss (on *paired training set*), with an adversarial loss and the cycle loss.

However, the main difficulty here is to have generators and discriminators of a similar level that can train mutually in adversarial fashion. We believe this was a key weak point in the implementation of this method. The multiple models were not performing similarly enough to improve simultaneously. In addition, the amodality of the task introduced another issue. Recovering the image from two separate masks was not performing well as the model poorly recognized which hand was upfront.

5 CONCLUSION

In this report, we presented subsequent improvements to the base structure of the U-Net to successfully handle the hand amodal segmentation task. Initially, we verified how improvements to the model architecture allow us to achieve better performance. Then, we leveraged combination of losses and the unpaired data to obtain more accurate and reliable segmentations. More precisely, we initialized the weights under imperfect data circumstances, and we showed how this improved the score of the model. Finally, we obtained our best result by adopting an adversarial approach to help the model producing labels that look like valid hands. The final architecture achieved the best score of 0.55804 on the public test set and produced visually satisfactory results.

REFERENCES

[1] Sven Bambach, Stefan Lee, David J Crandall, and Chen Yu. 2015. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proceedings of the IEEE International Conference on Computer Vision*. 1949–1957.

[2] Alejandro Betancourt, Pietro Morerio, Emilia Barakova, Lucio Marcenaro, Matthias Rauterberg, and Carlo Regazzoni. 2017. Left/right hand segmentation in egocentric videos. *Computer Vision and Image Understanding* 154 (2017), 73–81.

[3] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1251–1258.

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)* (2013).

[5] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *arXiv preprint arXiv:1406.2661* (2014).

[6] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. 2018. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934* (2018).

[7] Byeongkeun Kang, Kar-Han Tan, Nan Jiang, Hung-Shuo Tai, Daniel Tretter, and Truong Nguyen. 2017. Hand segmentation for hand-object interaction from depth map. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 259–263.

[8] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[9] Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019. Dice Loss for Data-imbalanced NLP Tasks. arXiv:arXiv:1911.02855

[10] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. Citeseer, 3.

[11] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. 2020. InterHand2. 6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. *arXiv preprint arXiv:2008.09309* (2020).

[12] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. 2019. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3014–3023.

[13] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[14] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10428–10436.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[16] Samarth Shukla, Luc Van Gool, and Radu Timofte. 2019. Extremely Weak Supervised Image-to-Image Translation for Semantic Segmentation.

[17] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. 2018. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 3–11.

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2020. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.

[19] Christian Zimmermann and Thomas Brox. 2017. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*. 4903–4911.

A TASK VISUALIZATION

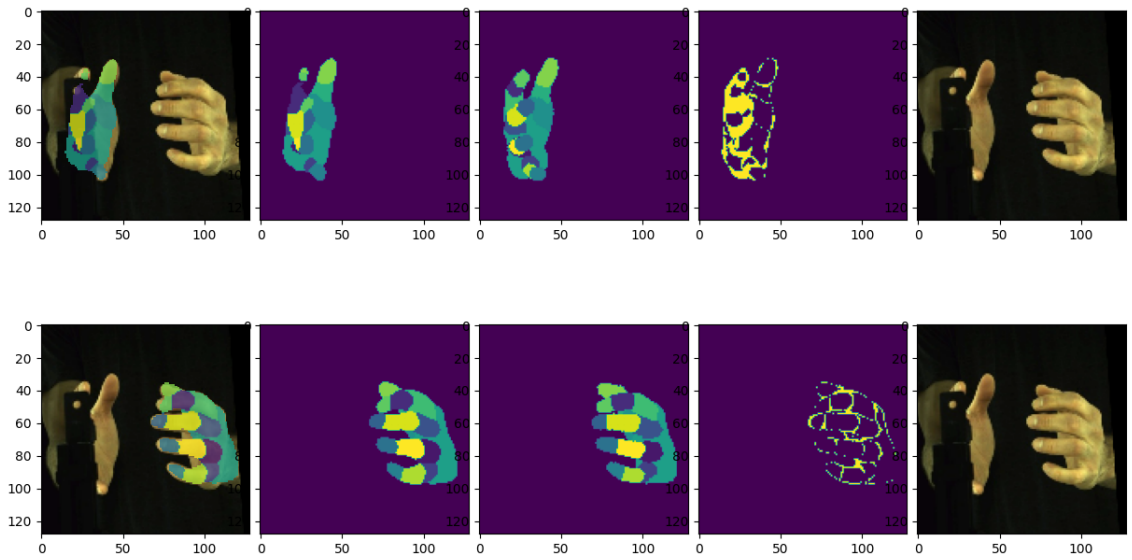

Figure 1: Visual representation of the amodal segmentation task. From left to right: input image with prediction overlaid, prediction, groundtruth, error map, input image.

B ARCHITECTURE VISUALIZATION

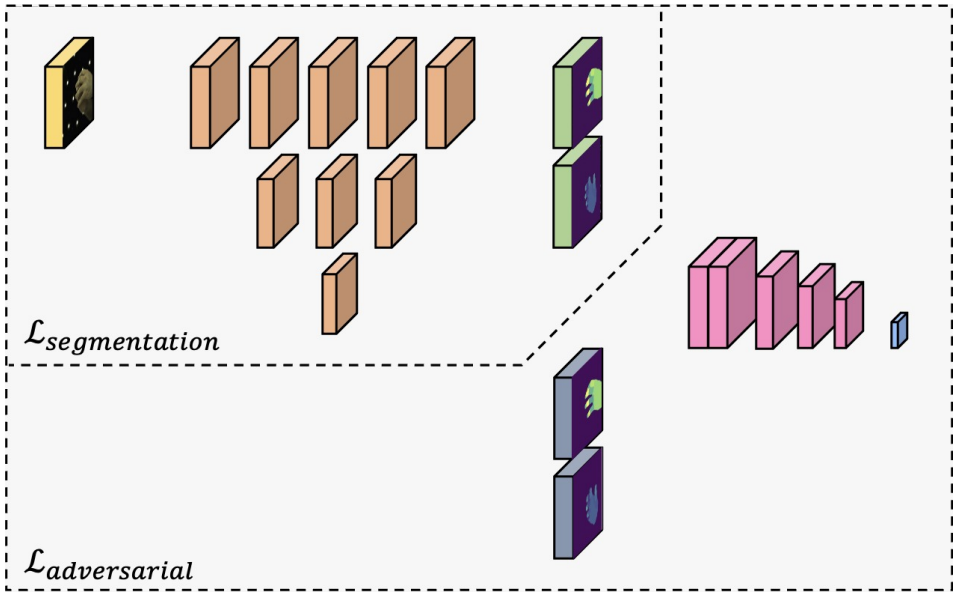

Figure 2: Visual representation of the training procedure. On one hand, during phase 1, the UNet++ (orange blocks) is trained with just the segmentation loss. On the other hand, during phase 2, the discriminator (pink blocks) is used to compute an adversarial loss. The UNet++ is then trained on a weighted average of segmentation loss and adversarial loss.